HAL
open science

Modélisation du raisonnement géométrique pour la programmation des robots

Jocelyne Troccaz

To cite this version:

Jocelyne Troccaz. Modélisation du raisonnement géométrique pour la programmation des robots. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1986. Français. NNT : . tel-00321468

HAL Id: tel-00321468

https://theses.hal.science/tel-00321468

Submitted on 15 Sep 2008

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

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

THESE

Présentée à

L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

pour obtenir le grade de
docteur de l'Institut National Polytechnique de Grenoble
(arrêté ministériel du 5 juillet 1984)
"Informatique"

par

PERTIN Jocelyne épouse TROCCAZ

□□□□□

MODELISATION DU RAISONNEMENT GEOMETRIQUE POUR LA PROGRAMMATION DES ROBOTS.

□□□□□

Thèse soutenue le 14 mars 1986 devant la commission d'examen.

G. VEILLON	**Président**
G. GIRALT **J.C. LATOMBE** **C. LAUGIER** **F. MARTINEZ**	**Examinateurs**

Résumé

Cette thèse est relative à la programmation automatique des robots. La première partie porte sur l'analyse des besoins en modélisation géométrique pour la Robotique et la définition d'un modèle approprié (représentations multiples, univers dynamique, représentation de l'incertain). La seconde partie présente une méthode originale de raisonnement géométrique pour l'automatisation de la préhension dans le contexte d'une tâche d'assemblage. Finalement, nous proposons dans une troisième partie de la thèse, une architecture de système pour la programmation automatique des robots de manipulation.

Abstract

This work takes place in the general context of the automatic programming of assembly robots. In a first part of the thesis, we analyze the geometric modeling requirements of Robotics and then submit a relevant model (multiple representation, spatial evolution modeling, uncertainty modeling). The second part of the thesis focuses on geometric reasoning; an original method for dealing with purposive automatic grasping is described. In the last part of this document, the general architecture of a system for automatic robot programming is proposed.

Remerciements

Malgré tout le mal que j'ai eu à rédiger cette thèse, je crains bien que cette page ne soit la plus difficile tant j'aimerais y mettre toute ma sincerité.

Un travail de thèse est avant tout une épreuve collective, c'est pourquoi je tiens à remercier tout d'abord, ceux qui supportent quotidiennement mes mouvements d'humeur isomorphes aux mouvements de grêve de ma muse:

- Pierre, mon mari, sacrifié à l'autel de la recherche, dont le soutien constant est, pour moi, d'un secours inestimable,
- ces messieurs de l' "Equipe Intelligence Artificielle et Robotique" du LIFIA dont l'amitié autant que le cadre scientifique auxquels ils contribuent font de cette équipe un lieu de travail privilégié.

Cette équipe n'existerait sans doute pas si Jean-Claude Latombe n'avait pas su allier ses compétences à son energie inépuisable pour en faire un pôle attractif. Je tiens à lui exprimer ma profonde gratitude pour cet enthousiasme communicatif qu'il met dans son travail et pour la part prépondérante qu'il a prise à la conception de ce manuscrit, tant dans sa forme que dans son contenu.

A Pierre,
A mes amis de l'équipe,
A Jean-Claude.

Je tiens également à remercier:

- Gérard Veillon, Professeur à l'Institut National Polytechnique, Directeur de l'ENSIMAG, qui me fait l'honneur de présider le jury de cette thèse et qui, toujours comme aujourd'hui, a pris à cœur les recherches menées au LIFIA, et a permis de les valoriser par l'enseignement,
- Georges Giralt, Directeur de Recherche au CNRS, Responsable du Projet National "Automatique et Robotique Avancées", pour l'aide qu'il a su apporter aux chercheurs de l'équipe à de multiples occasions et pour les nombreuses opportunités qu'il m'a offertes, de faire partager mes recherches,

- Jean-Claude Latombe, Professeur (détaché) à l'Institut national Polytechnique, Président de la Société ITMI (Industrie et Technologie de la Machine Intelligente) pour avoir dirigé cette thèse et avoir accepté de juger ce travail,
- Christian Laugier, Chercheur à l'INRIA, Responsable de l'équipe, qui a guidé mes premiers pas dans la recherche et qui m'a initiée aux grands mystères de la modélisation et du raisonnement géométriques,
- Francis Martinez, Professeur à l'Université Scientifique, Technologique et Médicale de Grenoble pour avoir accepté de porter un œil critique sur ce manuscrit.

Georges Giralt, Jose-Luis Gordillo, Philippe Jorrand, Yves Lagoude, Augustin Lux, Emmanuel Mazer, Pierre Puget et Pascal Theveneau m'ont permis par leurs nombreuses remarques et suggestions d'améliorer cette rédaction; je tiens à leur exprimer, à tous, ma reconnaissance.

Que soient également remerciés John Hollerbach et Kenneth Salisbury du M.I.T. pour la gentillesse avec laquelle ils m'ont envoyé des photographies illustrant leurs travaux, ainsi que toute l'équipe du service de reprographie pour la qualité de son travail.

Enfin, un grand merci à Pierre Puget et Pascal Theveneau pour l'aide précieuse qu'ils m'ont apportée pour la réalisation de ce manuscrit.

Jocelyne Troccaz

Sommaire

1 Rappel sur les niveaux de programmation en Robotique **5**
1.1 Exemple introductif: Carnets de voyage 5
1.2 Formalisation des niveaux 6
1.3 Bref état de l'Art par niveau 8
1.4 Objet de notre travail 10

2 Besoins en modélisation pour la programmation de niveau objet **13**
2.1 La problèmatique de niveau objet 13
2.2 Les besoins en modélisation 17
2.2.1 Un univers géométrique 17
2.2.2 Un univers physique 18
2.2.3 Un univers incertain 18
2.2.4 Un univers en évolution 19
2.3 Insuffisance des modèles actuels 19
2.3.1 Modèles de type CFAO 19
2.3.2 Modèles développés pour la programmation de niveau objet 25

3 SMGR: un système de modélisation pour la Robotique **27**
3.1 Structure de la modélisation dans SMGR 27
3.1.1 Le modèle géométrique des objets 28
3.1.2 Le modèle relationnel de l'univers physique 31
3.1.3 Modélisation de l'incertain 45
3.2 Module d'entrée de SMGR 49
3.2.1 Les objets élémentaires (module SMG) 50
3.2.2 Les structures articulées (module SMR) 50
3.3 Implantation 50
3.4 Limitations 51

4 Analyse bibliographique de la saisie **59**
4.1 Choix de l'outil 59
4.2 Choix de l'action 65
4.2.1 Recherche de prises 68
4.2.2 Etude d'accessibilité 69
4.2.3 Etude d'équilibre et de stabilité 72
4.2.4 Etude de compatibilité contextuelle 75

5 Méthode développée 83
5.1 Présentation de l'approche 83
5.1.1 Formalisation du problème de la saisie 83
5.1.2 Méthode proposée 85
5.2 Phase de prédiction: accessibilité locale et stabilité 89
5.2.1 Présentation de la méthode 89
5.2.2 Détermination des contacts potentiels 90
5.2.3 Construction des prises 92
5.2.4 Paramètres de préconfiguration de l'outil 98
5.3 Phase de vérification: accessibilité globale 99
5.3.1 Présentation de la méthode 99
5.3.2 Détermination des contraintes initiales 102
5.3.3 Recherche des obstacles à la pince 104
5.3.4 Propagation des contraintes 106
5.3.5 Méthodes de grossissement et coupe 108
5.3.6 Utilisation d'heuristiques pour la détermination de $P_{P/O}(\Pi)$ 113
5.4 Mise en œuvre 114
5.4.1 Interface géométrique avec les modules de raisonnement . . . 114
5.4.2 Implantation et expérimentation 117
5.4.3 Discussion des choix algorithmiques effectués 120

6 Discussion 123
6.1 Spécificité de l'approche 123
6.1.1 Par rapport au mode de saisie 123
6.1.2 Par rapport au modèle 123
6.1.3 Par rapport à l'outil 124
6.2 Inconvénients et avantages de notre méthode 125
6.2.1 Problèmes de compatibilité contextuelle 125
6.2.2 Saisie de structures 128
6.2.3 Application de l'approche à la "machine intelligente" 129
6.3 La préhension humaine 134
6.4 Raisonnement fonctionnel: vers une modèlisation 3d des objets . . . 136

7 Vers la non programmation des robots 139
7.1 Interêt de la programmation automatique 139
7.2 Historique rapide 140
7.3 La recherche sur les sous-problèmes de la programmation de niveau objet 140
7.4 Intégration d'un système de niveau objet 142
7.4.1 Le système TWAIN 143
7.4.2 Le système SHARP 144
7.4.3 TWAIN et SHARP 151

A SMGR: Exemples **169**
A.1 Les objets elémentaires 169
A.1.1 Les composants élémentaires 170
A.1.2 Définition du repère d'un objet élémentaire 171
A.2 Les structures articulées 173
A.2.1 Mode opératoire 173
A.2.2 Exemple 173

B Description analytique des articulations dans SMGR **177**
B.1 Paramètres de Denavit-Hartenberg 177
B.2 Représentation des liaisons entre corps 178

C Algorithme d'analyse d'accessibilité globale **181**

Liste des Figures

.1 Quelques fonctions de l'agent intelligent 2

1.1 Les pentacubes 6
1.2 Assemblage à réaliser 8
1.3 La programmation automatique 11
1.4 Assemblage manipulé par le système BUILD [Fahlmann 74] 12

2.1 Exemple de spécification de niveau objet en LM-Géo 14
2.2 Programme LM correspondant à la spécification précédente 15
2.3 Architecture simplifiée d'un système de niveau objet 17
2.4 Historique de la CAO [Requicha Voelcker 77] 20
2.5 La modélisation dans le système ACRONYM [Brooks 79] 21
2.6 Modélisation par énumération de l'espace occupé [Ballard] 22
2.7 Représentation par octrees 22
2.8 Représentation de type BR [Requicha Voelcker 82] 23
2.9 Représentation de type CSG [Requicha Voelcker 82] 23
2.10 Représentation par balayage hybride [Requicha Voelcker 82] 24

3.1 Exemple de représentation de type BR 30
3.2 Représentation de la topologie locale 31
3.3 Structure à articulations simples 33
3.4 Structure à articulations dépendantes 34
3.5 Manipulateur à structure de type parallèlogramme 35
3.6 Pince à structure de type parallèlogramme 36
3.7 Représentation d'une structure de type parallèlogramme 36
3.8 Structure arborescente de l'univers 37
3.9 Différents types de contacts 39
3.10 Représentation quantitative des relations dans les structures articulées 40
3.11 Représentation quantitative des relations dans les structures rigides . 41
3.12 Evolution du modèle relationnel au cours d'une manipulation 42
3.13 Représentation des incertitudes pour un contact plan sur plan 47
3.14 Représentation des incertitudes pour un contact cylindrique 47
3.15 Manipulation nominale 53
3.16 Manipulation réelle 53
3.17 Les objets manipulés et leur localisation dans l'espace de travail . . . 54

3.18 Modification des incertitudes de position lors de la saisie (1) 54
3.19 Modification des incertitudes de position lors de la saisie (2) 55
3.20 Une utilisation de SMGR: la simulation de programmes de manipulation 55

4.1 Saisie par adhérence [Bancon 82] 60
4.2 Saisie par serrage [Hirose 77] 60
4.3 Prothèse de main 61
4.4 Main développée par [Okada 77] 62
4.5 Main développée par Hitachi 63
4.6 "UTAH/M.I.T. Dextrous hand" 64
4.7 "Stanford/J.P.L. Dextrous hand" 66
4.8 Structure tendineuse de la main 67
4.9 "Linear Grasping Set" [Lozano 76] 71
4.10 "Polar Grasping Set" [Lozano 76] 71
4.11 Différents types d'instabilité 73
4.12 Méthode développée par [Hanafusa 77a] 77
4.13 L'approche de [Mason 82] 78
4.14 Différents types de saisie [Lyons 85] 79

5.1 Paramètres de saisie de deux objets 86
5.2 Prises instables 87
5.3 Effet de l'instabilité 87
5.4 Prise obtenue à partir d'une prise instable 88
5.5 Outil type à deux mors 89
5.6 Définition de la propriété de convexité locale 91
5.7 Convexité et accessibilité 92
5.8 Accessibilité locale d'une arète 93
5.9 Accessibilité globale d'une prise 93
5.10 Propriété de visibilité 95
5.11 Prise équilibrée par le frottement 96
5.12 Calcul de l'ecartement des mors 96
5.13 Faces localement accessibles ne constituant pas une prise 97
5.14 Construction de l'espace des configurations simplifiée [Lozano 79] . . 100
5.15 Espace balayé par les mors de l'outil 102
5.16 Recherche des obstacles dans [Wolter 84] 103
5.17 Calcul de la zône de contraintes initiales de positionnement des mors 104
5.18 Espace balayé par une partie d'outil 105
5.19 Recherche des zônes accessibles aux "tranches" d'outil et propagation à tout l'outil 107
5.20 Zône de préhension accessible pour les deux "tranches" considérées . 108
5.21 Zône d'ombre située "derrière" les obstacles 109
5.22 Grossissement d'obstacles 111
5.23 Recherche des obstacles par des coupes par un plan 112
5.24 Robot ROBITRON à quatre degrés de liberté 114

5.25 Recherche heuristique d'une direction dans les secteurs angulaires libres 115
5.26 Architecture du logiciel de saisie 119
5.27 Exemples d'objets traités 120
5.28 Représentions implicites ou explicites des obstacles 121

6.1 Pince à trois mors 125
6.2 Incompatibilité entre les environnements de saisie et de lacher d'un objet 126
6.3 Environnement de lacher d'un objet trop contraint 127
6.4 Prise incompatible avec le montage à réaliser 128
6.5 Choix de prise guidé par la tâche de manipulation 129
6.6 Manipulation de structures non rigides 130
6.7 Choix de prise pour une structure non rigide 130
6.8 Capteur de vision active tri-dimensionnelle [Borianne 84] 132
6.9 Recherche des obstacles par mise à jour d'une ligne de crête 133
6.10 Visualisation de courbes paramèriques par l'algorithme de [Williamson 72] 133
6.11 Coopération entre saisie et vision [Paillard 75] 136

7.1 Notion de pré-image [Lozano-Pérez 83] 142
7.2 Architecture du système SHARP 145
7.3 La synthèse de programme dans SHARP 147
7.4 Recherche de stratégie de montage [Theveneau 85] 149

B.1 Paramètres de Denavit-Hartenberg 178
B.2 Utilisation d'un repére intermédiaire (a) 179
B.3 Utilisation d'un repère intermédiaire (b) 180

Introduction

De tous temps l'être humain a essayé de créer des machines pouvant le remplacer dans des tâches diverses. Le développement de l'ordinateur a ouvert un domaine riche en recherches: l'Intelligence Artificielle.

Les premières recherches en la matière avaient pour but essentiel de donner au calculateur des capacités de raisonnement propres à lui permettre de reproduire certaines activités cognitives de l'être humain (essentiellement la compréhension de langues naturelles, les activités ludiques, la démonstration de théorèmes...). Ces différentes activités peuvent être ressenties comme les aspects "nobles" de l'Intelligence. Dans l'Encyclopædia Universalis, [Richard 68] l'oppose à la perception et à l'instinct. A notre sens, il n'est pas si simple de définir l'Intelligence en dissociant les fonctions intelligentes de "haut niveau" (réflexion plus ou moins consciente, dialogue entre systèmes intelligents...) de celles de "bas niveau" (cf. figure .1). Elles contribuent, en effet, de manière essentielle à l'adaptativité de l'individu à son environnement (au sens large) au travers de ses déplacements, de ses gestes et de l'intégration de ses sens dans son comportement.

Hubert Dreyfus, l'un des plus fervents détracteurs de l'Intelligence Artificielle, liait d'ailleurs ce qu'il considère comme le piétinement de la discipline à l'utilisation exclusive de l'ordinateur comme "machine intelligente". Il explique en effet que:

"l'*intelligence* sous-entend la *compréhension* au sens fort, et pour doter l'ordinateur de la faculté de comprendre, il faudrait aussi lui fournir ce "sens commun" dont disposent les êtres humains adultes par le simple fait qu'ils possèdent un corps, qu'ils sont en interaction avec le monde matériel, enfin, qu'ils ont été formés à une culture" [Dreyfus 72].

Le développement de la robotique donne, pour cela, une dimension nouvelle aux recherches en Intelligence Artificielle. On pressent cependant que les robots d'aujourd'hui ne sont que l'émergence de la machine informatique de demain douée d'autonomie, d'intelligence, de dextérité ainsi que de capacités de perception et de communication.

Le développement à long terme d'une telle machine qui nécessite de s'ouvrir à de nombreuse disciplines, a motivé la création de l'IMI (Institut de la Machine

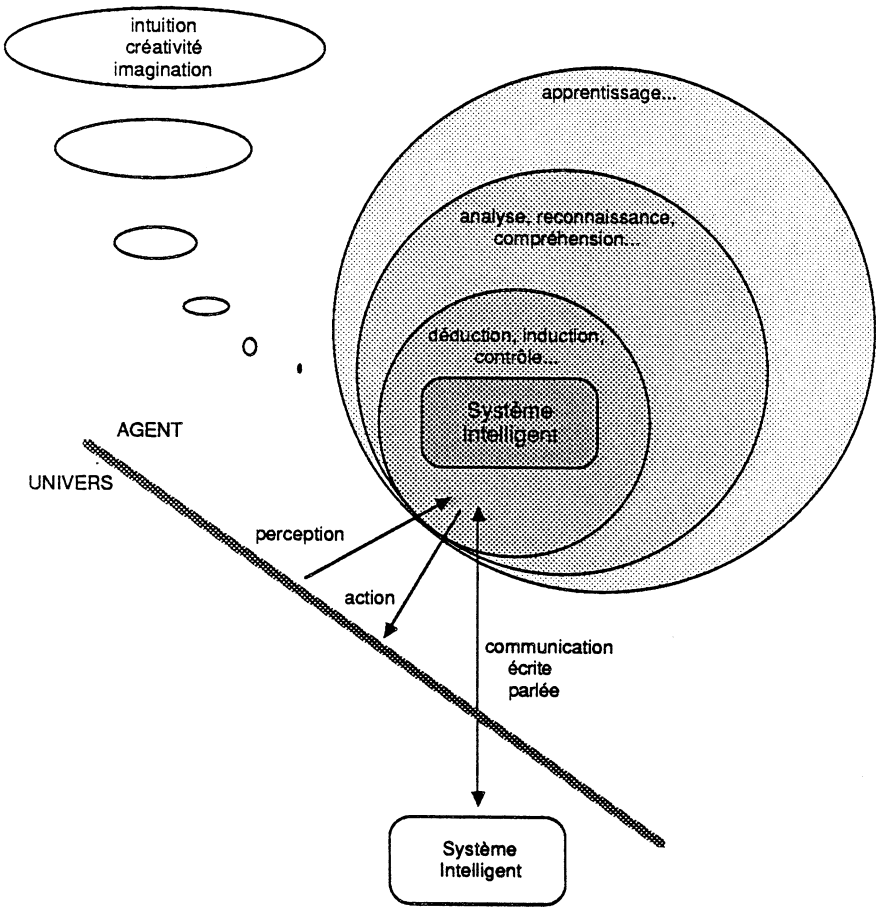


Figure .1: Quelques fonctions de l'agent intelligent

Intelligente) par le regroupement de laboratoires aux compétences complémentaires [IMI 85].

Dans ce cadre, l'objectif global de l'équipe d'Intelligence Artificielle et Robotique du LIFIA, laboratoire de l'IMI, concerne la Robotique Intelligente. Il se traduit par trois axes principaux de recherche.

La modélisation du raisonnement se tourne vers les aspects "les plus cognitifs" de l'Intelligence Artificielle (raisonnement fonctionnel, génération de plans, conception, apprentissage, raisonnement sur l'incertain...).

Les développements en Robotique ont dans un premier temps exploré la discipline en tant que telle et par cela ont permis d'acquérir un savoir-faire lié à des problèmes de "bas niveau" de la programmation des robots et de la maitrise du sens tactile. Ils se sont en outre matérialisés par la réalisation d'un support d'expérimentation [Mazer 81] essentiel à la validation des modèles et méthodes élaborées dans le cadre général de la "machine intelligente". Depuis quelques années

ces recherches se sont orientées vers l'utilisation du robot comme partie intégrante d'un système intelligent. Pour cela, la compréhension de l'"Intelligence du geste" est fondamentale.

La vision par ordinateur a évolué de manière analogue; après avoir étudié la vision comme discipline "technologique" et logicielle (développements de divers capteurs et procédés d'analyse), notre équipe envisage depuis quelques années, la vision par ordinateur comme le moyen de perception privilégié du système intelligent. Elle s'intéresse donc aux problémes de reconnaissance de formes et de compréhension de scènes. Ces recherches visent à maitriser ce que nous appellerons "Intelligence des sens".

Enfin, à ces trois premiers axes de recherche, vient s'ajouter la Robotique mobile essentielle au développement d'un système intelligent réellement autonome et adaptatif. Cette mobilité apporte, outre ses problèmes propres (navigation, modélisation de l'environnement...), une dimension nouvelle (temporelle) à la perception, visuelle ou autre.

Le développement d'un robot intelligent nécessite de maitriser trois notions essentielles: l'espace, le temps, l'incertain.

Il est à noter que, dans l'évolution des espèces, la construction dans le système nerveux central d'une représentation des objets (dans leur structure spatiale et leurs rapports mutuels) constitue une étape clé du développement de l'Intelligence. Lorenz soutient, en effet, que:

"Il est plus que vraisemblable que l'ensemble de la pensée de l'homme a tiré son origine de ces opérations détachées de l'activité motrice et situées dans l'espace "représenté", et même que cette fonction originelle constitue la base de nos actes de pensée les plus élevés et les plus complexes" [Lorenz 70].

Le développement de la main préhensile chez les primates se déplaçant dans les arbres a joué un rôle essentiel dans la construction de cet espace "représenté" puisque la préhension ne pouvait, dans ce contexte, se satisfaire de mécanismes de type essai-erreur.

La modélisation de l'espace et le raisonnement sur ce modèle constituent à ce titre, un thème de recherche fondamental.

Nous nous intéressons, dans ce manuscrit au problème de **modélisation de l'espace** et au **développement conjoint de modèles de raisonnement pour la préhension** dans le contexte de la robotique d'assemblage.

La première partie de ce document nous permettra dans un premier temps de justifier plus formellement l'intérêt de notre travail en nous ressituant par rapport

aux recherches déjà effectuées dans le cadre de la Robotique intelligente. Nous évoquerons brièvement les problèmes de modélisation liés aux différentes composantes du robot intelligent puis définirons plus exactement les besoins en modélisation nécessaires au raisonnement sur l'espace; nous montrerons que les deux autres notions (temps et incertain) sont sous-jacentes à cette modélisation de l'espace. Nous présenterons ensuite un système de modélisation de l'univers matériel répondant autant que faire se peut à chacun de ces besoins.

Nous examinerons ensuite le problème de la saisie automatique d'un objet par un organe de préhension dans le contexte de réalisation d'une tâche donnée. Après avoir analysé les différents travaux sur la préhension dans le contexte de la Robotique, nous présenterons notre méthode. Nous examinerons ensuite ses limitations et proposerons quelques ouvertures.

Enfin, nous présenterons le système SHARP, travail collectif, destiné à intégrer les différents résultats de notre équipe dans le contexte de la programmation automatique des robots d'assemblage (c'est à dire la production automatique des programmes de commandes de robots à partir de spécifications de haut niveau de la tâche d'assemblage et de modèles de type CAO de l'environnement de manipulation).

L'intérêt pratique de développer de telles recherches est étroitement lié à l'évolution de l'industrie dans les dix dernières années.

Si l'automatisation a déjà fondamentalement évolué grâce à l'introduction dans l'industrie des robots programmables, l'enjeu économique actuel nécessite des produits à moindre coût, et dont la mise en production se fasse dans des délais plus courts (la durée de vie des produits diminuant constamment).

L'automatisation de la programmation des robots peut jouer un rôle déterminant puiqu'elle se traduirait par la production plus rapide de programmes de commande de robots plus fiables. La main d'œuvre serait en outre, moins nombreuse et moins spécialisée.

La résolution de ce problème constituera, de plus, un réel pas vers la Robotique autonome.

Chapitre 1

Rappel sur les niveaux de programmation en Robotique

1.1 Exemple introductif: Carnets de voyage

Empruntons la "quatrième dimension" et transportons nous dans un univers parallèle peuplé de robots intelligents (réellement intelligents). Approchons nous d'une de leurs habitations et observons: dans l'une d'elles un robot s'amuse au milieu d'un désordre tout à fait humain; il joue avec des pentacubes[1] qui lui ont été offerts par l'être humain Axle Munshine [Godard] au cours de l'un de ces voyages. Mais soudain, dans ce calme apparent, apparait un autre protagoniste au comble de l'excitation enjoignant au premier de ranger l'endroit; alors Maurice, c'est ainsi que nous l'appellerons, plein de tristesse range ses pentacubes dans leur boîte puis ferme la boîte.

Tournons ici cette page de vie d'une rare intensité émotionelle et essayons d'imaginer ce qui se passe dans la "tête" de Maurice (excusez cet anthropomorphisme outrancier) pour réussir cette tâche difficile.

Cet exemple est volontairement simpliste: Maurice évolue sans se servir de ses sens, dans un univers inerte (i.e. modifié par les seules actions de Maurice) dans lequel les notions d'incertain et de temps sont ignorées.

Nous supposerons, d'autre part, que Maurice raisonne de façon hiérarchique; nous n'aborderons donc pas le traitement des échecs pouvant survenir aux différents niveaux de raisonnement et pouvant remettre en cause des choix effectués à d'autres niveaux. Cette hypothèse simplificatrice facilitera notre exposé dont le but est d'évoquer les grandes lignes de recherche en Robotique Intelligente.

[1] Les pentacubes sont des objets consitués de cinq cubes élémentaires qu'il faut assembler (sans laisser d'espace vide) de façon à réaliser une figure géométrique donnée. L'équivalent 2D est le jeu des pentaminos (cf. figure 1.1).

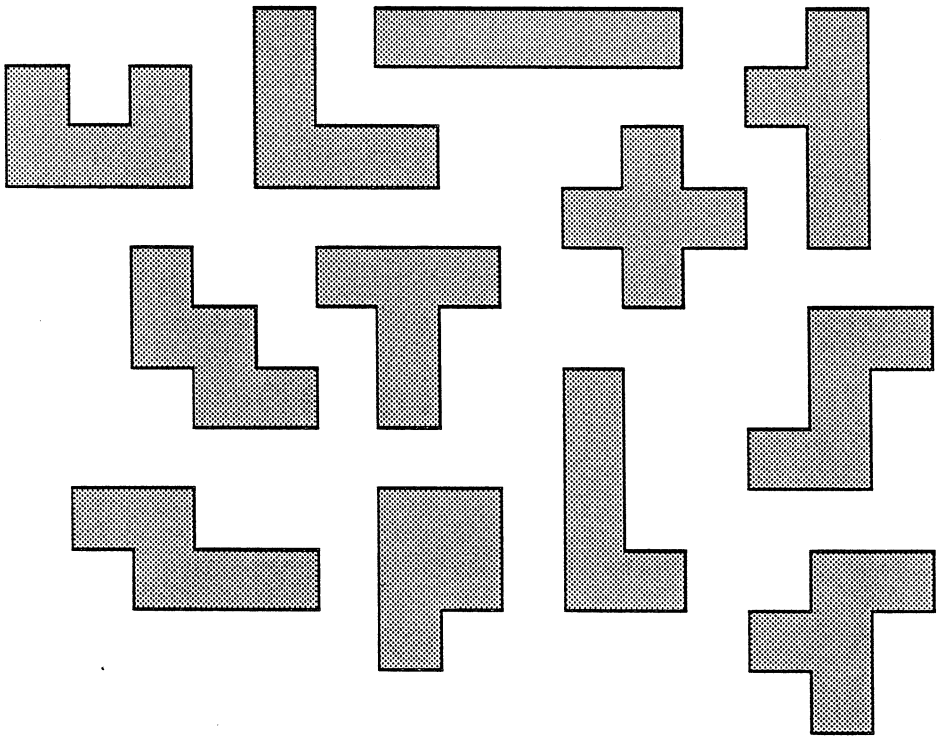


Figure 1.1: Les pentacubes

1.2 Formalisation des niveaux

On peut examiner ce problème sous un jour relativement général. Considèrons en effet, un agent manipulant les objets d'un univers donné et ayant à satisfaire un but (les objets manipulés peuvent évidemment être de natures très différentes: objets physiques, musicaux, mathématiques...).

Un but se traduit par un état de l'univers.

La satisfaction de ce but (donc la réalisation de cet état) se fait au moyen d'une séquence d'actions modifiant l'état de l'agent et par conséquent celui de l'univers.

Ce problème peut être décomposé en trois grandes étapes.

La première étape correspondant à un premier niveau de raisonnement concerne la **conception** d'un objet de l'univers répondant au but fixé.

Les deux étapes suivantes s'occupent de la réalisation de cet objet.

L'agent dispose en effet, après franchissement de la première étape, d'un modèle de la situation finale de l'univers dans lequel il évolue. Se pose alors, la question

du passage de l'état courant de cet univers à cet état final. Il faut donc, dans un premier temps, spécifier une suite d'états intermédiaires conduisant à l'état final et donc à la satisfaction du but. La construction de cette suite d'états est fondée sur un raisonnement sur des actions appliquées aux objets de l'univers. Il faut également choisir les paramètres des actions (si paramètres il y a). Cette seconde étape correspond à une phase de **planification**.

La troisième étape concerne la phase d'**exécution**. Elle correspond à un affinage des transitions déterminées par la planification. Ce niveau permet de passer des actions de "haut niveau" (actions caractérisées par leurs effets sur l'état de l'univers) à un ensemble équivalent d'actions de base (actions caractérisées par leurs effets sur l'état de l'agent).

Dans l'exemple introductif, pour lequel les actions de base sont des envois de consignes aux articulations des membres de Maurice, ces trois étapes se matérialisent par quatre niveaux (les deux niveaux intermédiaires correspondant à la phase de planification).

1/ niveau objectif: ranger les pentacubes

Maurice a dans un premier temps à concevoir un objet de l'univers physique répondant aux contraintes d'allocation spatiale exprimées par le but "ranger les pentacubes". Il doit donc manipuler volumes et formes pour déterminer l'organisation finale des pentacubes dans leur boîte c'est à dire l'état objectif de l'univers.

2/ niveau tâche: réaliser l'assemblage conçu (cf. figure 1.2)

Il faut ensuite déterminer les étapes de réalisation de l'état objectif de l'univers. Maurice doit donc déterminer une séquence d'actions lui permettant d'atteindre la situation finale à partir de la situation de désordre initial. Il raisonne alors sur des actions sur les objets de l'univers (par exemple prendre, transporter, positionner); il doit pour cela connaître les modalités d'application et les effets des actions, indépendamment de la façon de les exécuter.

3/ niveau objet: prendre le "I", amener le "I" en position, monter le "I", prendre le "W"...

Cette phase d'ordonnancement effectuée, Maurice doit se poser des questions plus concrètes concernant la paramètrisation des actions (puisque paramètres il y a): à savoir comment manipuler ses pentacubes? i.e. comment les saisir, selon quelle trajectoire les transporter, et comment les positionner dans la boite? Si les réponses potentielles à ces questions sont très nombreuses, le désordre et l'encombrement régnant à proximité de Maurice vont réduire considérablement ses possibilités de mouvement.

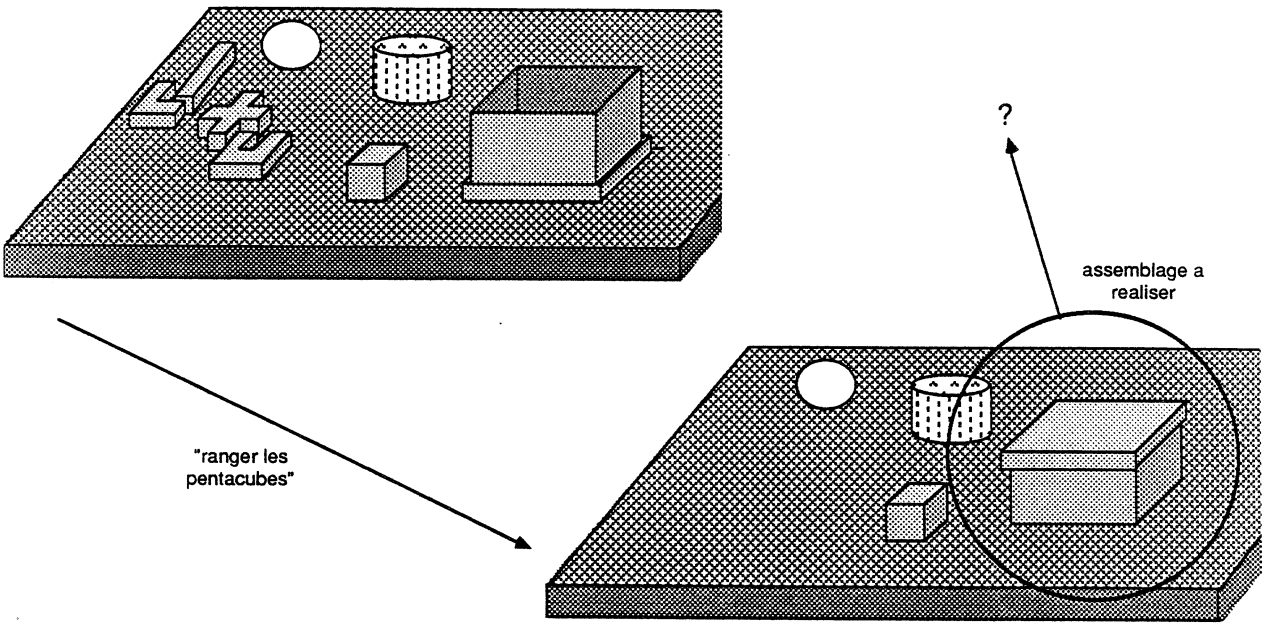


Figure 1.2: Assemblage à réaliser

Il doit, en effet, engendrer une suite de positions "sûres" (i.e. dont on est certains que la réalisation ne produira aucune collision). Pour cela, il lui faut connaitre parfaitement l'univers géométrique dans lequel il évolue et la sémantique géométrique des actions qu'il a planifiées.

4/ niveau agent: prendre le "I" par les faces f_1 et f_2 en arrivant verticalement, refermer les doigts jusqu'à tenir le "I", transporter le "I" de la position P_1 à la position P_2 en passant par les configurations $C_1, C_2, ..., C_n, ...$

Maurice doit maintenant déterminer plus précisément les trajectoires de sa main partiellement déterminées au niveau précédent, et les traduire en termes de commandes à envoyer aux articulations de ses membres.

A ce niveau, le raisonnement repose sur une modélisation de la cinématique et de la dynamique des membres de Maurice.

1.3 Bref état de l'Art par niveau

Le problème de conception évoqué au niveau objectif laisse une large part aux *processus créatifs*. S'il est étudié dans différents domaines (création musicale [Rodet 84], [Cadoz 84], mathématiques ...), c'est dans le domaine de la conception d'objets

physiques "utilitaires" que les recherches semblent les plus avancées. En effet, la conception repose dans ce cas sur un processus relativement rationnel. Différentes approches fondées sur des méthodes d'Intelligence Artificielle, sont ou ont été développées. La première consiste à raisonner sur la fonction des objets pour en déduire leur forme [Freeman 71], [Ingrand 84b], [Brady 84]. La seconde approche utilise le savoir-faire de la conception dans certains domaines particuliers [Latombe 77].

Le niveau tâche faisant appel aux *processus décisionnels*, correspond à un domaine de recherche très ouvert de l'Intelligence Artificielle: la génération de plans d'actions. Là encore deux approches prédominent. La première qui reste de loin la plus courante respose sur une modélisation logique du but et des actions (en termes de leurs conditions d'application et de leurs effets sur l'univers); cette modélisation est généralement faite grâce au formalisme de la logique du premier ordre. Cette approche regroupe tous les planificateurs de la famille STRIPS [Fikes 71]. Elle est adaptée lorsque l'univers se prète bien à une modélisation logique. La seconde approche repose sur une modélisation du plan d'actions lui-même et sur un raisonnement sur les conséquences du choix des actions sur ce plan. L'intérêt d'une telle approche a été démontrée dans le contexte de la fabrication mécanique [Lagoude 85].

Le niveau objet constitue en soit un terme de recherche fondamental en Robotique. Il s'agit en effet, de reproduire un **savoir-faire de la manipulation** . Après une première tentative relativement infructueuse de certains laboratoires vers les années 75 dûe à une approche globale du problème [Lozano 76], [Liebermann 77], [Finkel 74], les laboratoires se sont tournés vers la résolution de sous-problèmes de la programmation de niveau objet (principalement la saisie, le transport et le montage d'objets). A l'heure actuelle, l'effort se porte à nouveau sur la conception de systèmes de niveau objet (cf. chapitre 7).

Le niveau agent a pour but de *donner des capacités d'action et de perception* à l'agent. Il a vu deux axes principaux de recherche se développer parallèlement.

Le premier correspondant à la partie haute du niveau agent, répond à un besoin industriel pressant: l'automatisation flexible [Nitzan 76]. Des langages de manipulation (LM [Latombe 81b], AL [Finkel 74], VAL [Shimano 84], AML [Taylor 82]) ont ainsi vu le jour dans différents laboratoires. Ces langages permettent de commander un manipulateur en termes de déplacements et d'actions de son effecteur (ces langages sont dits de niveau effecteur), en laissant une large part à l'interaction avec des capteurs. Différents types de perception sont étudiées dans les laboratoires (toucher, vision...). Certains de ces langages (le langage LM développé au LIFIA, par exemple) connaissent un essor industriel.

La commande des manipulateurs (partie basse du niveau agent) fait encore l'objet de nombreuses recherches essentiellement tournées vers l'étude de la dynamique. Les actes de l'International Symposium of Robotics Research de l'année 85 [ISRR 85] sont représentatifs des recherches menées en la matière.

1.4 Objet de notre travail

Dans le contexte de la Robotique d'assemblage, la phase de planification est plus généralement connue sous le terme de **programmation automatique**. Elle regroupe une phase de planification des actions et une phase de planification de l'environnement. Ce dernier point a pour but de déterminer la configuration de l'espace de travail, les systèmes de maintien de pièces, d'amené/évacuation de pièces, les outils de manipulation... (cf. figure 1.3).

La construction d'un système de programmation automatique constitue un objectif à moyen (voire long) terme de l'équipe "Intelligence Artificielle et Robotique" du LIFIA. Nous nous intéressons dans ce paragraphe uniquement à la planification des actions (niveaux tâche et objet).

Si les premiers générateurs de plans d'actions ont été conçus pour être appliqués à la Robotique d'assemblage (STRIPS [Fikes 71]), ils ne se sont intéressés qu'au niveau tâche. Un univers physique simpliste et des possibilités d'actions restreintes permettent, en effet, d'avoir une même modélisation logique de l'univers et des actions et par là, de court-circuiter le niveau objet. Une telle approche n'est pas satisfaisante dans le contexte de l'assemblage pour lequel le choix des actions et leur ordonnancement dépend en grande partie de critères de plus bas niveau nécessitant une modélisation réaliste de l'univers physique. Elle s'est en outre soldée par de grandes difficultés lors de la connexion de STRIPS au robot SHAKEY.

Certains systèmes ont essayé d'aborder globalement le problème de la planification des actions en considérant simultanément les niveaux tâche et objet. Le système BUILD [Fahlmann 74], s'il opère dans le monde des blocs présente un aspect plus réaliste du point de vue physique que les générateurs de plans de la famille STRIPS. Les actions disponibles dans BUILD sont celles de STRIPS, mais BUILD permet de transporter des sous-assemblages composés d'objets autres que des cubes (cf. figure 1.4). Les buts sont des assemblages à réaliser.

BUILD se distingue des autres générateurs de plans du monde des blocs par une *analyse des effets logiques et physiques des actions*. Une étude de stabilité est en effet effectuée chaque fois que l'on décide de la création d'un sous-assemblage afin de déterminer si l'action est physiquement viable. Le système manipule donc simultanément une représentation logique et une représentation physique de l'univers et raisonne sur les deux niveaux pour effectuer le choix des actions.

BUILD traite cependant des problèmes très simples; les problèmes de saisie et d'encombrement spatial ne sont pas considérés et l'assemblage se fait dans un espace équivalent à un espace bi-dimensionnel dans lequel les effets physiques des actions sont encore aisément évaluables.

Nous pensons qu'un système de programmation automatique devra suivre une

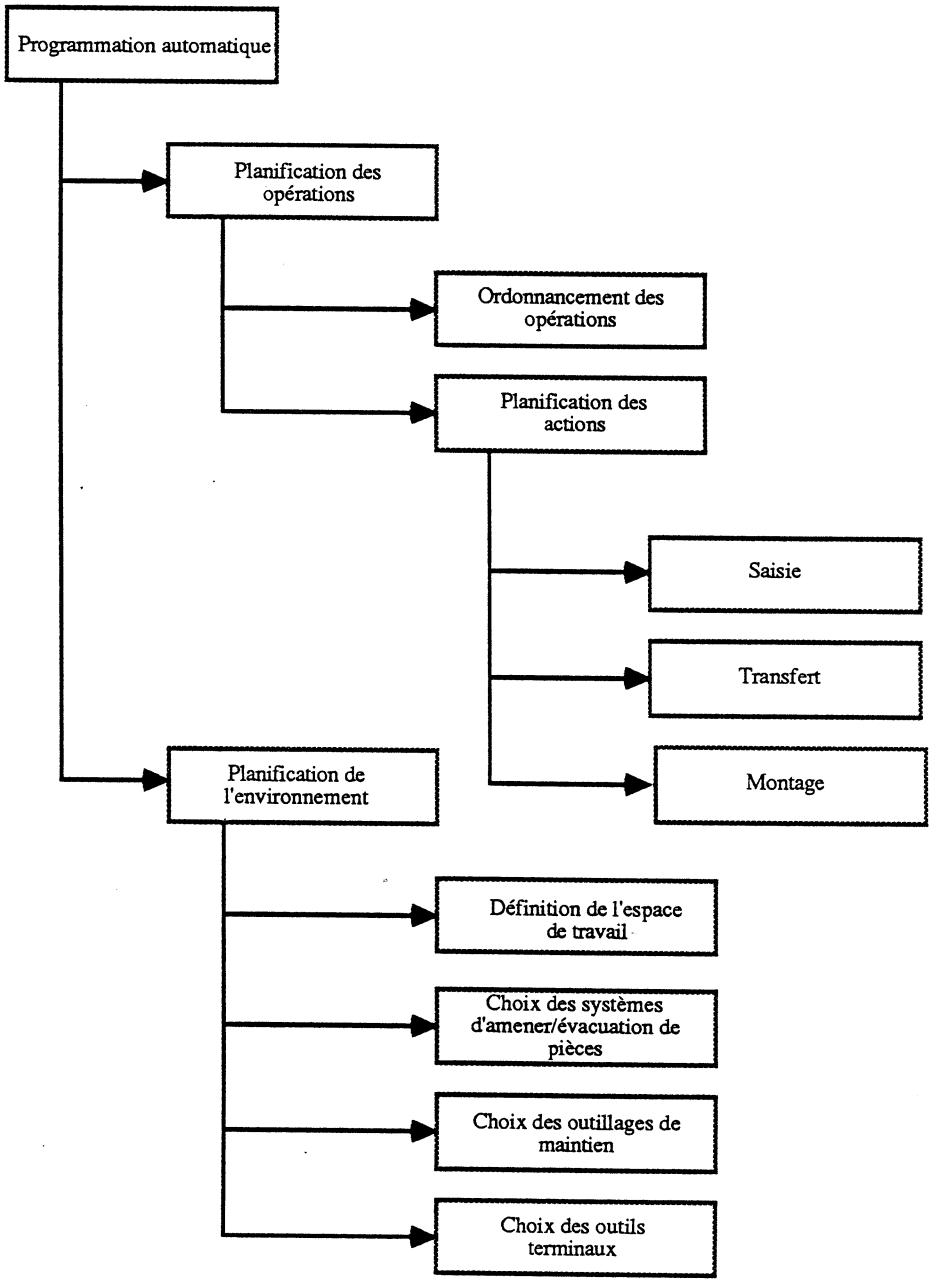


Figure 1.3: La programmation automatique

approche similaire à celle de BUILD; il ne pourra donc être conçu tant que la maitrise du niveau objet ne sera pas acquise. C'est pourquoi une partie des efforts du laboratoire LIFIA se porte sur la conception d'un système prototype de programmation de niveau objet, le système SHARP.

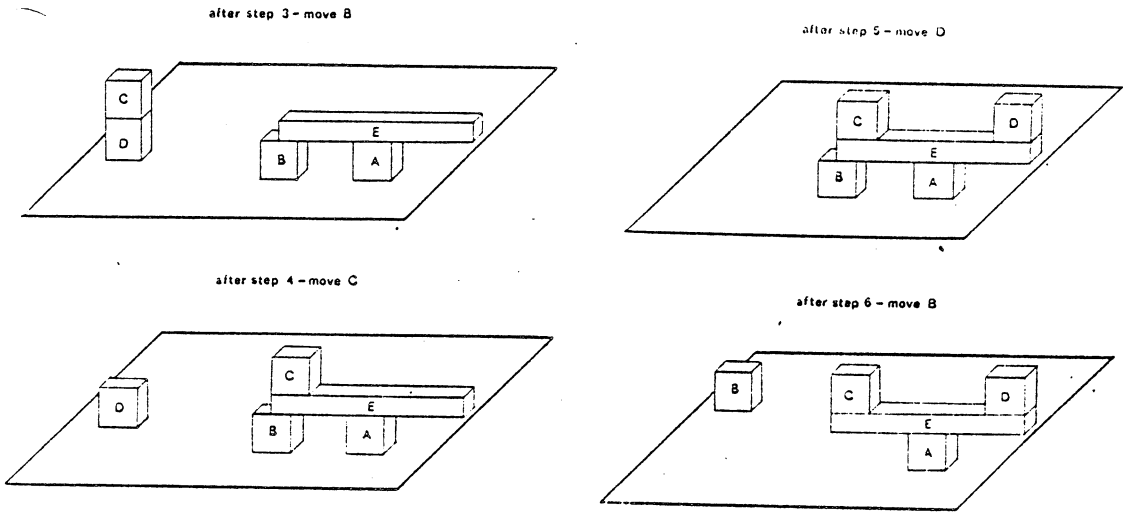


Figure 1.4: Assemblage manipulé par le système BUILD [Fahlmann 74]

Le développement d'outils de modélisation adéquats est essentiel pour pouvoir conduire le **raisonnement géométrique et spatial** inhérent à ce niveau de programmation. Il ne se fait pas uniquement sur des **configurations statiques de l'univers physique**, mais aussi sur les transitions existant entre deux états successifs déterminés au niveau tâche. Cet **état évolutif** (ou "dynamique") **de l'univers** est l'un des points clé de la modélisation au niveau objet.

Nous présentons dans les deux chapitres suivants, une proposition de système de modélisation de niveau objet et sa matérialisation: le système SMGR.

Chapitre 2

Besoins en modélisation pour la programmation de niveau objet

2.1 La problèmatique de niveau objet

Considèrons le problème général d'un montage d'assemblage; comme nous l'avons vu brièvement, ce montage va être décomposé en une séquence d'actions permettant d'atteindre des états intermédiaires matérialisés par des objets intermédiaires appelés sous-assemblages. Bien que cette hypothèse ne soit pas toujours valide dans le contexte général de l'assemblage, nous supposerons que le passage d'un sous-assemblage à son successeur est fait par un seul manipulateur opérant sur un objet unique (éventuellement un sous-assemblage).

Un système de programmation de niveau objet a pour but de planifier l'exécution des actions du manipulateur. Il aura en entrée une description de la tâche à effectuer sous forme de la séquence de ces états intermédiaires. La description d'un état est généralement faite par un ensemble de relations de montage entre les objets concernés (cf. figure 2.1).

Le résultat attendu est une suite d'actions exécutables par un manipulateur (déplacements et actionnements d'outils) éventuellement guidées par des données de capteurs: figure 2.2.

Le problème de la réalisation de chacune de ces transitions entre les états intermédiaires nécessite de résoudre trois sous-problèmes résultant de la décomposition de la transition en transitions plus fines: choix de la prise de l'objet à monter, choix de la trajectoire de transfert de la pièce de sa position initiale à une position proche de sa position finale, synthèse des mouvements de mise en correspondance de la pièce sur le sous-assemblage en place.

La recherche de prise doit respecter des contraintes fortes d'accessibilité; la saisie doit se faire sans engendrer de collisions avec l'environnement ou avec des parties de

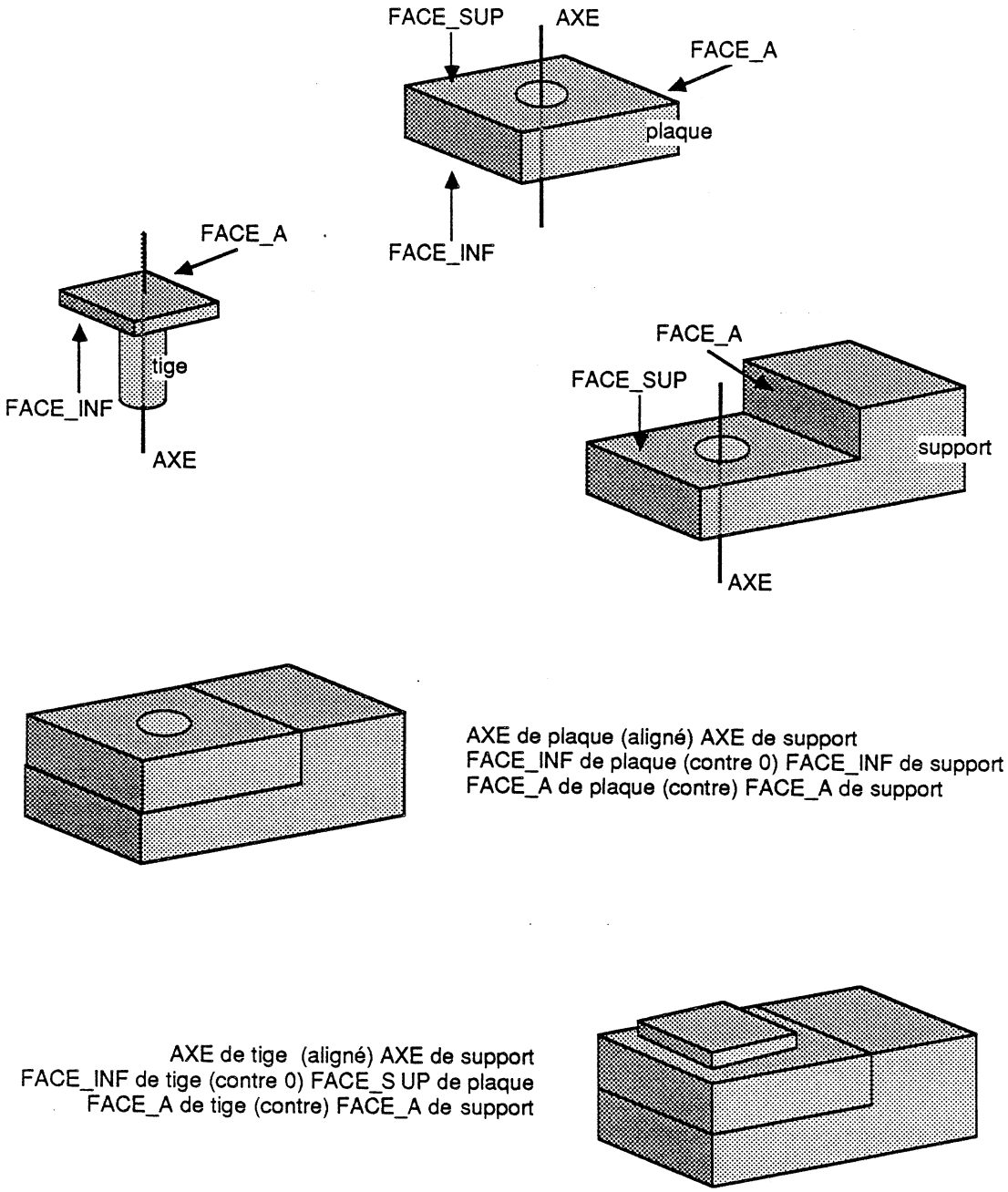


Figure 2.1: Exemple de spécification de niveau objet en LM-Géo

```
PROGRAMME MONTAGE;
CO Le repere ROBOT est situe en bout de pince et les valeurs de
     variables d'etat MX, MY, MZ, FX, FZ sont ramenees en ce point;
CO Declarations;
REPERE SUPPORT, FACE_A_DE_SUPPORT, PLAQUE, FACE_INF_DE_PLAQUE,
            APPROCHE_SUPPORT, TIGE, INIT;
REEL SEUIL_FX, SEUIL_FZ, SEUIL_CONTACT, EPS;
BOOLEEN MIS_EN_PLACE;
EXT PROCEDURE SAISIE (REPERE, REEL);
DEBUT
CO Une partie des initialisations;
SEUIL_FX:=4.; SEUIL_FZ:=2.; SEUIL_CONTACT:=5.; EPS:=10.;
APPROCHE_SUPPORT:=FACE_A_DE_SUPPORT*TRANSLAT(VX,10.);
FIXER VITESSE A 0.8;
INIT:=ROBOT;
CO Saisie et positionnement de la plaque;
SAISIE (PLAQUE,70.);
LIER PLAQUE A ROBOT;
DEPLACER PLAQUE A APPROCHE_SUPPORT*TRANSLAT(VZ,10.);
FIXER VITESSE A 0.05;
MIS_EN_PLACE:=FAUX;
TANTQUE NON MIS_EN_PLACE FAIRE
        DEPLACER PLAQUE A APPROCHE_SUPPORT*TRANSLAT(VZ,-10.)
                    JUSQUA FZ > SEUIL_FZ;
        CO Test sur MX et correction de mouvement;
        SI MX > EPS ALORS DEPLACER PLAQUE DE ROT(VX,PI/2)
                    EN MAINTENANT
                    FZ = SEUIL_CONTACT JUSQUA MX < EPS;
        CO Meme traitement pour MY;
        SI FZ > SEUIL_FZ ALORS MIS_EN_PLACE:=VRAI;
FINFAIRE;
MIS_EN_PLACE:=FAUX;
TANTQUE NON MIS_EN_PLACE FAIRE
        DEPLACER PLAQUE A FACE_A_DE_SUPPORT*TRANSLAT(VX,-10.)
                    EN MAINTENANT FZ=SEUIL_CONTACT,MX=0,MY=0
                    JUSQUA FX > SEUIL_FX;
        CO Test sur MZ et correction de mouvement (cf. MX);
        SI FX > SEUIL_FX ALORS MIS_EN_PLACE:=VRAI;
FINFAIRE;
OUVRIR PINCE SANS ATTENTE;
DELIER PLAQUE DE ROBOT;
DEPLACER ROBOT DE TRANSLAT(VZ,100.);
FIXER VITESSE A 0.8;
DEPLACER ROBOT A INIT;
CO Saisie et insertion de la tige ;
...
FIN;
```

Figure 2.2: Programme LM correspondant à la spécification précédente

l'objet saisi lui-même. La prise choisie doit en outre, permettre le positionnement de l'objet sur le sous-assemblage. Des contraintes plus faibles portant sur la stabilité de la prise (i.e. de l'objet saisi dans l'outil) doivent également être considérées de façon à introduire aussi peu d'incertitude que possible sur la position de l'objet lors de la saisie.

Les trajectoires de transfert doivent être calculées en respectant les contraintes d'encombrement spatial de l'environnement; elles doivent être aussi optimales que possible (distance, rapidité, précision) tout en tenant compte des caractéristiques du manipulateur considéré (espace atteignable, précision des différents degrés de liberté...).

Les mouvements fins de montage doivent être engendrés de façon à assurer la réalisation de l'état final de la transition en dépit des erreurs de positionnement des objets et de l'imprécision du manipulateur. La recherche de stratégies de perception ou la génération de mouvements compliants est donc un complément essentiel de la recherche de stratégies de montage.

La solution de ces trois sous-problèmes passe par la résolution de deux problèmes plus fondamentaux:

- le **calcul de positions sûres pour un objet articulé évoluant dans un univers encombré d'obstacles**: il s'agit de déterminer les positions par lesquelles le robot doit passer pour accomplir la tâche spécifiée (positions de prise des objets, trajectoires de transport des objets, trajectoires de montage).
- la **prise en compte des incertitudes géométriques sur les positions des pièces et du manipulateur**: lorsque celles-ci sont supérieures aux jeux de montage autorisés, il s'agit d'élaborer des stratégies de manipulation (perception, compliance) permettant de réduire les incertitudes au dessous des jeux autorisés.

A la complexité intrinsèque de ces deux points essentiels, s'ajoutent des problèmes d'interdépendances. Ainsi, le choix d'une prise stable pourra éviter l'emploi ultérieur de capteurs; d'autre part, elle influencera grandement le choix d'une stratégie de montage. De même, le choix de la prise est conditionné par les relations d'assemblage à établir (une face devant être mise en contact avec une partie du sous-assemblage ne peut pas participer à une prise); certaines prises pourront également être mieux adaptées à l'exercice de certaines forces nécessaires à la réalisation de l'assemblage.

Au vue de ces quelques remarques introductives, nous pouvons esquisser l'architecture d'un système de niveau objet (cf. figure 2.3).

Trois modules de planification permettront de résoudre les trois problèmes d'accessibilité (saisie, transport, mouvements fins). Ces modules s'appuieront sur un

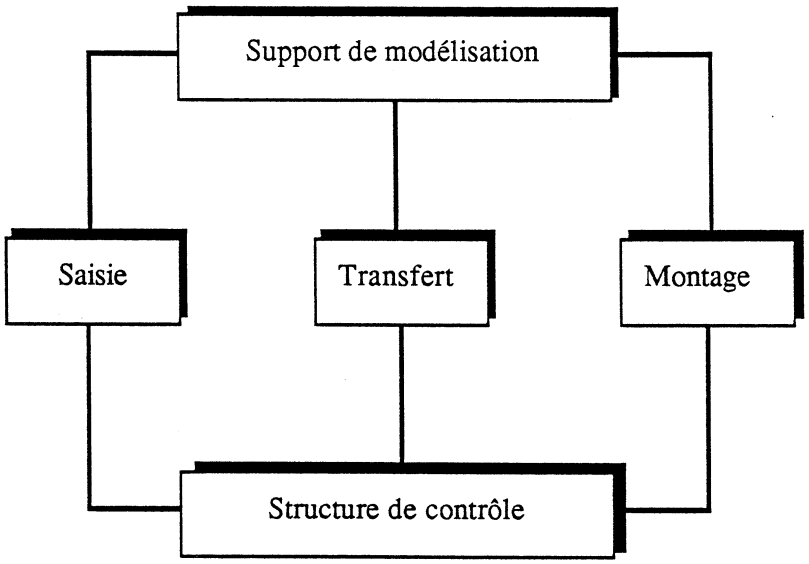


Figure 2.3: Architecture simplifiée d'un système de niveau objet

support de modélisation dont les grandes fonctions sont présentées au paragraphe suivant. Ils seront activés par un module de contrôle permettant de propager à la fois les contraintes d'accessibilité et les contraintes d'incertitude et par là de traiter les problèmes d'interdépendance entre les différents modules. Nous reviendrons sur cette architecture dans le chapitre 7.

2.2 Les besoins en modélisation

2.2.1 Un univers géométrique

Les trois sous-problèmes de la programmation de niveau objet nécessitent des méthodes de recherche de trajectoires sans collision. Cette recherche peut se faire dans un univers de manipulation très contraint; c'est le cas de la saisie et des mouvements fins d'assemblage. Elle peut également se dérouler dans un univers peu contraint comme dans le cas des mouvements de transfert. Ces différents degrés de contraintes imposées par le contexte conduisent à des méthodes de résolution différentes ayant en commun un raisonnement géométrique et spatial.

Il s'agit dans le premier cas de raisonner sur la géométrie et la morphologie des objets et de l'outil de préhension; dans le second cas, il faut raisonner sur l'encombrement spatial de l'univers, en utilisant par exemple une représentation

explicite de l'espace libre. Un support de modélisation dans un système de niveau objet doit permettre ces deux types de raisonnement très différents.

2.2.2 Un univers physique

Les raisonnements géométriques mis en jeu ne seraient pas complets si l'on ne tenait pas compte de la physique de l'univers dans lequel on évolue. Il est souhaitable, par exemple, de pouvoir modéliser la mobilité d'un objet ou d'avoir les informations permettant de prévoir un glissement potentiel d'un objet dans les mors de la pince. Pour cela, nous devrons connaitre les caractéristiques physiques des objets manipulés (centre de gravité, masse, moments d'inertie, coefficients de friction des matériaux, description des états de surface...).

Dans le cas de la manipulation de sous-assemblages non fixés (par exemple, des cubes superposés), ces informations seront essentielles aux planificateurs de trajectoires; ils devront, en effet, tenir compte des contraintes (essentiellement dûes à la pesanteur) liant les objets du sous-assemblage, dans leur recherche d'une solution; la trajectoire choisie devra assurer l'existence du sous-assemblage tout au long de l'exécution de l'action planifiée.

2.2.3 Un univers incertain

Il faut aussi, dans un système de modélisation cherchant à représenter un univers physique, prendre en compte l'incertain sous toutes ses formes: incertitude sur les dimensions des objets, déplacements imprécis du manipulateur, mesures de capteurs entâchées d'erreur... Le raisonnement sur cet incertain permet, par l'utilisation appropriée des capteurs, de produire des programmes de manipulation utilisables en univers réel.

Deux types d'incertitude intervenant à différents niveaux dans la programmation automatique nous semblent essentiels: l'incertitude de forme (ou **tolérance**) directement liée aux problèmes de dispersion de fabrication et l'incertitude de localisation (ou **incertitude**) dûe aux imprécisions des opérations de manipulation et de localisation.

La première intervient assez localement au niveau de la programmation des opérations de saisie ou d'assemblage ou pour la planification de mouvements à compliance. Elle peut revêtir trois formes:

- tolérance sur le dimensionnement des objets
- tolérance sur la forme des objets permettant de traduire les imperfections de surface, défauts de parallélisme...
- tolérance sur la structure topologique des objets puisque une arète vive n'existe pas en fabrication mécanique.

La seconde intervient à tous les niveaux de la planification et c'est de sa bonne évaluation que dépend le succès d'un programme de manipulation. Elle porte sur la position et l'orientation des objets.

2.2.4 Un univers en évolution

La programmation de niveau objet spécifie une succession d'états de l'univers. Un support de modélisation doit donc permettre une représentation des évolutions de l'univers physique. Ainsi, en plus des caractéristiques géométriques et physiques de l'objet, toute information le concernant aura pour but de spécifier un état de l'univers. Cet état est essentiellement un état relationnel représentant des classes d'objets réunis dans des structures de natures diverses (permanentes ou temporaires, rigides ou non). Une structure rigide temporaire pourra, par exemple, être formée par le manipulateur et un objet saisi, le temps du transport de cet objet. Une autre structure pourra représenter le lien gravitationnel entre un objet et le support sur lequel il est posé. Une structure permanente non rigide pourra être constituée par les liens entre les différents constituants d'un manipulateur.

2.3 Insuffisance des modèles actuels

2.3.1 Modèles de type CFAO

Nous ne présenterons pas ici l'historique de la CAO (cf. figure 2.4) mais certaines étapes de son évolution, significatives par rapport à nos problèmes de modélisation. Ainsi, si les premiers travaux dans le domaine visaient uniquement à représenter les objets 3D par des objets unifilaires 2D (les "wireframes"), un premier pas vers des modèles évolués a été franchi avec la réalisation du sytème BUILD [Braid 73] dans lequel étaient incluses les notions de volume et de positionnement de la matière. BUILD fut le précurseur de nombreux systèmes actuels (Build2, Romulus, Design... [Hillyard 82]). Une seconde étape aussi importante fut l'évolution de la CAO vers la CFAO enrichissant par là la nature des informations présentes dans les modèles pour donner aux objets manipulés un caractère fonctionnel ("features" [Descotte 81], [Henderson 84a], [Henderson 84b]) absent jusque là. Un exemple de tel système est BUILD-4 [Jared 83].

Ce type de modélisation ne nous suffit pas, essentiellement parce qu'il est tourné vers la représentation d'objets individuels. Certaines des représentations utilisées dans de tels systèmes semblent cependant bien adaptées à la description analytique de l'univers géométrique.

Examinons rapidement les principales représentations utilisées dans les systèmes de CAO [Requicha 80], [Requicha 79]:

1. les modèles génériques (cf. ACRONYM [Brooks 79] qui apportent une description symbolique concise sur la forme des objets) basés sur la notion de famille d'objets (cf. figure 2.5),

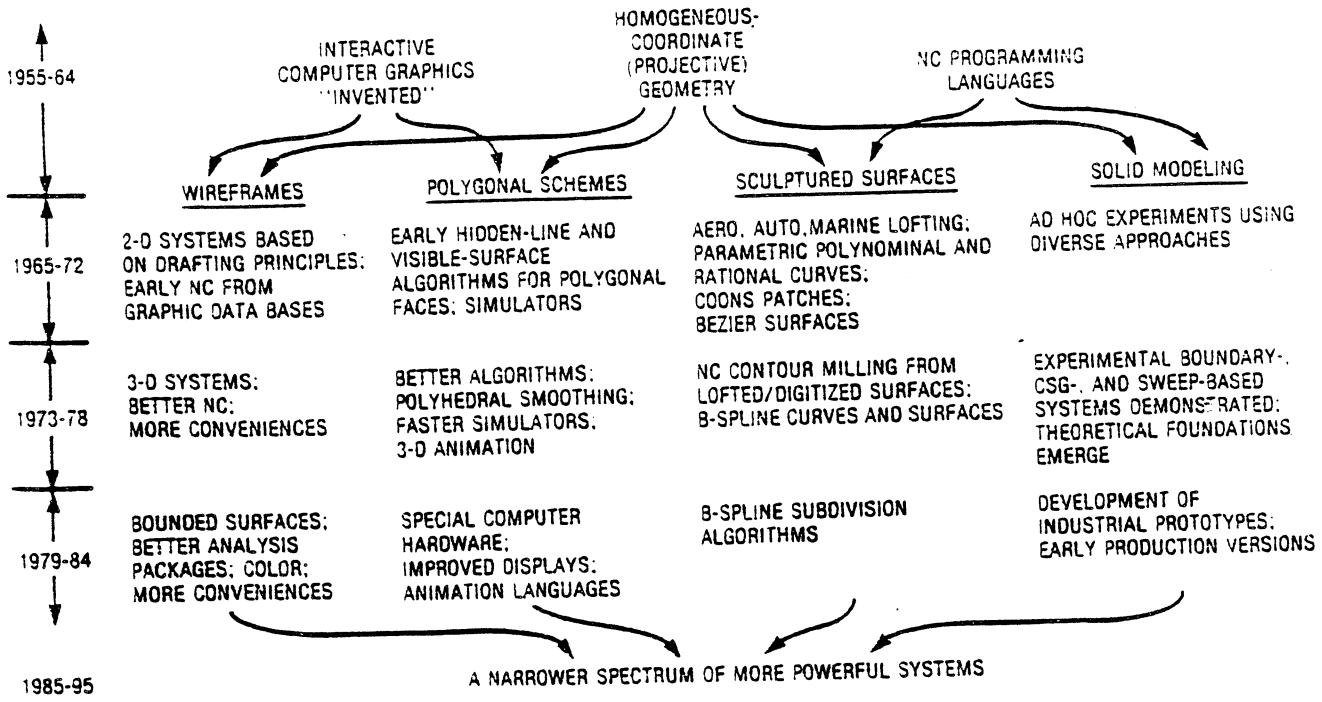


Figure 2.4: Historique de la CAO [Requicha Voelcker 77]

2. les modèles basés sur l'énumération de l'espace occupé, cet espace étant généralement décomposé en cubes de taille fixe (application en architecture, en tomographie) (cf. figure 2.6),

3. les modèles basés sur une décomposition des objets en cellules, ces cellules pouvant être obtenues par division récursive des volumes (par exemple les octrees [Meagher 81] - cf. figure 2.7) ou bien par triangulation,

4. les modèles par représentation des frontières (B.R.),

5. les modèles basés sur une construction ensembliste (C.S.G.) des objets (cf. PADL [Voelcker 78]),

6. les modèles basés sur une représentation des objets obtenus par balayage (représentation en cônes généralisés dans ACRONYM ou par espaces fibrés [Boulle 83].

Parmi ces six grandes familles, seules les trois dernières sont principalement utilisées:

1. la **"Boundary representation (B.R.)"** ou représentation par les frontières dans laquelle un objet est représenté par son enveloppe extérieure, sous la forme d'une liste de faces (planes, cylindriques...); chaque face est décrite par

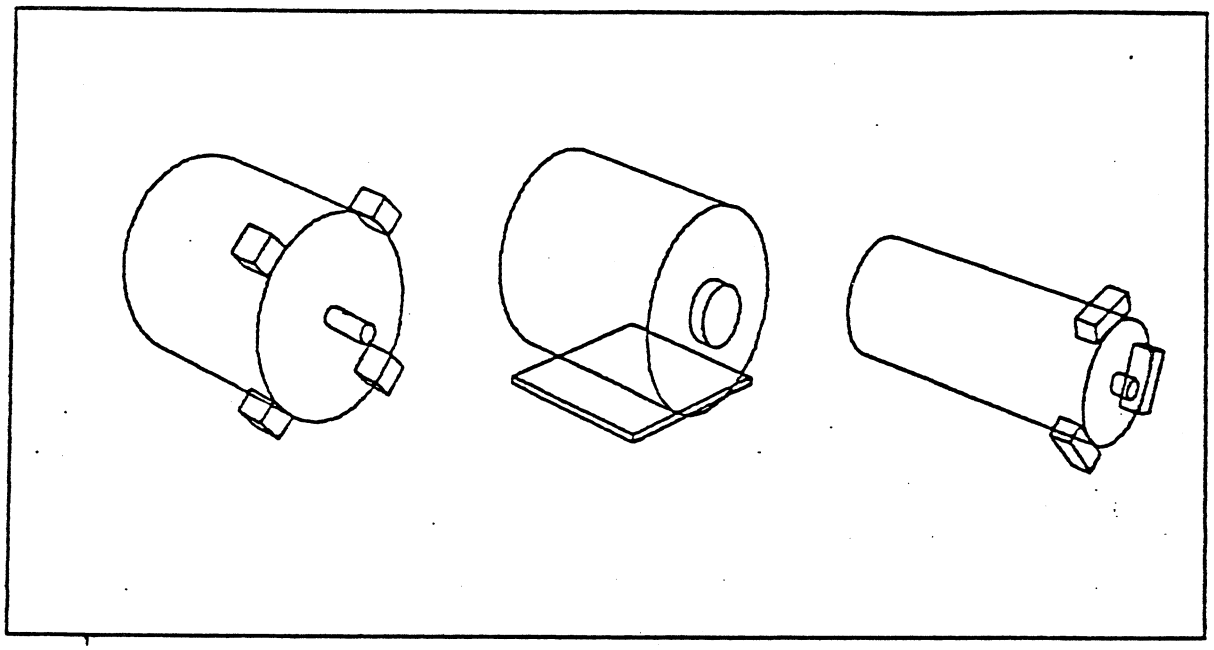
```
Node: GENERIC_ELECTRIC_MOTOR_CONE
    CLASS:              SIMPLE_CONE
    SPINE:              Z0014
    SWEEPING_RULE:      CONSTANT_SWEEPING_RULE
    CROSS_SECTION:      Z0013

        Node: Z0014
            CLASS:      SPINE
            TYPE:       STRAIGHT
            LENGTH:     MOTOR_LENGTH

        Node: CONSTANT_SWEEPING_RULE
            CLASS:      SWEEPING_RULE
            TYPE:       CONSTANT

        Node: Z0013
            CLASS:      CROSS_SECTION
            TYPE:       CIRCLE
            RADIUS:     MOTOR_RADIUS
```

A generalized cone model of a generic electric motor body.



Three specializations of the generic class of small electric motors.

Figure 2.5: La modélisation dans le système ACRONYM [Brooks 79]

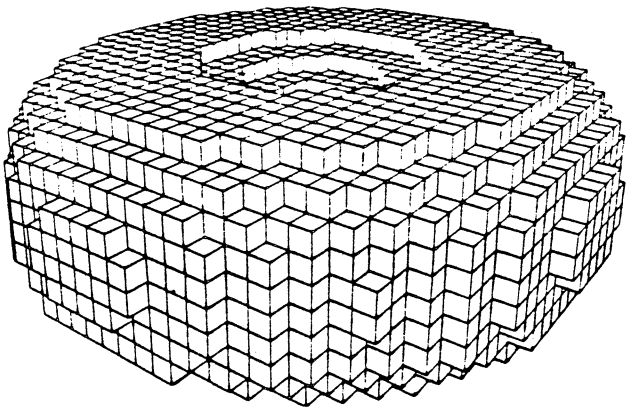


Fig. 9.17 A solid (the shape of a human red blood cell) approximated by a volume occupancy array.

Figure 2.6: Modélisation par énumération de l'espace occupé [Ballard]

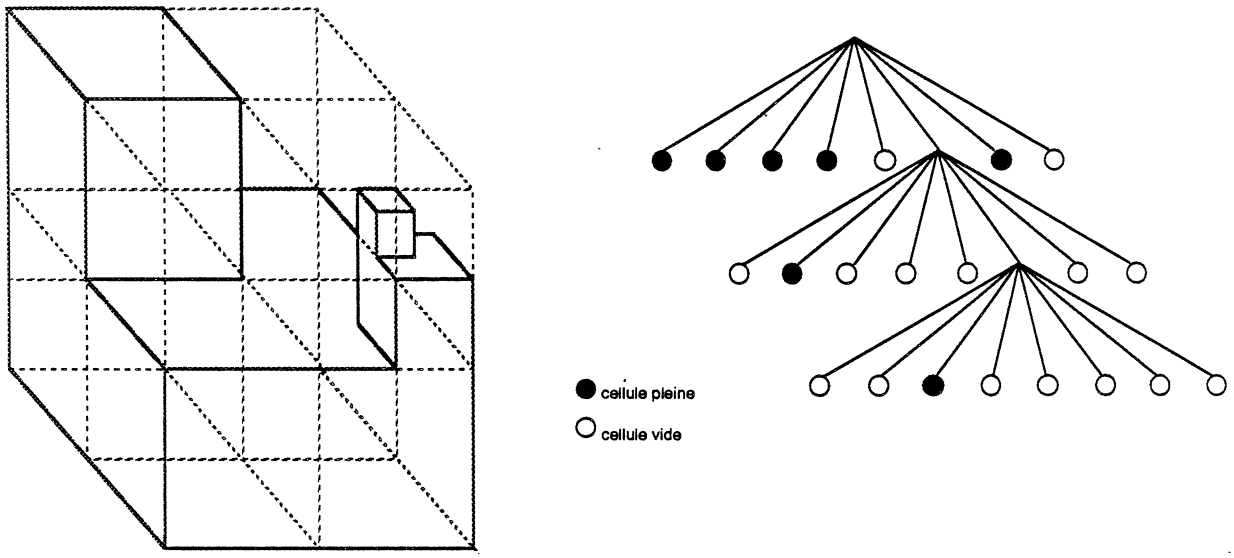


Figure 2.7: Représentation par octrees

les arètes qui la délimitent (droites, circulaires...) et des paramètres visant à mieux définir la position de la matière par rapport à cette face (généralement la normale extérieure à la face). De même chaque arète est définie par ses extremités. Cette représentation des objets est donc fondée sur un découpage de l'enveloppe de l'objet en entités, sur la description de ces entités et sur la localisation de la matière par rapport à chaque entité (cf. figure 2.8).

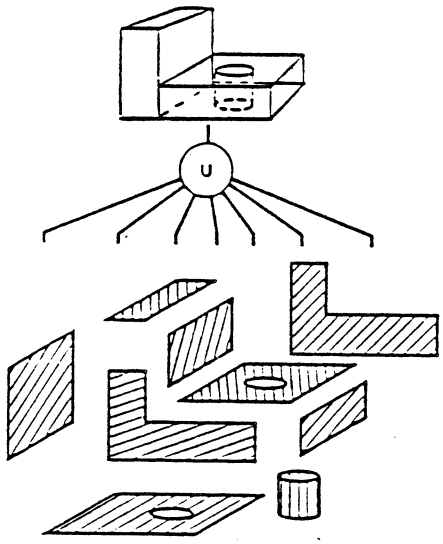


Figure 2.8: Représentation de type BR [Requicha Voelcker 82]

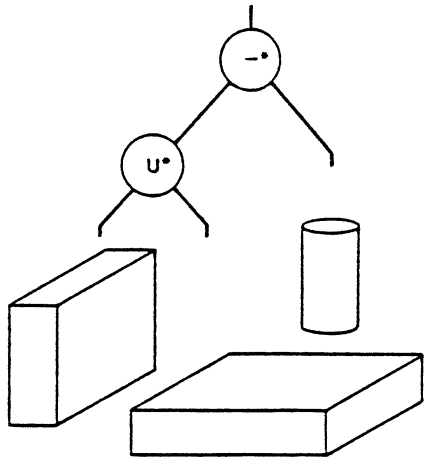


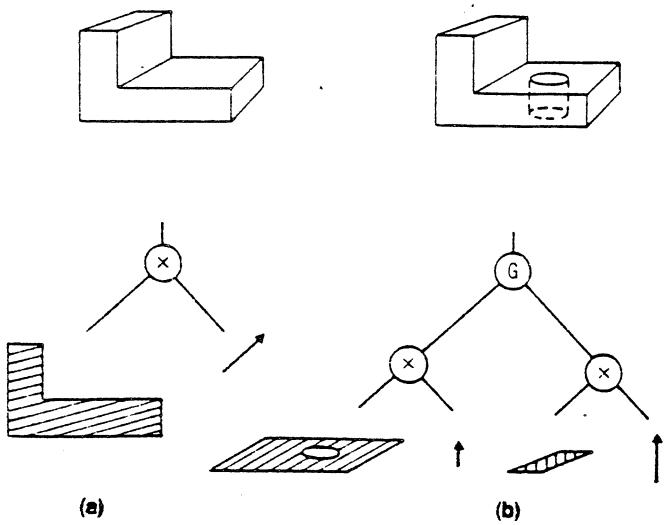
Figure 2.9: Représentation de type CSG [Requicha Voelcker 82]

2. la représentation de type **Constructive solid geometry** (**C.S.G.**) qui consiste à définir un objet comme une suite ordonnée d'intersections, d'unions et de différences de sous-objets. Une base de données de volumes élementaires permet de définir cônes, cylindres, cubes etc... Ces volumes élementaires sont géneralement décrits grâce à la représentation B.R. (cf. figure 2.9).

3. la représentation par cônes généralisés (ou **balayage**) dans laquelle les objets (ensemble solide de points) sont représentés comme un produit cartésien d'une surface de base et d'une trajectoire. Cette représentation est souvent completée par des opérateurs de collage et de perçage des volumes (balayage hybride) (cf. figure 2.10).

Certaines de ces représentations sont bien adaptées à la définition d'objets (CSG, balayage); d'autres sont idéales pour la manipulation d'objets géométriques complexes (B.R. par exemple). Un des intérêts d'utiliser de telles représentation est que leurs propriétés formelles (validité, unicité, complétude...) et informelles (concision, efficacité dans un domaine...), ainsi que les problèmes de conversion entre elles ont déjà fait l'objet de recherches approfondies [Requicha 80].

Les modèles issus du développement de la CFAO ont considérablement enrichi ces représentations permettant d'extraire centres de gravité, moments d'inertie, états de surface, toutes caractéristiques physiques utiles à la fabrication des pièces



Pure (a) and hybrid (b) sweep representations.

Figure 2.10: Représentation par balayage hybride [Requicha Voelcker 82]

mécaniques.

Certains systèmes de CFAO incluent également une représentation des tolérances de fabrication (PADL [Voelcker 77],[Voelcker 78], [Requicha 79], ROMULUS [Romulus]...). Cette modélisation est évidente du fait de l'utilisation classique des tolérances en fabrication mécanique. Elle nous semble assez difficilement transposable au problème de la robotique pour lequel son utilisation est moins claire. Requicha [Requicha 83] a tenté de formaliser les tolérances pour les systèmes de CFAO (fabrication, assemblage...). Cette formalisation est très intéressante en tant que telle mais a tous les défauts d'une théorie élaborée sans application pratique précise en vue. Elle ne tient, par exemple, aucun compte des besoins en Robotique et nous semble de ce fait totalement inutilisable dans le contexte de l'assemblage automatisé.

Il est clair que la notion d'évolution est généralement absente des systèmes de CAO tournés vers la seule modélisation d'objets individuels. Certains modèles CFAO ont cependant été "étendus" à la Robotique afin de pouvoir modéliser des manipulateurs et de fournir des outils d'aide à la mise en place de cellules flexibles; c'est le cas des systèmes CATIA [Borrel 83] ou GMSolid [Boyse 82], [Tilove 83]. Malheureusement, ces modèles restent assez pauvres du point de vue de la modélisation "relationnelle" (limitée à la représentation de mécanismes) et sont plutôt adaptés à une Robotique de "bas niveau" – simulation et apprentissage graphiques –, par opposition à la programmation de niveau objet.

2.3.2 Modèles développés pour la programmation de niveau objet

En ce qui concerne les systèmes de niveau objet, deux attitudes s'opposent; la première consiste à ne pas se poser le problème de l'intégration au niveau des moyens de modélisation; c'est par exemple, le cas du système ATLAS [Lozano 84] dont chacun des modules utilise une représentation adéquate généralement assez simple (polyèdres essentiellement).

Plus précisément, les planificateurs de la saisie et des mouvements fins se sont généralement contentés de représentations simples inspirées des modèles CAO: des polyèdres, cylindres décrit grâce à une représentation de type BR.

La planification de trajectoires de transfert d'un objet a fait l'objet de deux types d'approche:

- La première consiste à représenter les obstacles au déplacement du mobile. Elle a conduit à utiliser des modèles classiques de type polyédrique ou de type "octrees" (cf. [Maruyama 72], [Lozano 79] ou [Ahuja 80]).
- La seconde est fondée sur la notion d' "espace libre". Elle a conduit à l'implantation de modèles de type "pavage de l'espace" consistant à représenter explicitement l'espace en termes de cellules libres et de cellules entièrement ou partiellement occupées [Lozano 81]. Un autre type de représentation fondé sur un découpage de l'espace libre en cônes généralisés a été introduit par Brooks pour la recherche de trajectoires sans collision [Brooks 83].

L'un des défauts majeurs de tels modèles est d'être conçus dans le cadre d'une application spécifique donc difficilement utilisable à d'autres fins. Notre problème requiert une plus grande "généralité"; le modèle utilisé devant cependant fournir de façon directe ou non, des informations plus spécifiques.

A l'inverse de cette première tendance, les concepteurs du système AUTOPASS [Liebermann 77] se sont efforcés de développer un support de modélisation adapté à la programmation de niveau objet. Le système résultant, GDP, est cependant assez simpliste au niveau de la modélisation géométrique; ceci vient du fait que les efforts se sont essentiellement portés sur l'"aspect global" d'un système de niveau objet sans examiner en détail les différents sous-problèmes de la programmation de niveau objet; GDP est de ce fait tourné vers l'aspect évolutif de la modélisation de niveau objet.

En ce qui concerne la modélisation de l'incertain en Robotique, très peu de choses existent. Le système ACRONYM [Brooks 79] développé dans le domaine de la vision par ordinateur permet de décrire des tolérances sur la taille, la position et la structure des objets. Ces informations sont tout à fait adaptées aux processus de reconnaissance propres à la vision.

Dans le cadre du système de programmation de niveau objet ATLAS, Brooks [Brooks 82b] s'est intéressé au développement d'un modèle de propagation des incertitudes de positionnement pour la programmation en Robotique; ce modèle est basé sur le calcul symbolique. Il ne s'appuie sur aucun système de modélisation ce qui alourdit considérablement les calculs. Par exemple, la position d'un objet n'est pas considérée comme un repère et nécessite l'introduction de nombreuses variables, donc d'équations supplémentaires pour la représentation de contraintes.

Dans le cadre des systèmes de niveau objet, seul le système GDP [Wesley 80] servant de support de modélisation à AUTOPASS, permet de modéliser et de mettre à jour les relations intervenant dans les différents états de l'univers de manipulation. Cette modélisation est basée sur l'utilisation d'une structure de graphe dont chaque nœud est un objet ou un sous-assemblage et dont les arcs portent des relations (mouvements relatifs, relations d'assemblage...). Malheureusement, à notre connaissance, le système AUTOPASS étant resté à l'état de spécification, la validité de GDP reste théorique.

Chapitre 3

SMGR: un système de modélisation pour la Robotique

Dans ce chapitre nous décrivons le système SMGR qui a été conçu et réalisé pour servir de support de modélisation au système SHARP de programmation de niveau objet.

3.1 Structure de la modélisation dans SMGR

Nous avons vu au chapitre 2 que les besoins en modélisation pour la programmation de niveau objet étaient nombreux et différents. Chaque besoin peut être localement satisfait soit dans les modèles CFAO, soit dans des représentations spécifiques d'une application particulière. Le développement d'un système de niveau objet passe par la conception d'un support de modélisation répondant aussi bien que possible à *tous* ces besoins. C'est l'objectif du système SMGR: Système de Modélisation Géométrique et Relationnelle pour la Robotique.

Il nous faudra pour cela faire un compromis entre généralité du modèle et efficacité des algorithmes de traitement en nous orientant vers un **modèle multi-représentation**. Cette multi-représentation répondra aussi bien à la diversité de nature des informations à modéliser qu'à celle des traitements mis en œuvre.

SMGR peut être décomposé en deux parties principales. La première s'attache à décrire les solides présents dans l'univers; ce sont des objets élémentaires: non constitués d'élements en relation (articulaire ou autre). A ces solides sont attachés un ensemble de caractéristiques physiques. Nous appelons ce modéle, **modèle géométrique des objets**. Il est basé sur l'utilisation combinée d'une représentation générale et de représentations plus spécifiques.

SMGR comporte également un **modèle relationnel de l'univers** qui permet de rassembler les objets de cet univers dans des structures de natures diverses, permanentes ou temporaires. Ce modèle est principalement décrit au moyen d'une

arborescence dans laquelle les nœuds représentent des objets et les arcs dénotent une relation d'appartenance. Les objets élémentaires constituent les feuilles de cette arborescence.

Bien que nous pensions qu'il soit utile d'inclure une information sur l'incertain au niveau de ces deux modèles, dans un premier temps, seul le modèle relationnel contiendra une telle information.

3.1.1 Le modèle géométrique des objets

Les types de représentation

Certaines des représentations utilisées dans le domaine de la CAO sont, comme nous l'avons vu, bien adaptées à la définition des objets; d'autres sont idéales pour le traitement géométrique. Dans un souci d'ergonomie nous avons donc préféré dissocier les représentations liées à la définition et à l'utilisation des modèles. La définition de volumes élémentaires se fait au moyen de primitives de base ou par balayage. Ces volumes sont ensuite combinés ou modifiés grâce à des opérateurs de collage, perçage, ou de coupe par un plan.

Les représentations utilisées par le système de programmation de niveau objet répondent à deux besoins distincts. Le premier de ces besoins correspond à l'examen détaillé des objets dans le cadre de leur saisie ou de leur montage; cet examen est lié à l'évaluation de nombreuses propriétés géométriques des objets (locales ou non). De tels traitements nécessitent l'emploi de représentations analytiques aussi complètes et exactes que possible. C'est la **représentation B.R.** qui répond le mieux à ces besoins. Parallèlement à ces traitements géométriques rigoureux, les modules de recherche de trajectoires sans collision ont besoin de représentations plus approximatives des objets fournissant une information plus spécifique. Ce type d'information peut être plutôt numérique (c'est le cas de l'**approximation sphérique des objets**) ou plutôt symbolique (si l'on utilise les **cônes généralisés**).

La présence *simultanée* de ces représentations est essentiellement motivée par une économie de temps calcul. En effet, les sous-problèmes sont de nature différentes (calcul de trajectoires sans collisions, saisie automatique...), et les représentations les mieux appropriées aux calculs à effectuer sont donc différentes (polyèdrique pour le graphique, sphérique pour la recherche de trajectoires en univers peu contraint...). Il est donc utile de conserver plusieurs types de représentations (et au besoin de degrés de finesse dans celles-ci) afin de ne pas avoir à les recalculer.

La représentation sphérique est essentiellement utilisée pour optimiser certains calculs ne nécessitant pas une étude détaillée de par leur contexte (par exemple calcul de trajectoires sans collision dans un univers très peu contraint). Enfin, la représentation de type cônes généralisés, fournit une représentation symbolique et concise sur la forme des objets. Ce type d'informations peut être utile pour certaines

prises de décisions (par exemple, choix d'une partie à saisir) ou pour des opérations de mises en correspondance telles que celles employées en vision par ordinateur.

Les informations codées pour caractériser les représentations de type sphérique et cônes généralisés sont classiques (centre et rayon pour les sphères, contour, axe et distance ou angle de balayage pour les cônes généralisés). Celles qui sont enregistrées pour la représentation de type B.R. permettent de caractériser séparement la topologie des objets représentés et leur géométrie.

Le modèle topologique et géométrique associé à la représentation B.R.

La **géométrie** décrit la **position** des éléments constituant un solide; la **topologie** décrit les **relations** de connection et de hiérarchie entre ces éléments.

La topologie de chaque objet du modèle est décrite par une décomposition du type:

$$\text{objet} \rightarrow \text{composant} \rightarrow \text{face} \rightarrow \text{arète} \rightarrow \text{sommet}$$

Cette décomposition conduit à une hiérarchisation du modèle associé aux objets.

Les composants représentent des parties d'objets approximées soit par des volumes élementaires de type cylindrique, cônique ou sphérique, soit par des polyèdres quelconques. Les faces peuvent de ce fait être planes, cylindriques, sphériques ou côniques, les arètes rectilignes ou circulaires.

Afin de conserver une représentation homogène pour chaque objet, le système associe une approximation polyédrique (plus ou moins fine) aux composants à surface courbe. Ce mécanisme est nécessaire pour certains algorithmes généraux tels que les fonctions de calculs d'interférence entre objets ou la visualisation graphique.

Nous donnons, dans la figure suivante (cf. figure 3.1), un exemple simple de description géométrique d'un objet composé d'un cylindre et d'un parallèlépipède. La finalité de cette représentation B.R. est de pouvoir localiser la matière par rapport à l'objet à tous les niveaux de la hiérarchie. Pour cela nous conservons les informations suivantes:

- nature du composant (vide ou plein)
- propriétés topologiques (concavité par exemple) de toutes les entités (ces propriétés pouvant être utiles pour la détermination automatique des prises d'un objet)
- normale extérieure aux faces planes
- orientation des arètes (convention "matière à droite")

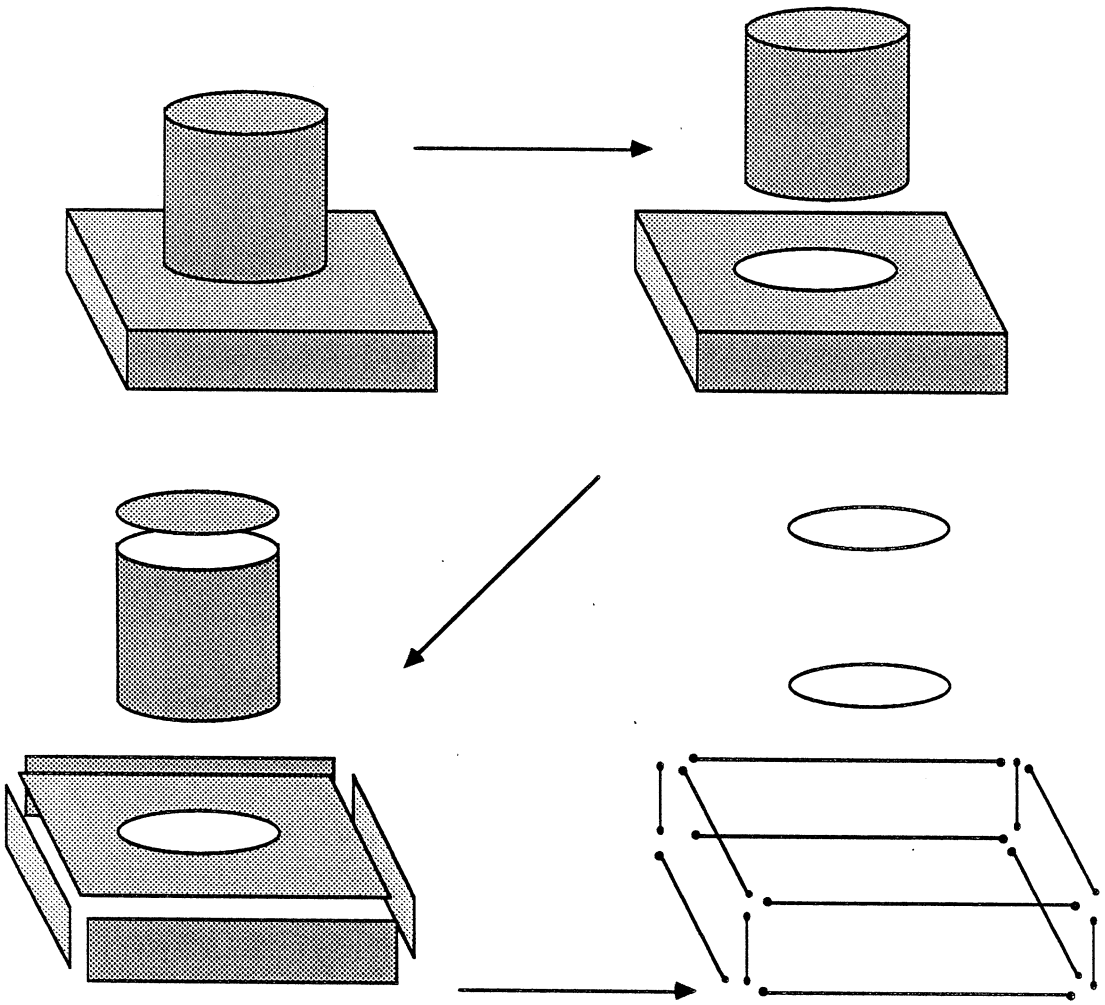


Figure 3.1: Exemple de représentation de type BR

La topologie locale est représentée par un chaînage de type "winged-edge". Ce chaînage [Cameron 81] consiste à représenter explicitement les relations de "voisinage" entre élements du modèle (cf. figure 3.2). Ainsi pour une arète, nous aurons pour information topologique locale un chaînage avec les deux faces possèdant cette arète ainsi qu'avec ses arètes voisines.

Ce chainage, s'il fournit des informations redondantes par rapport aux précédentes, est cependant très utile pour certains algorithmes géomètriques travaillant dans un environnement local. Il permet ainsi un parcours "intelligent" du modèle en évitant d'examiner l'ensemble des entités.

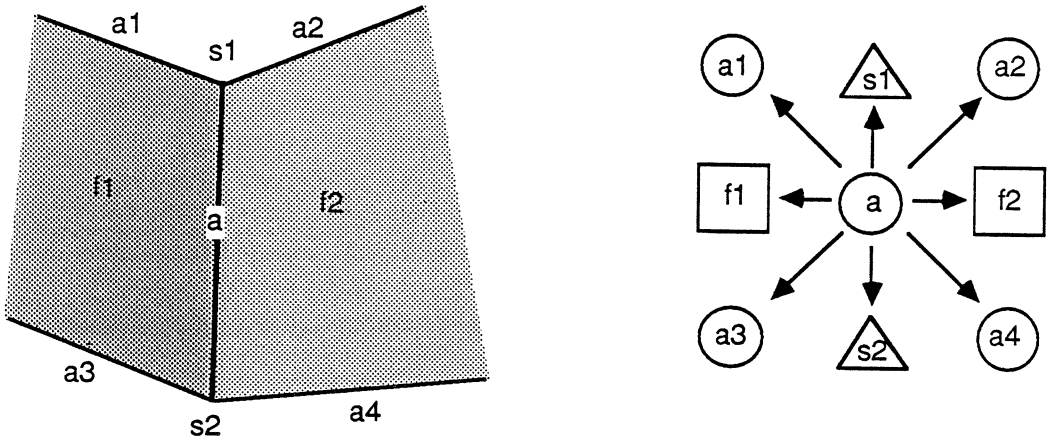


Figure 3.2: Représentation de la topologie locale

Grâce à ce chainage, nous avons donc, outre le parcours descendant de l'arborescence (implicite dans la structure de données), la possibilité d'utiliser un parcours ascendant ou par niveau.
La représentation géométrique associée consiste en une quantification des paramètres des élements géométriques: centre et rayon pour une sphère, rayon, hauteur et orientation de l'axe pour un cylindre... (cf. [Pertin 84]).

3.1.2 Le modèle relationnel de l'univers physique

Les relations dans l'univers physique

"Point de terme si absolu ou si détaché qu'il n'enferme des relations et dont la parfaite analyse ne mène à d'autres choses et même à toutes les autres"

Leibniz

Si ce principe, dit des relations internes, est tout à fait discutable d'un point de vue purement philosophique, il nous semble convenir parfaitement à la réalité de l'univers physique pour lequel le mouvement (ou l'immobilité) de tout objet n'est que par les relations de celui-ci à tous les objets de l'univers.

Le comportement d'un objet dans un environnement physique n'est donc prévisible qu'au travers des contraintes qu'il y subit. Savoir modéliser ces contraintes est une des clés du raisonnement de "sens commun" essentiel à la manipulation et

si difficile à reproduire dans les systèmes artificiels.

Dans le contexte de l'assemblage, ces contraintes se manifestent principalement par des contacts entre les objets (manipulant ou manipulés). Cependant, seuls certains de ces contacts sont pertinents en ce qui concerne le maintien de la cohérence du modèle de l'univers; ce sont les contacts actifs qu'un objet impose à un autre objet par l'application d'une force (pesanteur ou serrage). Ils se traduisent par des restrictions partielles ou totales du mouvement des objets. Ils imposent en outre, des contraintes de positionnement et d'orientation relatifs pour les objets en relation. La mise en relation d'objets pendant l'assemblage conduit à mener un raisonnement sur des regroupements d'objets que nous appellerons structures: par exemple, la saisie ou le transport d'un sous-assemblage nécessitent la prise en compte et le respect des contraintes entre les objets pendant la durée de l'action.

Le modèle relationnel comporte donc:

- une description des structures créées par les relations,
- une description "qualitative" des relations en terme de contacts actifs entre objets,
- une description "quantitative" des relations en terme de contraintes de positionnement relatif entre objets d'une structure.

Les structures composant l'univers

Différents types de structures doivent être modélisés; elles peuvent être articulaires ou rigides. Le premier type correspond à la modélisation de manipulateurs ou d'assemblages articulés. Les structures rigides permettent de modéliser les autres assemblages ou de représenter des liaisons physiques dûes, par exemple, au transport d'un objet par un manipulateur ou à la gravitation maintenant un objet sur un autre.

Appelons respectivement ces structures, **structures articulées** et **structures rigides**.

Trois types de *structures articulées* de base doivent être modélisés:

1. les **structures à articulations simples**: c'est le cas d'un bras manipulateur dans lequel chaque objet composant la chaîne cinématique est lié au précedent par un ou plusieurs degrés de liberté indépendants. Chaque degré de liberté peut être soit rotoïde, soit prismatique (cf. figure 3.3). D'autres liaisons plus complexes peuvent être décrites à l'aide de ces deux types d'articulations. Une liaison de type rotule (trois degrés de liberté en rotation) peut par exemple être représentée par trois liaisons à axes concourants. Dans ce cas deux solides fictifs (repères purs) seront introduits dans la chaine cinématique par le système.

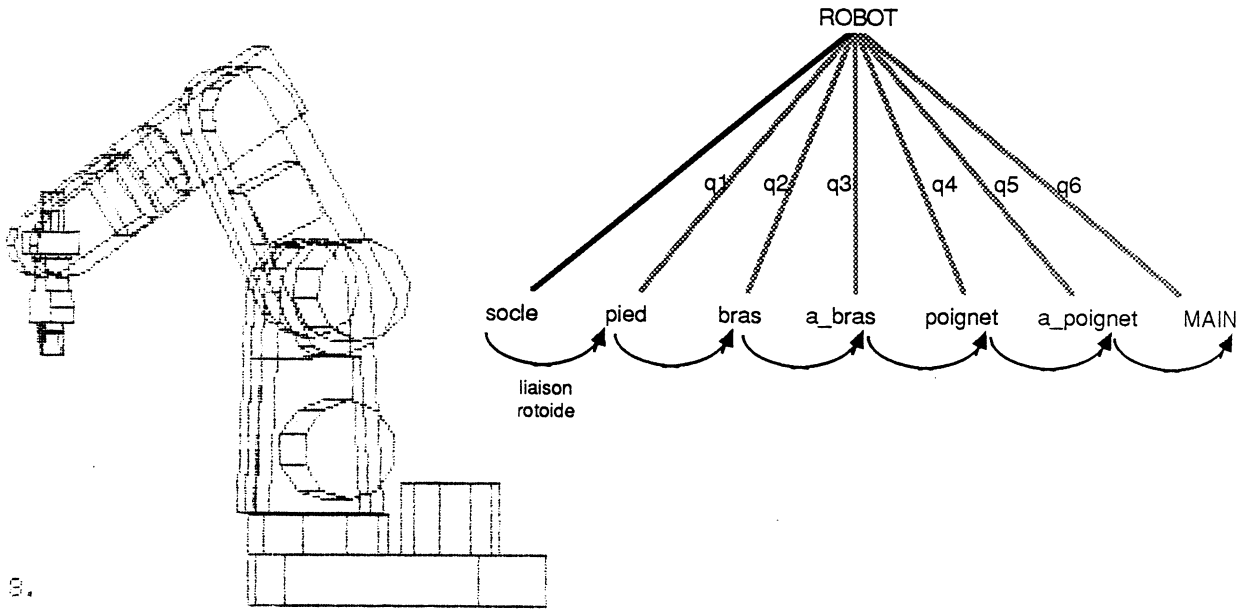


Figure 3.3: Structure à articulations simples

2. les **structures à articulations dépendantes**: de tels objets possèdent des degrés de liberté qui sont fonctionnellement liés. Un exemple simple est la liaison prismatique double (liaison biprismatique ou plus généralement polyprismatique); c'est le cas d'une pince de robot à deux mors parallèles: la commande de mouvement reçue par la pince (valeur q) est répartie sur les deux mors qui doivent se déplacer des quantités -q/2 et +q/2 (cf. figure 3.4). Un autre exemple est la liaison rotoide double (liaison birotoïde ou plus généralement polyrotoïde).

3. les **structures à articulations couplées**: certains robots possèdent dans leurs mécanismes des structures passives asservies à d'autres articulations commandées par des moteurs. Ces articulations passives sont commandées directement par la structure mécanique du manipulateur. C'est le cas, par exemple, de la boucle cinématique de type parallèlogramme (cf. figure 3.5). Ce type de boucle passive permet d'améliorer les performances mécaniques du bras (en particulier rigidité et précision). Nous trouvons également ce type de liaison que nous nommerons c-rotoïdes, dans certains organes de saisie (cf. figure 3.6).

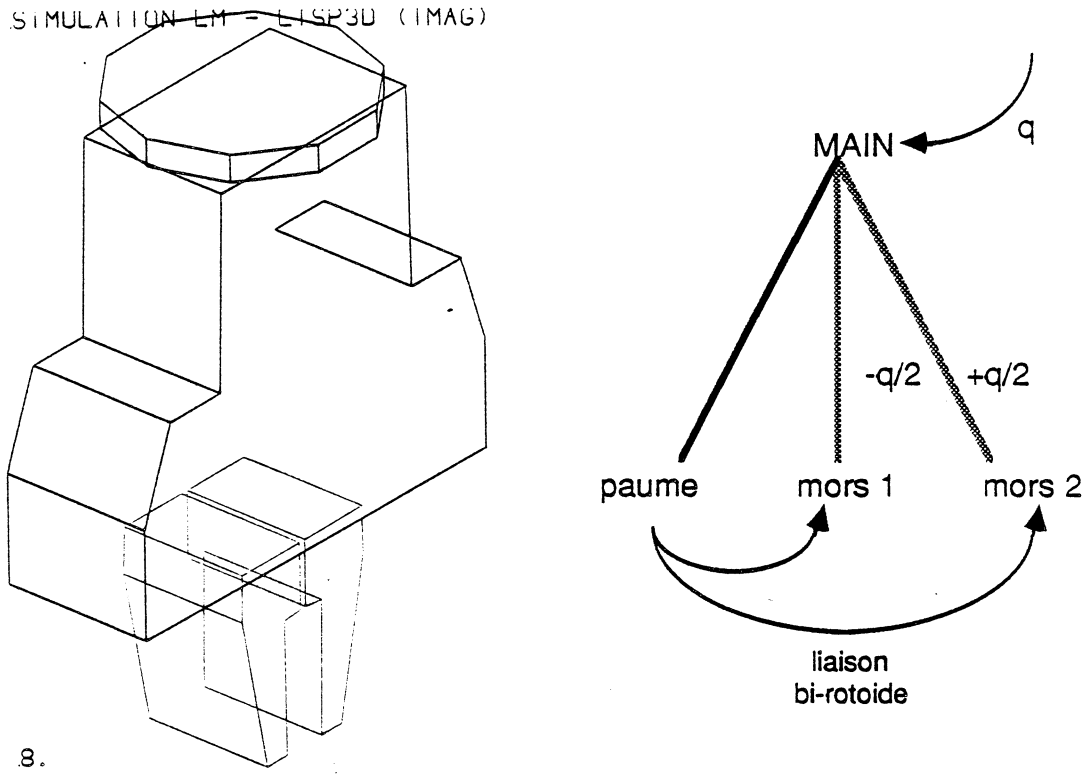


Figure 3.4: Structure à articulations dépendantes

Comme nous le voyons sur la figure 3.7 ce type d'articulation peut être représentée en chaine cinématique ouverte.

Ces liaisons permettent de conserver des propriétés géomètriques entre les objets composant la structure mécanique et ceci malgré les mouvements des degrés de liberté: parallèlisme des corps c_1 et c_4 (ou c_2 et c_3) dans le premier exemple, parallèlisme des deux mors dans le second exemple. Ainsi les déplacements des corps passifs peuvent-ils être déduits directement des contraintes géomètriques de parallèlisme entre les divers corps.

Il existe également plusieurs types de *structures rigides*.

- Certaines de ces structures sont à rapprocher de celles qui sont créées ou détruites par des instructions particulières telles que les instructions LIER et DELIER du langage LM (cf. [Miribel 84]). Les objets regroupés par ce type de liaisons forment des structures rigides c'est à dire des groupements d'objets solidaires dans leurs déplacements. Ces structures sont **éphémères**. Elles sont par exemple créées par des opérations de saisie et n'ont d'existence que le temps nécéssaire aux manipulations considérées (typiquement: transport d'un objet ou opérations éxécutées par un outil particulier). Elles peuvent également traduire les contraintes gravitationnelles liant un objet à son support.

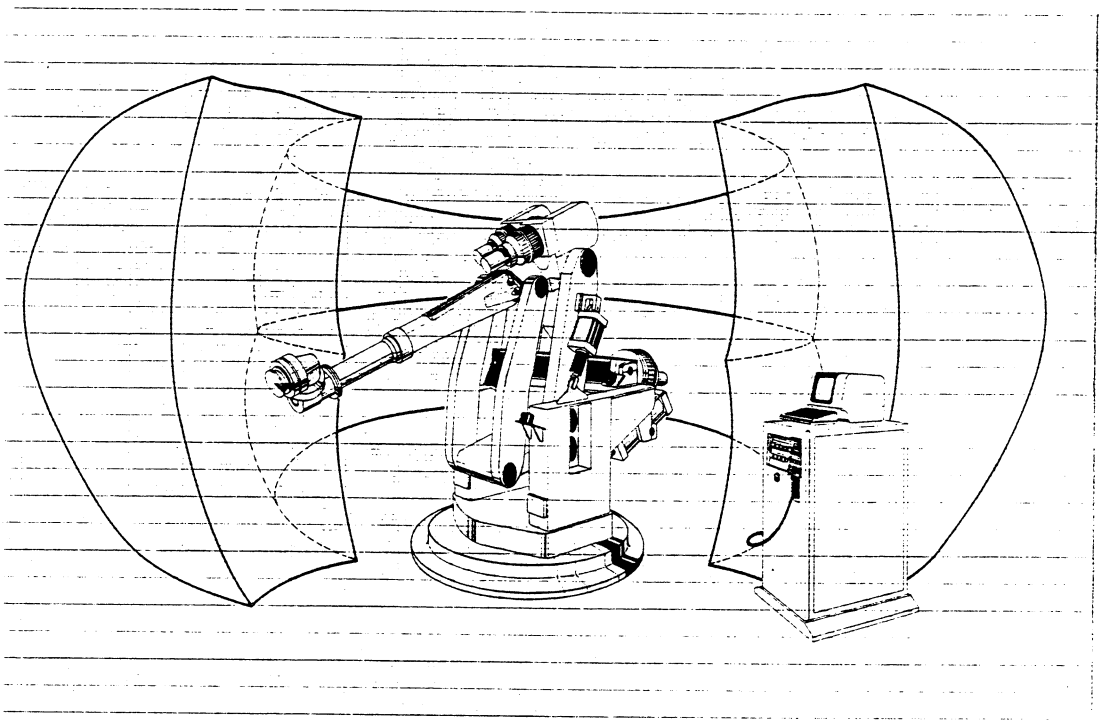


Figure 3.5: Manipulateur à structure de type parallèlogramme

- D'autres structures ont un caractère **permanent**; elles représentent par exemple des groupes de pièces mécaniques assemblées entre elles de manière irréversible (par collage en particulier). Ces structures rigides permanentes peuvent être considérées comme des solides à part entière et modélisées de même. Nous assimilerons dans la suite de l'exposé structure rigide à structure rigide éphémère.

Représentation des structures

Lors de la planification d'un assemblage, il est nécessaire de considèrer un système de référence absolue. Nous privilégierons donc un objet de l'espace de travail appelé univers: un banc de montage, le sol de l'atelier...

Les solides présents dans l'univers sont de deux natures: certains d'entre eux sont "isolés" (uniquement en relation avec la référence absolue) alors que d'autres participent à la création de structures rigides ou articulées.

Toute structure est représentée au moyen d'une arborescence (graphe connexe sans cycle, à une racine). Sa racine est un objet fictif permettant d'effectuer un accès direct; les nœuds issus de la racine sont les objets constituant la structure. L'objet univers est, lui aussi, une structure, rigide, contenant les objets "isolés". De ce fait, la représentation globale de l'espace de travail peut être modélisée sous la

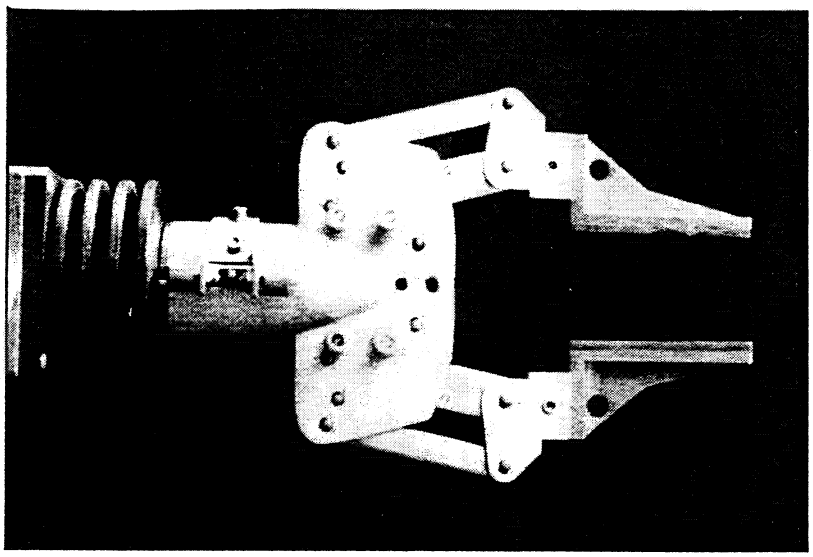


Figure 3.6: Pince à structure de type parallèlogramme

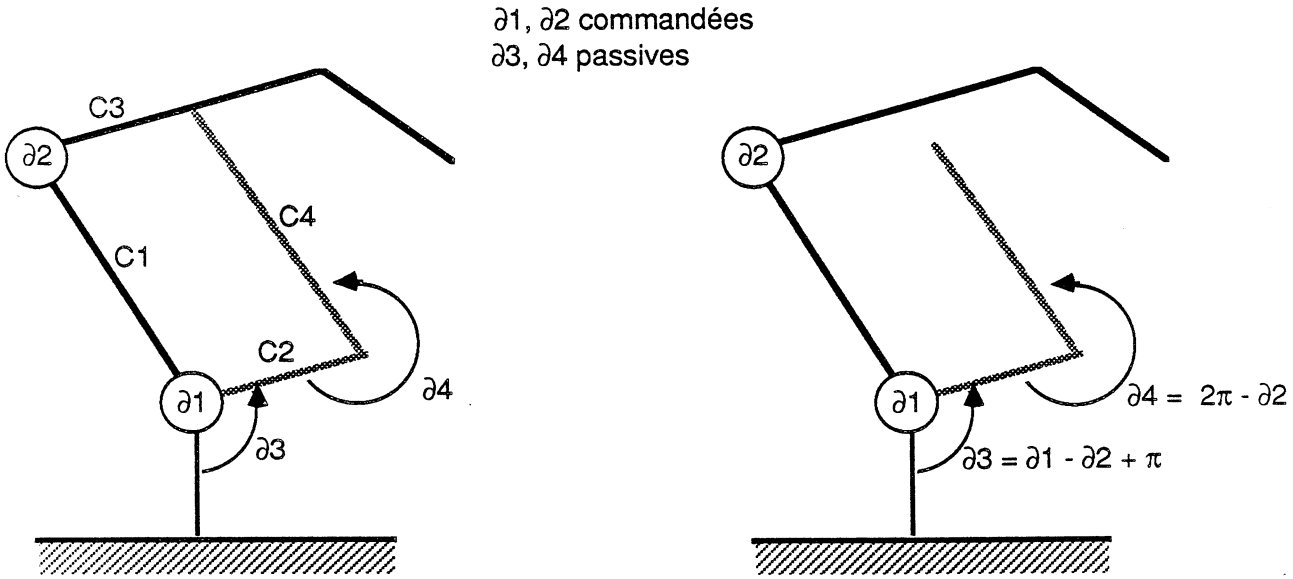


Figure 3.7: Représentation d'une structure de type parallèlogramme

forme d'une arborescence (cf. figure 3.8).

Cette représentation est essentielle pour l'utilisation du modèle; c'est en effet, la seule représentation directement exploitable dans le modèle relationnel de par sa structure arborescente.

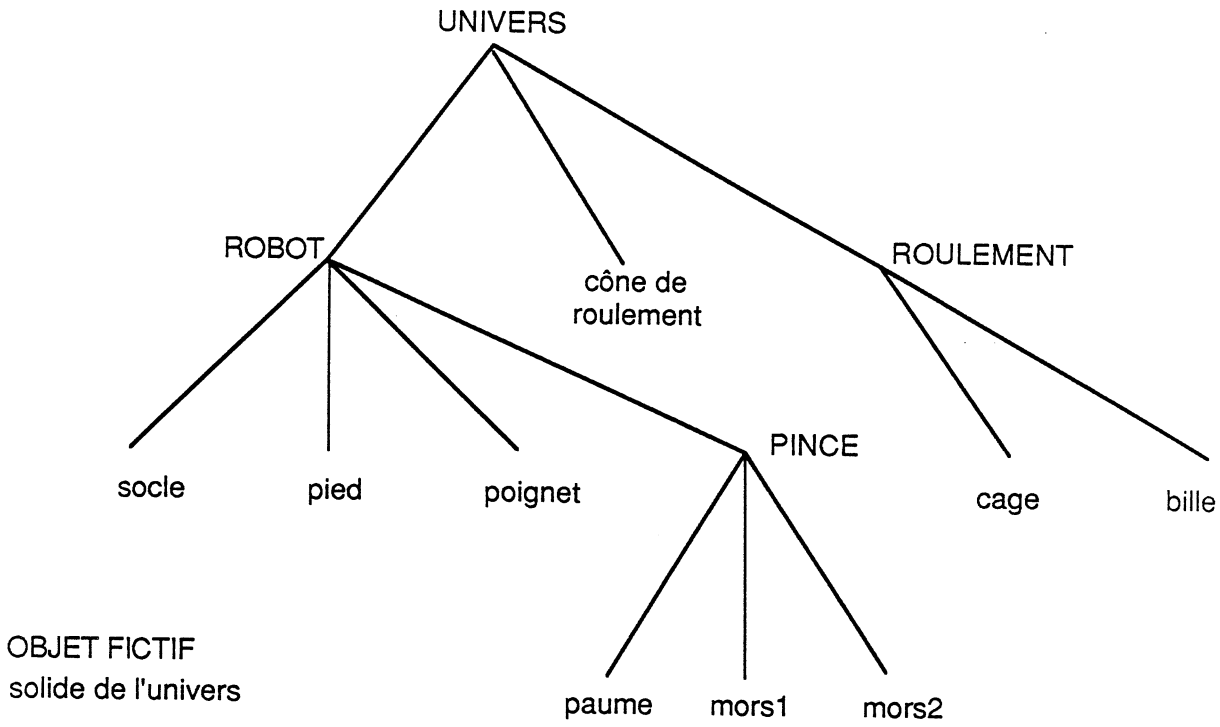


Figure 3.8: Structure arborescente de l'univers

Représentation qualitative des relations

Cette représentation est fondée sur la notion de contact actif entre deux objets. Elle a pour but essentiel de permettre le maintien de la cohérence du modèle de l'univers physique tout au long de la planification (qu'elle peut guider) ou de la simulation d'un montage. C'est un problème extrêmement complexe que nous commençons seulement à aborder par le biais de la modélisation et qui n'est pas encore intégré à nos modules de raisonnement (géométrique et spatial).

Un contact peut être modélisé au moyen d'un arc menant de l'objet le produisant à l'objet le subissant. Cet arc supporte différentes informations visant à préciser la nature du contact: sa cause - pesanteur, serrage, assemblage, compliance... - (cf. figure 3.9), les entités géométriques des objets mises en contact (essentiellement des faces). Cette information sera utilisée de façon à vérifier en quoi la planification d'un mouvement est susceptible de modifier un contact et de provoquer le mouvement non désiré d'un objet (glissement, chute). Cette représentation définit un graphe orienté connexe greffé sur l'arborescence structurelle. Elle n'existe cependant, à

l'heure actuelle, qu'à l'état de spécification.

Représentation quantitative des relations

Cette représentation a pour but de caractériser quantitativement les relations par la connaissance des positions relatives des objets appartenant à une structure pendant toute la durée de vie de celle-ci.

Chaque nœud de l'arborescence structurelle doit être physiquement localisé dans l'univers. Si la position absolue des objets dans l'univers est l'information généralement recherchée, elle nécessite la connaissance des positions relatives des objets obtenues grâce aux relations entre ces objets. C'est cette information relative que nous conserverons de par sa nature invariante au sein d'une structure. Il faut pour cela, dans chaque structure, positionner un objet par rapport à une référence. Pour des raisons de facilité d'accès et de mise à jour du modèle, celle-ci ne sera pas le père de cet objet dans l'arborescence.

Dans le cas de structures articulées la recherche d'une référence pour les différents éléments du mécanisme correspond, en fait, à une reconstruction de la chaîne cinématique. Chaque élément est en effet référencé par rapport à son prédecesseur dans la chaine cinématique. Ce sera son frère immédiatement plus agé dans la structure arborescence, dans le cas de structures à articulations simples ou couplées; dans le cas de structures à articulations dépendantes, ce sera son frère ainé. Cette représentation permettra ainsi, de calculer séquentiellement les positions des objets de la structure correspondant aux mouvements du manipulateur (cf. figure 3.10).

Afin de réduire les temps de parcours de graphe, lors de la mise à jour du modèle relationnel de l'univers, chaque structure rigide est représentée par un ensemble d'objets secondaires rattachés à un objet principal (cf. figure 3.11). Tout objet de la structure sera référencé par ce dernier qui est choisi par le système parmi les objets composant la structure au moment de sa création. Dans le cas où l'un de ces objets est un mécanisme articulé ou appartient à un tel mécanisme (pince du robot par exemple), c'est lui qui est choisi. Dans tout autre cas, le choix est indifférent.

Le chaînage ainsi créé forme une forêt (graphe acyclique) plaquée sur l'arborescence structurelle. Le positionnement relatif des objets est exprimé au moyen de transformations géométriques représentées sous forme de matrices de coordonnées homogènes ou de paramètres (pour les structures articulées commandées). Cette représentation s'appuie sur la notion de repères cartésiens; ceux-ci sont attachés à tout objet (solide ou structure). Dans le cas d'une structure rigide, le repère associé est celui de l'objet principal; pour les structures articulées, il s'agit du repère du fils ainé dans la sous-arborescence correspondante. Tout solide est associé à un repère cartésien au moment de sa construction.

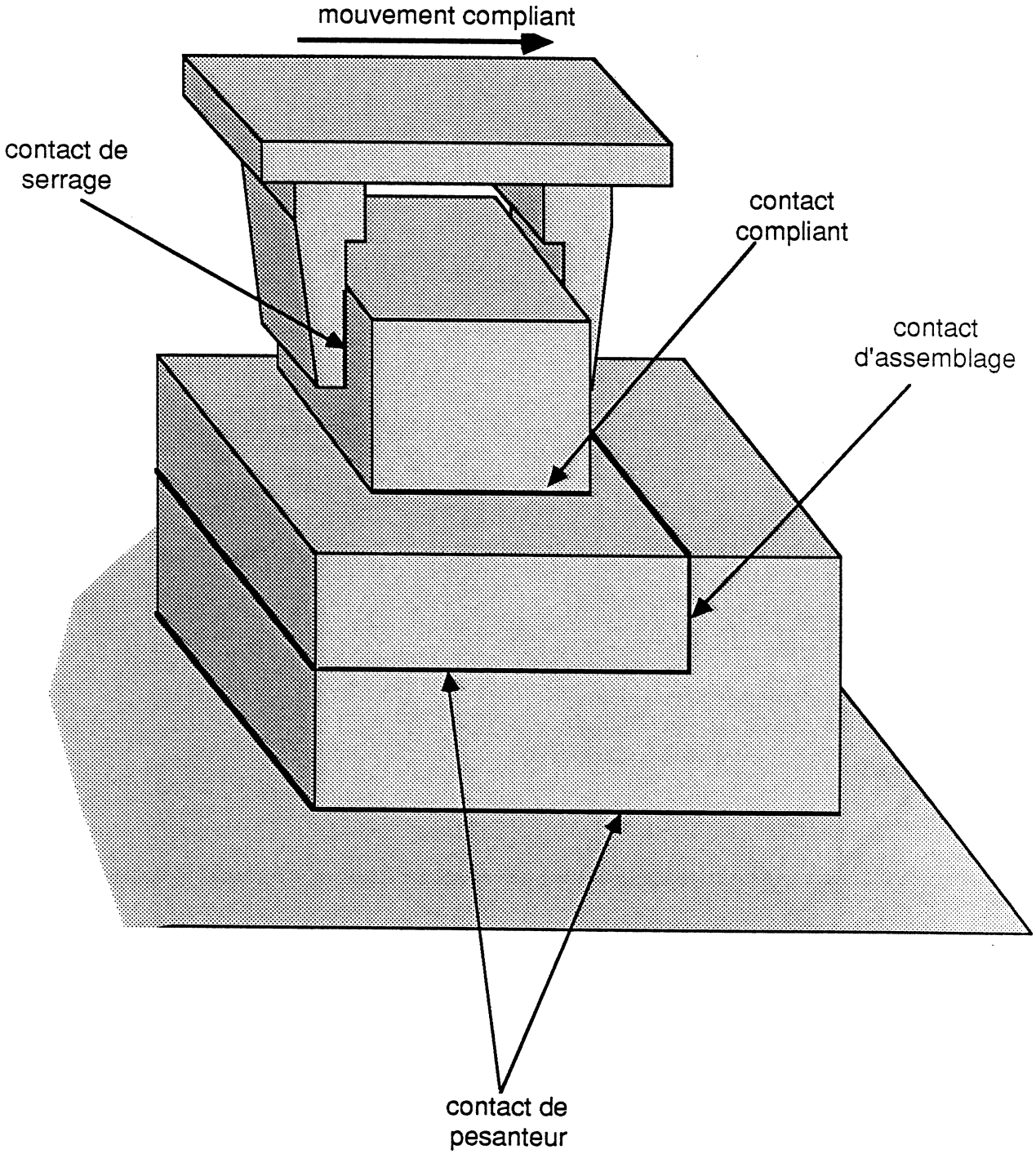
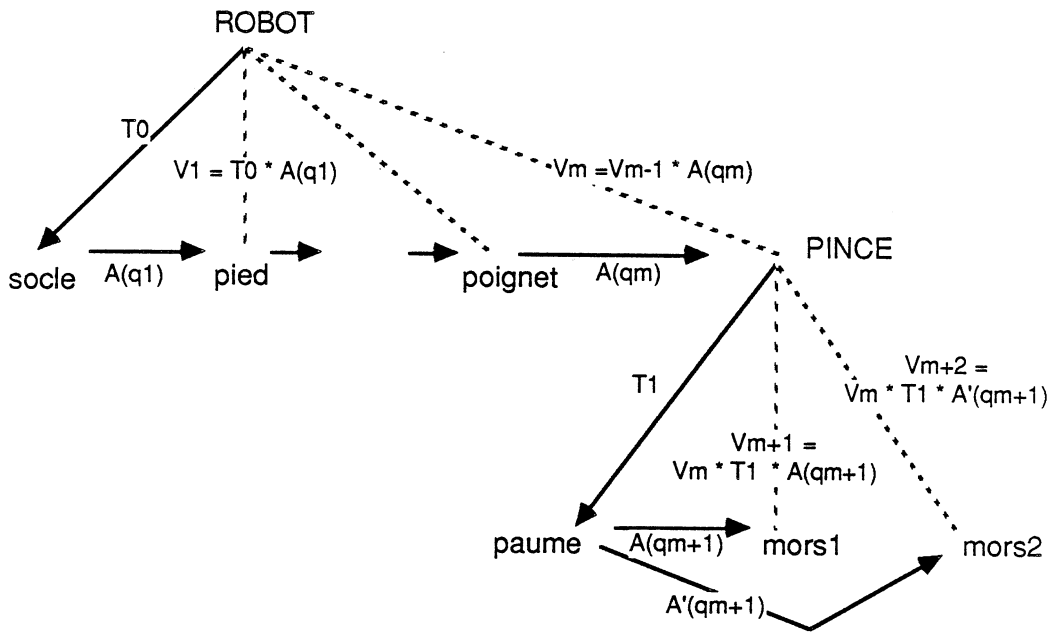


Figure 3.9: Différents types de contacts



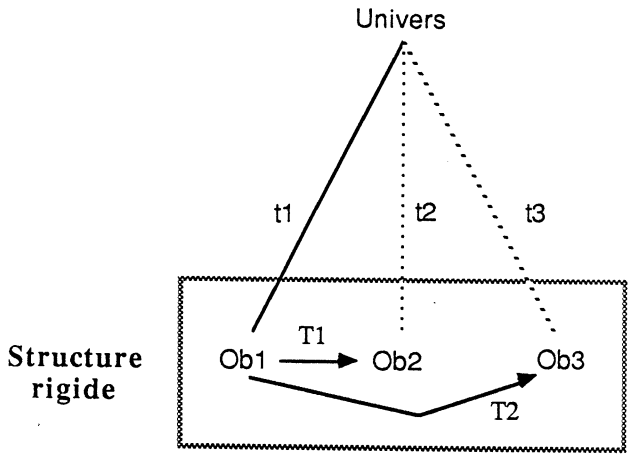
T0, T1: transformations constantes
A(qi): transformations paramètrées
Vi: transformations variables (calculées)

Figure 3.10: Représentation quantitative des relations dans les structures articulées

Les fonctions de manipulation du modèle relationnel

Les opérations de mise à jour du modèle relationnel nécessitent l'implantation de fonctions de manipulation des structures mises en jeu. Cette mise à jour concerne d'une part la gestion des structures (création, destruction) et d'autre part la gestion des relations (principalement, sa mise à jour quantitative).

La création d'une structure rigide permanente correspond à l'apparition d'un nouveau solide. Aucune arborescence n'est créée dans ce cas; seul un nouvel objet élémentaire apparait remplaçant les objets à partir duquel son modèle géométrique est construit (opération de collage sur les modèles géométriques).



t1: transformation modifiable (déplacement de la structure rigide)

t01,t02,t03: valeurs des transformations t1,t2,t3 au moment de la création des liaisons

T1,T2 transformations constantes

$T1=t1^{-1} \bullet t02$, $T2=t1^{-1} \bullet t03$

Figure 3.11: Représentation quantitative des relations dans les structures rigides

Dans le cas des structures rigides éphémères apparaissant à la saisie d'un objet ou lors d'un assemblage, une sous-arborescence est créée. Sa racine est un objet fictif dont le nom est engendré par le logiciel. L'objet principal est choisi selon les règles mentionnées ci-dessus.

Les structures articulées nécessitent une mise à jour des informations de positionnement à chaque déplacement du manipulateur. Si ce déplacement est décrit par la suite de coordonnées généralisées correspondant, ces valeurs sont directement injectées dans le modèle. Dans le cas où le mouvement est décrit grâce à la position de l'organe terminal du mécanisme, un calcul de cinématique inverse est nécessaire pour mettre à jour ces paramètres. Il est à noter que dans un souci d'efficacité nous conservons la position de l'organe terminal dans le repère associé à la structure articulée. Cette information est utile aux mises à jour, notamment lorsque cet organe terminal intervient dans une structure rigide.

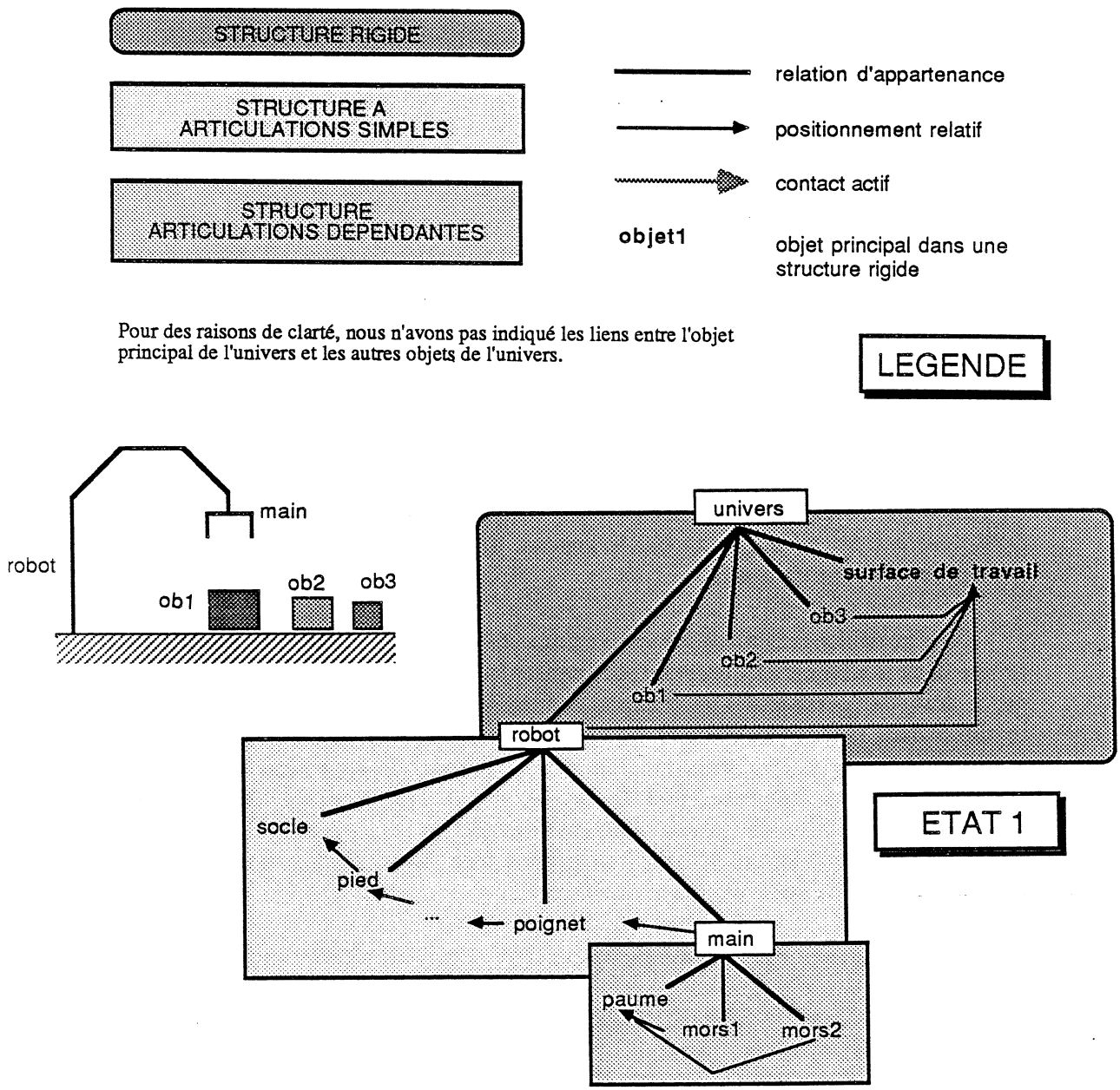
Dans les structures rigides, les transformations qui permettent de passer de

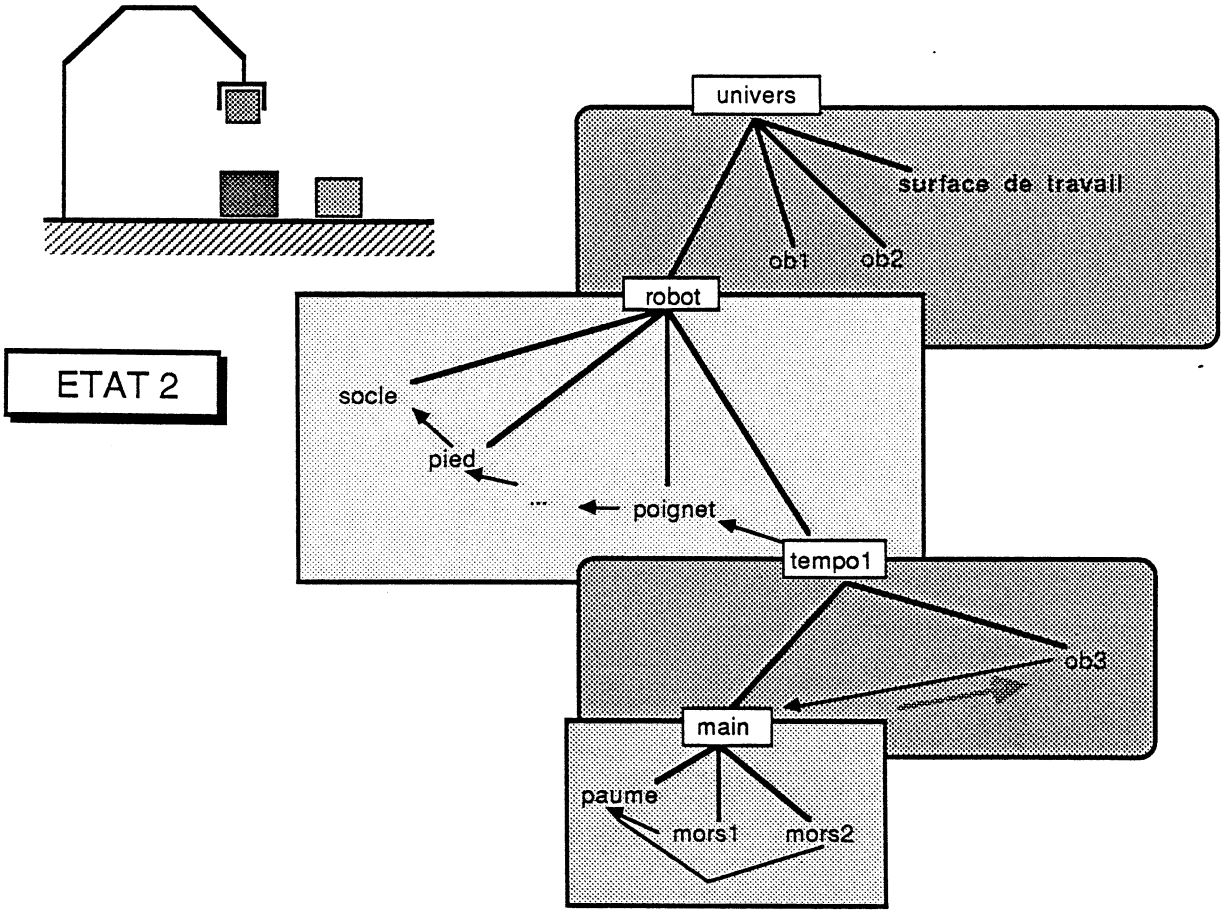
l'objet principal aux objets secondaires sont calculées par le système lors de la création des liaisons ou lorsqu'une réorganisation de la structure est nécessaire (par exemple: introduction de la pince du robot dans la structure, suppression de l'objet principal, ou fusion avec une autre structure).

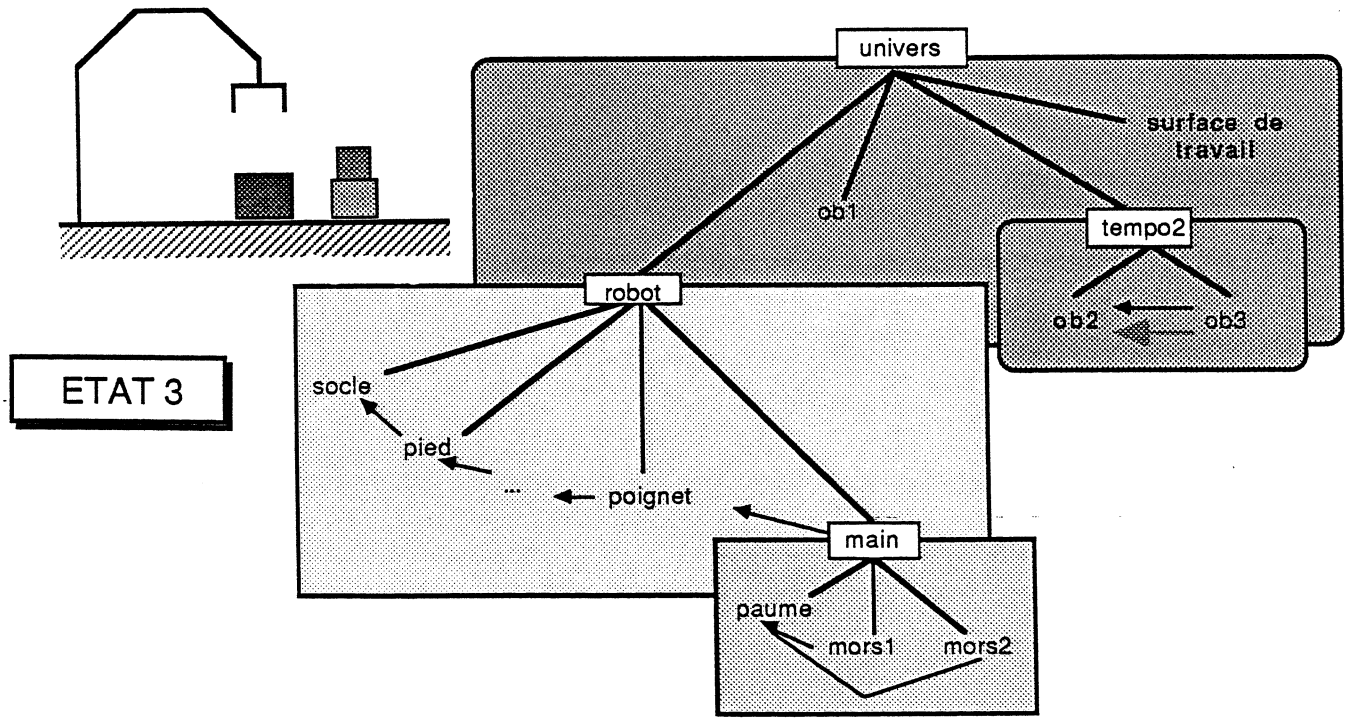
Une description précise des modèles géométrique et relationnel pourra être trouvée dans [Pertin 84].

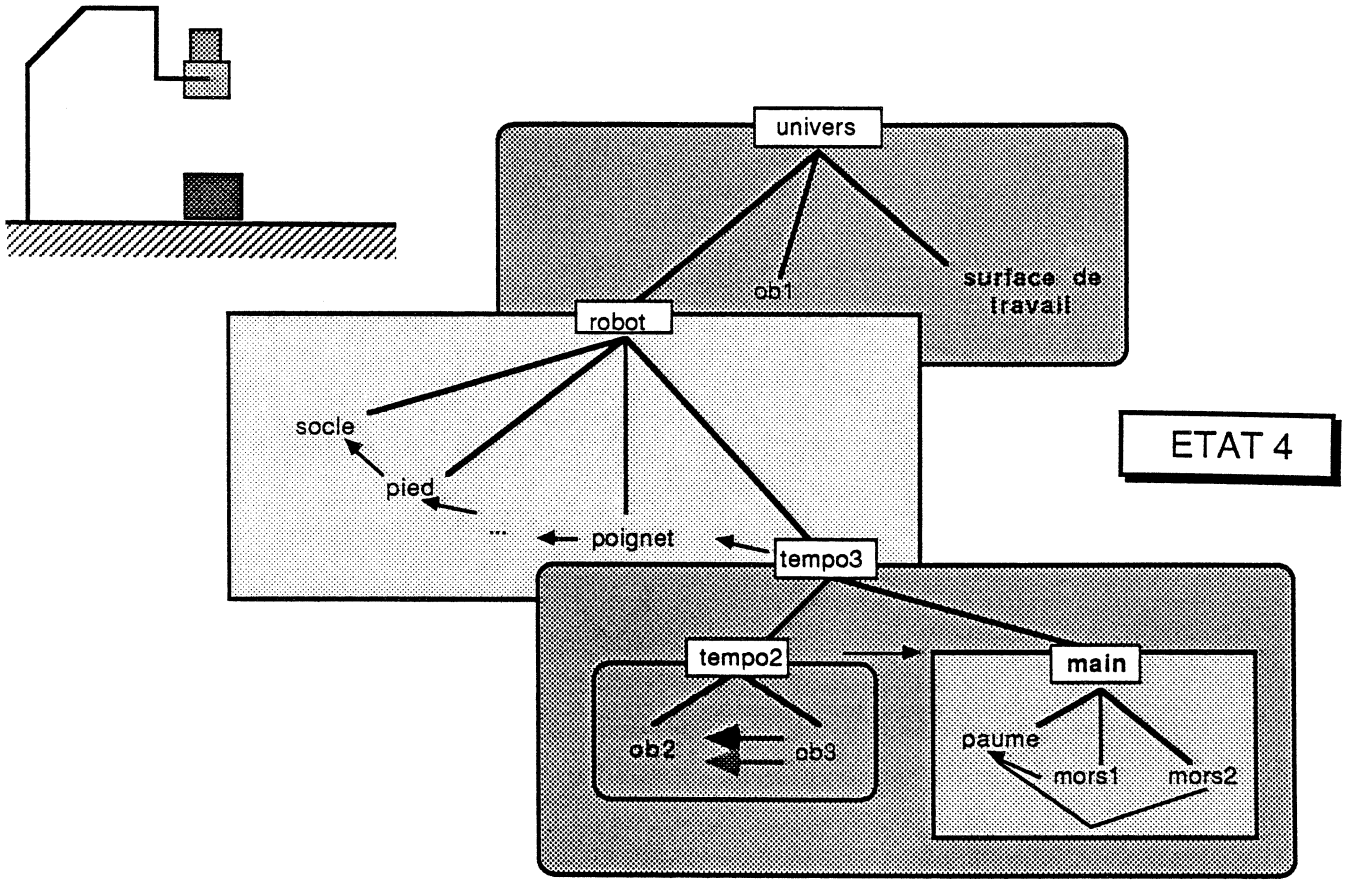
La figure suivante (cf. figure 3.12) montre l'évolution du modèle relationnel au cours d'une manipulation.

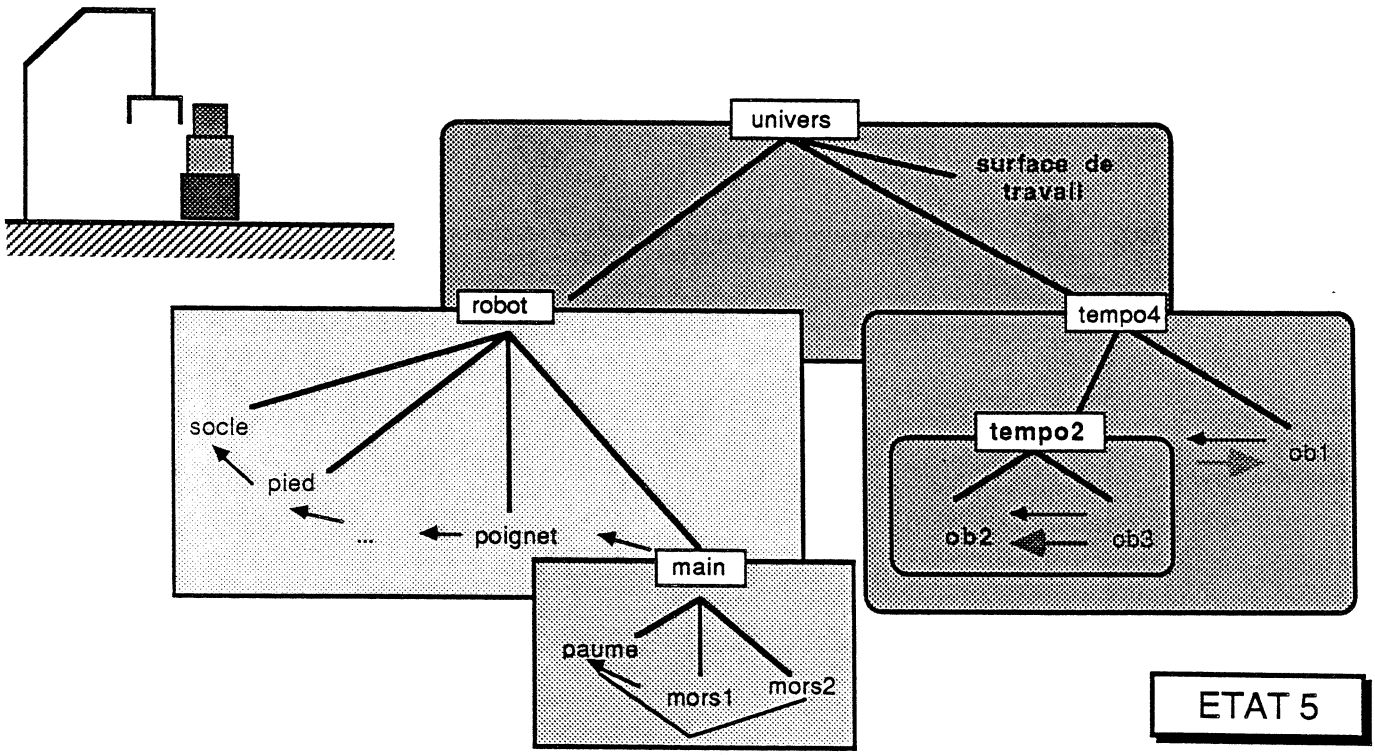
Figure 3.12: Evolution du modèle relationnel au cours d'une manipulation




univers
surface de travail
ob1
ob2
robot
ETAT 2
socle
pied
...
poignet
tempo1
ob3
main
paume
mors1
mors2


univers
surface de travail
ob1
tempo2
ob2
ob3
robot
ETAT 3
socle
pied
...
poignet
main
paume
mors1
mors2


univers
ob1
surface de travail
robot
socle
pied
...
poignet
tempo3
tempo2
ob2
ob3
main
paume
mors1
mors2
ETAT 4


univers
surface de travail
robot
socle
pied
...
poignet
main
paume
mors1
mors2
tempo4
tempo2
ob1
ob2
ob3
ETAT 5

3.1.3 Modélisation de l'incertain

Certains récents développements en modélisation au LIFIA ont porté sur la modélisation de l'incertain. Ces travaux guidés par le développement conjoint de modèles de représentation et de raisonnement se sont matérialisés sous la forme d'un modèle de représentation et de raisonnement sur les incertitudes de position [Puget 85], [Pertin 86].

La programmation des robots nécessite très souvent de connaître les positions relatives de deux objets et de savoir avec quelle précision ces objets sont positionnés l'un par rapport à l'autre. Nous présentons ici un formalisme permettant de supporter ce type de raisonnement.

Comme nous l'avons vu précédemment, nous ne nous attachons actuellement qu'aux incertitudes de position. La position d'un objet est définie par une transformation géométrique (composition d'une translation pure et d'une rotation pure) qui appliquée au repère de référence (celui de l'objet univers) donne le repère lié à l'objet.

Raisonner sur les positions d'objets revient donc à raisonner sur des repères liés par des transformations. Nous noterons $\Re$ les différents repères, $\Re_0$ étant le repère fixe de référence de l'univers. La position relative de deux repères $\Re_i$ et $\Re_j$ est définie par une transformation géométrique T_{ij} telle que $\Re_j$ soit le transformé de $\Re_i$ par T_{ij} :

$$\Re_j = \Re_i * T_{ij}$$

L'inversion et la composition des transformations permettent de calculer les positions relatives T_{ij} de deux repères quelconques $\Re_i, \Re_j$ à partir de leur position absolue T_{0i} et T_{0j} . Ces transformations ne permettent que de donner des positions théoriques. Une **erreur** sur une position se traduit par un "écart" entre la position théorique et la position réelle. Cet écart, ou erreur de position est la transformation géométrique entre position théorique et position réelle.

On notera ε_{ij} l'erreur de position du repère $\Re_i$ par rapport au repère $\Re_j$. Tenant compte du terme d'erreur, la position réelle de $\Re_j$ par rapport à $\Re_i$ sera $T_{ij} \star \varepsilon_{ij}$:

$$\Re_j = \Re_i \star T_{ij} \star \varepsilon_{ij}$$

Représentation des incertitudes

Nous considérons qu'**une incertitude est un ensemble d'erreurs possibles**. La manipulation des incertitudes, donc d'ensembles de transformations géométriques, nécessite le choix de représentations appropriées.

Deux types de représentations sont utilisés suivant le type de calcul exécuté (cf. [Pertin 86]).

- Le premier type consiste à représenter une transformation sous forme d'une matrice 4×4 de coordonnées homogènes. Cette représentation est bien adaptée aux calculs numériques sur les transformations (inversion, composition, produit...) et sera utilisée à cet effet.

- Le second repose sur une décomposition de la transformation en une translation et une rotation; la représentation correspondante est un triplet: vecteur de translation, axe de rotation (vecteur unitaire), angle de rotation. Elle se révèle mieux adaptée que la précedente à la manipulation symbolique d'incertitudes. En effet, elle est plus proche de l'expression des degrés de liberté existant entre deux objets positionnés l'un par rapport à l'autre.

  Une incertitude devrait, en toute généralité, être représentée par un sous-ensemble du produit cartésien $\Re^3 \times S(1) \times [-\pi, +\pi]$ (où S(1) représente la sphère unité). Tout sous-ensemble de ce produit cartésien sera approximé par le plus petit sous-ensemble le recouvrant, du type:

  $Tr \times U \times A$

  - Tr peut être une sphère, un disque ou un segment
  - U sera soit la sphère dans son ensemble, soit formé d'un vecteur unique
  - A sera de la forme $[-\alpha, +\alpha]$ où α est un angle "petit" devant l'unité.

  Cette restriction se traduit par une perte d'informations donc une surcontrainte des incertitudes compensée cependant par le fait que la représentation correspond à un grand nombre de situations géométriques mettant en jeu différents types de contacts.

Voici à titre d'exemple, trois situations et les incertitudes correspondantes:

1. Objet reposant sur un plan z = Cste (cf. figure 3.13)

   L'incertitude de position d'un tel objet est liée à ses translations potentielles dans le plan de contact. Son orientation autour de l'axe z est sujette à incertitude. D'où:

   - Tr est un disque de rayon δ_0 et d'axe z
   - U est le vecteur unitaire porté par l'axe z
   - $A = [-\alpha_0, +\alpha_0]$

2. Objet "libre" dans l'espace

   - Tr est la sphère de rayon δ_0 centrée sur la position théorique de l'objet
   - U est la sphère unité $S(1)$
   - $A = [-\alpha_0, +\alpha_0]$

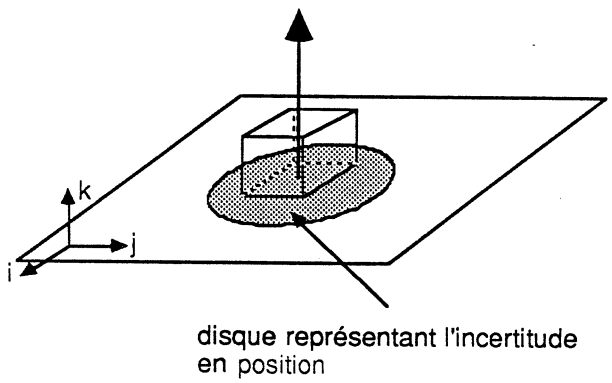


Figure 3.13: Représentation des incertitudes pour un contact plan sur plan

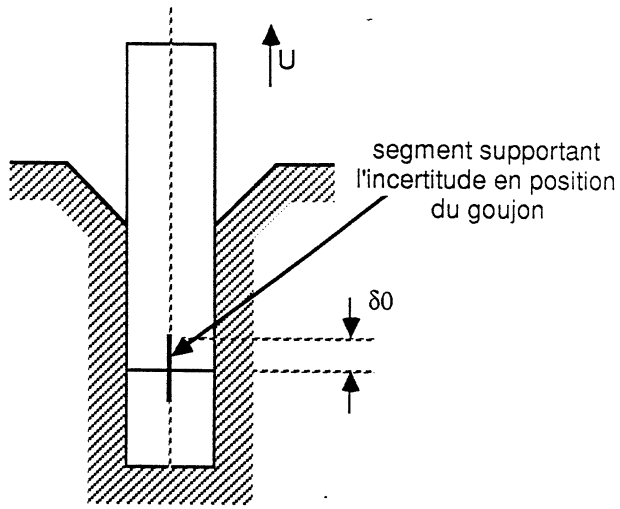


Figure 3.14: Représentation des incertitudes pour un contact cylindrique

3. Goujon partiellement inséré dans un alésage (cf. figure 3.14)

   Lorsque la relation de contact entre les faces cylindriques des goujon et alésage est vérifiée:

   - Tr est un segment porté par l'axe de l'alésage, de longueur $2\delta_0$ et de milieu la position théorique du goujon
   - U est le vecteur unitaire porté par l'axe d'insertion
   - $A = [-\alpha_0, +\alpha_0]$

Mise à jour du modèle

La représentation matricielle est utilisée comme représentation de travail. La représentation par triplet, sera superposée aux arcs du modèle relationnel (elle n'est pas encore intégrée à SMGR). Elle permet la manipulation symbolique d'incertitudes.

Le modèle est initialisé (avant manipulation) grâce à certaines données du programme concernant les incertitudes de position relative ou absolue des objets.

Le modèle doit être exploré et complèté pour connaître la précision d'un positionnement relatif de deux objets. Il doit être mis à jour à chaque exécution d'une action (déplacement du manipulateur ou lecture de capteurs). Ces opérations nécéssitent de résoudre trois problèmes:

1. Connaissant la position T_{ij} et l'incertitude I_{ij} d'un repère $\Re_i$ par rapport à un repère $\Re_i$, quelle est l'incertitude réciproque I_{ji} ?

2. Connaissant les positions T_{ij} et T_{jk} et les incertitudes I_{ij} et I_{jk} , d'un repère $\Re_j$ par rapport à un repère $\Re_i$ et d'un repère $\Re_k$ par rapport à un repère $\Re_j$, quelle est l'incertitude résultante I_{ik} de $\Re_k$ par rapport à $\Re_i$?

3. Etant donnés trois repères $\Re_i, \Re_j, \Re_k$ et les incertitudes I_{ij}, I_{jk}, I_{ik} , lorsqu'on modifie I_{ij} (par une mesure de position relative par exemple), comment propager cette information sur I_{ik} et I_{jk} ?

Il n'existe généralement pas de solutions simples à ces trois problèmes fondamentaux de la propagation géométrique des incertitudes; c'est pourquoi des approximations de calcul sont utilisées. Nous ne détaillons pas ici ces calculs (cf. [Pertin 86] pour plus de détail) mais en présentons une utilisation intuitive au travers d'un exemple.

Exemple de propagation géométrique d'incertitudes

Considérons la séquence de manipulation représentée dans les figures 3.15, 3.16.

Situation de départ:

Les deux objets sont posés sur le plan de travail (objet univers). Leur position ainsi que celle de l'effecteur du robot est définie par rapport au repère $\Re_0$ lié à l'objet univers. A chacun de ces objets C_1 et C_2 est attaché un repère $\Re_1$ et $\Re_2$ (resp.). Le repère lié à l'effecteur du robot est $\Re_{Rob}$ (cf. figure 3.17).

Les incertitudes absolues sont les suivantes:

$$I(\Re_{Rob}, \Re_0) = Sphere(O_{Rob}, \delta_0) \times S(1) \times [-\alpha_0, +\alpha_0]$$

$$I(\Re_1, \Re_0) = Disque(O_1, \delta_1) \times \{\overrightarrow{k_0}\} \times [-\alpha_1, +\alpha_1]$$

$$I(\Re_2, \Re_0) = Disque(O_2, \delta_2) \times \{\overrightarrow{k_0}\} \times [-\alpha_2, +\alpha_2]$$

Saisie de C_1 :

Lors de la saisie de C_1 , on supposera que le robot ne bougera pas: son incertitude sera toujours la même. Par contre, C_1 pourra "glisser" au moment de la fermeture de la pince. La saisie modifiera donc l'incertitude - absolue et relative - attachée à C_1 . Son hypervolume d'incertitude relative par rapport au robot sera modifié de la façon suivante:
- en translation: projection sur le plan des mors. La composante sur la normale à ce plan s'annule.
- en rotation: projection sur l'axe normal au plan des mors. Seule la rotation autour de cet axe est inchangée, les autres s'annulent.

Il s'agit en fait d'une projection de l'hypervolume d'incertitude relative $I(\Re_1, \Re_{Rob})$ sur le plan des mors (cf. figures 3.18, 3.19).

Parallèlement, une liaison temporaire est créée dans le modèle entre C_1 et robot (objet principal de la structure rigide) et la position relative de C_1 par rapport au robot est enregistrée.

Si l'on veut alors connaître l'incertitude absolue $I(\Re_1, \Re_0)$ sur la position de C_1 , il faut combiner $I(\Re_0, \Re_{Rob}$ et $I(\Re_1, \Re_{Rob})$. Cette dernière restera - normalement - constante jusqu'au lâcher.

Posage de C_1 sur C_2 :

Le posage est terminé lorsque le contact entre C_1 et C_2 est plan. Avant le lâcher de C_1 , on peut vérifier, par calcul, si l'incertitude relative des deux objets est suffisamment faible (composition de $I(\Re_1, \Re_{Rob}), I(\Re_{Rob}, \Re_0)$ et $I(\Re_0, \Re_2)$.

Lâcher de C_1 :

La liaison entre C_1 et le robot, matérialisant la structure rigide temporaire, est détruite. On connait après le lâcher l'incertitude relative de C_1 par rapport à C_2 . L'incertitude en position selon l'axe portant $\overrightarrow{k_2}$ est nulle. L'incertitude en rotation autour des vecteurs $\overrightarrow{i_2}$ et $\overrightarrow{j_2}$ est nulle. Les autres composantes de l'incertitude restent constantes. C'est là encore une projection de l'hypervolume d'incertitude relative $I(\Re_1, \Re_2)$ sur la surface de contact (cf. figure 3.13).

3.2 Module d'entrée de SMGR

Le module d'entrée du système SMGR (Système de Modélisation Géomètrique et Relationnelle) est formé de deux modules. Le premier permet de définir des objets rigides (module géomètrique SMG); pour cela l'utilisateur dispose de primitives de construction et d'opérateurs géomètriques. Le second module permet de décrire des objets complexes (module relationnel SMR) comportant des liaisons articulées ou rigides.

SMGR comporte une version simplifiée, SMGRaphic permettant de créer un modèle moins complet compatible avec le logiciel graphique LISP-3D [Laugier 82b]. Notons cependant que ces modèles graphiques peuvent également être obtenus par des fonctions de SMGR à partir du modèle complet dans le cas d'une utilisation plus large des objets modélisés. Les deux versions du système sont d'utilisation similaire.

Une description détaillée du module d'entrée de SMGR peut être trouvée dans [Pertin 84].

3.2.1 Les objets élémentaires (module SMG)

Chaque objet élémentaire est construit selon un schéma hybride. Il peut être un volume élémentaire: un volume primitif (parallèlépipède, cylindre, ...) ou un volume obtenu par balayage. Il peut également être obtenu à partir d'opérations géométriques sur des objets existant (coupe, collage...). Contrairement aux modèles CSG basés sur la même approche, notre modèle est résolu au moment de la construction puisque la représentation CSG est difficilement exploitable pour les calculs géométriques. On transformera donc la structure arborescente correspondante en une représentation BR.

Tout objet élémentaire créé par SMG possède un repère cartésien qui lui est propre (repère de construction). L'opération de création consiste: (1) à définir les volumes élémentaires, (2) à leur associer des repères et (3) à les combiner ensemble à l'aide des opérateurs géométriques fournis par le système.

On trouvera en Annexe A l'exemple d'une telle création.

3.2.2 Les structures articulées (module SMR)

Le logiciel SMR permet de modèliser les chaînes cinématiques ouvertes. Pour cela, l'utilisateur peut positionner deux corps C_i et C_{i+1} (précèdemment crées par SMG) l'un par rapport à l'autre. Le mode opératoire utilisé comporte deux étapes:

1. La définition du type de liaison utilisé (rigide, rotoïde, prismatique,...)
2. La description des paramètres de la liaison (position, degré de liberté): le positionnement des deux corps est réalisé par une caractérisation de la position et de l'orientation du repère $\Re_{i+1}$ par rapport au repère $\Re_i$.

Les informations relatives aux positions et orientations sont traduites en termes des quatre paramètres de Denavit-Hartenberg $(\alpha_i, a_i, \theta_i, r_i)$ d'une matrice constante (cf. Annexe B).

Nous présentons dans l'Annexe A des exemples de modélisation de mécanismes articulés.

3.3 Implantation

SMGR est écrit en MACLISP et est implanté sur un calculateur CII-HB-70.

Le choix du langage LISP a été essentiellement motivé par la structure de listes permettant de représenter aisément les structures arborescentes qui sont à la base de notre modèle.

De plus, la possibilité d'utiliser des listes de propriétés LISP est tout à fait appropriée à la définition des liaisons inter-objets. Dans l'implantation du modèle

relationnel, la structure arborescente est en effet représentée par des listes et les valeurs de transformations ou autres informations portées par les arcs sont enregistrées sous forme de propriétés associées aux atomes représentant les nœuds de l'arborescence. L'information relative aux incertitudes sera elle aussi codée de cette manière.

Certains des nœuds formant l'arborescence sont engendrés automatiquement par le système (par des fonctions de type GENSYM) lors de la création de structures éphémères.

Les listes de propriétés permettent également l'attachement de caractéristiques physiques aux feuilles de l'arbre et facilite la coexistence de plusieurs types de représentations. Elles permettent, d'autre part, d'avoir des modèles aisément extensibles ou modifiables.

L'assimilation données/programme facilite la création d'articulations passives par l'association au degré de liberté passif d'une fonction dépendant du degré de liberté actif.

En outre, le choix du langage LISP a été fait dans l'idée d'une connection facilitée avec les modules de raisonnement géométrique ou plus généralement avec tout logiciel d'Intelligence Artificielle écrit en LISP.
Le module d'entrée de SMGR est connecté avec le logiciel graphique tridimensionnel LISP3D qui permet de travailler sur des postes graphiques divers: terminal graphique RAMTEK 6211, terminaux compatibles TEKTRONIX, terminal de synthèse d'images HELIOS/GETRIS.

Si l'utilisation principale de SMGR reste la programmation de niveau objet, il a également été connecté au système de simulation graphique ISR [Laugier 84] permettant ainsi de valider le modèle relationnel. Deux films de simulation de tâches d'assemblage décrites en langage LM ont été réalisés sur des supports vidéo ainsi que sur des films 16mm (cf. figure 3.20).

3.4 Limitations

Les limitations évidentes de notre système sont dûes à la pauvreté du module d'entrée qui a assez peu à voir avec les systèmes de CAO usuels. Si l'on arrive à représenter grâce à notre modèle une classe d'objets relativement importante, la phase d'entrée est assez fastidieuse. Ce point n'est cependant pas très important puisque SMGR pourrait être interfacé avec un système de CAO puissant (ROMULUS ou EUCLID par exemple); les représentations utilisées en entrée sont, en effet, de même nature. Les données de SMGR, en ce qui concerne les objets élémentaires, pourraient également être construites grâce à l'utilisation d'un capteur visuel; une telle approche a déjà été abordée au LIFIA dans [Borianne 84], qui utilise un cap-

teur de vision active 3D reposant sur l'émission d'un plan lumineux produit par un laser. D'autres laboratoires s'y sont également intéressés (par exemple l'INRIA [Boissonnat 85]).

Au niveau du modèle à proprement parler restent deux points noirs. Le premier concerne la représentation qualitative des relations qui reste à l'heure actuelle assez pauvre et dont l'exploitation pour maintenir la cohérence du modèle relationnel n'est pas encore abordée. Le second est l'absence de modélisation des tolérances; nous attendons pour développer un tel modèle d'avoir mieux conceptualiser leur rôle pour la programmation de niveau objet.

On peut également regretter l'absence d'informations géométriques de plus haut niveau directement utilisables dans les algorithmes de traitement géométrique (parallélisme, distance...). Ces limitations apparaitront plus clairement dans le contexte de l'automatisation de la saisie. Nous y reviendrons en §5.4.1 où une interface géométrique entre SMGR et le processus de raisonnement géométrique sera présentée.

Dans le contexte de la programmation automatique, on peut accepter le postulat de ***modélisation a priori*** de l'univers. En effet, dans ce cas l'univers de manipulation est supposé controlé complètement par le système qui peut en maitriser l'évolution (aux aléas du monde physique près). Dans le contexte plus général du robot autonome, il est complètement irréaliste de poser un tel postulat car le robot n'est que l'un des agents d'un univers évoluant en grande partie indépendamment de lui. L'apport de la modélisation dynamique de l'environnement est alors fondamental [Crowley 85]. Cette modélisation repose sur un apprentissage actif de l'univers basé sur l'utilisation de capteurs divers. La difficulté de cette modélisation dynamique provient essentiellement de l'extraction de la notion d'objets à partir des données de capteurs. Si ceci ne remet pas en cause les besoins en modélisation liés à la Robotique, il est essentiel de développer des méthodes permettant de lier modèles dynamiques et modèles de type CAO; de telles recherches débutent au LIFIA.

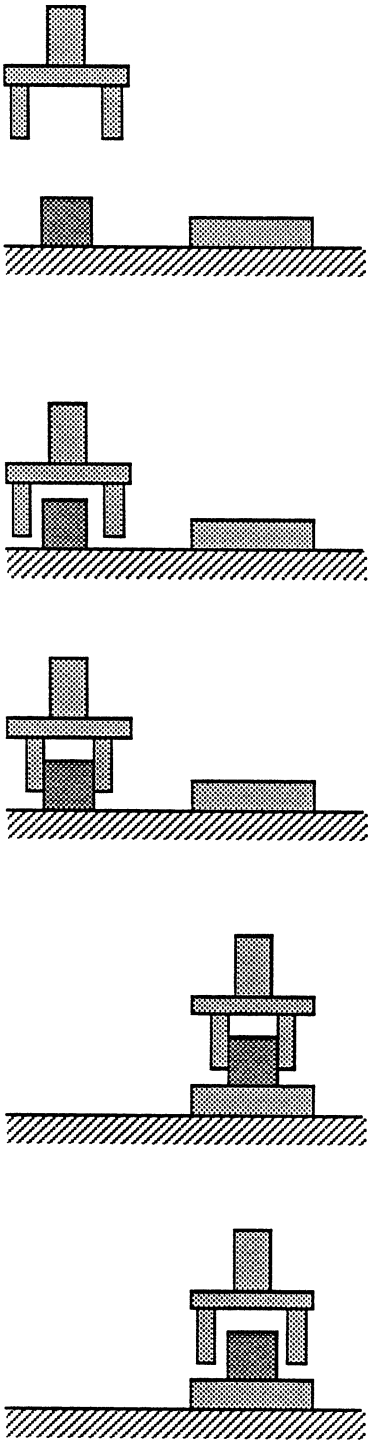


Figure 3.15: Manipulation nominale

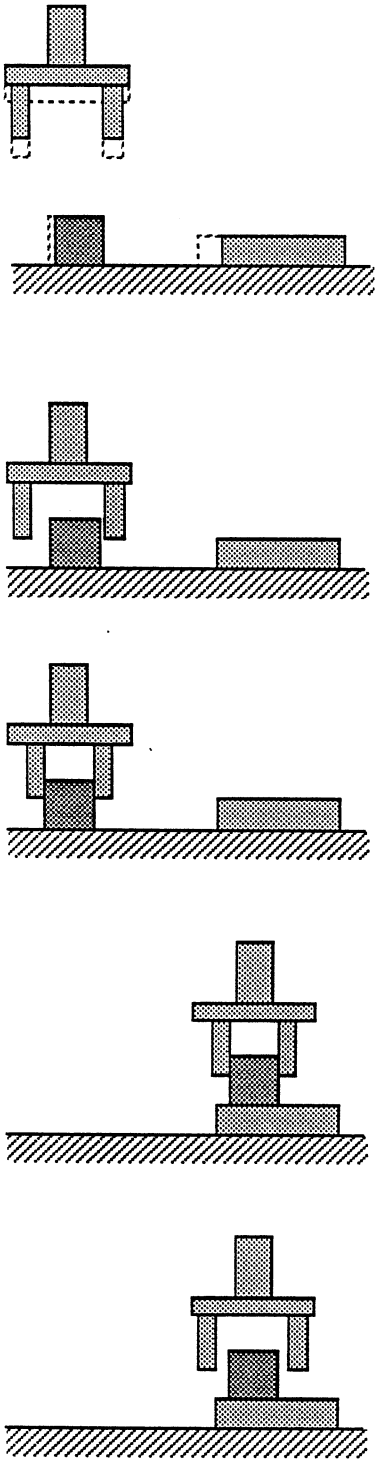


Figure 3.16: Manipulation réelle

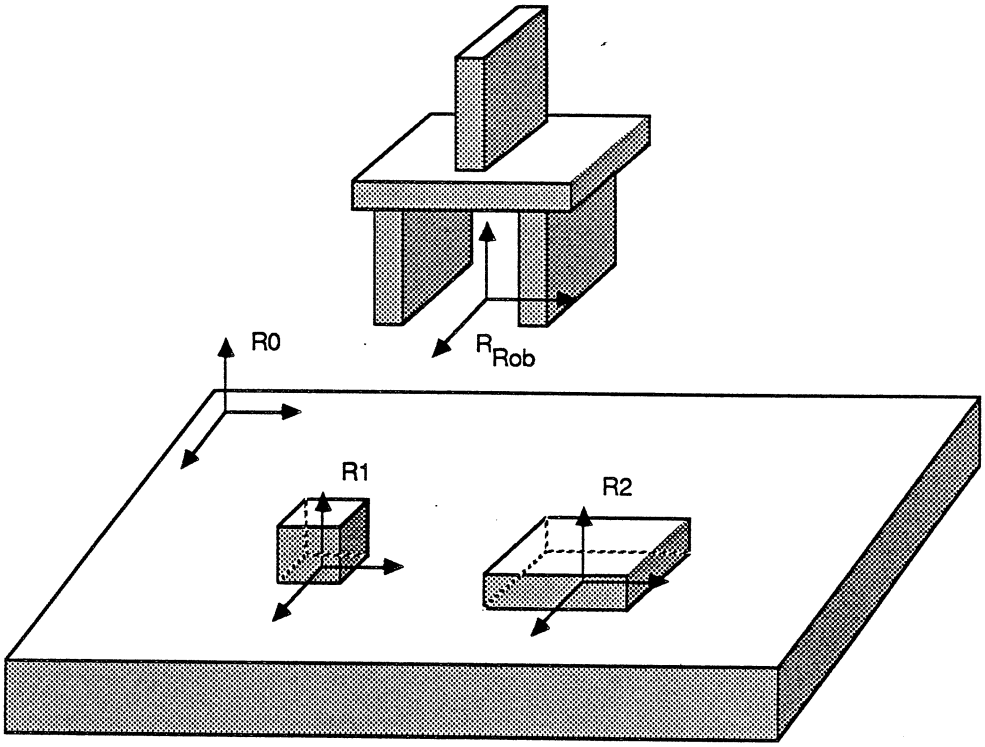


Figure 3.17: Les objets manipulés et leur localisation dans l'espace de travail

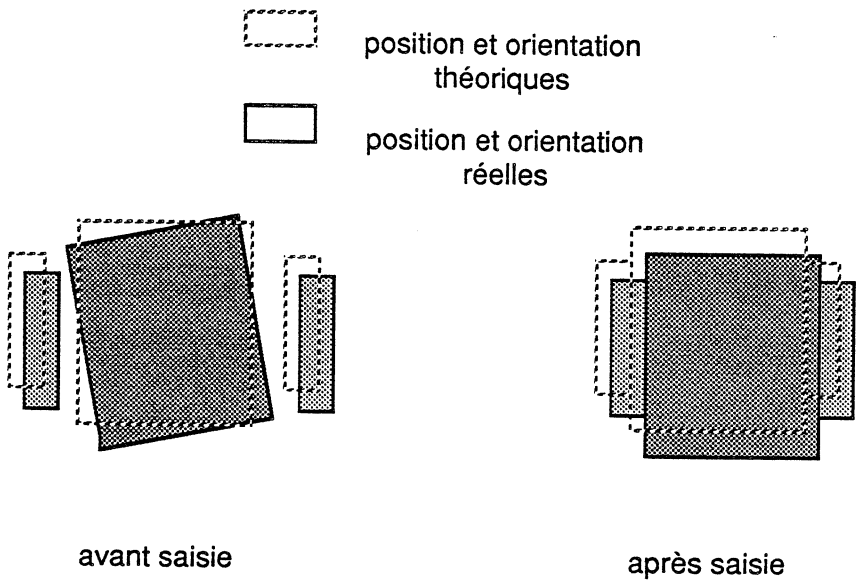


Figure 3.18: Modification des incertitudes de position lors de la saisie (1)

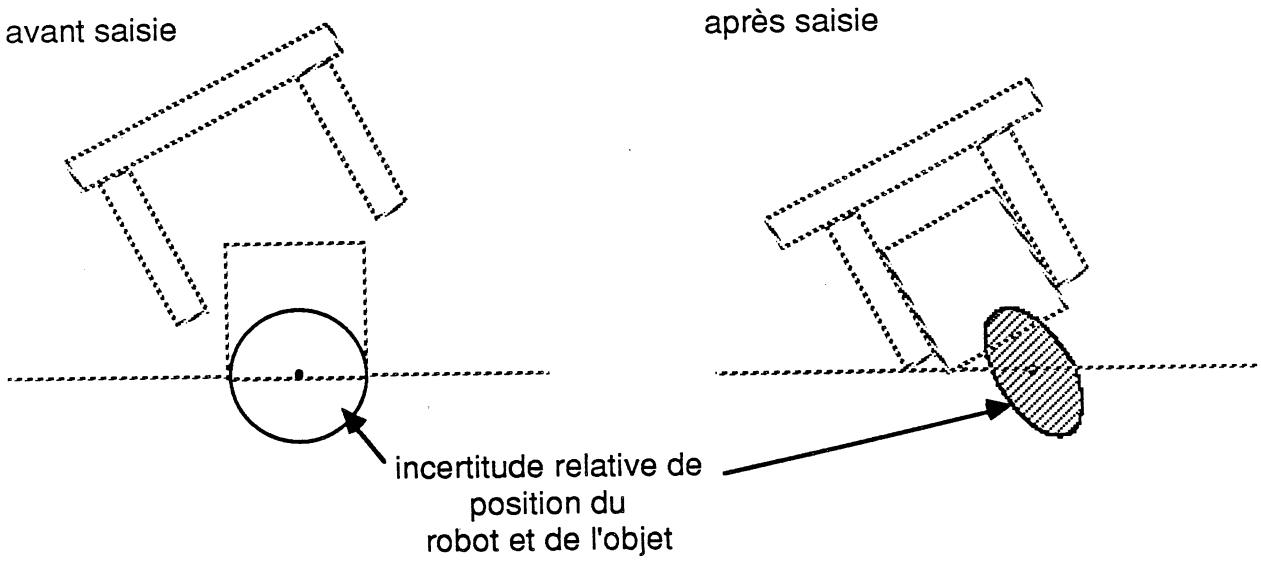


Figure 3.19: Modification des incertitudes de position lors de la saisie (2)

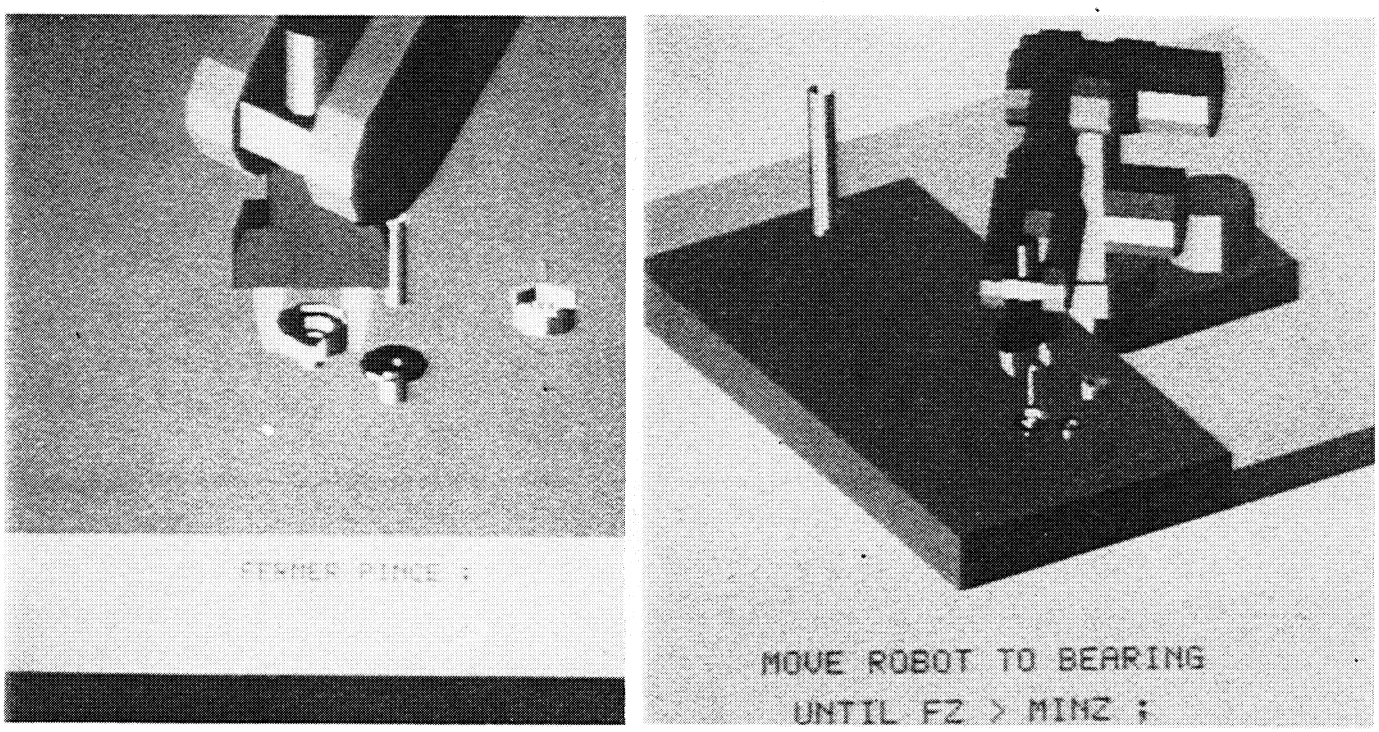


Figure 3.20: Une utilisation de SMGR: la simulation de programmes de manipulation

Cette partie de la thèse aborde le problème de l'automatisation du raisonnement géométrique dans le cadre de la réalisation d'un système de programmation de niveau objet par le biais d'une étude de cas approfondie: la saisie d'objets.

Outre son intérêt évident dans le contexte général de la modélisation du raisonnement géomètrique et spatial, la saisie est comme nous le verrons par la suite, l'un des problèmes clés pour la réalisation d'un système de programmation de niveau objet.

Après avoir analysé les diffèrentes approches de l'automatisation de la saisie (développement d'outils spécifiques, résolution logicielle), nous exposerons notre méthode de résolution et ses originalités.

Nous avons développé notre méthode pour la saisie à l'aide d'une pince à deux mors parallèles. Néanmoins, nous en tirerons un enseignement quant à l'élaboration d'une méthode générale de saisie basée sur le raisonnement géométrique et spatial.

Chapitre 4

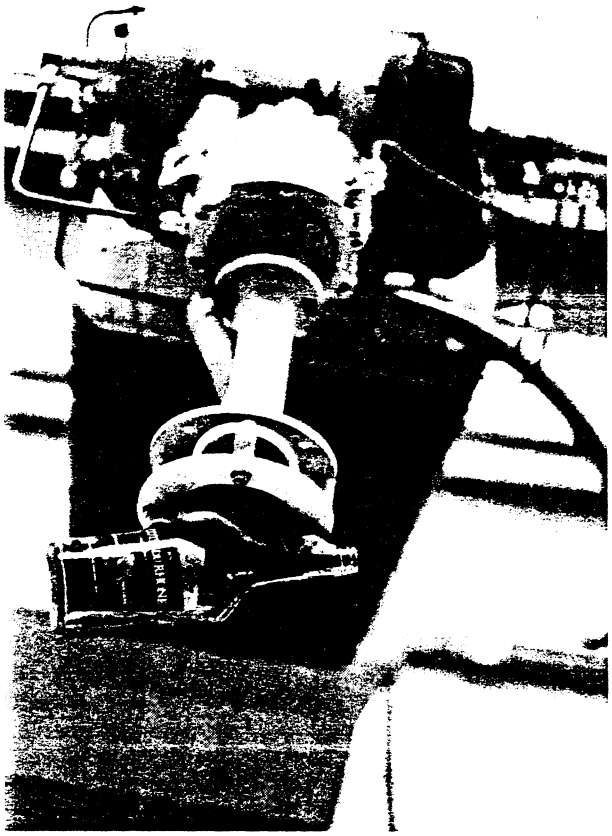
Analyse bibliographique de la saisie

Les travaux concernant la saisie peuvent être scindés en deux ensembles: le premier contient toute recherche ou réalisation liée au choix de l'outil de préhension (ce problème est dû à l'absence d'outils à la fois polyvalents et industriellement viables); le second type de travaux concerne le processus décisionnel lié au choix de l'action de saisie et de ses différents paramètres.

Bien que notre travail se raccroche à ce second ensemble, nous examinerons successivement les travaux les plus significatifs de ces deux aspects complémentaires du problème de la saisie.

4.1 Choix de l'outil

On peut distinguer deux classes d'outils résultant de deux approches différentes selon qu'elles sont tournées vers l'industrie ou la recherche. La première classe est directement liée à l'automatisation de la production. Celle-ci s'est portée dans un premier temps sur l'automatisation lourde de chaines de production de grande ou moyenne série. Les outils issus de cette automatisation ont souvent été des outils "ad-hoc" adaptés à la pièce (ou classe de pièces) produite(s) et ayant de ce fait, en général peu de versatilité. Peu à peu, les besoins de flexibilité ont amené la conception d'outils un peu plus génériques; certains sont liés à un mode préhension unique que nous appellerons adhérence (ou succion): les ventouses, préhenseurs magnétiques et préhenseurs à éléments expanseurs. Ce dernier type d'outil comporte généralement une membrane souple que l'on gonfle à l'air comprimé et qui s'adapte à la conformation de la pièce; celle-ci est maintenue en position en faisant le vide dans la membrane qui agit alors comme une ventouse. L'air est quelquefois remplacé par des billes métalliques, le tout étant rigidifié sous l'effet d'un champ magnétique [Bancon 82] (cf. figure 4.1). Un autre mode de saisie, le serrage, est lié à une grande classe d'outils appelés pinces; celles-ci sont généralement assez diverses, le nombre de mors et leur géométrie étant plus ou moins spécifiques des objets



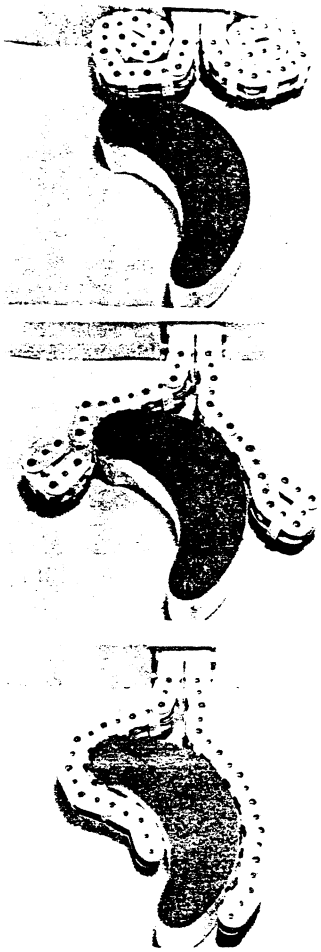


Figure 4.1: Saisie par adhérence [Bancon 82]

Figure 4.2: Saisie par serrage [Hirose 77]

manipulés. Parmi ces pinces, certaines présentent des mors flexibles ("mécanismes mous" [Hirose 77]) qui s'adaptent à la géométrie de la pièce et permettent de lui appliquer une pression constante (cf. figure 4.2).

Tous ces outils restent cependant assez spécifiques par rapport aux objets manipulés ou aux possibilités de prises. Les efforts de la recherche ont donc porté sur la conception de ce que l'on peut appeler des "outils universels" ("dextrous hand") assurant un maximum de versatilité par rapport aux objets et aux tâches considérés.

L'apport de la recherche en ingénierie bio-médicale pour la conception de prothèses a été important [Jacobsen 82], [Ignazi 78] essentiellement au niveau des technologies mises en œuvre. La conception de prothèse de bras comporte en effet des impératifs stricts (économiques, de poids, de facilité de commande...). Les prothèses de main sont du fait de ce dernier point très simplifiées (pinces d'aspect plus ou moins esthétique - cf. figure 4.3): il est bien évident que les ersatz de com-

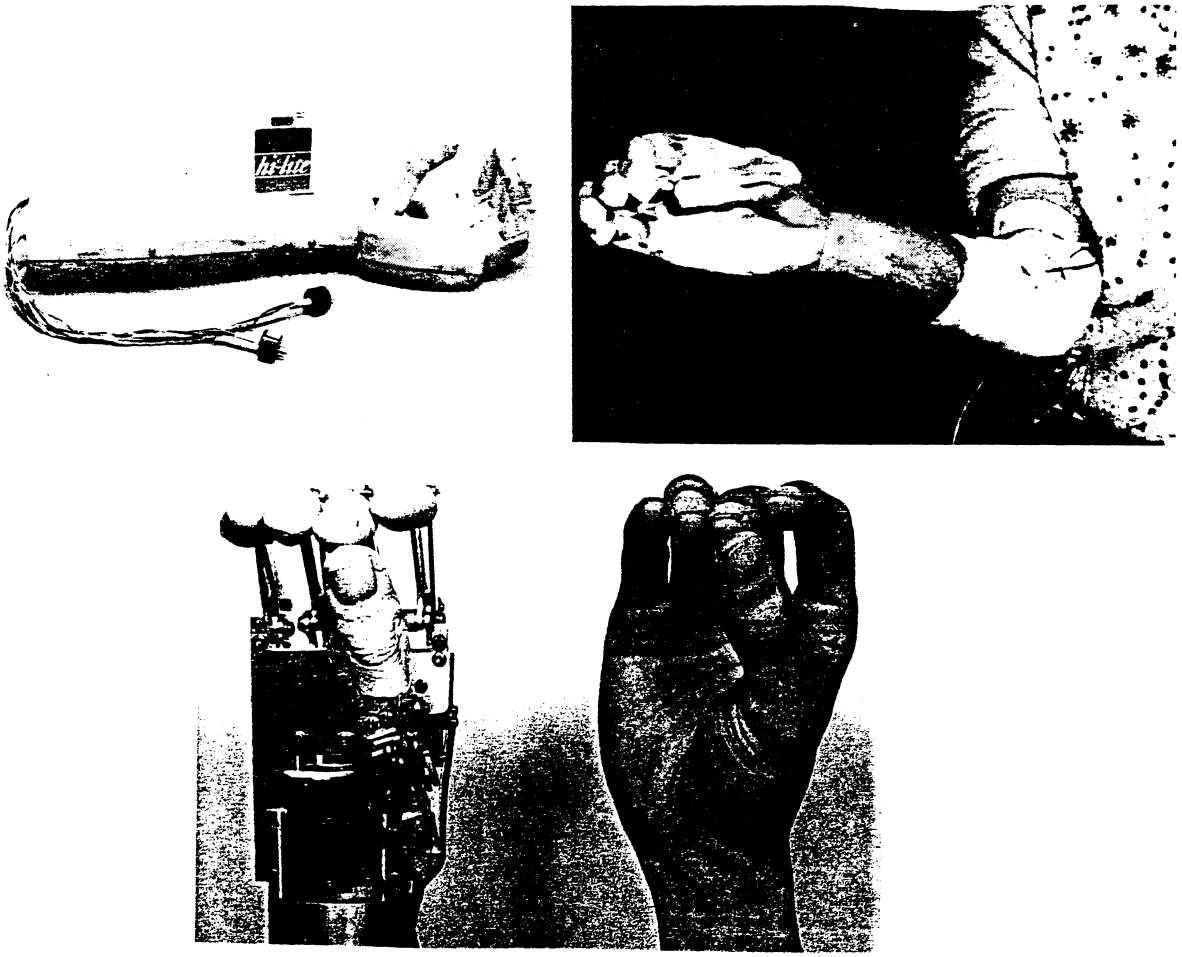


Figure 4.3: Prothèse de main

mande (myoélectrique, vocale, mécanique...) dont dispose un homme handicapé ne permettent pas de commander des mains multi-doigts, multi-phalanges. Cependant de nouvelles voies de recherche fondées sur l'utilisation de capteurs peuvent radicalement modifier cet état de faits. Un tel exemple de main est développé à Montpellier [Seroussi 82]; cette main est équipée d'une peau sensible permettant de contrôler, sans intervention de l'handicapé, l'ajustement final de l'objet dans la main. Une autre approche consiste à associer la commande de la prothèse au mouvement du globe occulaire accompagnant le geste de saisie; cette démarche qui est tout à fait homologue à celle de la saisie réelle permettra sans aucun doute de concevoir des mains artificielles au prix d'une miniaturisation du hardware nécéssaire.

L'approche "robotique" est différente dans le sens où l'outil conçu peut avoir toute la flexibilité des moyens mis en œuvre puisque les problèmes d'encombrement matériel ne se posent pas avec la même importance. Toutes sortes de mains ont vu le jour essayant de copier l'organe humain, la tendance générale étant à développer des mains à trois doigts assurant une assez grande classe de prises et une bonne stabilité de l'objet. Citons par exemple les mains développées par Okada [Okada 77],

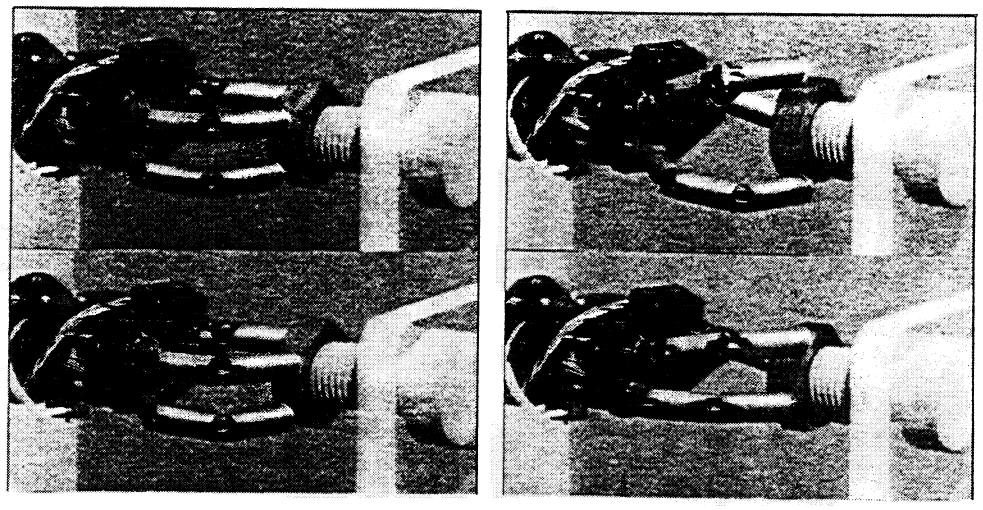


Figure 4.4: Main développée par [Okada 77]

[Okada 79] (cf. figure 4.4) et par Hitachi [Jourob 84] (cf. figure 4.5). Cette dernière est intéressante technologiquement parlant car l'actionnement des trois doigts est effectué grâce à un ensemble de fils en alliage (titane/nickel) à mémoire (Shape Memory Alloy); de tels fils tendus par des ressorts en position de repos de la main présentent l'intéressante propriété de revenir à une longueur donnée par échauffement électrique; il n'y a de ce fait pas d'actionneur et la main présentant de bonnes performances (rotation de 90 degrés par seconde) reste très légère et peu encombrante. Elle présente cependant quelques inconvénients dûs à un long refroidissement des fils; la main retrouve de ce fait assez lentement sa configuration initiale [High Tech 85].

Une autre réalisation, la main à cinq doigts développée de concert par l'UTAH University et le M.I.T. [Jacobsen 82], [Jacobsen 84a], [Jacobsen 84b], utilise des tendons en matériaux composites (Kevlar/Dacron) à longue durée de vie actionnés pneumatiquement (cf. figure 4.6).

Un autre type d'outil á trois doigts moins anthropomorphique a été développé par [Guinot 85]. Cet outil présente l'intéressante caractéristique d'être équipé d'une commande évoluée: déplacement conjoint des doigts dans un répertoire de tâches données, commande en force.

A l'opposé, la commande des mains précédentes, se fait principalement par l'adressage de consignes de position à chacune des articulations; parfois, il est également possible d'utiliser une commande en force ou de faire appel à une bibliothèque de configurations clés de la main. C'est de toute façon une commande d'assez bas niveau.

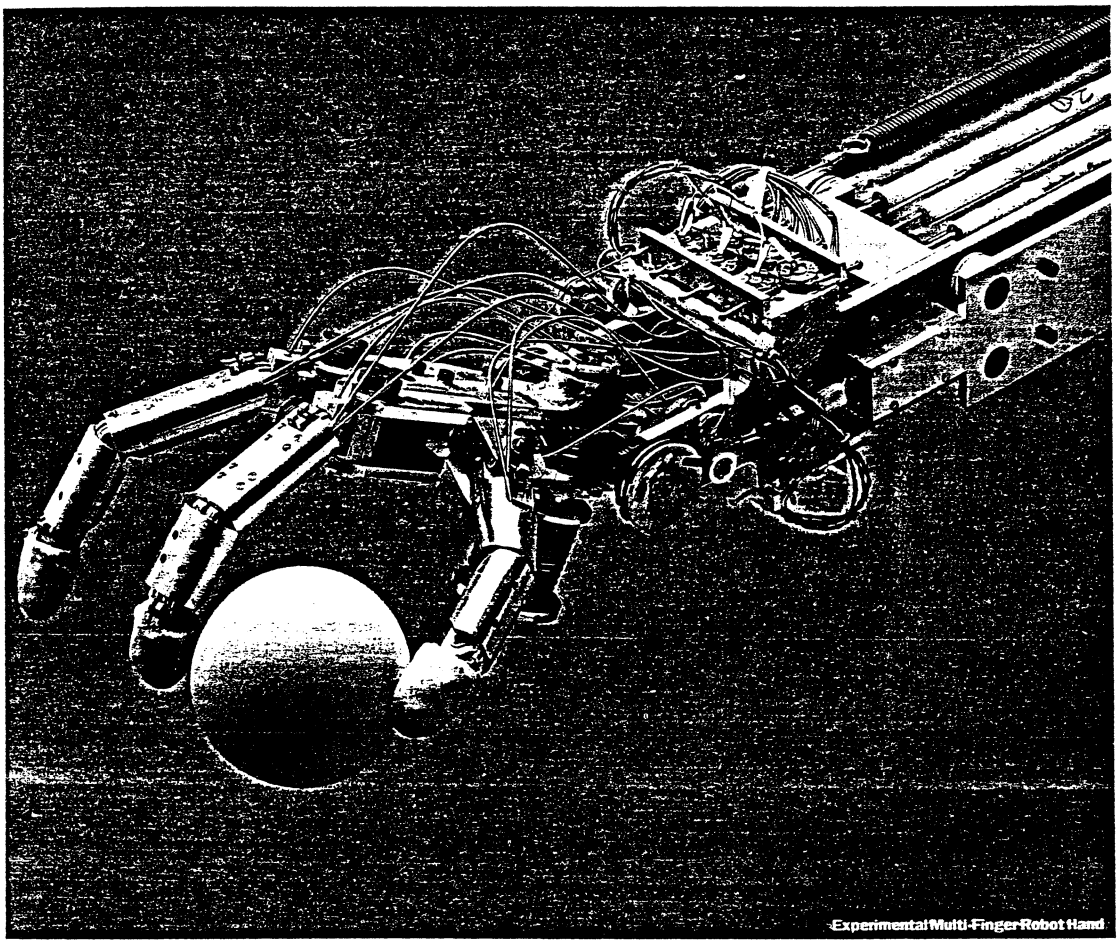


Figure 4.5: Main développée par Hitachi

Toutes ces réalisations semblent cependant oublier de se poser quelques bonnes questions: le problème du choix de la commande en est une mais le choix d'une morphologie particulière en est une autre encore plus importante. Les réalisations existantes ont en effet trouvé leur motivation dans un désir de reproduire l'organe humain ou un ensemble de configurations de la saisie humaine; il nous parait extrêmement important de se poser la question du choix de l'organe de saisie par rapport aux tâches que l'on veut effectuer avec cet outil.

A ce titre, une approche intéressante de la conception fonctionnelle d'un outil de préhension est présentée dans [Salisbury 82]. L'analyse descendante est fondée sur une étude de stabilité et une phase d'optimisation. La première phase permet d'adopter une configuration particulière de main (nombre de doigts et de phalanges) en fonction de la mobilité désirée d'un objet dans la main; la seconde phase détermine les paramètres physiques de la main (positionnement des doigts, dimensionnement, choix des degrés de liberté...) grâce à divers critères d'optimisation: prises potentielles, espace atteignable par les doigts, risques de collisions des doigts... Cette

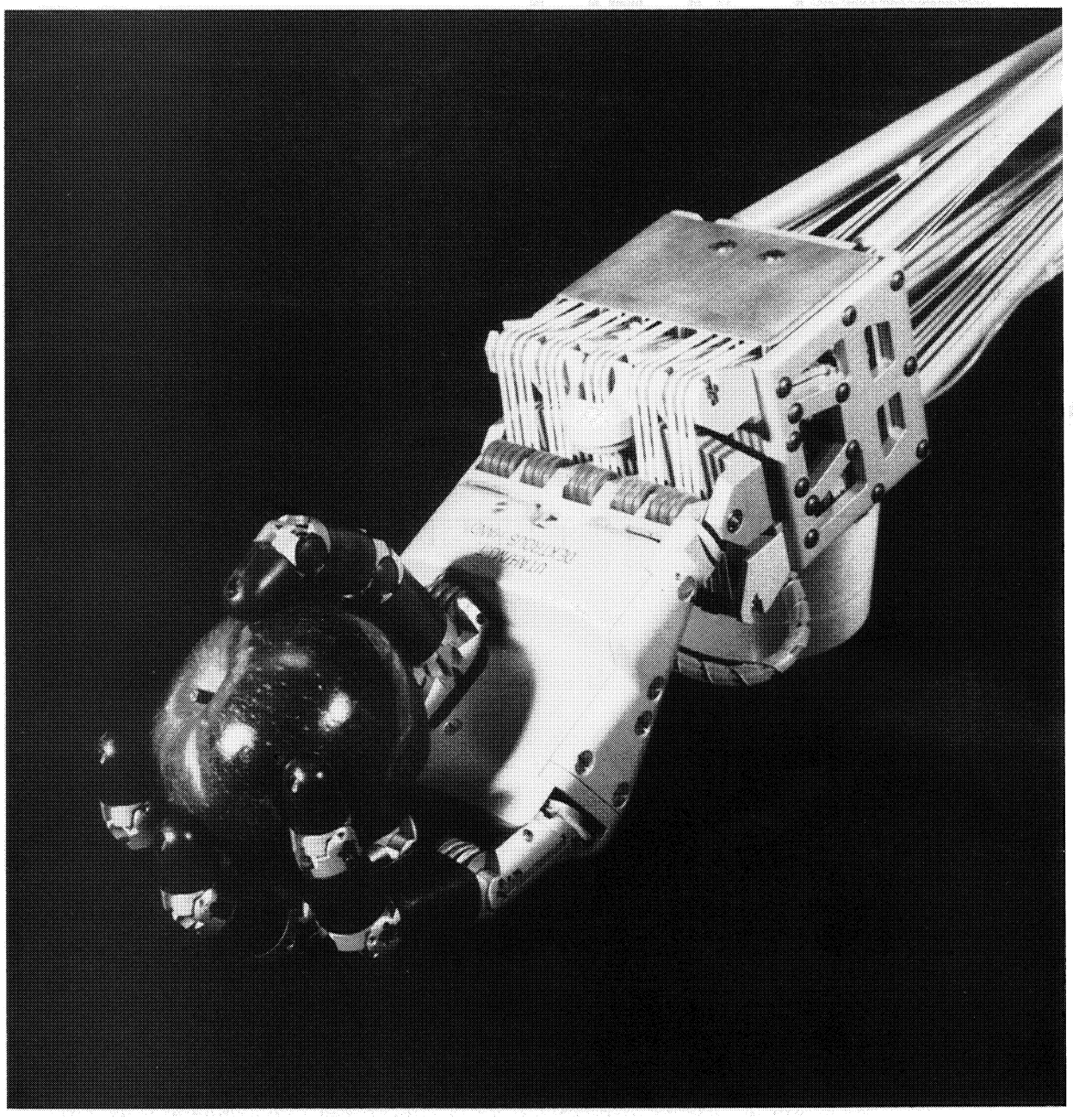


Figure 4.6: "UTAH/M.I.T. Dextrous hand"

recherche a conduit le laboratoire d'Intelligence Artificielle de Stanford University et le Jet Propulsion Laboratory a concevoir un outil de préhension à trois doigts, chacun de ces doigts ayant trois degrés de liberté (cf. figure 4.7); l'actionnement de ces doigts se fait à l'aide de tendons; les entrées sont les positions des extremités digitales ou la valeur des variables articulaires et des consignes de forces.

Bien que la dernière approche soit à distinguer des précédentes, le point faible évident est le manque d'intelligence de ces merveilles mécaniques (ou plutôt de leur système de commande). Il est, en effet, actuellement impossible d'utiliser toute la versatilité de tels organes grâce à une commande de haut niveau; c'est en ce sens que doit évoluer une partie des recherches. Une autre direction de recherche intéressante est suggérée dans [Himeno 83]. Cette suggestion est basée sur l'examen de la commande du doigt humain; il est remarquable de constater que celle-ci n'est pas une simple commande hybride en force et position; un troisième type de commande existe, que les auteurs appellent "tonus control". Cette possibilité est dûe à la présence, dans chacun des doigts, d'actionneurs redondants jouant des rôles antagonistes (cf. figure 4.8); cet antagonisme permet de modifier la rigidité du doigt sans en changer les commandes en force et position. C'est cette troisième commande qui nous permet de manipuler une très grande variété d'objets (en poids essentiellement). Les objets très lourds étant maintenus, non pas par l'application de forces intenses (comme c'est généralement le cas en Robotique) mais par une rigidification des articulations. Cette approche, qui affleure dans les systèmes robotiques incluant des articulations passives (structure à parallèlogramme par exemple) nous semble prometteuse dans le cadre de la conception d'un outil de préhension à caractère universel. Il nous semble que c'est d'ailleurs là, un des intérêts essentiels à développer des mains dont les doigts soient des structures articulées.

4.2 Choix de l'action

Considèrons le problème de choix de l'outil résolu; le problème décisionnel de saisie se pose alors en de multiples termes; dans le cas d'un objet, d'un outil et d'une tâche donnés, ces termes sont les suivants:

1. Quelle prise choisir (i.e. comment saisir l'objet)?
2. La prise choisie est-elle accessible?
3. La prise choisie est-elle en équilibre stable (i.e. l'objet ne risque-t-il pas de bouger dans la main)?
4. la prise choisie est-elle compatible avec le contexte de manipulation (i.e. permet-elle de réaliser la tâche spécifiée)?

Plusieurs méthodes ont été développées afin de traiter avec plus ou moins de succès les sous-problèmes correspondant: (1) la recherche de prises, (2) l'étude d'accessibilité, (3) l'étude d'équilibre et de stabilité et (4) l'étude de compatibilité

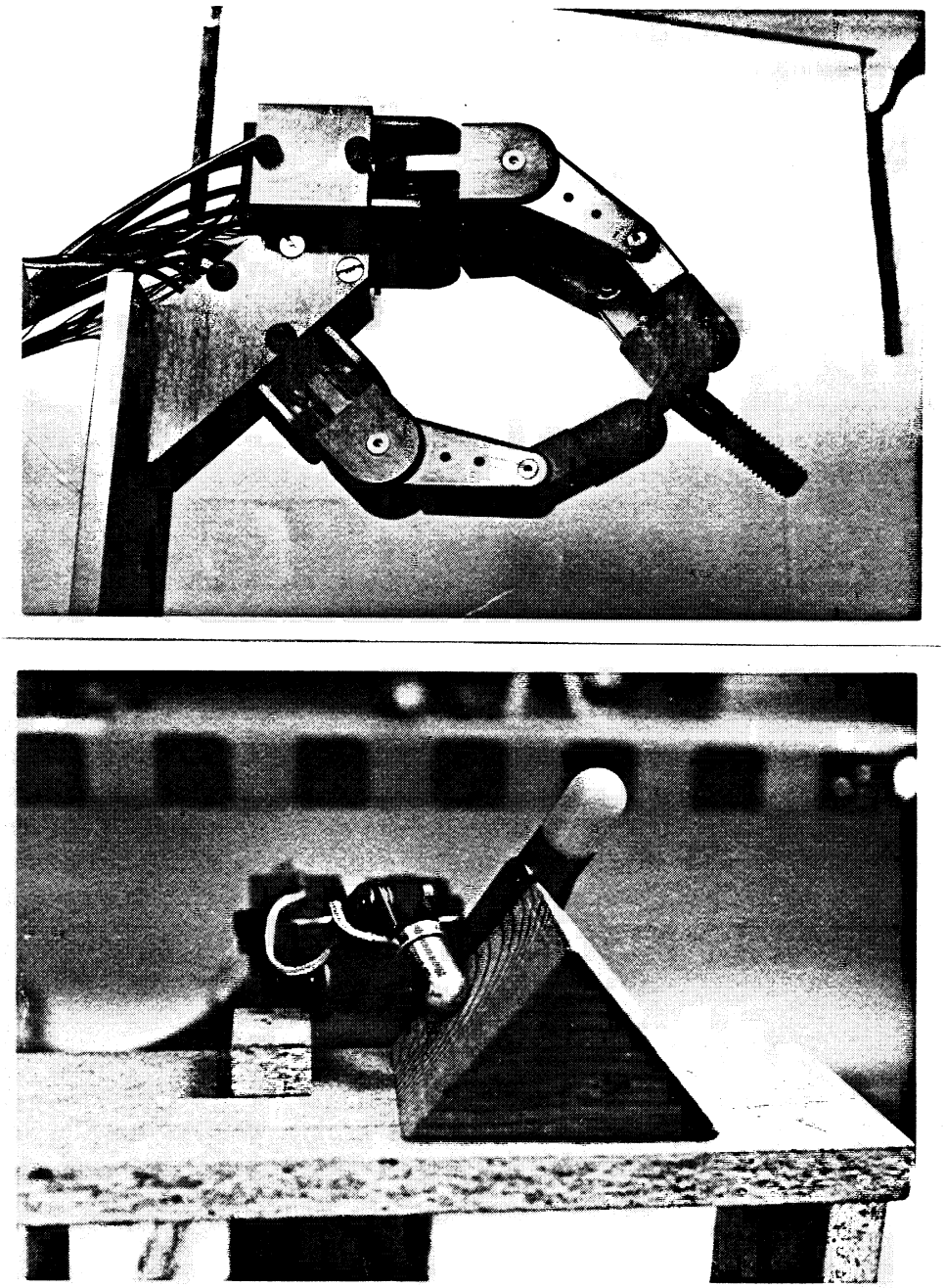


Figure 4.7: "Stanford/J.P.L. Dextrous hand"

contextuelle.

Ces méthodes utilisent deux types d'approches correspondant à deux problèmes distincts de la manipulation. On peut en effet distinguer deux phases avant la ma-

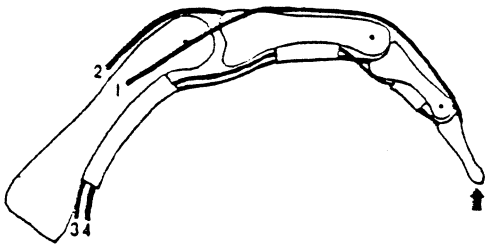


Fig.1
Modelling of a finger, lateral view.
1:Intrinsic muscles. There are two muscles (1.R and 1.L). 1.L is hidden at the opposite side. See horizontal view at Fig.4.

2:M. extensor digitorum

3:M. flexor digitorum profundus

4 : M. flexor digitorum superficialis

Points at the joints indicate centers of rotation.

Figure 4.8: Structure tendineuse de la main

nipulation proprement dite: la première a pour but d'amener l'objet de son lieu de stockage a son lieu de montage. Il est clair que dans un contexte industriel, cette phase n'est pas considérée lorsque l'on dispose d'un système d'alimentation de pièces (tapis roulant, palette, bol vibrant...). Cette opération de déstockage nécessite généralement l'exécution d'une opération de saisie en univers très contraint (généralement vrac ou semi-vrac).

La seconde opération consiste à saisir l'objet considéré en vue d'une manipulation particulière. Elle nécessite la prise en considération du contexte de manipulation (opération à effectuer avec l'objet et environnements initial et final de la manipulation). Nous appellerons cette phase, "saisie sous contexte".

A l'opposé de la phase de saisie sous contexte, la phase de destockage n'a pas besoin d'une connaissance précise du contexte, mais uniquement d'une connaissance de l'environnement de destockage. La prise engendrée n'a pas non plus besoin d'être connue de façon exacte. Cette phase se concentre sur le problème de recherche de prises, en étant quelquefois guidée par la recherche d'une réponse à la troisième question (la prise est-elle en équilibre stable?).

L'approche classique utilisée pour ce faire consiste à développer des logiciels opèrant sur des modèles construits à partir de données de capteurs. Ces modèles

sont imparfaits et incomplets: silhouette dans le cas d'un capteur optique, quelques arètes d'un objet dans le cas d'un capteur tactile.

Une des caractèristiques des méthodes utilisées pour résoudre ce problème d'acquisition d'objets est le droit à l'échec. Celui-ci vient quelquefois d'une mauvaise interprétation des données sensorielles; il provient plus souvent des limitations, intrinsèques au choix du capteur, imposées sur le type des informations disponibles.

Une telle approche n'est pas possible en ce qui concerne la phase de saisie sous contexte qui nécessite une parfaite connaissance du contexte de manipulation et de la prise choisie. Cette phase a pour objectif de répondre impérativement aux deux premières questions et autant que possible aux deux autres.

La puissance des méthodes mises en œuvre est évidemment liée au potentiel de modélisation associé. Il est clair que des modèles construits à l'aide de capteurs ne peuvent que fournir des réponses très partielles à nos quatre questions. Si une solution satisfaisante peut être engendrée grâce à un modèle de type CAO, seule une modélisation plus riche permet d'appréhender le traitement de la phase de saisie sous contexte dans toute sa complexité.

4.2.1 Recherche de prises

On peut ici distinguer deux approches complémentaires: la première consistant à utiliser des "bibliothèques" de prises, la seconde permettant d'engendrer un ensemble de prises potentielles par examen de l'objet à manipuler. Si la différence entre ces approches peut sembler subtile, disons qu'elle réside dans le degré d'explicitation des prises possibles.

A l'extrême, on associe au modèle d'un objet l'ensemble des endroits où l'on peut le prendre; on peut également associer ces prises à des "formes" caractéristiques de l'objet (cylindres, parallèlépipèdes par exemple). Enfin, on peut chercher à regrouper certaines des entités géométriques élémentaires (faces, arètes, sommets) de l'objet permettant de le saisir grâce à un outil donné; cette recherche est généralement guidée par des schémas-type de prise, associés à un outil particulier.

Dans le cas des bibliothèques de prises, l'"approche capteur" est assez peu utilisée; il s'agit, en effet, de reconnaitre un objet tri-dimensionnel généralement partiellement visible. Une telle approche est présentée dans [Ayache 84]; elle consiste à reconnaitre un objet parmi d'autres objets pouvant l'occulter partiellement et à choisir une prise pour cet objet dans une bibliothèque de prises; ce choix est fait grâce aux informations liées à l'accessibilité fournies par l'image. Dans le cas de l'utilisation d'un support de modélisation de type CAO, un ensemble de prises est associé à des volumes élementaires de l'objet; les modèles sont généralement assez pauvres: cylindres et parallèlépipèdes dans [Lozano 76], parallèlépipèdes seulement dans [Wingham 77] ou [Binford 79].

La recherche d'éléments de prise (second type d'approche) a donné lieu à de nombreuses réalisations fondées sur l'utilisation de capteurs. Elles sont généralement destinées à la saisie "hors contexte" dans des environnements très contraints (phase de déstockage des pièces).

A l'Université de Rhode Island, [Kelley 84] ont développé plusieurs méthodes fondées sur des algorithmes de vision pour acquérir des pièces manufacturées dans des vracs. Ces méthodes permettent la saisie de pièces orientées de manière aléatoire dans un environnement très contraint grâce à une pince de type "vacuum-cup" (préhenseur à éléments expanseurs) ou à une pince à deux mors parallèles. Dans le premier cas, un algorithme de vision permet de détecter des régions de texture homogène correspondant à des surfaces presque planes utilisables pour placer la pince "vacuum-cup" [Kelley 84]. Dans le second cas, ce sont des couples d'arètes opposées détectés dans l'image, qui sont transmises par la vision au système de saisie [Birk 81]. Hanafusa et Asada [Hanafusa 77a] utilisent également des informations de type visuel pour commander leur système de préhension. Nous verrons un peu plus loin que ce choix est guidé par une analyse heuristique de la stabilité du système main/objet. Boissonnat [Boissonnat 81], [Boissonnat 82] a développé et implanté une méthode permettant de calculer des paramètres de prises stables pour des objets situés dans un vrac. Pour cela, il extrait les entités morphologiques de l'objet, utilisables pour la saisie au moyen d'un capteur de vision tri-dimensionnelle à base de laser. D'autres types de capteurs sont utilisés dans des approches similaires: Todd [Todd 81] et Romiti [Romiti 81] se servent de capteurs tactiles, Espiau [Espiau 80] de capteurs de proximité et Ikeuchi [Ikeuchi 84] d'un système de vision stéréoscopique.

Les systèmes utilisant un support de modélisation fondent leurs méthodes sur une analyse morphologique de l'objet à saisir; cette analyse permet de choisir une prise parmi une liste de schémas types (par exemple: saisie face-face, face-arète, arète-arète ... dans le cas d'une pince à deux mors). Cette approche confère une assez grande généralité aux prises engendrées pour un outil donné (généralement une pince à deux mors parallèles). Elle a été introduite par [Laugier 81], [Laugier 83b] et reprise dans [Laugier 83a] et [Wolter 84], [Volz 83].

4.2.2 Etude d'accessibilité

Cette seconde phase consiste à déterminer si une prise choisie au pas précédent est effectivement réalisable, eu égard à l'encombrement spatial environnant. Elle permet généralement de mieux spécifier la prise considérée par application des contraintes spatiales imposées par l'environnement. Cette étude a donné lieu a assez peu d'implantations fondées sur l'utilisation de capteurs, ceux-ci (généralement vision bi-dimensionnelle) ne fournissant aucune information liée à l'accessibilité si ce n'est celle, évidente, des parties de l'objet visibles dans la direction de "regard" (accessibilité visuelle). Une seule implantation (à notre connaissance) utilise des

capteurs de vision pour extraire une information riche; [Ikeuchi 84] obtiennent, en effet, grâce à leur système de vision stéréo une information tri-dimensionnelle exploitable pour l'étude d'accessibilité.

Toutes les méthodes s'appuyant sur un support de modélisation reposent sur le même principe: on fixe (plus ou moins) les direction et orientation de l'organe de saisie et on détermine l'espace balayé par celui-ci au cours de ses mouvements; cet espace contient les obstacles susceptibles de s'opposer à ces mouvements.

L'outil (généralement une pince à deux mors parallèles) est décomposé en différents éléments et l'étude se fait pour chacun de ces éléments. Les différences essentielles des diverses méthodes résident d'une part dans la décomposition de la pince (et dans la propagation des solutions à tout l'outil) et d'autre part dans la recherche des obstacles à chacun des éléments précédents.

[Wolter 84] utilisent pour la recherche de ces obstacles une méthode classique de calcul d'interférences entre objets (cf. [Ahuja 80], [Boyse 79], [Comba 68], ou [Maruyama 72] pour plus de détail). L'approche introduite par Udupa [Udupa 77], et désormais classique grâce à Lozano-Pérez [Lozano 76] consiste à réduire un objet en mouvement à un point de référence et à rechercher une trajectoire parmi des obstacles modifiés en conséquence; cette approche, dont l'intérêt n'est plus à démontrer, a été reprise dans [Wingham 77] et [Laugier 83a]. Ces derniers (i.e. nous) réduisent le problème tri-dimensionnel à un problème bi-dimensionnel en approximant l'espace balayé par un ensembles de tranches (coupes par un plan); si l'utilité d'une telle méthode est discutable (et discutée: cf. [Wolter 84]) pour un schéma de pince classique (mors approximés par deux parallèlépipèdes), son intérêt sera clairement illustré dans ce manuscrit, pour un outil plus général (cf. §5.3).

La propagation des solutions pour chacun des éléments de la pince est généralement peu ou mal traitée [Wingham 77], [Wolter 84], [Lozano 76]. Les deux premiers considèrant uniquement les mors de la pince comme susceptibles de provoquer une collision, le second considèrant également le plat de la pince mais faisant une propagation de contraintes d'accessibilité qui nous semble incorrecte. En effet, dans ce dernier travail, Lozano-Perez travaille dans l'espace des configurations de l'outil (raisonnement sur X et Y dans le cas de la figure 4.9, raisonnement sur R et θ dans le cas de la figure 4.10). Les prises sont, elles, exprimées dans l'espace cartésien par rapport à un repère lié à l'objet. La méthode exposée passe sous silence les problèmes de transitions entre ces différents systèmes de coordonnées.

Dans tous ces cas, la pince est approximée par un ensemble de parallèlépipèdes permettant ainsi de résoudre implicitement le problème de la propagation des contraintes d'accessibilité, locales à une partie de l'outil. En effet, si l'on considère chacun de ces éléments d'outil comme un ensemble de tranches, rechercher les obstacles à un de ces éléments équivaut à rechercher les obstacles à chacune de ces tranches et à propager les solutions à tout l'élément: l'assimiler à un parallèlépipède

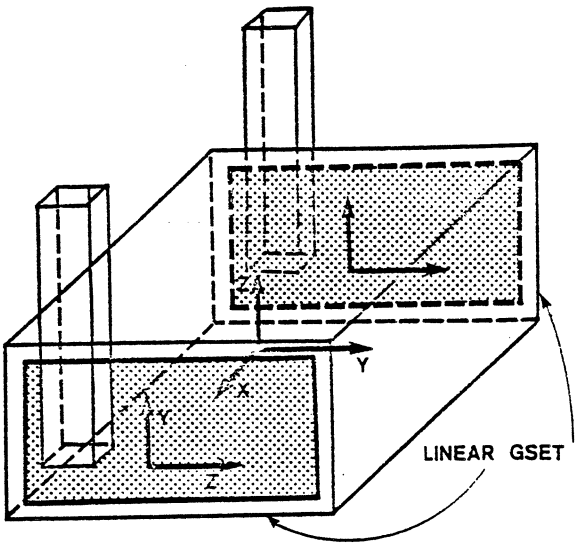


Figure 4.9: "Linear Grasping Set" [Lozano 76]

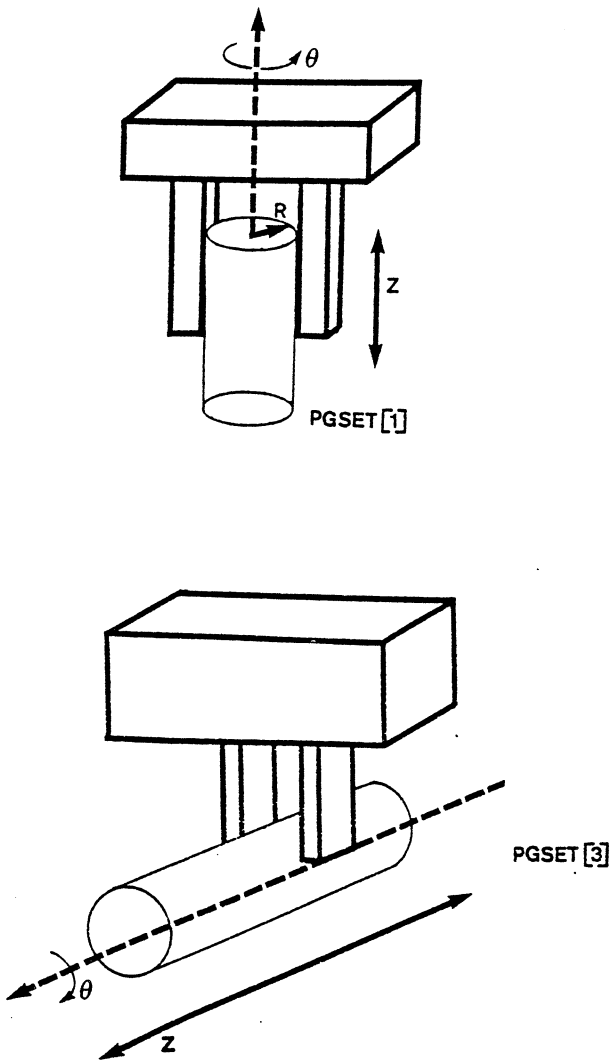


Figure 4.10: "Polar Grasping Set" [Lozano 76]

(ou à un objet pseudo-2D) correspond à un mécanisme de propagation trivial et permet donc de confondre les problèmes bi- et tri-dimensionnels.

Notre approche déjà utilisée dans [Laugier 83a] associant une modélisation de l'outil par points de référence et une discrétisation de l'espace balayé se distingue (par rapport au mécanisme de propagation) des autres méthodes citées; elle permet d'effectuer une étude plus fine des problèmes d'accessibilité et de considérer des outils plus réalistes.

Citons également, une méthode un peu en marge, présentée dans [Lyons 85]

reprenant des principes utilisés pour résoudre le problème plus général de la recherche de trajectoires sans collision pour un manipulateur. Dans le cadre d'un système de commande adaptative de manipulateur, [Khatib 80] a développé un type de modélisation de l'environnement par champ de forces; la position à atteindre constitue pour le robot une source attractive, les obstacles et autres manipulateurs éventuels des sources répulsives, l'étude de ces potentiels permettant d'engendrer des trajectoires sans collision pour le manipulateur. De manière similaire, Lyons associe aux objets de l'environnement et à chacune des trois configurations clés de sa main (à trois ou cinq doigts) une modélisation de ce type lui permettant de l'amener vers l'objet à saisir et de la positionner sur cet objet.

4.2.3 Etude d'équilibre et de stabilité

Avertissement: Si dans ce paragraphe nous veillerons à distinguer étude d'équilibre et étude de stabilité, ces deux termes sont généralement regroupés sous le vocable général de "stabilité". La stabilité considérée ici recouvre la signification usuelle donnée à ce terme.

Principe de Le Chatelier (loi du déplacement de l'équilibre) [Yavorski]:
Si un système en équilibre stable est soumis à une action extérieure faisant sortir ce système de son équilibre, l'équilibre se déplace dans le sens de l'atténuation de l'effet de cette action.

Le problème consiste donc dans un premier temps a engendrer des prises équilibrées (l'objet est soumis à un ensemble de forces et de moments s'annulant en son centre de gravité) et dans un second temps à s'assurer que les prises conduisent autant que possible à des équilibres stables. Dans ce dernier cas, la présence d'incertitude sur la localisation relative de l'objet et de l'outil sera résolue par la stabilité elle-même.

Un obstacle majeur à l'évaluation exacte des forces et moments appliqués à l'objet, résulte d'une part de la mauvaise connaissance des directions et intensités des forces appliquées et de la difficulté inhérente à la prise en compte des phénomènes de friction.

Différentes approches tentent cependant de résoudre ce problème de façon formelle. Citons [Holzmann 85] qui, pour décider de l'équilibre statique d'un objet saisi, effectuent une étude des forces et moments appliqués dans le cas d'une saisie par trois points (les forces appliquées étant normales à la surface de l'objet).

Citons également la méthode de [Wolter 84] dont l'originalité est de considèrer, non pas des points de contact, mais des surfaces de contact et d'évaluer l'équilibre du système main-objet; nous en présentons les grandes lignes. [Wolter 84] considèrent trois types de mouvements parasites: le glissement en translation ou en rotation et le "twisting" (cf. figure 4.11).

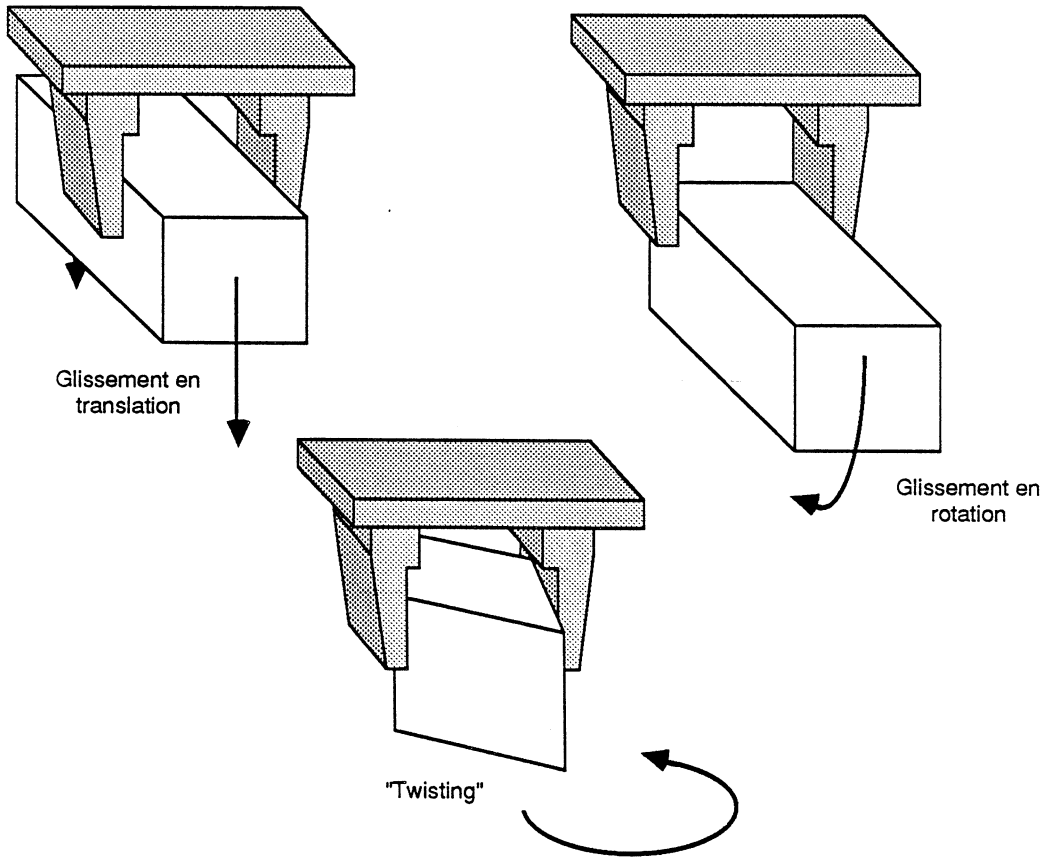


Figure 4.11: Différents types d'instabilité

Le glissement en translation dépend uniquement des forces appliquées par les mors et du coéfficient de friction statique entre l'objet et la pince; la mesure de cette instabilité ne constitue donc pas un critère d'ordonnancement sur les prises engendrées. Il peut cependant servir à déterminer simplement si la saisie d'un objet au moyen d'un outil donné est réalisable.

Le glissement en rotation de l'objet dans les mors dépend du torseur de force appliqué à l'objet, de la forme des contacts entre objet et outil, du coéfficient de friction entre ces deux corps et enfin de la pression appliquée à l'objet. Comme le montrent [Wolter 84], le calcul du torseur faisant tourner l'objet dans les mors est très complexe: un modèle du calcul de ce torseur est défini pour une surface de contact élémentaire; il faut ensuite déterminer aussi exactement que possible les surfaces de contact entre pince et objet et réaliser une intégration du modèle précédent sur ces surfaces. Le problème d'assymètrie de ces surfaces est un écueil supplémentaire pour cette évaluation.

Enfin, pour le "twisting", [Wolter 84] présentent, là encore, un modèle complexe d'évaluation de la stabilité nécéssitant entre autres des calculs d'enveloppes convexes.

Les paramètres traduisant les deux derniers types d'instabilité sont ainsi acquis par l'utilisation d'un modèle théorique assez rigoureux. Ces paramètres sont alors pondérés par des coéfficients empiriques pour être ensuite combinés au moyen d'une fonction heuristique. Les résultats fournis par cette fonction restent assez décevants (sic). C'est pourquoi Wolter, Volz, Woo portent leurs efforts actuels sur l'emploi d'une méthode plus rigoureuse en ce qui concerne l'exploitation de ces paramètres pour le choix de la prise.

Deux attitudes principales découlent de la difficulté d'évaluer rigoureusement l'équilibre et la stabilité des prises. La première consiste à utiliser des méthodes complètement heuristiques ignorant généralement les problèmes de friction (par exemple [Lozano 76], [Laugier 81], [Laugier 83b], [Hanafusa 77a]). La seconde approche repose au contraire sur la modélisation des problèmes de friction [Fearing 84], [Brost 85].

Dans le premier cas, les méthodes reposant sur un support de modélisation géométrique essayent d'engendrer des prises stables (en préférant des contacts plans [Lozano 76], [Laugier 81]), et équilibrées (en maximisant les surfaces de contact donc les forces de friction [Laugier 83a]).

Hanafusa et Asada présentent une méthode originale destinée à engendrer des prises stables pour la saisie d'objets bi-dimensionnels à l'aide leur outil de préhension comportant trois doigts élastiques (cf. figure 4.12). La méthode repose sur une recherche de minima d'énergie potentielle à partir d'un modèle bi-dimensionnel construit grâce à un système de vision. Ces minima sont recherchés, considèrant que le centre de la main est au barycentre du profil. Un minimum correspond à trois points du contour que l'on suppose correspondre à des régions stables du contour. Après avoir approximé ces régions par des arcs de cercles, les auteurs tentent réellement de minimiser le potentiel en cherchant des points de contact sur ces contours tout en conservant le centre de la main dans un voisinage du barycentre (pour des problèmes d'équilibre évidents).

Baker [Baker 85] présentent une variante de cette méthode pour un outil similaire; la différence essentielle réside dans un écartement variable des doigts permettant de résoudre les problèmes de glissements non considérés dans l'approche précédente.

La seconde classe de méthodes est, comme nous l'avons esquissé précédemment, basée sur une approche réaliste du problème considèrant que les problèmes de friction sont essentiels mais que la connaissance des forces est souvent erronée. Les travaux présentés dans le contexte de la saisie par [Fearing 84] et [Brost 85] sont une très

intéressante illustration des travaux de Mason sur la friction ([Mason 85] pour une approche intuitive, [Mason 82], [Mason 84] pour plus de détail et de rigueur).

L'originalité de cette méthode consiste à réduire les incertitudes de localisation de l'objet sans utilisation de capteurs mais par une suite de mouvements adéquats (cf. figure 4.13); ceux-ci sont engendrés par un étude des phénomènes de friction de l'objet sur le plan sur lequel il repose, les direction et intensité des forces de friction étant supposées inconnues.

Fearing applique cette approche au cas de la saisie d'un objet bi-dimensionnel pour un outil à deux doigts produisant des contacts ponctuels. Schématiquement, la technique consiste à positionner les deux doigts sur l'objet, l'un d'eux se trouvant dans le cône de friction et l'autre non. Le glissement provoqué par ce dernier provoque une rotation qui tend à ramener les deux doigts dans le cône de friction (création d'un équilibre). L'introduction d'un troisième doigt permet de contrebalancer les effets de la gravitation ou d'effectuer des changements de prises sur le même principe. Brost développe une approche similaire dans le cas de la saisie par un outil à deux mors plans. Nous invitons le lecteur intéressé à se reporter à [Fearing 84], [Mason 84] et [Brost 85].

4.2.4 Etude de compatibilité contextuelle

Dans le cadre d'une saisie intentionnelle (i.e. guidée par un but plus général que celui de tenir l'objet), il est crucial de tester la compatibilité de la prise choisie avec ce but.

Dans le domaine de la manipulation, cette compatibilité revêt deux aspects: un aspect lié à la description de bas niveau de la situation finale (description géométrique de l'environnement final par exemple), le second étant attaché à une description plus qualitative de la situation finale (par exemple: saisie d'un stylo en vue d'une opération d'écriture). Nous appellerons respectivement ces compatibilités de niveau objet et de niveau tâche.

La compatibilité de niveau objet est généralement traitée par les systèmes basés sur un support de modélisation, la complexité résultante n'étant qu'algorithmique. [Wolter 84] considèrent simultanément les objets des environnements initial et final comme obstacles potentiels et cherchent une trajectoire pouvant servir de trajectoire d'approche et de déproche. Lozano-Perez présente dans [Lozano 81] une méthode analogue; dans ce cas, l'algorithmique est cependant beaucoup plus complexe car Lozano-Pérez considère tout le manipulateur; il travaille pour cela, dans l'espace des configurations de celui-ci (ensemble des p-uplets décrivant l'espace atteignable d'un manipulateur à p d.d.l.). Les configurations du manipulateur dans les environnement initial et final de saisie étant différentes, les obstacles de l'environnement final doivent être transformés avant d'être réunis avec ceux de l'environnement initial. Les systèmes à base de capteurs n'abordent pas ce problème bien qu'il n'y ait là aucune impossibilité évidente; de tels systèmes sont plus généralement utilisés

pour un destockage des pièces (saisie non contextuelle).

La compatibilité de niveau tâche est à peine abordée dans les systèmes existants; c'est un aspect que l'on peut rattacher à une approche "fonctionnelle" de la saisie dans laquelle les critères de choix n'ont plus grand chose à voir avec les informations de bas niveau présentes dans un système usuel de modèlisation type CAO. Nous examinerons, pour illustrer ces propos, deux approches très intéressantes. La première d'entre elles, dûe à Hanafusa et Asada [Hanafusa 77b] permet d'ajuster les paramètres de la saisie en fonction de la tâche à accomplir, les prises potentielles étant engendrées de manière heuristique (cf. §4.2.3). Il s'agit principalement du choix effectif de la prise parmi les minima de potentiels et de la détermination de l'intensité des forces appliquées pour chacun des doigts. Ils répondent à un répertoire de tâches usuelles telles que les mouvements à compliance, la mise en position d'une pièce ou encore les opérations d'insertion.

La seconde méthode [Lyons 85] (cf. [Iberall 84] pour plus de détail, [Iberall 85] pour la justification anthromorphique de l'approche) repose une modélisation plus qualitative de la tâche. Elle se place dans le contexte de la saisie par un outil universel (main à trois ou cinq doigts). Cette approche est tout à fait similaire à l'approche supposée utilisée pour l'être humain [Beaubaton 85b]. Elle est d'ailleurs très fortement inspirée de travaux antérieurs sur la saisie humaine [Napier 56]. Un type de saisie (parmi trois) est choisi grâce à des critères portant sur l'objet et sur la tâche. Ces critères qualitatifs sont la forme (plutôt rond ou plutôt long) et la taille (plutôt gros ou plutôt petit) de l'objet, ainsi que la précision et la fermeté requises pour effectuer la tâche. Les différentes combinaisons de ces "paramètres" conduisent à trois types de prises (englobante, latérale ou de précision) permettant une préconfiguration de la main (cf. figure 4.14). Comme nous l'avons vu en §4.2.2, la suite de l'analyse repose sur une méthode de calculs de potentiels; elle pourrait tout aussi bien déboucher sur l'utilisation des autres approches (Lozano-Perez, Wingham, etc...).

Cette approche nous semble extrêmement intéressante parce qu'elle est la seule à aborder le problème de la saisie au niveau objet. Elle analyse en effet, la morphologie de l'objet non pas sur la base d' un examen analytique de sa géométrie locale comme la plupart des méthodes précitées mais au travers de caractéristiques globales ("vrai" 3D) influant directement sur la façon de prendre l'objet. Elle permet en outre de considérer la compatibilité de niveau tâche.

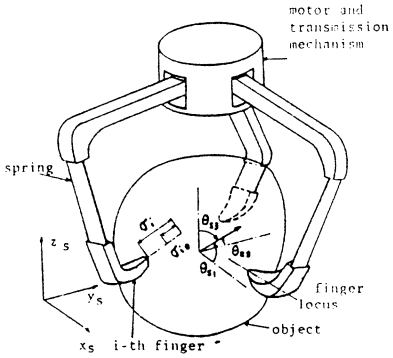


Figure 1
Schematic diagram of the prehension system.

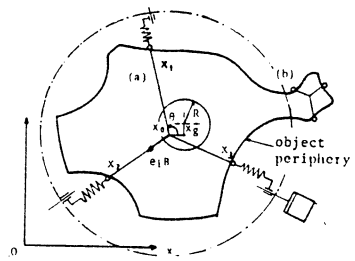


Figure 2
Desirable hand location for two-dimensional object.

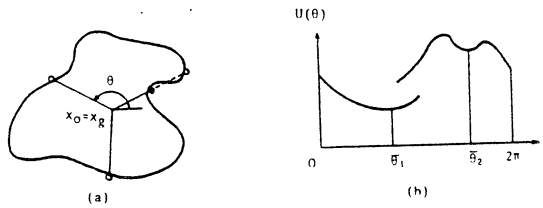


Figure 3
Local minimum of $U(\theta; x_o = x_g)$.

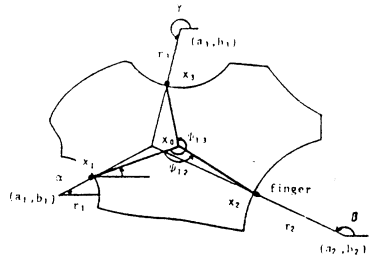


Figure 4
Approximation by circular arcs.

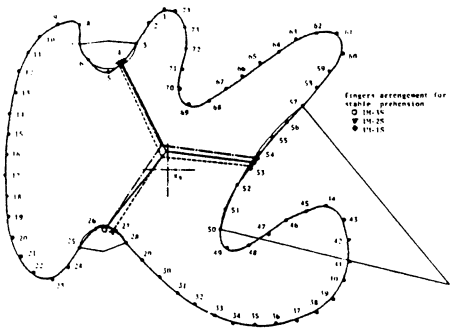


Figure 8
Object used in the experiment.

Figure 4.12: Méthode développée par [Hanafusa 77a]

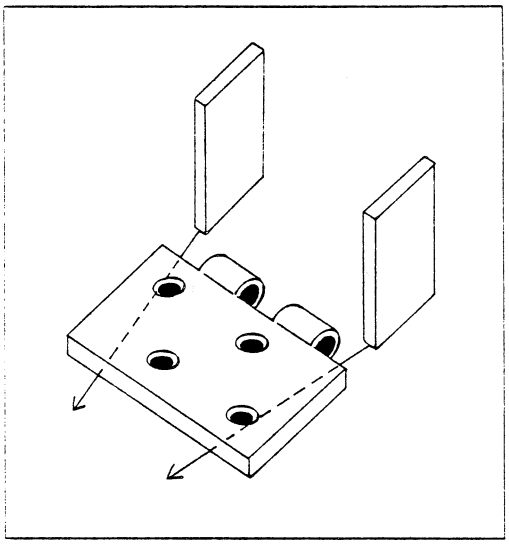


Figure 1. Hinge grasping. Each finger performs a uniform translation, without sensory feedback. Ultimately the hinge-plate will be aligned with the leading edges of the fingers, and centered between the fingers, eliminating all uncertainty in the position of the hinge-plate with respect to the fingers.

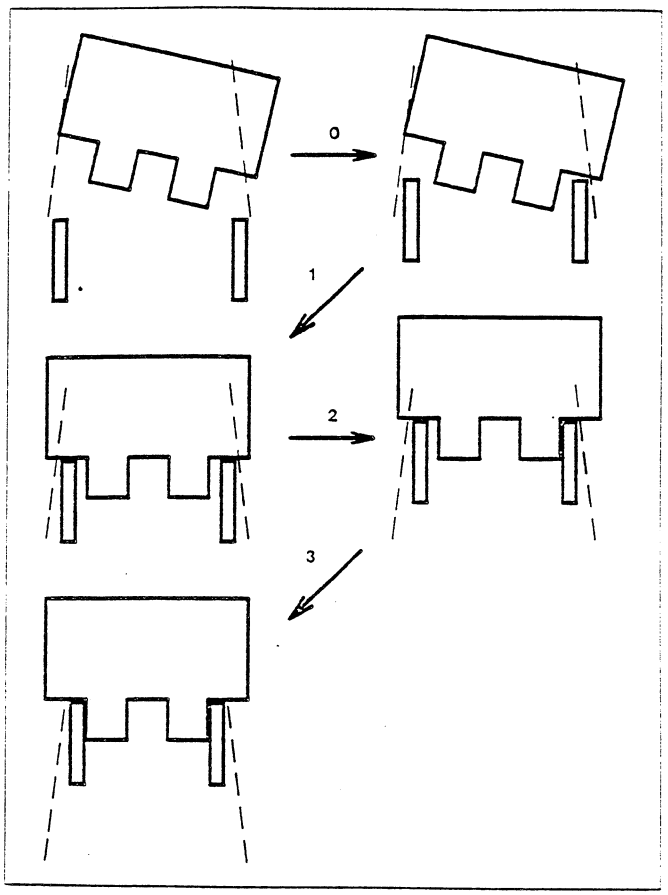


Figure 2. The hinge-plate grasping operation progresses in stages.

Figure 4.13: L'approche de [Mason 82]

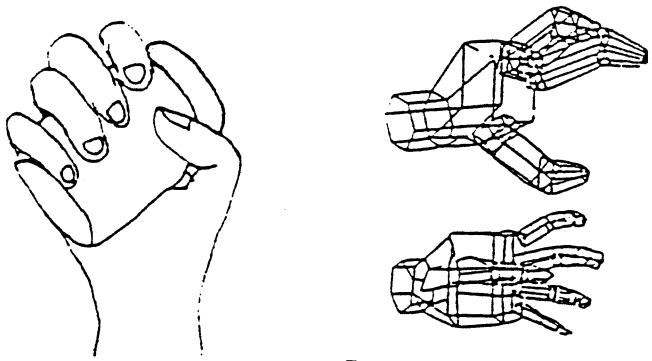


Figure 2: Encompass Grasp.

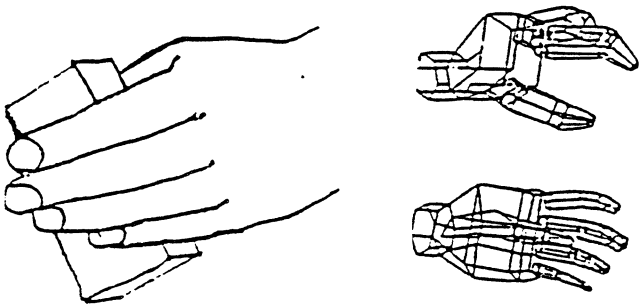


Figure 3: Lateral Grasp.

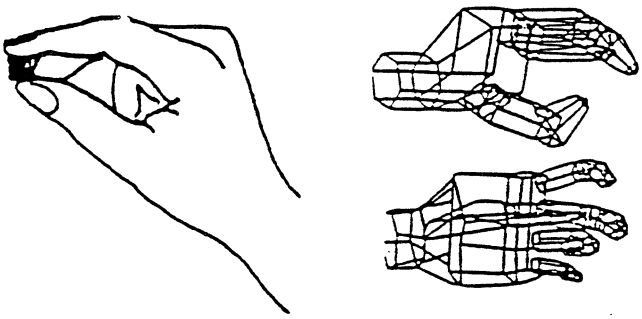


Figure 4: Precision Grasp.

Figure 4.14: Différents types de saisie [Lyons 85]

Les méthodes développées dans le cadre de la saisie sous contexte ont peu à peu évolué grâce à l'enrichissement du **concept de prise.**

Dans les premières méthodes on liait explicitement un ensemble de prises à chaque objet à saisir; la considération de l'objet comme paramètre de la recherche a permis dans un premier temps de lier le concept de prise à des volumes élementaires des objets (c'est le cas notamment de la méthode de Tomas Lozano-Pérez associant des prises, "grasp sets", à des parallèlépipèdes ou à des cylindres). Les entités composant les prises sont devenues petit à petit moins explicites; ainsi,la recherche d'un ensemble entités géométriques par examen morphologique de l'objet correspondait à une définition plus précise de la prise par un outil particulier (saisie par deux faces, ou une arète et une face par exemple).

Une étape essentielle à l'automatisation de la préhension consiste à ne plus formuler le concept de prise comme étant lié à un outil particulier mais à un mode de saisie; dans ce cas, les objet et outil considérés deviennent des paramètres de la recherche de prise. Il s'agit alors de déterminer la configuration de l'ensemble d'entités géométriques élémentaires de l'objet à mettre en contact avec l'outil, à partir de la connaissance de l'outil de préhension puis de rechercher ces ensembles dans l'objet.

L'un des objectifs de ce travail a été motivé par le désir de généraliser le concept de prise afin de nous orienter vers une méthode permettant une paramètrisation par l'outil de préhension. Pour ce faire, nous nous sommes basés sur les précédents travaux de notre équipe dans le domaine de la saisie automatique [Laugier 81,Laugier 83b]. Ce travail qui le premier a permis de définir le concept de prise et qui est fondé sur une analyse de la morphologie de l'objet à saisir, nous a considèrablement aidé dans notre approche du problème. La seconde partie de notre travail portant sur l'étude de compatibilité contextuelle de niveau objet a été motivée par les faiblesses des méthodes existantes en ce qui concerne le problème de propagation de contraintes d'accessibilité évoqué précedemment.

Chapitre 5

Méthode développée

5.1 Présentation de l'approche

La saisie est la réalisation d'une situation de contact entre l'outil et l'objet considérés. Une méthode générale pour automatiser cette opération ne peut se contenter d'une base formelle reposant sur le seul examen de la morphologie de l'objet à saisir. Au contraire, cette base formelle doit s'appuyer sur une analyse de l'outil et de ses relations à l'objet pour un mode de saisie donné.

Nous nous intéresserons ici, à un mode de saisie latérale caractérisé par un contact de la surface des "doigts" avec l'objet saisi (par opposition à une saisie par l'extremité des doigts.

5.1.1 Formalisation du problème de la saisie

Considèrons un objet O devant être saisi par un outil de préhension P. O évolue d'un environnement initial de saisie E_I à un environnement final de manipulation E_F . E_I et E_F contiennent l'ensemble des objets situés dans le voisinage de O (c'est à dire vérifiant des contraintes de distance par rapport à O). Il s'agit de répondre aux questions suivantes:

1. Comment prendre l'objet O au moyen de l'outil P?
2. Selon quelle direction aller le prendre dans E_I ?
3. Comment positionner P relativement à O au moment de la saisie?
4. Selon quelle direction s'éloigner de O après positionnement par P dans E_F ?

La recherche des réponses à chacune de ces questions nécessite de prendre en compte tous les obstacles que P est susceptible de rencontrer au moment de l'éxécution de la saisie c'est à dire tous les objets de l'ensemble $E_I \cup E_F \cup \{O\}$.

Nous supposerons d'une part que les trajectoires d'approche et de déproche de P sont rectilignes et d'autre part qu'il n'est pas utile de considérer le manipulateur dans sa totalité puisque l'on se trouve à proximité de O. L'approche vers O, la déproche à partir de O ainsi que son transport de E_I à E_F peuvent être résolus par des méthodes classiques de génération de trajectoires sans collision pour un manipulateur (voir [Lozano 80], [Gouzenes 84], [Faverjon 84] ou [Laugier 85a]). Le problème du montage de O sur le sous-assemblage présent dans E_F , s'il n'est pas indépendant de la saisie (cf. §5.2) sera lui-même résolu par des méthodes telles que [Valade 85] ou [Theveneau 85].

Saisir un objet, c'est tout d'abord **choisir les "parties" de cet objet devant être mises en contact avec l'outil de préhension**: elles constituent une **prise** Π . C'est également **déterminer les conditions de réalisation de ces contacts** (paramètres de préconfiguration de l'outil - écartement des mors par exemple - , directions d'approche et de déproche, position et orientation de P par rapport à O): ce sont les **paramètres de réalisation de la prise** Π .

La réalisation effective des contacts mis en jeu s'exprime au moyen d'un positionnement relatif de l'outil préconfiguré avant saisie. Dans le cas d'un objet et d'un outil caractérisés par par un repère cartésien, il s'agit de déterminer la transformation géométrique premettant de passer d'un repère à l'autre.

En toute généralité, la saisie doit être spécifiée de la manière suivante:

$$\mathbf{PS(O,P) = (\Pi, P_P(\Pi), P_{P/O}(\Pi))}$$

où:

- $\Pi = E_1, E_2, ..., E_m$

  $m \leq n$, où n est le nombre d'éléments de contact de l'outil (deux pour une pince à deux mors, quatre pour une pince à deux doigts bi-phalangés...)

  $E_i = (\{e_1, e_2, ..., e_k\}$, type de contact)

  où $i \in [1, m]$ et $e_j \in$ O pour $j \in [1, k]$, $k \geq 1$ - les e_j sont des éléments géométriques de l'objet (ses faces, arètes, sommets par exemple)

  Le type de contact est surfacique, linéïque ou ponctuel (il caractérise le nombre de degrés de liberté de O éliminés par la saisie)

- $P_P(\Pi)$ sont les paramètres de préconfiguration de la pince

- $P_{P/O}(\Pi) = ((O_P, X_P, Y_P, Z_P), D_D, D_A)$ sont les paramètres de positionnement de P par rapport à O

  où O_P , X_P , Y_P , Z_P (représentant le repère lié à P) et D_A et D_D (les vecteurs représentant les directions d'approche et de déproche) sont exprimés par rapport à O.

A titre d'illustration, nous présentons dans la figure 5.1 deux exemples de saisie d'un objet par un outil et les paramètres correspondant.

La détermination de $PS(O,P)$ se fait par application de contraintes strictes portant sur l'accessibilité de la prise et peut être guidée par certaines contraintes plus faibles liées à l'équilibre et à la stabilité du système main-objet. Certaines applications nécessitent, en effet, d'engendrer des prises "de bonne qualité" i.e. permettant de maîtriser l'évolution des incertitudes après la saisie.

Les contraintes fortes sont les suivantes:

C_1: Interférence $(P, O_P, X_P, Y_P, Z_P, \{O\}) = \emptyset$

C_2: Interférence $(P, M_A, X_P, Y_P, Z_P, E_I \cup \{O\}) = \emptyset$ pour $M_A \in droite(O_P, D_A)$

C_3: Interférence $(P, M_D, X_P, Y_P, Z_P, E_F \cup \{O\}) = \emptyset$ pour $M_D \in droite(O_P, D_D)$

où Interférence $(O_i, M_i, X_i, Y_i, Z_i, Ens)$ teste s'il existe une interférence entre l'objet O_i positionné en M_i et orienté selon (X_i, Y_i, Z_i) et les objets de l'ensemble Ens .

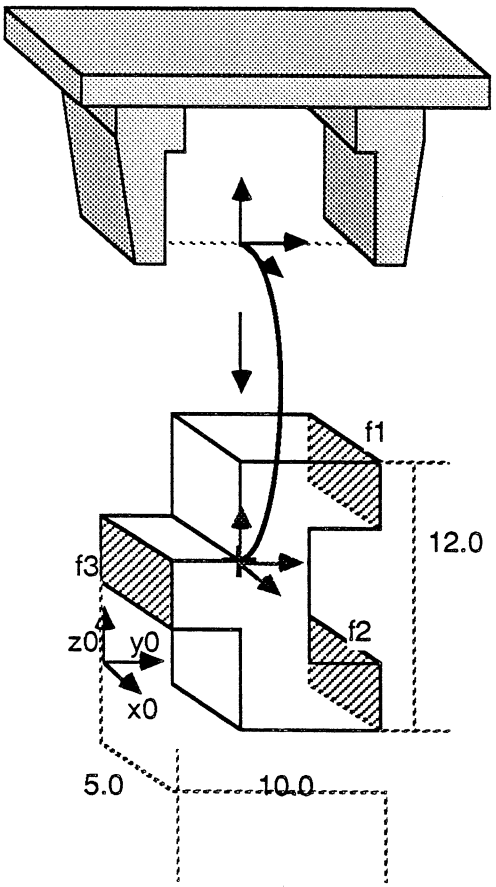
$droite(M,V)$ est la droite définie par le point M et le vecteur V .

Les contraintes faibles mesurent la résistance de la prise à deux types de mouvements:

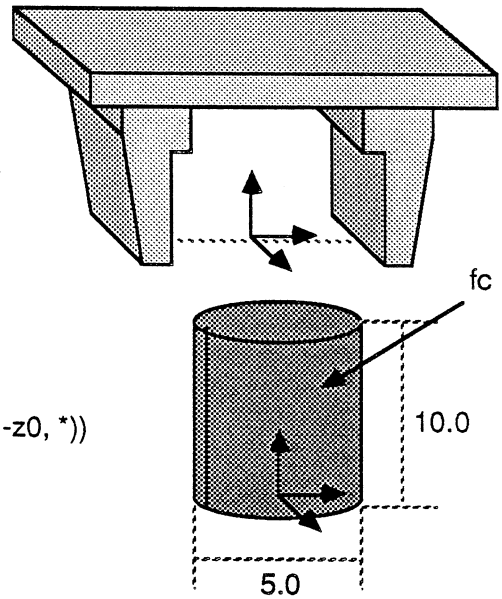
- les glissements en rotation dûs à l'application de forces externes: gravité, forces d'inertie dûes aux déplacements du système main/objet, et forces mises en jeu au montage
- le "twisting" provoqué au moment du serrage par les incertitudes de localisation de O et P lors de la réalisation de la saisie ou par la prise elle-même (cf. figure 5.2).

5.1.2 Méthode proposée

La programmation de niveau objet nécessite de maîtriser l'évolution des incertitudes tout au long du programme engendré pour assurer son succès en univers réel. En présence d'une petite incertitude initiale, une prise non équilibrée ou instable peut changer radicalement le positionnement relatif des outil et objet après saisie (cf. figure 5.3). L'évaluation rigoureuse des effets de la saisie (en terme d'incertitudes) est alors extrêmement complexe. C'est pourquoi il est préférable, dans ce contexte,



PS(O1, P) = ((((f1, f2) plan), ((f3), plan))
(fermer, 0.0)
(((2.5, 5.0, 6.0), x0, y0, z0), -z0, *))



PS(O2, P) = ((((fc), droite))
(fermer, 5.0)
(((0.0, 0.0, 5.0), x0, y0, z0), -z0, *))

Figure 5.1: Paramètres de saisie de deux objets

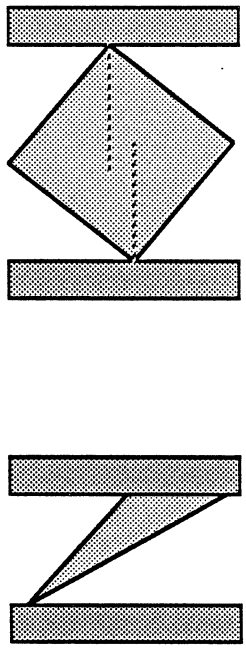


Figure 5.2: Prises instables

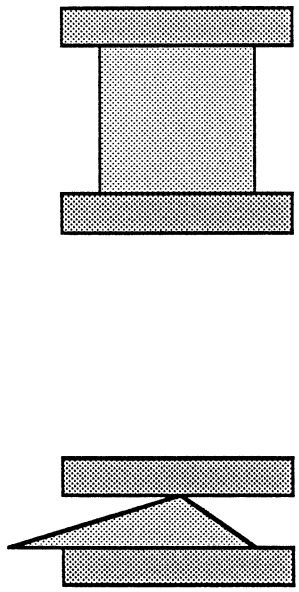


Figure 5.3: Effet de l'instabilité

de n'engendrer que des prises en équilibre stable.

La méthode développée se décompose en deux phases successives de traitement; elles sont respectivement consacrées à la prédiction de prises potentielles équilibrées et stables grâce à des critères morphologiques locaux et à la vérification contextuelle de ces prises.

La première phase est appelée phase d'**analyse d'accessibilité et de stabilité** ou A^2LS. Elle repose sur le **raisonnement géométrique**.

Elle a pour but d'engendrer un ensemble de prises potentielles stables. Cette étude est fondée sur une *analyse des contacts potentiels* entre O et P. Ceux-ci sont déterminés sur la base de critères d'accessibilité locale; ils sont ensuite réunis de manière à constituer des prises. La construction de ces prises est réalisée *grâce à l'utilisation de filtres géométriques directement extraits de la morphologie de l'outil et d'heuristiques liées à la stabilité désirée des prises.*

La seconde phase a pour but de complèter l'analyse d'accessibilité, limitée dans l' A^2LS au seul voisinage des contacts potentiels. Elle permet également de traiter, en partie, le problème de la compatibilité contextuelle de niveau objet. Elle repose sur une **analyse d'accessibilité globale** de la prise ou A^2G fondée sur le **raison-**

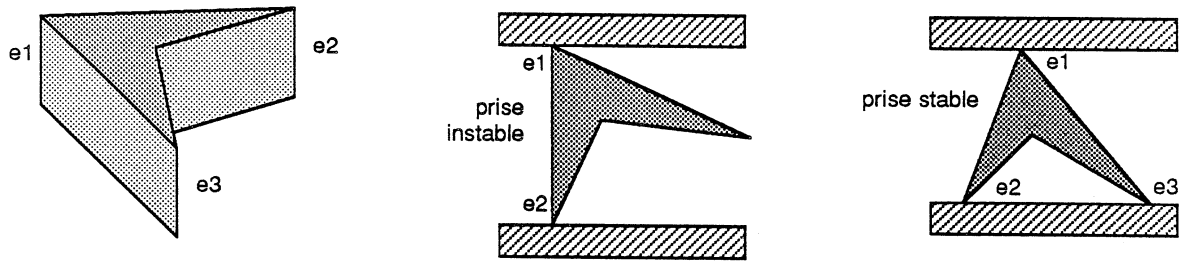


Figure 5.4: Prise obtenue à partir d'une prise instable

nement spatial.

Il s'agit de déterminer si une prise engendrée par examen local des contacts potentiels entre objet et pince est compatible avec les environnements initial et final de manipulation d'une part, et avec l'objet lui-même d'autre part. Pour ce faire, nous déterminons d'abord partiellement les valeurs de certains des paramètres de réalisation de la prise (orientation de la pince, directions d'approche et de déproche...). Nous vérifions ensuite que la pince ne rencontre pas d'obstacles dans ses phases de mouvement. Cette vérification peut conduire à une meilleure spécification des paramètres par application de nouvelles contraintes ou à une remise en cause, totale ou partielle, de la solution en cours de traitement. L'originalité de cette seconde phase réside dans l'utilisation d'une *représentation de l'outil par points de référence permettant de transformer un ensemble de contraintes d'accessibilité implicites tridimensionnelles en un ensemble de contraintes géométriques bi-dimensionnelles explicites sur la position de l'outil.*

A l'issue de ces deux phases les prises accessibles sont ordonnées par rapport à leur stabilité présumée. Cet ordre est dépendant du type des contacts mis en jeu (résistance au "twisting") et pour des contacts de même type de la distance de O_P au centre de gravité de l'objet (résistance au glissement en rotation).

Pour des raisons évidentes liées à la combinatoire de l'approche, nous n'engendrons, dans la première phase, que des prises dont les ensembles E_i (cf. §5.1.1) sont des singletons. En cas d'échec de l' A^2G pour les prises considérées, de nouvelles prises stables devront être engendrées à partir de prises instables en complètant les E_i correspondant (cf. figure 5.4).

Nous illustrerons notre approche dans le cas d'un outil spécifique et d'un type de modélisation particulier. L'outil considéré (cf. figure 5.5) reflète la morphologie

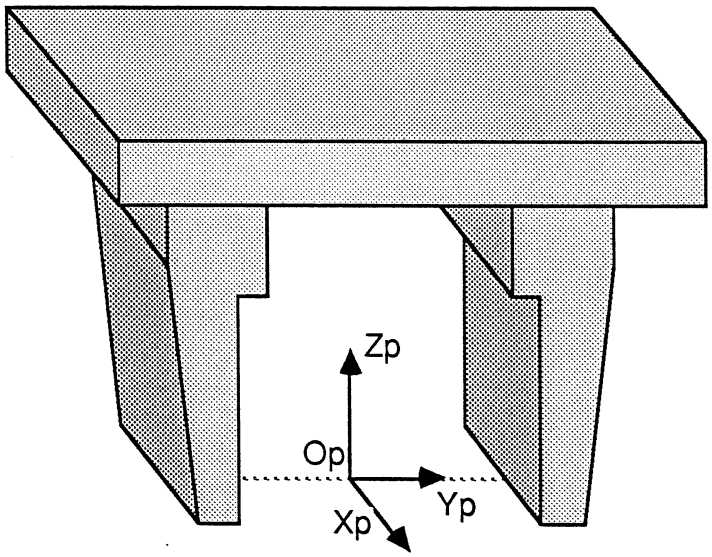


Figure 5.5: Outil type à deux mors

typique d'une pince à deux mors parallèles. Ce type d'outil est caracterisé par des contacts du type plan, droite ou point. La représentation de base utilisée pour les objets est de type B.R. (décomposition en polyèdres, cylindres et sphères). Nous verrons en §6.1 ce que ces hypothèses ont de restrictif par rapport à la méthode proposée.

Remarque: dans la suite de ce texte, nous appellerons entité (ou élément) géométrique EG d'un objet O tout élément de l'ensemble de ses faces, arètes, et sommets.

5.2 Phase de prédiction: accessibilité locale et stabilité

5.2.1 Présentation de la méthode

L' A^2LS a pour but de choisir une prise Π et de déterminer les paramètres de préconfiguration de l'outil $P_P(\Pi)$ (l'action et l'écartement des mors dans le cas de P). Elle permet en outre, dans un grand nombre de cas de contraindre partiellement les éléments de $P_{P/O}(\Pi)$ sur la base de l'examen morphologique de l'objet.

Cette première phase, qui peut être décomposée en deux grandes étapes, va réduire à moindre coût, l'arbre de recherche des prises en éliminant le plus vite possible, les solutions évidemment mauvaises (entités géometriques dont on est sûr qu'elles ne peuvent produire un élément de prise, prises non équilibrées ou instables).

Il faut d'abord recenser tous les éléments géométriques de O susceptibles de

fournir un contact avec un élément de contact de P: ce sont les faces, arètes et sommets de l'objet vérifiant certains critères d'accessibilité locale. Chacune de ces EG est l'unique élement d'un ensemble. Il faut ensuite réunir ces ensembles pour constituer une prise en utilisant les contraintes liées à la géométrie et à la morphologie de P et à la stabilité de la prise.

L' A^2LS est basée sur l'évaluation de propriétés géométriques des objets dont la principale est une analyse d'accessibilité locale liée à la convexité locale des entités considérées. L'intérêt de notre méthode réside essentiellement dans le formalisme induit par le raisonnement sur les contacts qui permet une représentation implicite des entités susceptibles de constituer une prise. Elle est, par ce fait, à opposer à la majorité (voire la totalité?) des méthodes existantes basées sur une description des schémas de prises potentielles sous forme des EG impliquées. Cette dernière approche conduit d'une part à restreindre l'ensemble des prises potentielles et d'autre part à l'associer plus directement à un type de modélisation des objets qu'à un type d'outil de préhension.

5.2.2 Détermination des contacts potentiels

La saisie se matérialise par un contact avec chacun des mors (généralement leur face interne ou externe) nécessitant une phase d'approche tangentielle avant le serrage. Il s'agit donc de déterminer les EG de l'objet permettant ce type de contact: celles qui vérifient certaines propriétés de convexité locale.

Considèrons la propriété de **convexité locale** présentée dans [Laugier 83b] (définie ici dans le cas 3D):

Définition 5.2.2.1 *Nous dirons qu'une partie E de l'enveloppe F d'un objet B est localement convexe (resp. concave) si il existe un plan d'appui local de F en E tel que la restriction de B (resp. du "vide") à un voisinage de E soit entièrement contenue dans l'un des demi-espaces définis par ce plan (cf. figure 5.6).*

Intuitivement, il est clair qu'une EG localement convexe permet de produire un contact avec un plan et qu'une EG localement concave ne le permet pas.

Malheureusement, certaines entités ne sont ni localement convexes, ni localement concaves. Il arrive donc parfois qu'après de lourds calculs (liés à la caractérisation analytique de cette propriété) on ne puisse pas conclure formellement à la possibilité ou à l'impossibilité d'avoir un contact avec une EG même si, intuitivement, la réponse est évidente (cf. figure 5.7).

Ceci nous a orienté vers la définition d'une propriété spécifique appelée **accessibilité locale**. Nous l'avons caractérisée analytiquement, uniquement, pour les différents types d'entités géométriques et elle mériterait d'être généralisée aux

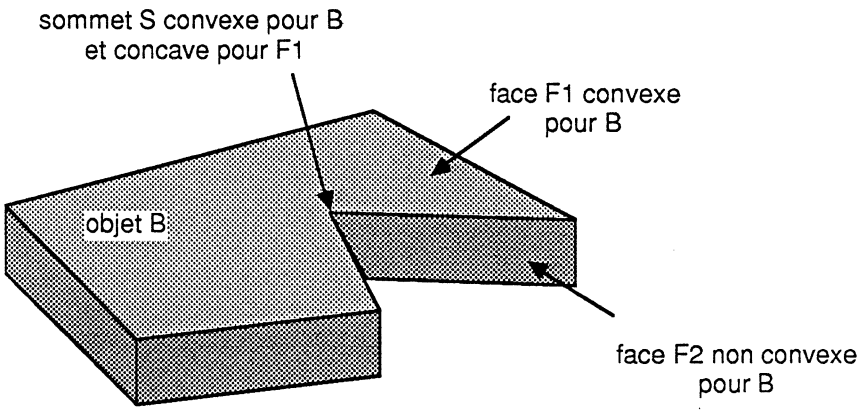


Figure 5.6: Définition de la propriété de convexité locale

différents types de contact. Nous présentons ici cette caractérisation analytique pour quelques types d'entités géométriques.

Définition 5.2.2.2 *Un objet O est localement accessible au voisinage d'une de ses faces planes F s'il existe une arète A de F telle que O soit localement accessible au voisinage de A .*

Définition 5.2.2.3 *Un objet O est localement accessible au voisinage d'un de ses sommets S s'il existe une arète A aboutissant à S telle que O soit localement accessible au voisinage de A .*

Définition 5.2.2.4 *Un objet O est localement accessible au voisinage d'une de ses arètes rectilignes A commune à deux faces planes F_1 et F_2 si le produit mixte $\overrightarrow{Va_1} \circ (\overrightarrow{N_1} \wedge \overrightarrow{N_2})$ est négatif ou nul. $\overrightarrow{N_1}$ et $\overrightarrow{N_2}$ sont les normales extérieures respectives des faces F_1 et F_2 ; $\overrightarrow{Va_1}$ est le vecteur porté par A et orienté avec la convention "matière à droite" par rapport à F_1 (cf. figure 5.8).*

On peut d'emontrer sur la base des définitions analytiques que la non accessibilité locale implique la concavité locale.

Dès lors qu'une EG ne vérifie pas cette propriété, nous sommes donc sûrs que tout contact entre elle et un élement de l'outil de préhension est impossible et qu'elle peut être rejetée. Cette propriété permet donc de décider de l'accessibilité potentielle d'un objet donné dans le voisinage d'une de ses entités de façon assez simple

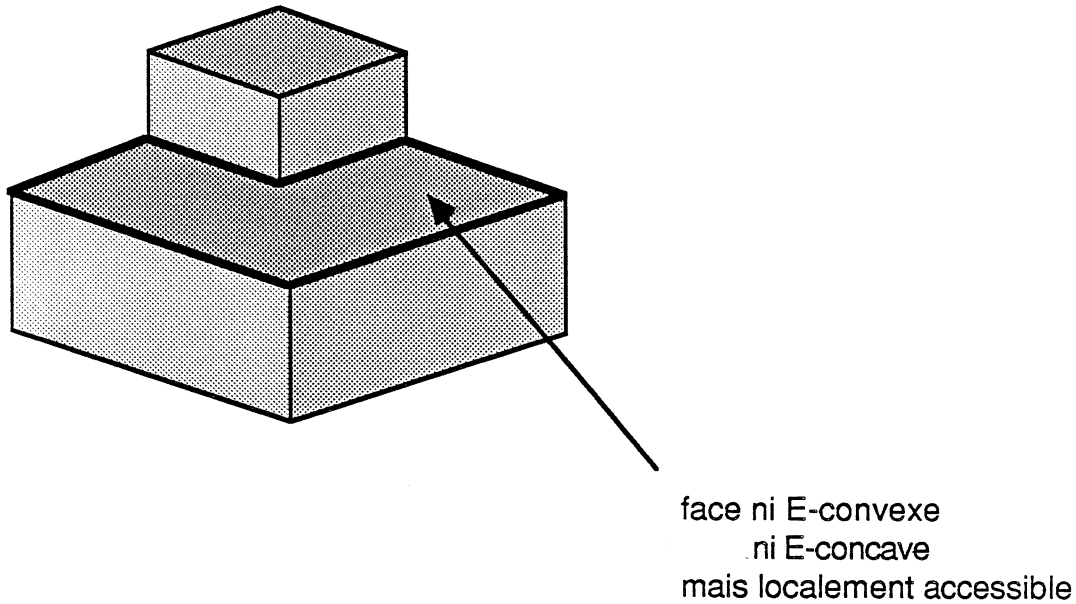


Figure 5.7: Convexité et accessibilité

(par calcul vectoriel). Elle permet d'autre part d'évaluer grossièrement le degré d'accessibilité locale: par exemple, une face plane dont toutes les arètes vérifient la propriété d'accessibilité locale aura généralement de meilleures chances de réussite qu'une autre face dont seulement quelques arètes vérifient cette propriété.

Notons cependant que cette propriété ne présage rien de l'accessibilité réelle de l'entité: une face plane localement accessible pourra ne pas permettre de positionner le mors à cause de faces voisines (cf. figure 5.9).

5.2.3 Construction des prises

La construction des prises repose sur l'utilisation de filtres géométriques dépendants de l'outil considéré et d'heuristiques fondées sur la stabilité des prises.

Appelons E_1 et E_2 deux ensembles d'EG de type quelconque. Il s'agit de déterminer s'il est possible de saisir l'objet simultanément par E_1 et E_2 i.e. si les deux mors de P peuvent être simultanément en contact avec les EG de E_1 et E_2 . Il faut par ailleurs s'assurer de la viabilité de la prise.

E_1 et E_2 formeront une prise si et seulement si:

- ils sont conformes aux paramètres morphologiques de l'outil

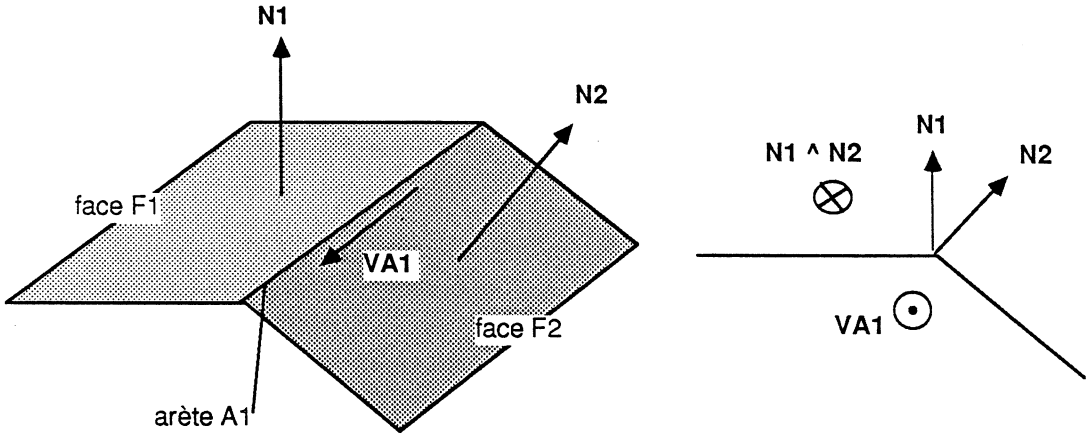


Figure 5.8: Accessibilité locale d'une arète

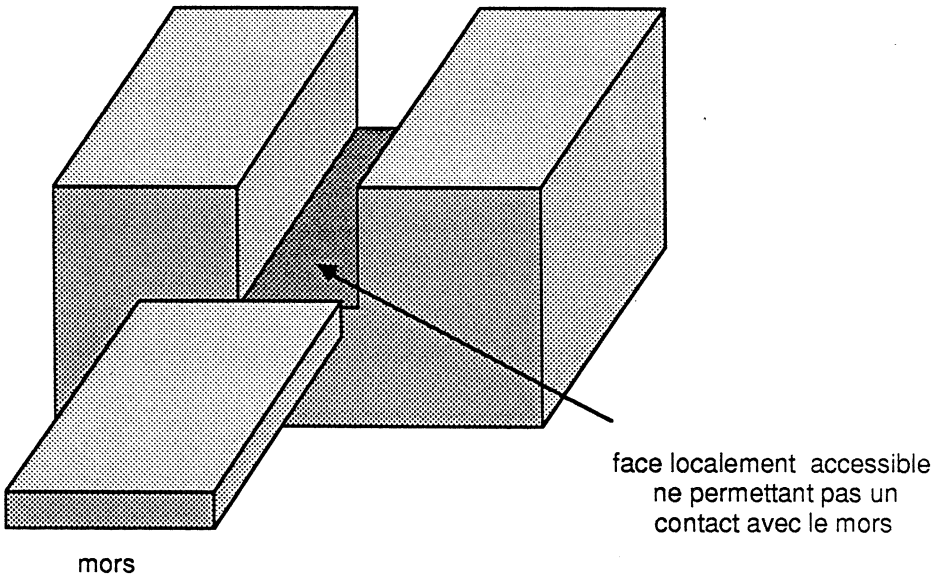


Figure 5.9: Accessibilité globale d'une prise

- le système main/objet est en équilibre
- l'équilibre n'est pas clairement instable

La première condition est une contrainte stricte qui se traduit par l'évaluation de propriétés géométriques et topologiques. Dans le cas de P: parallèlisme, distance, accessibilité locale plane et répartition de la matière.

La seconde condition correspond à une contrainte mécanique. Elle a pour but de vérifier l'absence de "twisting" dû à la prise elle-même. Il ne s'agit pas ici de déterminer rigoureusement les forces mises en jeu (frottement y compris), mais de trouver une propriété permettant simplement de vérifier que la prise met le système main/objet en équilibre. Cette propriété s'appelle visibilité.

Enfin, il nous faut éviter les prises instables qui risqueraient de provoquer un "twisting" en présence d'une petite erreur de positionnement relatif de P par rapport à O. Nous rejetons, à cet effet, toutes les prises sensibles à une rotation autour de X_P et/ou Z_P (ou d'axes parallèles): celles qui laissent au moins un degré de liberté en rotation autour de l'un de ces axes (par exemple, deux contacts linéïques parallèles). La rotation autour de Y_P n'a pas de conséquence sur le "twisting".

Cette heuristique permet de réduire de manière importante l'arbre de recherche des prises par une simple analyse des types de contacts mis en jeu. Elle repose sur les travaux formels de [Brost 85] portant sur la recherche de prises pour des objets 2D en présence de considérations sur les incertitudes et sur les forces de friction.

Parallèlisme

Ce critère de sélection est imposé par le parallèlisme des mors de P, donc des contacts à produire. Il est très facile à évaluer puisqu'il s'agit de tester un parallèlisme de plans, droites ou points.

Définition 5.2.3.1 *Deux ensembles d'EG de type quelconque E_1 et E_2 sont dits parallèles s'il existe au moins deux plans tangents aux contacts définis par E_1 et E_2 qui soient parallèles.*

Pour certaines prises (cylindrique, sphérique) il existe une infinité de tels plans.

Visibilité

Afin d'obtenir des prises équilibrées, il faut choisir le couple (E_1, E_2) de façon que la projection orthogonale sur un plan commun parallèle à leur plan tangent, des zônes de contact potentiel correspondant aient une intersection non vide(cf. figure 5.10): c'est la propriété de visibilité.

Cette propriété est trivialement vérifiée pour certaines prises (cylindrique, sphérique) sans qu'il soit nécessaire de définir ce plan. Dans les autres cas, le plan appelé **"plan de préhension"** est choisi à mi-distance des plans tangents aux contacts. Il représente ainsi l'ensemble des positions potentielles du point O_P . Il correspond à la détermination du vecteur Y_P .

Ce critère conduit à rejeter des prises qui pourraient malgré tout réussir lors de la manipulation grâce à la présence de forces de frottement (cf. figure 5.11). Cet

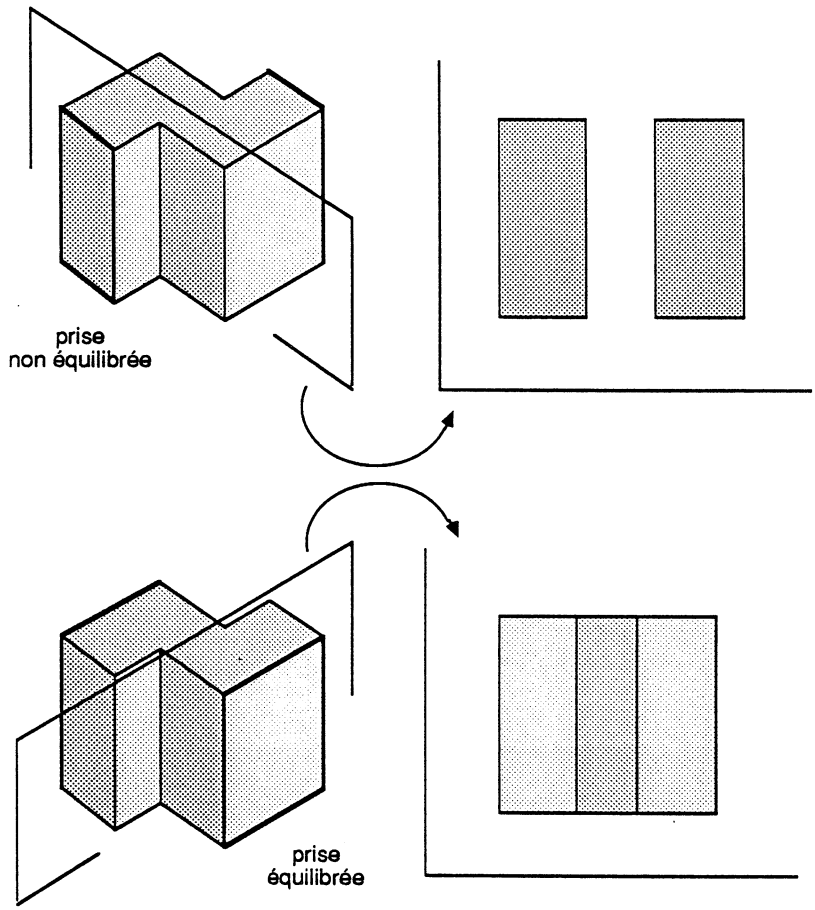


Figure 5.10: Propriété de visibilité

aspect n'est pas considéré dans ce travail à cause de l'intérêt évident de rejeter les prises potentiellement instables, eu égard au traitement des incertitudes.

Distance

E_1 et E_2 doivent se trouver à une distance telle que l'écartement nécessaire pour effectuer la saisie se trouve dans les bornes autorisées par le manipulateur considéré. La distance des entités de E_1 et E_2 est généralement simple à quantifier. Par contre, le calcul de l'écartement correspondant est fonction d'une part de cette distance et d'autre part de l'action à effectuer (O/F) et de la géométrie des éléments de E_1 et E_2 (cf. figure 5.12).

Accessibilité locale plane et répartition de la matière

La propriété d'accessibilité locale définie en §5.2.1 nous permet de savoir s'il existe au moins un plan d'appui local à une EG d'un objet. Il nous faut dans un deuxième temps trouver des plans d'appuis compatibles avec les différents E_i constituant la

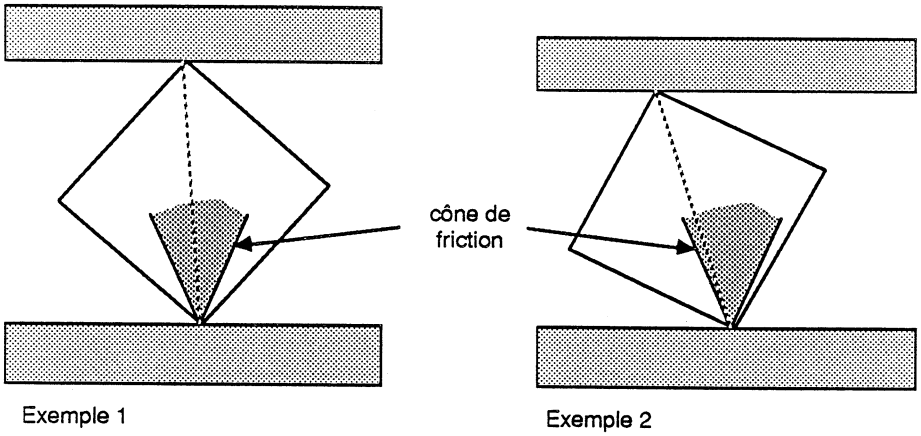


Figure 5.11: Prise équilibrée par le frottement

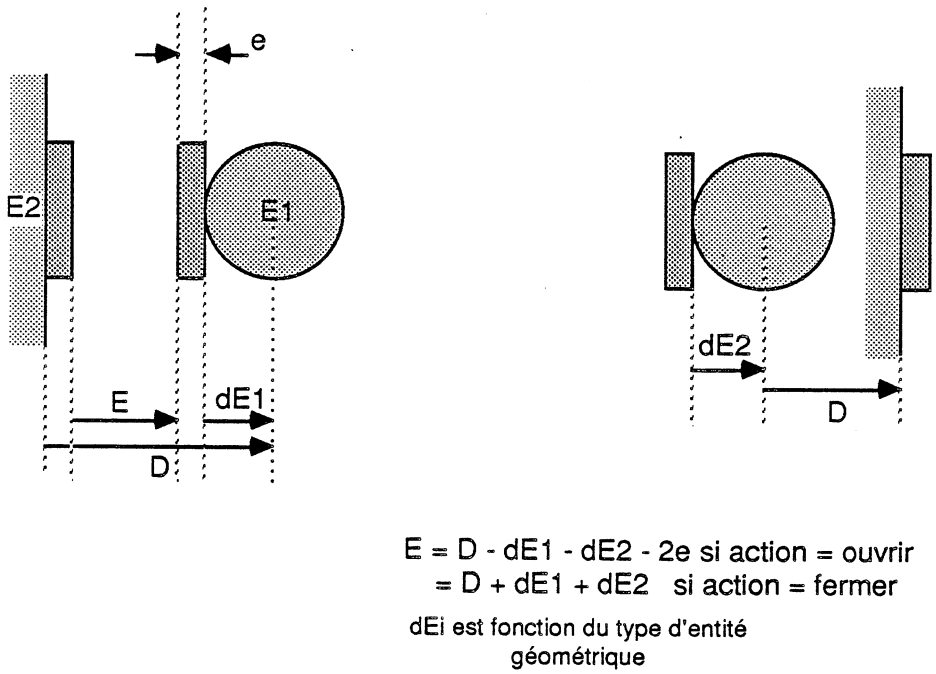


Figure 5.12: Calcul de l'ecartement des mors

prise, et nous assurer que la répartition de la matière au voisinage des contacts permet la saisie. Nous utilisons une seule propriété appelée P-accessibilité locale caractérisant la possibilité d'avoir un contact entre un E_i et un objet localement plan. Ce plan P est caractérisé par une normale $\overrightarrow{N_P}$ orientée vers l'extérieur de l'objet.

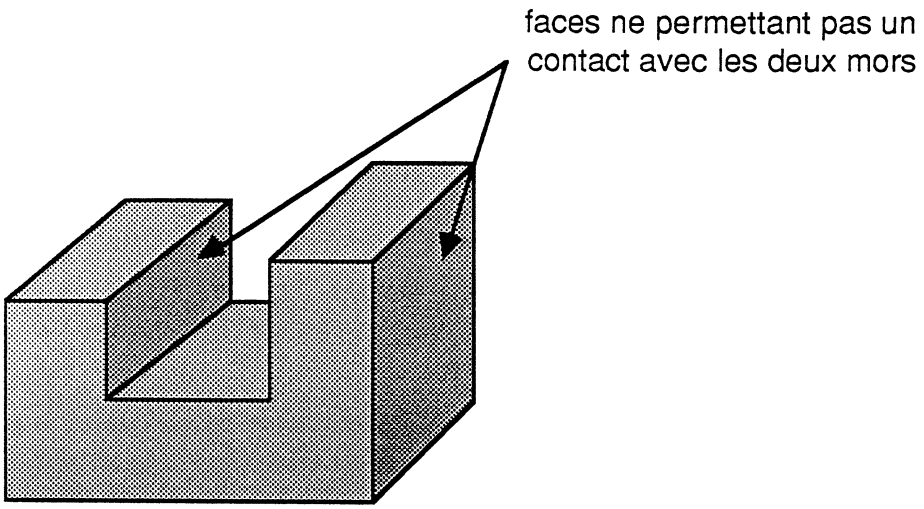


Figure 5.13: Faces localement accessibles ne constituant pas une prise

La P-accessibilité locale complète le parallèlisme qui ne faisait en effet aucune référence à la présence de matière (donc d'obstacles potentiels) dans le voisinage des EG (cf. figure 5.13). Nous donnons ici la caractérisation analytique de la **P-accessibilité locale** pour quelques EG particulières:

Définition 5.2.3.2 *Un objet O est localement P-accessible au voisinage d'une de ses faces planes F si:*

- *O est localement accessible au voisinage de F ,*
- *la normale extérieure à la face F est opposée à la normale* $\overrightarrow{N_P}$ *du plan P .*

Définition 5.2.3.3 *Un objet O est localement P-accessible au voisinage d'une de ses arètes A commune à deux faces* F_1 *et* F_2 *si:*

- *O est localement accessible au voisinage de A ,*
- *il existe deux réels négatifs a et b tels que* $\overrightarrow{N_P} = a\,\overrightarrow{N_1} + b\,\overrightarrow{N_2}$ *où* $\overrightarrow{N_1}$ *et* $\overrightarrow{N_2}$ *sont les normales extérieures respectives de* F_1 *et* F_2 *.*

Remarque: Les signes des coefficients a et b déterminent les positions relatives des vecteurs $\overrightarrow{N_P}, \overrightarrow{N_1}, \overrightarrow{N_2}$.

Définition 5.2.3.4 *Un objet O est localement P-accessible au voisinage d'un de ses sommets S si:*

- *O est localement accessible au voisinage de S ,*
- *le produit scalaire* $\overrightarrow{Va_S} \circ \overrightarrow{N_P}$ *est de signe négatif pour toute arète A de sommet S ;* $\overrightarrow{Va_S}$ *est le vecteur porté par A , orienté à partir de S .*

Remarque: Intuitivement, les arètes aboutissant à S doivent se trouver sur ou "au-dessus" du plan P .

Outre son utilisation pour la construction de la prise (E_1, E_2), cette propriété sera également utilisée pour la génération des E_i non réduits à un singleton. En effet, lorsque l'on veut complèter un ensemble E_i , on doit rechercher dans un voisinage de son unique élément, une ou plusieurs autres EG possèdant en commun avec lui un plan d'appui local (cf. figure 5.4).

Ordonnancement des filtres géométriques

Cet ordonnancement dépend en partie des contraintes d'antériorité existant entre les différents filtres de par leur définition. L'ordre final est obtenu par respect de ces contraintes, en essayant de minimiser les coûts algorithmiques; il s'agit donc de choisir d'abord, autant que possible, les filtres les moins coûteux et les plus discriminants en respectant ces contraintes.

Les contraintes d'antériorité sont les suivantes. La première contrainte porte sur la précédence du parallèlisme par rapport à la P-accessibilité locale puisque l'évaluation de la seconde propriété complète celle de la première. Le calcul de l'écartement des mors doit lui-même intervenir après l'évaluation de la P-accessibilité locale puisque cette dernière permet de déterminer l'action à effectuer. La visibilité n'a de contrainte que celle d'être évaluée après le parallèlisme. C'est une propriété assez discriminante mais également assez coûteuse.

En conséquence l'ordre choisi pour l'évaluation de ces propriétés est le suivant: (1) parallèlisme, (2) P-accessibilité locale, (3) distance et (4) visibilité.

5.2.4 Paramètres de préconfiguration de l'outil

Action

Celle-ci est déterminée par le positionnement de la matière au voisinage de E_1 et E_2. Considèrons le plan P_1 parallèle au plan de préhension et tangent à E_1. La normale $\overrightarrow{N_1}$ de P_1 est choisie de façon à pointer vers l'extérieur de l'objet O en E_1. Soit M_1 un point de E_1 et M_2 sa projection sur E_2 parallèlement à $\overrightarrow{N_1}$. L'action à effectuer est déterminée par le signe du produit scalaire $vecteur(M_1, M_2) \circ \overrightarrow{N_1}$ (ouvrir si positif, fermer sinon).

Dans le cas de prises formées d'un seul E_i, pour lesquels le plan de préhension n'est pas défini, l'action peut être trivialement déterminée (ouvrir pour un trou cylindrique, fermer pour une sphère pleine...).

Les prises associées à une fermeture de mors seront préférées à celles qui sont associées à une ouverture de mors pour des raisons évidentes de qualité de la saisie.

Ecartement des mors

Le calcul de l'écartement des mors est fait à partir de la distance des plans tangents aux contacts potentiels. Cette distance est extraite du modèle géométrique à partir des EG impliquées. Dans le cas d'une saisie par ouverture des mors, l'épaisseur de chacun des mors en sera soustraite pour obtenir l'écartement réel des mors.

5.3 Phase de vérification: accessibilité globale

5.3.1 Présentation de la méthode

Cette seconde phase a pour but de vérifier la cohérence d'une prise engendrée par l'A^2LS avec l'environnement de manipulation. Les paramètres de positionnement de l'outil et de réalisation de la prise non encore instanciés seront choisis au cours de l'A^2G.

Considèrons l'ensemble $E_I \cup E_F \cup \{O\}$ des objets de l'environnement; ces objets sont autant d'obstacles potentiels à l'outil P lors de la réalisation d'une prise Π. Le principe de base de cette seconde phase consiste à examiner l'espace décrit par l'outil au cours de ses mouvements afin de déterminer les obstacles s'y trouvant éventuellement, et à raisonner sur ceux-ci pour engendrer des trajectoires sans collision pour l'outil.

Classiquement, deux types de méthodes sont utilisées pour résoudre le problème de recherche de trajectoires sans collisions:

La première appelée méthode de type **génération et test** est assez simple dans son principe [Ahuja 80]; elle consiste à engendrer heuristiquement une trajectoire simple, à vérifier que la réalisation de cette trajectoire ne produit pas de collisions puis à la modifier localement en cas de détection de collisions.

La recherche de collisions se fait par étude de l'espace balayé par l'objet en mouvement puis calcul d'interférence ou bien par une suite de calculs d'interférence sur une trajectoire discrétisée. Les limitations intrinsèques à cette approche résident dans la modification locale et aveugle de la trajectoire engendrée; elle peut cependant être utilisée avec un relatif succès dans le cas de trajectoires simples pour un objet se mouvant avec peu de degrés de liberté (généralement un seul) dans un environnement assez peu contraint.

Le second type d'approche ou par **espace des configurations** est généralement utilisé pour planifier les mouvements des manipulateurs. Il est fondé sur une représentation explicite de l'espace libre. Dans le cas d'un manipulateur (ou d'un solide en mouvement) à p degrés de liberté, une configuration est un p-uplet de paramètres permettant de déterminer la position et l'orientation du robot (ou du solide en mouvement) dans un environnement donné. L'espace des configurations est l'ensemble

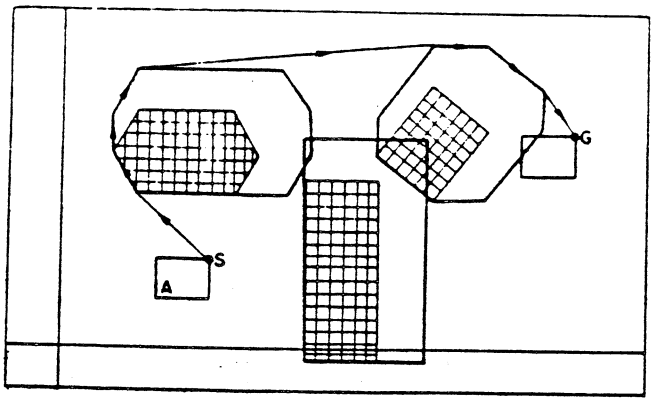


Figure 5.14: Construction de l'espace des configurations simplifiée [Lozano 79]

de ces p-uplets.

La recherche de trajectoires dans cet espace nécessite de déterminer quantitativement l'influence de chaque obstacle sur les différents constituants du manipulateur. Pour cela, il faut déterminer un ensemble d'obstacles dans l'espace des configurations équivalents aux obstacles présents dans l'espace réel. La recherche de trajectoires sans collisions consiste à rechercher un chemin pour un point (p-uplet) dans la partie libre de cet espace des configurations.

Chacune de ces deux méthodes prises indépendamment présente avantages et inconvénients par rapport à notre problème:

- L'espace balayé par l'outil est généralement aisément évaluable, puisque certaines contraintes d'orientation sont connues dès le choix de la prise. Malheureusement, puisque l'outil est non convexe et que ses degrés de liberté sont multiples, il est généralement difficile de raisonner sur cet espace. D'autre part, la position finale de l'outil est une inconnue du problème dont le choix est intimement lié à l'accessibilité de la prise et ne doit donc pas être arbitraire.
- L'approche par espace des configurations, peut nous permettre de prendre en compte les multiples degrés de liberté de l'outil. En outre, elle ne nécessite pas la connaissance de la position finale du mobile mais aide à la déterminer. Dans le cas où la dimension de l'espace des configurations est inférieure ou égale à la dimension de l'espace réel dans lequel évolue le mobile, la construction de cet espace peut être largement simplifiée (cf. figure 5.14) en utilisant des algorithmes de grossissement d'obstacles; dans le cas général c'est un problème très complexe.

Au vu de ces quelques remarques, nous avons adopté une approche mixte essayant au maximum de tirer parti des avantages de ces deux approches:

Il nous faut tout d'abord noter que la phase de mise en position de l'outil sur l'objet (ainsi que celle de retrait de l'outil) se fait généralement à orientation constante. C'est pourquoi dans un premier temps, nous contraindrons partiellement l'orientation de l'outil grâce aux contraintes "naturelles" imposées par la prise.

Nous rechercherons ensuite les obstacles à l'outil pour cette orientation donnée. Cette contrainte est imposée sur Y_P; ce choix se justifie amplement par le fait que Y_P est généralement déterminé par des critères intervenant dans la première phase. Dans les autres cas, une heuristique (présentée dans §5.3.6) permettra de faire ce choix. Y_P étant fixé, le plan de préhension l'est aussi et nous l'utiliserons par la suite de façon privilégiée afin de raisonner dans un univers de dimension 2. Les mouvements d'approche et de déproche seront effectués à Y_P constant.

Notre outil étant non convexe, nous le décomposerons donc en parties géométriquement significatives et approximativement convexes: les deux mors et le plat de la pince dans le cas de P. Nous étudierons ensuite l'espace balayé par chacun de ces mobiles. (cette information étant beaucoup plus riche que celle de l'espace globalement balayé par l'outil). Chacun de ces mobiles se déplace entre deux plans extrêmes, déterminés par Y_P, entre lesquels se trouvent les obstacles.

Plusieurs solutions s'offrent à nous pour raisonner sur ces obstacles. La première consiste à faire un calcul d'interférence entre l'espace balayé et les objets de l'environnement (méthode génération et test); il est cependant très difficile de traduire ces données en terme des positions et orientations valides du mobile (cf. figure 5.15).

Pour se départir de ce problème, [Wolter 84] approximent les mors de l'outil par des objets pseudo-2D (cf. figure 5.16) et peuvent se permettre ainsi d'effectuer l'étude des obstacles dans un plan. Les informations obtenues ainsi sont aisément traductibles en terme de positions valides pour l'outil (son orientation est fixée par ailleurs). Leur approche ne permet cependant pas de considèrer des outils tels que P.

A ce niveau nous préfèrons adopter une approche du type "espace des configurations". Pour cela, nous fixons heuristiquement l'orientation Z_P (si celle-ci n'est pas encore fixée) de façon à se ramener au problème d'un solide se déplaçant en translation; nous determinons alors l'espace des configurations correspondant. Cet espace doit être déterminé par grossissement des obstacles inversement à la forme du mobile; ce grossissement 3D étant relativement lourd (algorithmiquement) et les informations résultantes étant difficilement manipulables, nous avons choisi de discrétiser l'espace balayé en l'approximant par un ensemble de tranches correspondant à des tranches d'outil en mouvement; ceci nous permet d'effectuer le grossissement des obstacles dans un espace bi-dimensionnel. Le découpage étant adapté aux variations locales de géométrie de l'environnement, cette méthode présente généralement de meilleures performances algorithmiques qu'un grossissement 3D classique. Elle fournit en outre une information très fine sur les contraintes spatiales appliquées à l'outil.

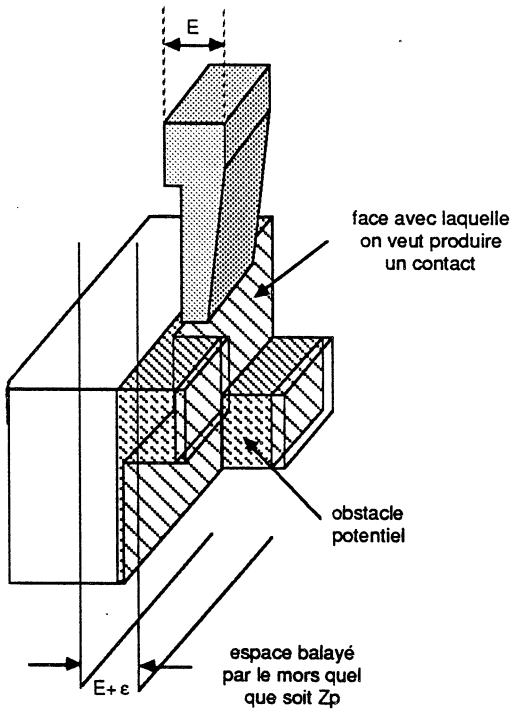


Figure 5.15: Espace balayé par les mors de l'outil

Les contraintes liées à chacun de ces éléments de P sont ensuite transformées en contraintes de positionnement pour l'outil P par un mécanisme simple de propagation de contraintes (cf. §5.3.4) autorisé par la réduction de chacune des parties élémentaires (tranches d'outil) à un point de référence. Le mécanisme de grossissement bi-dimensionnel permettant ce traitement est présenté en §5.3.5.
Le traitement de l'approche et de la déproche est fait en séquence en vérifiant respectivement les contraintes C_1 et C_2, et C_1 et C_3 (cf. §5.1.1).

L'algorithme utilisé lors de la seconde phase pour le traitement d'une prise déterminée par la première phase est présenté en Annexe C.

5.3.2 Détermination des contraintes initiales

La prise considérée est constituée d'un ensemble d'entités géométriques de l'objet à mettre en contact avec des parties de l'outil (les deux mors dans le cas de P). La simultanéïté géométrique de ces contacts restreint d'elle même l'ensemble des positions valides de l'outil. La détermination de cet ensemble de contraintes initiales de positionnement de l'outil conduit à restreindre le plan de préhension à une zône appelée **zône de préhension** définie de la façon suivante:

Définition 5.3.2.1 *Nous appelons zône de préhension, les parties du plan de préhension correspondant à des positions du point de référence de l'outil (O_P pour l'outil*

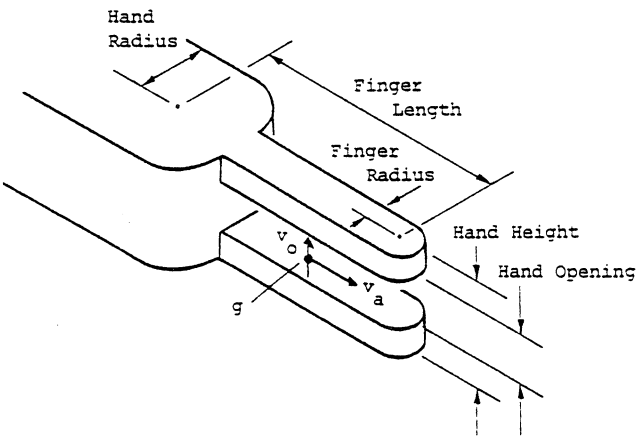


Figure 1 Hand model and parameters

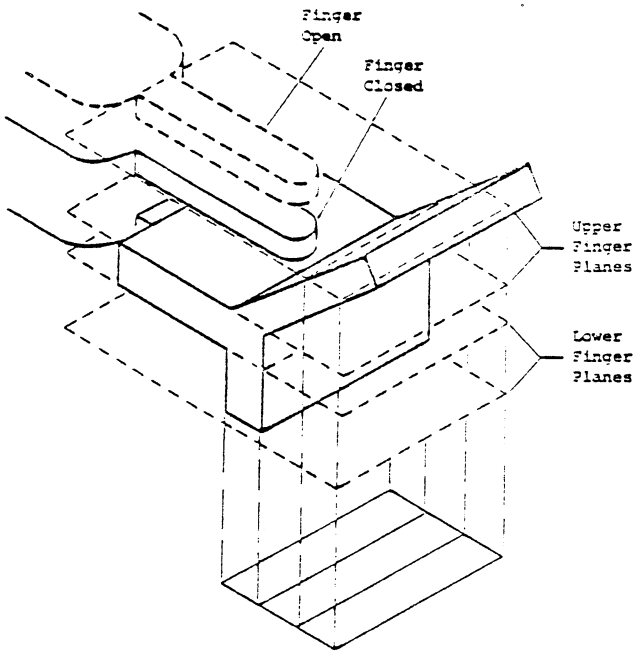


Figure 4 Cutting planes along top and bottom of fingers determine portion of part projected on to gripping planes.

Figure 5.16: Recherche des obstacles dans [Wolter 84]

P) assurant la réalisation simultanée des contacts représentés par les ensembles E_i d'EG.

Le calcul de cette zône nécéssite de fixer l'orientation de la pince en immobilisant Z_P et de choisir une direction D de mouvement (D_A ou D_D). Le choix de D est réalisé de manière heuristique et Z_P est choisie parallèle à la direction D. Ces choix pourront être remis en cause en cas d'échec lors de la recherche de la solution.

Soit Z_1 (resp. Z_2) l'ensemble des positions de O_P assurant un contact du $mors_1$ (resp. $mors_2$) avec l'ensemble d'EG E_1 (resp. E_2). La zône Z_0 de contraintes initiales est obtenue par intersection de Z_1 et Z_2. Les Z_i seront le résultat de l'intersection des ensembles de positions Z_{ij} de O_P permettant de maintenir un contact entre le $mors_i$ et une entité géométrique EG_j de E_i. Z_{ij} est obtenue par "grossissement" de l'entité EG_j inversement au $mors_i$ (cf. figure 5.17).

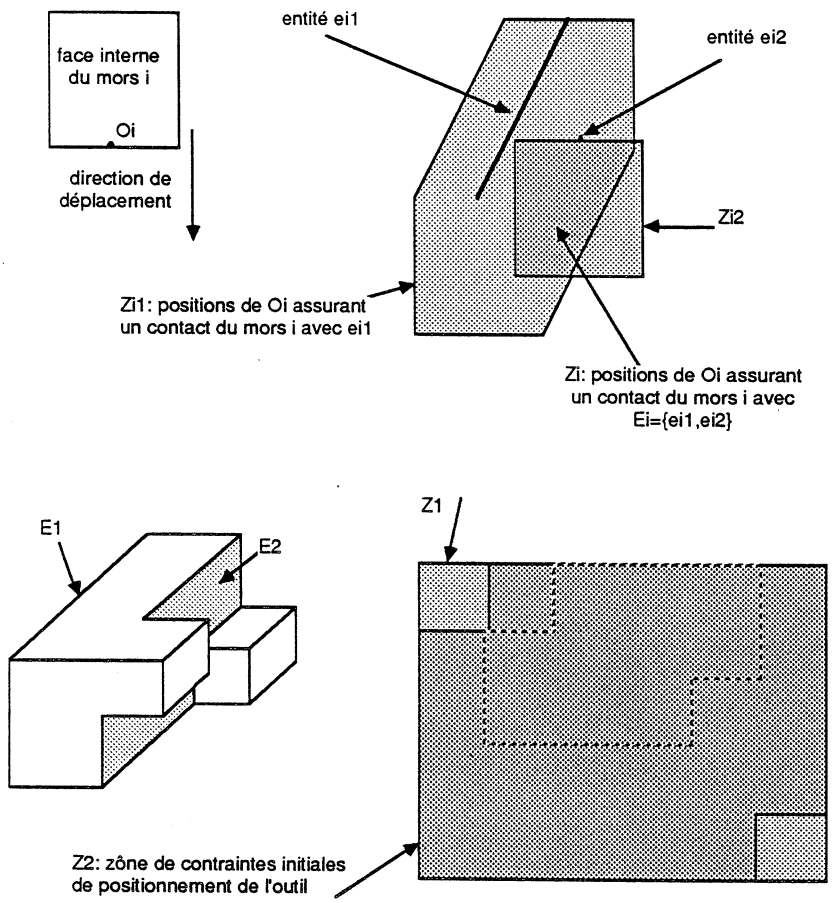


Figure 5.17: Calcul de la zône de contraintes initiales de positionnement des mors

5.3.3 Recherche des obstacles à la pince

Décomposition de la pince en parties élémentaires

L'idée de base de cette seconde phase est d'approximer l'outil de préhension par un ensemble de "parties élémentaires". Dans un premier temps [Laugier 83a], il nous a semblé naturel de rechercher les obstacles à chacune des parties principales (morphologiquement parlant) de la pince: la paume et les deux mors dans le cas de P. Nous appellerons ces parties, un peu abusivement, les convexes C_i de la pince. A chaque convexe C_i est lié un point de référence O_i défini dans le repère de la pince.

Soit Z_i la zône de préhension Z_0 ramenée au point de référence O_i de C_i: c'est à dire l'ensemble de contraintes de position de O_i correspondant aux contraintes de position de O_P représentées dans Z_0. Notre problème va consister à rechercher les obstacles à C_i, à restreindre Z_i en Z_i' et à en déduire un nouvel ensemble Z_0' de contraintes de positions pour O_P à partir de Z_i'.

Lorsque Y_P est fixé, le mouvement de C_i s'effectue entre deux plans parallèles

au plan de préhension. Les obstacles susceptibles de s'opposer au mouvement d'approche (resp. de déproche) de C_i correspondent à la partie de $E_I \cup \{O\}$ (resp. $E_F \cup \{O\}$) située entre ces deux plans parallèles (cf. figure 5.18). Notre idée a été de ne pas chercher un ensemble d'obstacles tri-dimensionnels à C_i mais un ensemble d'obstacles bi-dimensionnels à des tranches de C_i permettant ainsi de faciliter le raisonnement géométrique. Ce que nous appellerons tranche T_{ij} est en fait le résultat de la coupe de C_i par un plan parallèle au plan de préhension et situé entre les deux plans extrêmes limitant le mouvement de C_i. A chacune de ces tranches T_{ij} est lié un point de référence O_{ij} défini dans le repère de la pince.

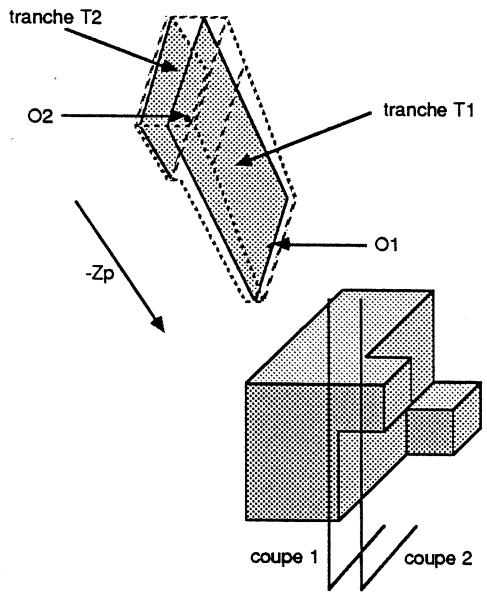


Figure 5.18: Espace balayé par une partie d'outil

Il nous faudra donc pour chaque tranche T_{ij} de C_i:

1. déterminer la zône Z_{ij} des contraintes portées sur la position de O_{ij} et correspondant aux contraintes courantes Z_i appliquées au point de référence O_i de C_i,

2. déterminer l'ensemble des contraintes Z'_{ij} obtenu par suppression dans Z_{ij} des positions de O_{ij} correspondant à des collisions de T_{ij} avec des obstacles,

3. en déduire un nouvel ensemble de contraintes Z'_i sur O_i

Les points (1) et (3) correspondent à une forme de propagation de contraintes géométriques qui sera décrite en §5.3.4. C'est ce même modèle de propagation qui est utilisé pour passer de Z_0 (contraintes sur l'outil) à Z_i (contraintes sur C_i) et vice-versa.

L'approximation de C_i par un ensemble de tranches T_{ij} résulte de l'approximation de l'espace balayé par C_i par un ensemble de coupes, parallèles au plan de préhension et en nombre variable; le nombre de ces coupes est déterminé par les variations locales de géométrie entre les deux plans (cf. §5.3.5).

Recherche des obstacles aux parties élémentaires

Le mécanisme de recherche des obstacles à une tranche T_{ij} est tout à fait simple. Soit E l'environnement de manipulation: $E_I \cup \{O\}$ ou $E_F \cup \{O\}$ selon le cas. Soit $OBST_{ij}$ l'ensemble des obstacles à T_{ij}, déterminé par coupe de E selon un plan P_{ij}. L'ensemble $OBST_{ij}$ est transformé par grossissement en un ensemble $OBST'_{ij}$ de façon à ramener le problème au point de référence O_{ij} de T_{ij}. La zône libre Z'_{ij} est obtenue par différence ensembliste de Z_{ij} et $OBST'_{ij}$ (cf. figures 5.19, 5.20).

Dans le cas d'un échec Z'_{ij} vide), un nouveau choix pour Z_P et pour la direction de mouvement sera effectué. Dans ce cas, l'ensemble des coupes n'aura pas à être réitéré puisqu'il ne dépend que de l'orientation du vecteur Y_P.

Le choix d'un nouveau Z_P et d'un nouveau D n'est pas toujours possible: c'est le cas notamment lorsque ces paramètres sont imposés par l'A^2LS; il peut également arriver que l'espace soit trop contraint et qu'il n'existe pas d'alternative pour ces paramètres. Dans ces deux cas, la prise considérée sera rejetée.

5.3.4 Propagation des contraintes

Trois raisons principales font que la propagation de contraintes se résout trivialement:

1. le mécanisme de grossissement permet de se départir du problème de forme de la tranche T_{ij} et de transformer les contraintes implicites sur le solide T_{ij} codées par des obstacles en un ensemble de contraintes implicites plus simples (plus directement interprétables) imposées à la position d'un point de référence O_{ij} de T_{ij},

2. la propagation d'un ensemble de contraintes implicites ou explicites C_1 portées sur la position d'un point M_1 est facilement transformable en un ensemble de contraintes C_2 portées sur la position d'un point $M_2 = F(M_1)$ par application de la fonction F à tous les points de C_1,

3. la détermination de la fonction F est rendue triviale dès lors que le plan de préhension est utilisé comme plan d'étude. F est toujours la composée d'une translation et d'une projection orthogonale, chacune d'elles étant éventuellement l'identité.

Comme nous le voyons, la propagation des contraintes de position entre les différents points de référence revient simplement à l'application de translations et

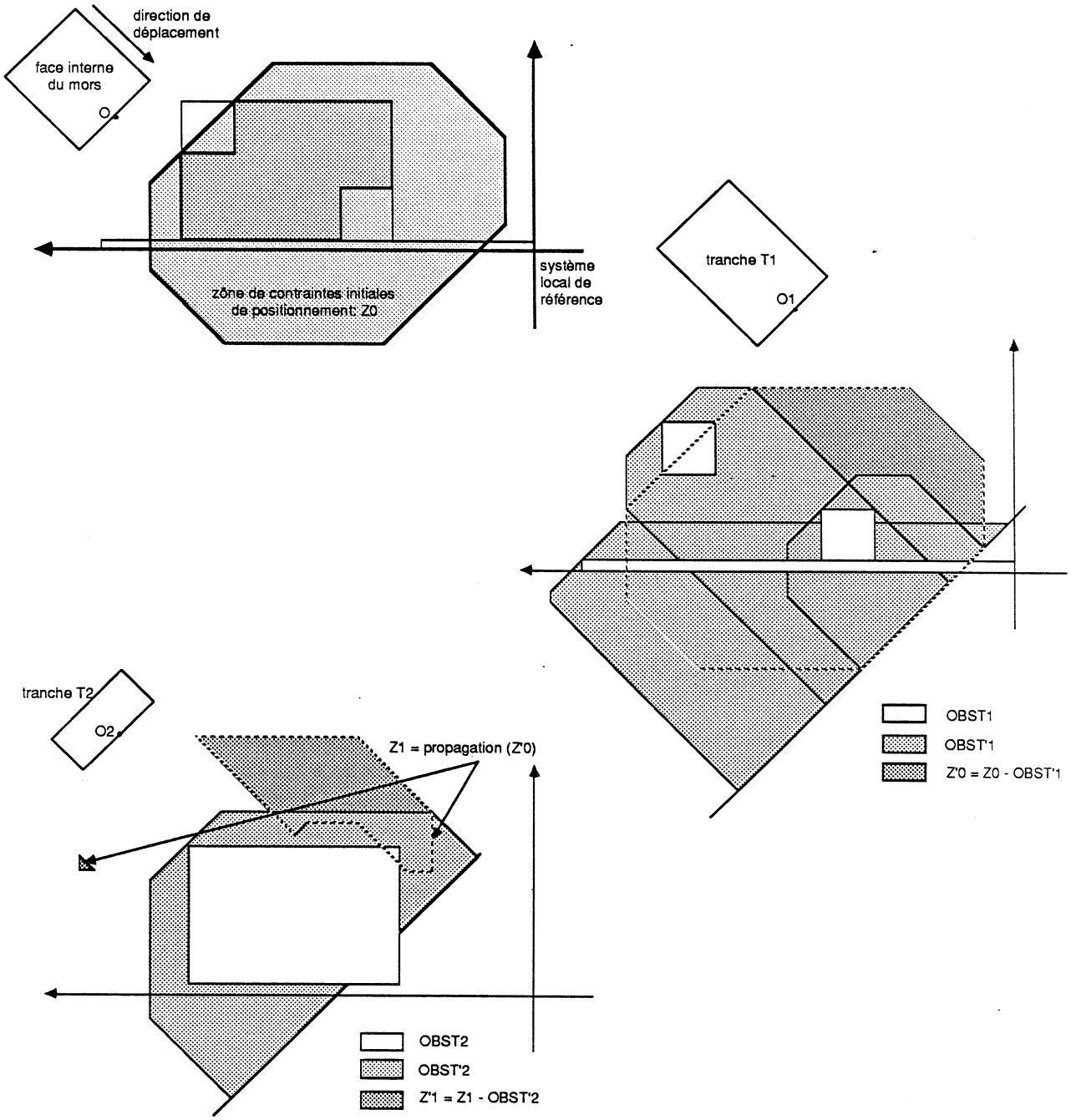


Figure 5.19: Recherche des zônes accessibles aux "tranches" d'outil et propagation à tout l'outil

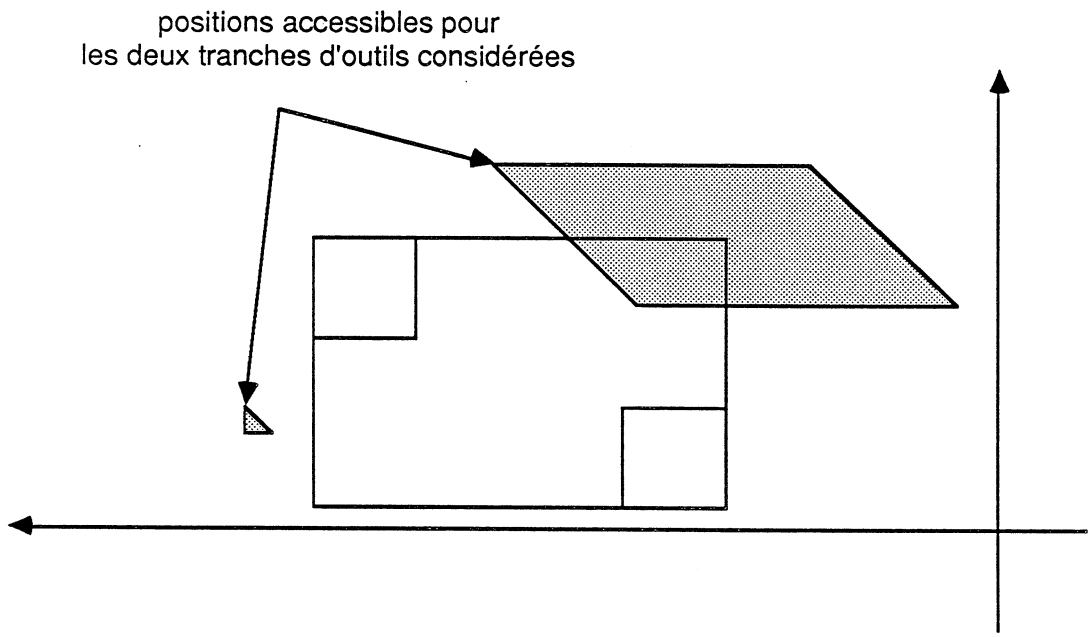


Figure 5.20: Zône de préhension accessible pour les deux "tranches" considérées

projections orthogonales aux zônes manipulées.

Si ce modèle de propagation parait très simple, c'est que le modèle de représentation choisi le permet; la restriction de l'outil à un ensemble de points de référence autorise en effet la transformation d'un ensemble de contraintes implicites représentées par des obstacles sur lequel il très difficile de raisonner en un ensemble de nouvelles contraintes dont la manipulation devient triviale. Nous mettons ici le doigt sur le problème fondamental du choix du modèle de représentation d'un problème donné; cet exemple montre que ce choix est crucial et que c'est de lui que dépend la complexité de la résolution.

5.3.5 Méthodes de grossissement et coupe

Méthode de grossissement

Ce mécanisme de grossissement est à la base de la phase d'A^2G. C'est en grande partie grâce à lui que le modèle de propagation est si simple. Il repose sur la constatation suivante: il est plus difficile de rechercher une trajectoire pour un objet mobile que de transformer le problème et de rechercher une trajectoire pour un point de ce solide. Cette remarque se doit d'être nuancée: elle est vraie au premier degré, mais ne l'est plus lorsque que l'on essaye de définir le sens du mot "transformer". Cette transformation est relativement aisée dans certains cas comme celui de la recherche

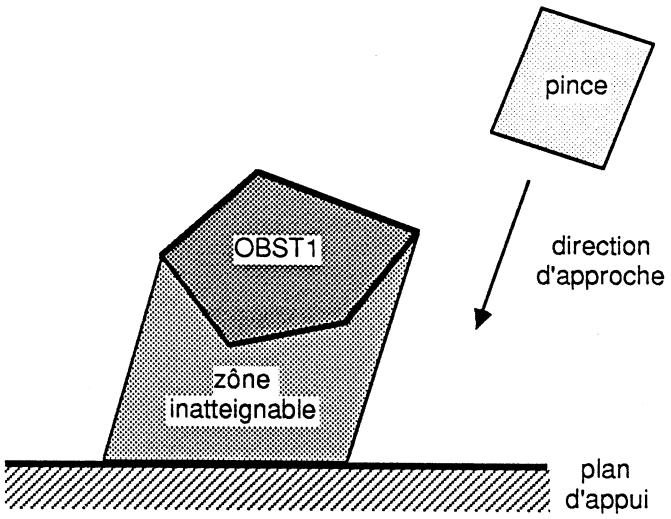


Figure 5.21: Zône d'ombre située "derrière" les obstacles

de trajectoire pour un objet bi-dimensionnel se déplaçant en translation; dans le cas général c'est un problème extrèmement complexe [Lozano 80]. Bien heureusement pour nous, notre application colle parfaitement au cas idéal présenté ci-dessus.

Soient donc M notre objet mobile et $OBST$ l'ensemble des obstacles susceptibles de s'opposer au mouvement de translation de M. La recherche d'une trajectoire pour M parmi $OBST$ est un problème équivalent à celui de la recherche de trajectoire pour un point de référence de M parmi un ensemble $OBST'$ défini comme suit: $OBST'$ est obtenu par grossissement de $OBST$ inversement à M.

Une définition plus formelle de ce principe peut être trouvée dans [Lozano 80]. Ce mécanisme est couramment utilisé dans les problèmes généraux de génération de trajectoires sans collisions ([Laugier 85a] par exemple).

Dans notre cas, nous distinguerons deux types de grossissement: l'un correspondant à la recherche de la zône Z_0 de contraintes initiales et l' autre à la recherche d'obstacles. Cette distinction est dûe au fait que les obstacles obtenus par coupe provoquent une ombre dans le plan considéré puisque tous les points situés derrière eux (par rapport à la direction de mouvement) sont inatteignables (cf. figure 5.21). Le principe du grossissement reste cependant le même.

Le grossissement est relativement simple à effectuer dans le cas d'objets convexes; deux solutions s'offrent donc à nous: approximer l'objet par un convexe puis le grossir ou bien décomposer l'objet en éléments convexes et grossir chacun d'eux. La première solution produit un résultat approximatif alors que la seconde conduit

à augmenter sensiblement la complexité algorithmique de la phase de grossissement.

Examinons à présent le mécanisme de grossissement dans le cas des obstacles. Le mobile en mouvement provient de la coupe d'un convexe de la pince par un plan; c'est un rectangle (dans le cas de P). Il est orienté selon la direction Z_P et se déplace dans la direction D. Chaque obstacle convexe est représenté par une liste d'arètes $(A_1, A_2, ..., A_n)$. Il est grossi de façon à pouvoir réduire la tranche T_{ij} à son point de référence O_{ij}.

Appelons θ_i l'angle entre Z_P et A_i ($i = 1$ à n). L'algorithme pour grossir l'obstacle applique tout d'abord une translation à toutes les arètes A_i telles que $\theta_i \in [0, \pi]$ (les autres arètes ne sont pas atteignables par le rectangle mobile). Le vecteur $\overrightarrow{V_i}$ de translation appliquée à A_i est défini comme suit:

$$\begin{cases} \overrightarrow{V_i} = (\frac{w}{2}, 0) \text{ si } \theta_i \in [0, \frac{\pi}{2}] \\ \overrightarrow{V_i} = (0, 0) \text{ si } \theta_i = \frac{\pi}{2} \\ \overrightarrow{V_i} = (-\frac{w}{2}, 0) \text{ si } \theta_i \in]\frac{\pi}{2}, \pi] \end{cases} \quad w \text{ est la largeur de la "tranche" d'outil considérée} \tag{5.1}$$

Ceci peut être résumé par l'expression mathématique suivante exprimée dans le repère de l'objet T_{ij}:

$$\overrightarrow{V_i} = \frac{\overrightarrow{V}}{\| \overrightarrow{V} \|} \times \frac{w}{2}$$

avec:

$$\overrightarrow{V} = D \wedge (D \wedge A_i) \times cos\theta_i$$

Soient $A_j, A_{j+1}, ..., A_m, ..., A_k$ la séquence d'arètes telles que $\theta_i \in [0, \pi], \forall i \in [j, k]$; A_j est la première arète ayant $\cos\theta_i < 0$ et A_m la dernière; A_k est la dernière arète ayant $\cos\theta_i < 0$.

Soit N_i (resp. N_i') l'origine (resp. l'extrémité) de l'arète A_i; $\forall i \in [j, k], N_i$ (resp. N_i') est translaté en un point M_i (resp. M_i').

L'obstacle grossi est alors obtenu en chainant les points:
$M_j, M_{j+1}, ..., M_m, M_m', M_{m+1}, M_{m+2}, ..., M_{k-1}, M_k, M_k'$ et en reliant les points M_j et M_k à une ligne L par des arètes parallèles à D. L est la droite orthogonale à la direction d'approche passant par le point de la zône de préhension le plus "éloigné" dans cette direction (cf. figure 5.22).

Dans le cas de P, le grossissement est largement simplifié par le fait que toute tranche d'outil est un rectangle; dans un cas plus général, il nous faudra utiliser des

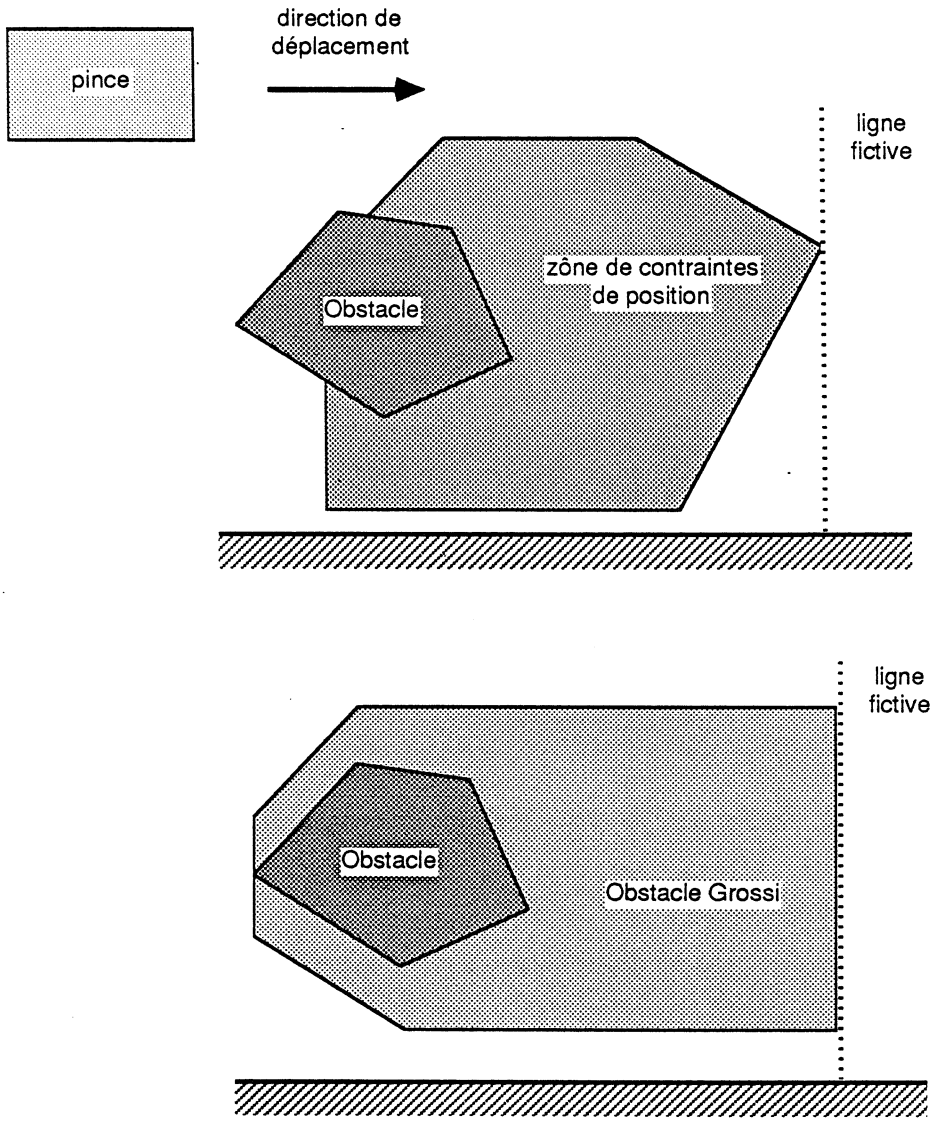


Figure 5.22: Grossissement d'obstacles

méthodes classiques de grossissement (cf. [Lozano 80]).

Remarque: La solution adoptée pour résoudre le problème de convexité des objets à grossir est différente selon que l'on s'occupe de la zône de préhension ou que l'on recherche les obstacles à la prise; elle consiste à grossir exactement la zône Z_0 initiale (second schéma) et à surcontraindre le problème en grossissant l'enveloppe convexe des obstacles obtenus par coupe (premier schéma). De cette manière, nous perdons quelques solutions mais nous sommes sûrs que les solutions trouvées sont valides.

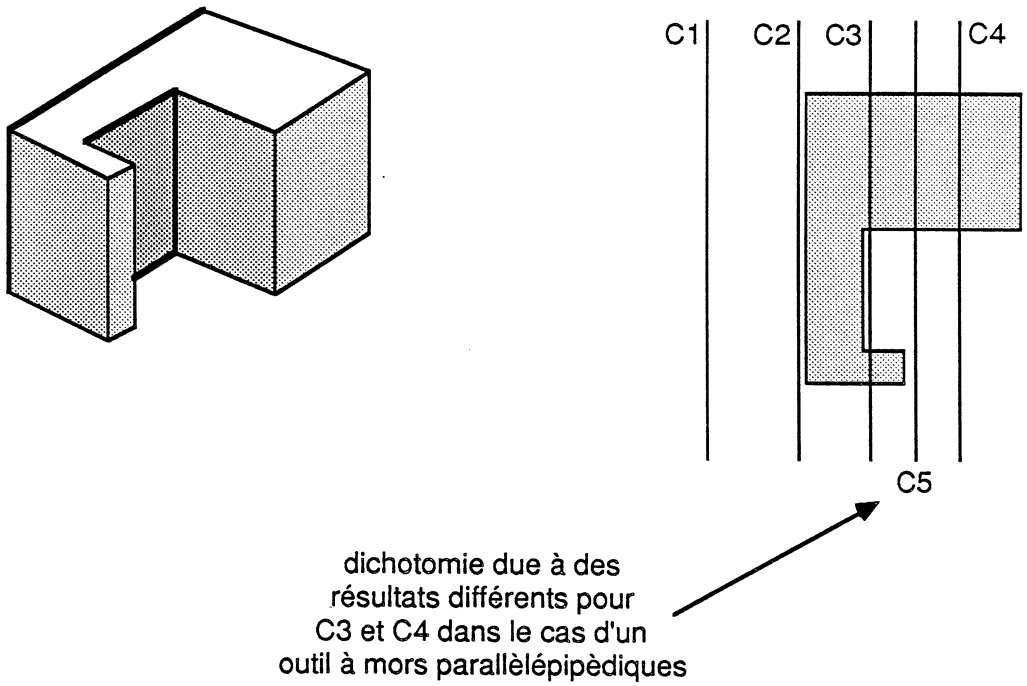


Figure 5.23: Recherche des obstacles par des coupes par un plan

Algorithme de coupe

La méthode de coupe de l'espace balayé fonctionne sur le principe de la dichotomie. La fréquence de coupe dépend des variations géométriques locales de l'environnement d'une part et de l'outil d'autre part. Elle repose sur une hypothèse de variation continue des géométries.

Lorsque l'on a effectué deux coupes de l'environnement par des plans P_1 et P_2, la décision de couper par un plan P_{12} médiateur de P_1 et P_2 peut provenir de 2 causes:

- P_1 et P_2 fournissent des résultats différents et proviennent donc d'une variation de l'environnement
- les tranches T_1 et T_2 de l'outil sont différentes, ce qui correspond à une variation de la morphologie de l'outil nécéssitant une étude plus fine.

Ces coupes sont réalisées jusqu'à atteindre des résultats "stables" ou une résolution minimale (cf. figure 5.23). Il est clair que la finesse de description de l'outil influe sur les performances de l'algorithme et qu'il nous faut faire un compromis entre la qualité des résultats et le temps de calcul nécéssaire.

L'algorithme de coupe à proprement parler n'est pas présenté ici car il est de peu d'intérêt par rapport à la description de notre travail. Sa seule originalité est de reposer sur une analyse de voisinage d'entités géométriques réduisant la complexité algorithmique. Il utilise pour cela les informations liées à la topologie locale des entités (cf. §3.1.1)

5.3.6 Utilisation d'heuristiques pour la détermination de $P_{P/O}(\Pi)$

Comme nous l'avons vu précédemment, le choix heuristique peut intervenir en deux endroits:

- pour la détermination de Y_P lorsque ce dernier n'est pas contraint par l'A^2LS,
- pour la détermination de D (D_A ou D_D) et de Z_P en début de l'A^2G ou bien en cas d'échec lors de la résolution.

Les choix de Y_P et D reposent sur les mêmes principes heuristiques. Z_P est choisie égale à D. Ce choix peut sembler arbitraire; il correspond cependant à une réalité pratique (ou plutôt à une habitude de manipulation). Le choix de Y_P et D est basé sur une étude grossière des obstacles potentiels à la pince. Cette étude est effectuée dans un plan dans lequel des secteurs angulaires libres sont déterminés. La direction est choisie dans ces secteurs libres.

Remarque: Dans certains cas, le choix de D (donc de Z_P) est imposé par la structure du manipulateur considéré. C'est le cas lorsque le poignet ne possède qu'un degré de liberté en rotation (cf. figure 5.24).

Choix des plans d'étude pour Yp

Y_P n'est pas contraint dans le cas de prises réduites à un singleton. C'est typiquement le cas lors de la saisie d'un cylindre ou par un trou cylindrique ou lors de la saisie d'une sphère. Dans le cas où la direction Z_P est contrainte (par exemple pour un trou cylindrique) nous effectuons des coupes perpendiculaires à Z_P au voisinage de l'entrée du trou. Dans tout autre cas, nous effectuons ces coupes parallèlement à la surface d'appui de l'objet et au-dessus de l'objet par rapport à ce plan. Ces coupes représentent quelques-uns des obstacles à l'outil. Chacun des plans est lié à un point de référence (correspondant à la projection du centre de la sphère ou à l'intersection avec l'axe du cylindre.

Choix des plans d'étude pour D

Dans le cas de D, le plan de préhension est évidemment privilégié. Les plans choisis correspondent aux plans des coupes qui sont effectuées pour la recherche d'une solution. Le point de référence dans chacun des plans est la projection du centre de gravité de l'objet.

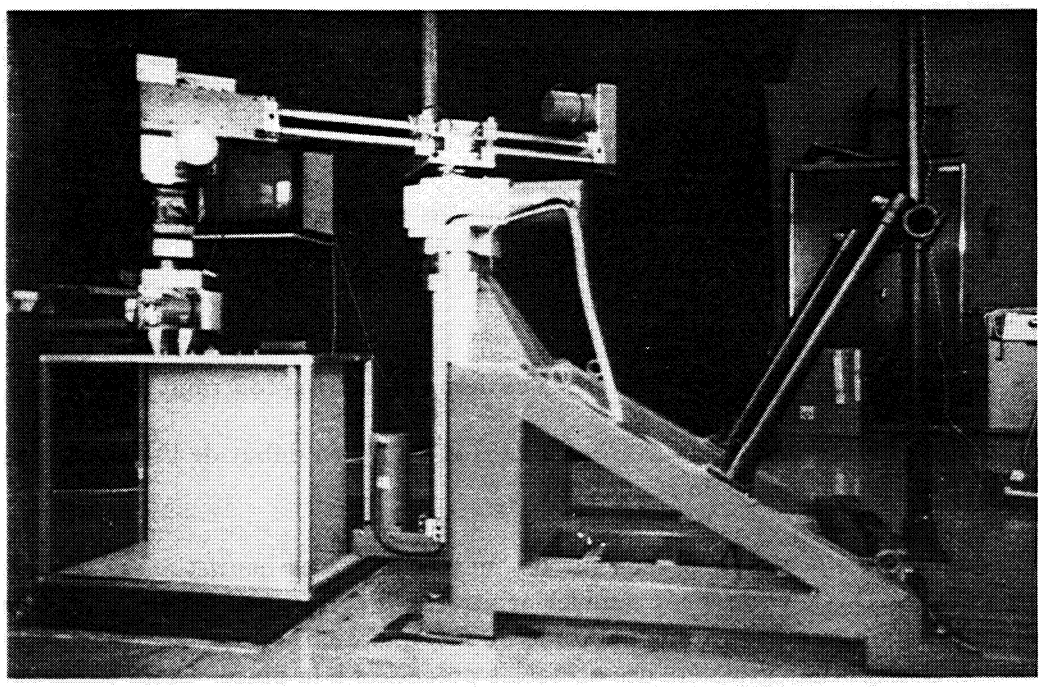


Figure 5.24: Robot ROBITRON à quatre degrés de liberté

Recherche de la direction (D ou Yp)

L'ensemble des coupes obtenues précédemment représente les obstacles au mouvement de la pince (de manière très approximative dans le cas de Y_P). Un ensemble de secteurs angulaires libres va être déterminé sur chacun des plans à partir du point de référence correspondant ((cf. figure 5.25). Un ensemble de secteurs angulaires correspondant à l'union des contraintes représentées par les ensembles précédents est obtenu par intersection de ceux-ci. La direction D peut alors être déterminée dans le plus grand des secteurs de l'ensemble. Ce sera la médiatrice du secteur angulaire le plus grand ou bien la direction verticale si elle n'en est pas trop éloignée. Le choix de Y_P est un peu différent puisqu'il s'agit de positionner les deux mors de l'outil; il nous faut donc trouver deux secteurs angulaires en "vis à vis" (contenant une même droite: cf. figure 5.25). Dans le cas où de tels secteurs n'existeraient pas la prise serait rejetée. C'est également le cas si l'ensemble de secteurs libres est vide où si ces secteurs sont trop petits.

5.4 Mise en œuvre

5.4.1 Interface géométrique avec les modules de raisonnement

Les modules de raisonnement géométrique développés aussi bien pour l'automatisation de la Robotique de manipulation que dans d'autres contextes (vision par

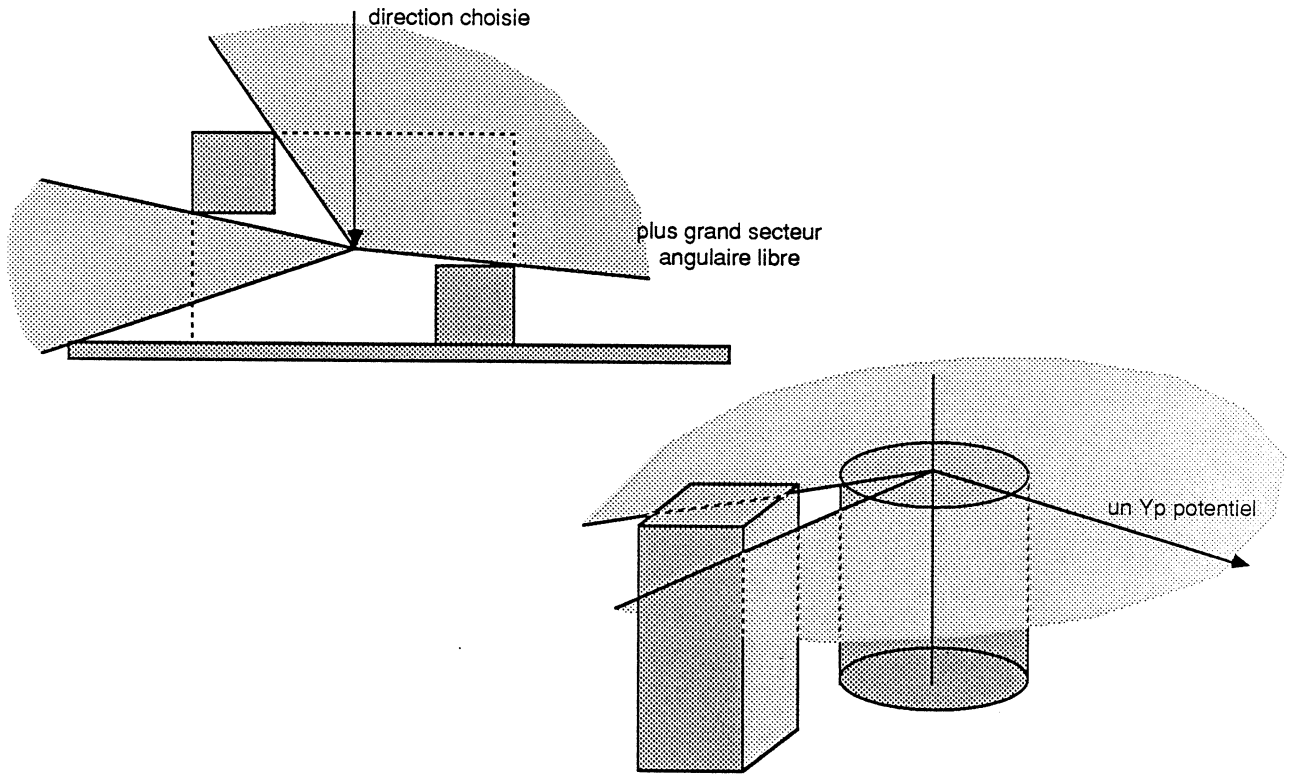


Figure 5.25: Recherche heuristique d'une direction dans les secteurs angulaires libres

exemple) sont généralement fondés sur l'évaluation de propriétés et d'opérateurs géométriques de base (parallèlisme, orthogonalité, tangence, distance, projections...). Ils opèrent sur des entités des espaces affine et euclidien (plan, droite, vecteur...).

Même si dans leur principe, les méthodes mises en œuvre peuvent être assez générales, elles présentent fréquemment certaines limitations provenant de la traduction directe des propriétés en fonction d'une représentation particulière des objets.

C'est pourquoi il parait important de séparer au mieux le raisonnement géométrique du modèle utilisé grâce à la définition d'une interface adéquate. Celle-ci doit permettre de lier un "langage" géométrique manipulant les entités géométriques précitées à l'ensemble de fonctions d'accès spécifiques au modèle utilisé.

Par exemple, le parallèlisme sera exprimé dans le "langage" de la façon suivante:

(TYPEPLAN e_1)
(TYPEPLAN e_2)
(colinéaire (*normale* e_1) (*normale* e_2))
$\Longrightarrow$
(PARALLELE e_1 e_2)

(TYPEPLAN e_1)
(TYPEDROITE e_2)
(colinéaire (*normale* e_1) (*vecteur* e_2))
$\Longrightarrow$
(PARALLELE e_1 e_2)

.../...

Les fonctions en *italique* sont des fonctions d'interfaçage entre le "langage" et les fonctions d'accès au modèle. Les fonctions en baton sont des fonctions élémentaires de calcul géométrique opérant sur des points, vecteurs, droites etc... . Les MAJUSCULES correspondent aux propriétés géométriques définies ou utilisées par le "langage".

L'introduction de tout nouveau type d'objet dans le modèle nécessite alors uniquement la modification des fonctions d'interfaçage. Ceci correspond à une économie de travail non négligeable dès qu'un système réunit plusieurs modules de raisonnement géométrique.

L'intérêt principal de cette approche est de permettre à des modules de raisonnement géométrique utilisant des représentations éventuellement différentes de partager un même langage et de mêmes informations de nature géométrique.

Elle présente un autre intérêt lié à l'aspect déductif de la géométrie. Rares sont en effet les modules de raisonnement géométrique utilisant, en plus des définitions de la géométrie, les théorèmes qui en découlent. Le "langage" pourrait ainsi être transformé en base de connaissances géométriques exploitée par un démonstrateur de théorèmes.

Pour le parallèlisme, on rajoutera par exemple le théorème suivant:

(ORTHOGONAL e_1 e_3)
(ORTHOGONAL e_2 e_3)
$\Longrightarrow$
(PARALLELE e_1 e_2)

Ceci autorisera un certain non-déteminisme dans la vérification des propriétés géométriques.

A l'heure actuelle, seul l'aspect interfaçage a été considéré et cela a donné lieu à l'implantation d'un langage-interface exprimé sous forme déclarative. Par exemple:

(FACEPLANE e_1)
(FACEPLANE e_2)
(colinéaire (*normale-extérieure* e_1)
(*normale-extérieure* e_2))
$\Longrightarrow$
(PARALLELE e_1 e_2)

Les fonctions en *italique* sont des fonctions d'accès au modèle. Les fonctions en baton sont des primitives de calcul géométrique. Les MAJUSCULES correspondent à des propriétés définies ou utilisées par le langage-interface.

Ce langage-interface permet d'ores et déjà de déconnecter les modules de raisonnement géométrique du modèle de représentation des objets. Il a été implanté en LOGLISP, langage qui mèle un environnement LISP (MACLISP) à un environnement logique, permettant ainsi la cohabitation des connaissances géométriques (de nature plutôt déclarative) avec nos modèles et leurs fonctions de manipulation (procédurales).

[Arbab 85] présentent une approche voisine dans un environnement complètement logique. Or, il est beaucoup plus facile de faire du déclaratif en LISP que du procédural en PROLOG. Ils se heurtent donc à la lourde tâche de reécrire des fonctions de calcul géométrique élémentaires et de définir leur modèle dans cet environnement logique.

Un travail beaucoup plus fondamental est présenté par [Kapur 85], [Mundy 85]. Son ambition est de fournir un support formel de raisonnement géométrique pour la Robotique et la Vision. Une connaissance géométrique codée grâce au calcul des prédicats du premier ordre, combinée à des règles de reécriture permet plusieurs types de déduction (raisonnement algébrique sur des polynômes, raisonnement logique utilisant le principe de résolution et calcul formel).

5.4.2 Implantation et expérimentation

L'exposé de notre approche a été fait dans le cadre général de prises mettant en jeu un nombre quelconque d'EG. L'implantation n'a cependant été réalisée que pour des prises pour lesquelles les E_i sont des singletons. D'autre part, l'efficacité de l'heuristique pour la recherche d'une direction - Z_P ou Y_P - (cf. §3.6) n'a pas été évaluée. Nous avons, en effet, adopté une stratégie plus simple consistant à choisir des directions simples (approche verticale ou horizontale, Y_P aléatoire).

La saisie a été implantée sur un CII-HB70 (système Multics) sous la forme de deux modules distincts traitant respectivement l'A^2LS et de l'A^2G . Ces deux modules partagent à la fois des primitives de calcul géométrique, des fonctions

géométriques plus élaborées et les fonctions d'accès au modèle des objets.

Le module de recherche de prises est écrit en LOGLISP. Etant fondé sur le raisonnement géométrique, il utilise de façon privilégiée le langage-interface décrit au paragraphe précédent. La recherche de prises est elle-même codée sous forme déclarative.

La facilité d'écriture du noyau de l'A^2LS est payée par la lenteur d'exécution inhérente au langage LOGLISP (quelques 10 minutes pout engendrer les 3 prises potentielles d'un cube). C'est pourquoi nous envisageons de traduire cette partie en LISP en lui associant un moteur d'inférence simple.

L'algorithme utilisé pour le calcul d'accessibilité globale est écrit en MACLISP. Il utilise directement les fonctions géométriques et les primitives géométriques sans passer par le langage-interface puisque le raisonnement spatial mis en œuvre n'a pas besoin des connaissances géométriques qui le constituent (cf. figure 5.26).

Le travail le plus important dans cette implantation a consisté à écrire l'ensemble de fonctions et primitives géométriques utilisées par les deux modules.

Les primitives géométriques opèrent sur les entités des espaces affine et cartésien. Elles contiennent tous les calculs vectoriels, angulaires, les distances, projections, intersections, tests de colinéarité, perpendicularité etc... . Les erreurs introduites soit au moment de la modélisation des objets, soit à cause de calculs imprécis imposent d'effectuer tous les tests à un ε près; le bon choix de sa valeur dépend uniquement de l'intuition de l'utilisateur du programme.

De nombreuses autres fonctions géométriques plus complexes sont nécessaires. Certaines opèrent dans l'espace 2D, d'autres dans l'espace 3D - intersection, union, différence d'objets 2D (polygônes par exemple), appartenance d'un point à un objet 2D, enveloppe convexe, grossissement, coupe d'un objet 3D par un plan - pour ne citer que les plus utilisées. Si toutes ces fonctions sont très simples dans leur principe, leur implantation se heurte à la prise en compte d'une multitude de cas particuliers qu'il faut traiter de façon homogène. Les opérateurs ensemblistes en sont un très bon exemple (cf. [Pertin 84] pour plus de détail sur la méthode employée).

Ces fonctions s'appliquent à des objets polygonaux et polyèdriques. Cette restriction, uniquement liée à l'implantation de la seconde phase et non à son principe, se justifie amplement par le fait que des fonctions géométriques plus générales existent dans les systèmes de CAO et que leur implantation dans le cadre de cette thèse aurait été de fort peu d'intérêt.

L'expérimentation a été faite sur des modèles d'objets et d'outil assez simples (cf. figure 5.27). Nous envisageons, dans un premier temps de développer et de

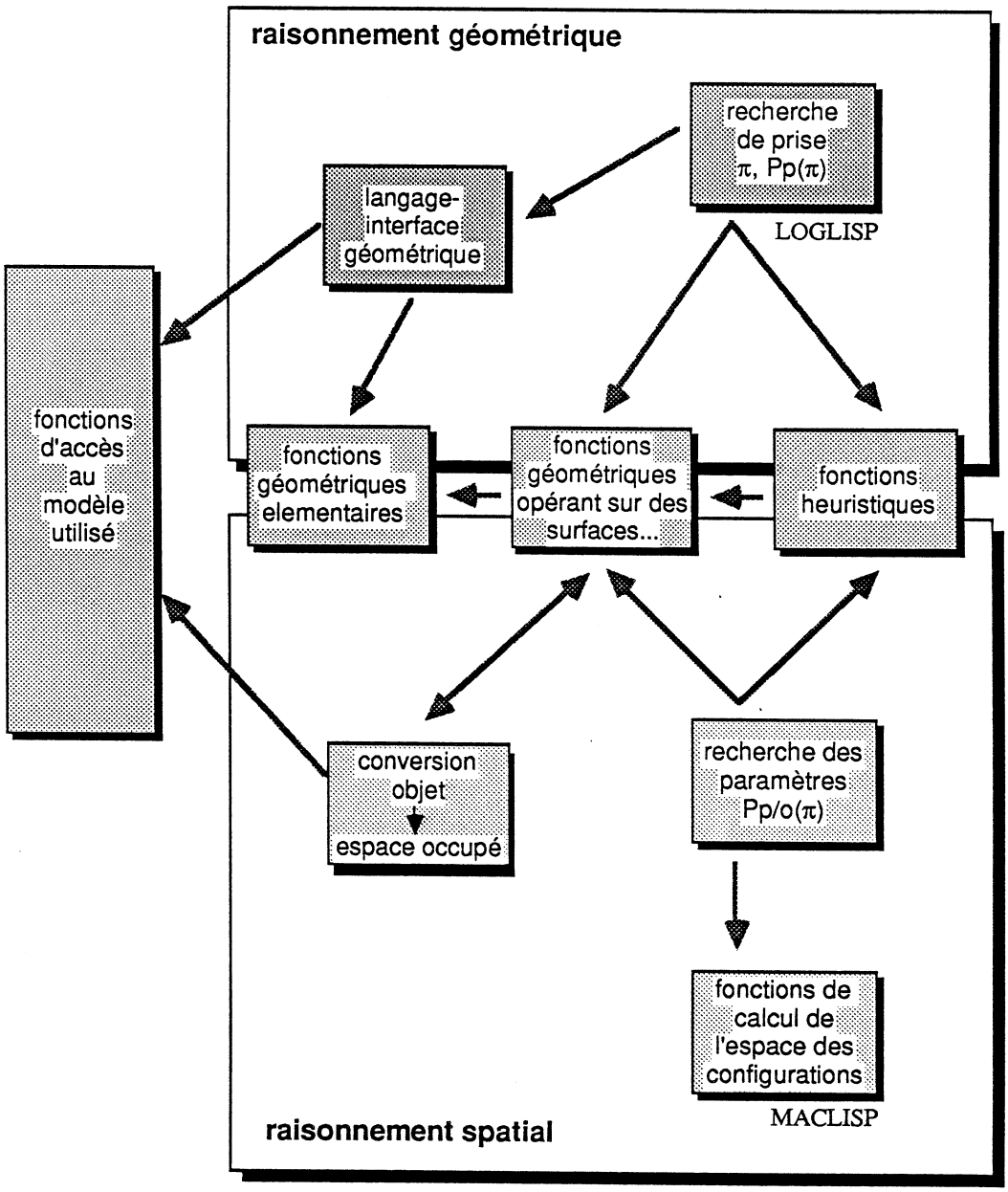


Figure 5.26: Architecture du logiciel de saisie

comparer différentes heuristiques de choix de trajectoires et de conclure ce travail par sa connection effective à un manipulateur.

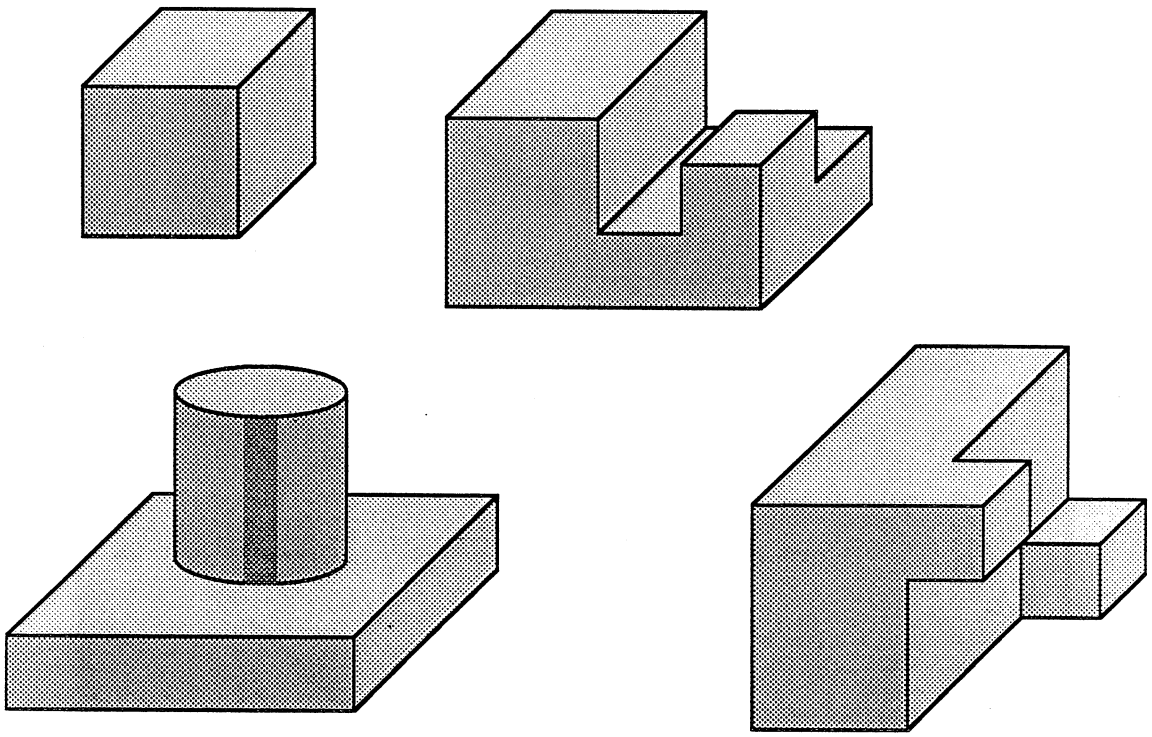


Figure 5.27: Exemples d'objets traités

5.4.3 Discussion des choix algorithmiques effectués

Comme nous l'avons vu précédemment, notre approche pour tester de l'accessibilité globale d'une prise est fondée sur une méthode de propagation de contraintes géométriques. Dans cet exposé, il est apparu que nous avons fait le choix consistant à propager des solutions locales d'une partie de l'outil à l'outil dans son entier: ce sont des contraintes explicites sur les positions du point de référence de l'outil.

Nous aurions également pu, dans le processus de propagation, raisonner non plus en termes de solutions pour un point de référence de l'outil mais en termes d'obstacles au mouvement de ce point: ce sont alors des contraintes implicites sur la position du point de référence (cf. figure 5.28).

Dans le premier cas, une double propagation de l'ensemble des solutions valides est nécessaire (pour se placer dans le repère local à la partie considérée puis pour revenir au système de référence global de l'outil). Dans le second cas, la propagation ne se fait que dans un sens. Il est clair que ce processus de propagation étant

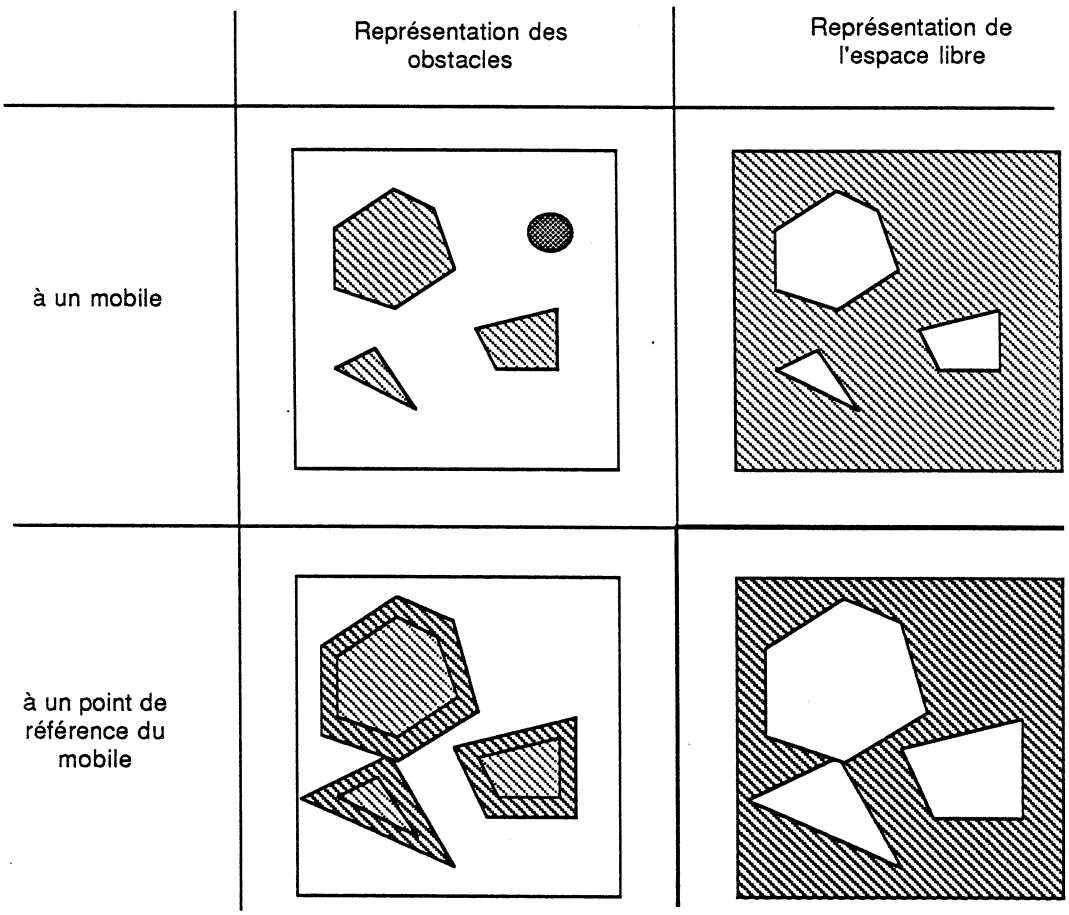


Figure 5.28: Représentions implicites ou explicites des obstacles

réitéré pour chaque tranche d'outil, un gain de temps substantiel doit résulter de l'implantation de la seconde approche.

Si l'approche décrite dans ce document est la propagation de solutions, c'est premièrement parce que, historiquement, elle fût notre première approche du problème; mais c'est aussi et surtout, parce qu'elle permet d'illustrer plus clairement les différents degrés d'explicitation dans la représentation des contraintes sur les mouvements d'un solide. Son implantation serait tout à fait immédiate sur la base de l'algorithme existant.

Considérons maintenant la phase de traitement correspondant à une tranche

T_{ij} de l'outil; deux possibilités s'offraient également à nous: était-il préférable de manipuler n obstacles distincts ou un obstacle résultant de l'union de ceux-ci?

Ceci se traduit par les choix algorithmiques suivants:

n grossissements	n grossissements
n-1 unions	
1 différence	n différences

ou encore

n-1 unions	n grossissements
1 grossissement	
1 différence	n différences

ce qui,dans le cas de P, est équivalent puisque l'union et le grossissement sont deux opérations commutatives.

Il est assez difficile d'évaluer la complexité de tels algorithmes puisque celle-ci évolue en fonction du nombre des sommets et arètes traités et que ce nombre fluctue à cause des opérations ensemblistes. Ceci étant, union et différence étant deux paramètrisations du même algorithme et le processus de grossissement étant relativement lourd (si l'on considère des objets non convexes), la première solution nous semble intuitivement être la meilleure. Une étude plus précise serait cependant nécessaire pour confirmer ou infirmer la validité de ce choix.

Chapitre 6

Discussion

6.1 Spécificité de l'approche

6.1.1 Par rapport au mode de saisie

Notre méthode a été développée dans le contexte d'une saisie latérale mettant en jeu un contact avec les surfaces des doigts de l'outil considéré. L'extension de l'approche à un autre type de saisie (par le bout des doigts par exemple) nécessiterait d'étendre les propriétés employées dans la première phase à de nouveaux types de contacts. Cette généralisation ne semble pas présenter de difficultés majeures et devrait ainsi permettre d'étendre les possibilités de notre approche.

6.1.2 Par rapport au modèle

Résoudre chacun des sous-problèmes de l'automatisation de la saisie nécessite de disposer de différents types de représentations. La recherche de prise est intimement liée à la notion d'objet; l'étude d'accessibilité s'appuie sur la notion d'espace occupé; l'étude d'équilibre et de stabilité a besoin d'une modélisation de l'univers en termes mécaniques; enfin la compatibilité contextuelle repose sur une modélisation de la tâche en termes qualitatifs et quantitatifs (géométriques et mécaniques). Il n'existe aucun type de modèle propre à fournir toutes ces représentations; en conséquence les méthodes développées se restreignent souvent à l'un ou à l'autre de ses sous-problèmes. A l'heure actuelle, l'outil de modélisation le plus polyvalent est celui de type CAO (par opposition aux modèles perceptifs ou purement mécaniques). Malgré cette "universalité", nous avons veillé à déconnecter au mieux, notre approche du mode de représentation utilisé.

Elle nous semble relativement peu spécifique, dans son fondement, par rapport au type de modélisation employé puisqu'elle repose sur un raisonnement sur la saisie effectué indépendamment de la représentation utilisée.

La première phase s'appuie, en effet, sur un raisonnement sur les différents types de contacts pouvant intervenir entre l'outil et l'objet saisi. Le raisonnement sur ces

contacts (de type plan, droite ou point) est d'autant plus déconnecté du mode de représentation des objets qu'il utilise de façon privilégié, le langage-interface que nous avons développé.

Le principe de raisonnement spatial utilisé dans la seconde phase est également peu sensible au mode de représentation du fait d'une reconstruction de l'espace occupé.

Si cet aspect donne à notre méthode une relative originalité par rapport aux travaux du même type, il est clair que cette généralité est un peu affaiblie par l'implantation qui en a été faite. Nous avons traité d'objets composés de polyèdres, de cylindres et de sphères. D'autres types d'objets nécessiteraient la généralisation de la propriété d'accessibilité locale permettant d'engendrer les contacts potentiels. La seconde phase a été implantée sous la forme d'algorithmes opérant sur des objets polyèdriques; la généralisation de tels algorithmes a déjà été faite dans les systèmes de CAO et ne constitue donc pas un obstacle à l'amélioration de la méthode.

6.1.3 Par rapport à l'outil

Comme pour le mode de saisie, l'extension de notre méthode à d'autres types d'outils impliquerait une généralisation par rapport aux contacts mis en œuvre (dans le cas de doigts cylindriques par exemple). Notre approche peut cependant s'appliquer à un certain type d'outils sans modifications préalables: la classe des outils à mors plans.

Chacun des filtres utilisés dans la première phase est lié à une ou plusieurs caractéristiques de l'outil et traduit une partie de l'analyse des contacts; cette approche parait donc relativement flexible. De là, le pas qui consiste à penser que la génération de ces filtres peut être "automatisée" pour ce type d'outils (mors plans) est vite franchi. L'idée que nous avons de cette génération automatique est qu'elle ne permettra pas d'inventer de nouveaux filtres mais plutôt d'instancier un ensemble de filtres paramètrables grâce aux caractéristiques de l'outil considéré. Les trois filtres de Visibilité, P-accessibilité locale et écartement sont suffisamment génériques pour permettre cette paramètrisation. Le filtre de Parallèlisme se doit d'être remplacé par un critère plus général portant sur des contraintes angulaires; le Parallèlisme en serait un cas particulier. Il nous semble que cet ensemble de filtres doit être suffisant puisque lié à une méthode générale.

L'examen des caractéristiques des parties de la pince significatives pour l'A^2LS (par exemple les faces internes et externes des mors dans le cas de P) permettra de déterminer les paramètres des filtres utilisés. L'ordonnancement des filtres sera fait en respectant les contraintes d'antériorité existant entre eux.

Cet ensemble de 4 filtres nous parait bien adapté à des outils classiques comportant 2 ou 3 mors. Ce n'est évidemment pas directement le cas des outils plus

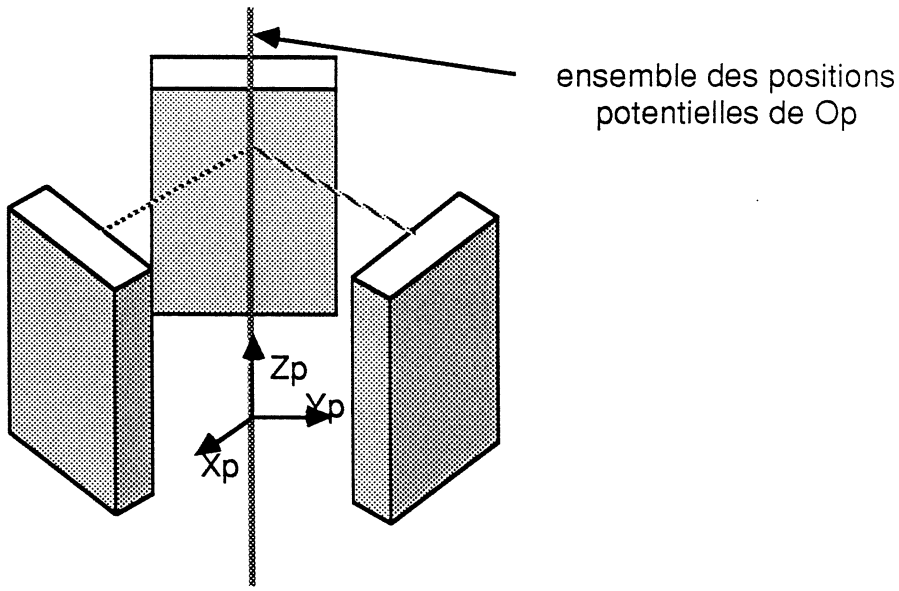


Figure 6.1: Pince à trois mors

complexes (anthropomorphiques par exemple). Ce type d'outil comporte en effet de nombreux degrés de liberté et les mouvements de chaque doigt sont relativement indépendants; ils ne contraignent donc pas suffisamment la recherche d'une prise. Notons cependant que ce genre d'outil se comporte globalement comme une pince plus classique et que le positionnement fin de chacun des doigts peut être assuré par un contrôle tactile.

La seconde phase repose une méthode de propagation de contraintes qui n'est pas limitée à un outil particulier. Nous pouvons affirmer, au contraire, qu'elle s'en trouvera facilitée dans le cas d'outils "classiques" plus complexes que P. En effet, dans ce cas, l'objet sera plus contraint en position et orientation par rapport à l'outil; il nous faudra donc engendrer moins de solutions potentielles. Examinons par exemple un type de pince à 3 mors correspondant à la figure 6.1. Dans ce cas, les vecteurs Y_P et Z_P seront déterminés par le triple contact dès l'A^2LS. Les positions potentielles de O_P se situeront non plus sur un plan de préhension mais sur une droite de préhension. La phase d'A^2G sera donc plus rapide puisque ne comportant pas de choix heuristiques et parce que plus contrainte.

6.2 Inconvénients et avantages de notre méthode

6.2.1 Problèmes de compatibilité contextuelle

Notre méthode nous permet de traiter une certaine forme de compatibilité contextuelle de niveau objet puisque, dans la seconde phase, nous choisissons une prise compatible avec les environnements de saisie et de montage de l'objet. Certaines

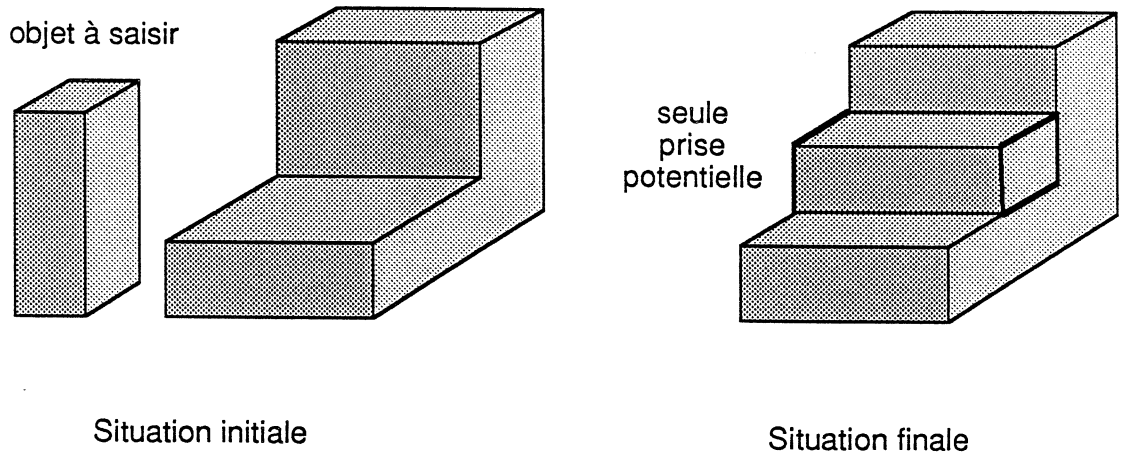


Figure 6.2: Incompatibilité entre les environnements de saisie et de lacher d'un objet

formes de compatibilité ne sont cependant pas traitées dans notre approche.

Changements de prise

Il peut arriver que les ensembles de prises associés aux positions initiale et finale de l'objet ait une intersection vide (cf. figure 6.2). Il faut, dans ce cas, effectuer un changement de prise, soit grâce à un second manipulateur, soit par posage et reprise de l'objet.

Très peu de chercheurs se sont penchés sur ce problème; citons le travail de [Dauchez 83] se plaçant dans le contexte de la coopération de deux manipulateurs. La méthode développée repose sur l'utilisation d'une bibliothèque de prises liées à l'objet manipulé; ces n prises sont représentées par des repères cartésiens $R_1, R_2, ..., R_n$. Une matrice A modélise les possibilités de tenue simultanée de l'objet par les deux manipulateurs.

$A(i,j) = 1$ si M_1 tenant l'objet par R_i, M_2 peut le saisir par R_j
$A(i,j) = 0$ sinon

où M_1 et M_2 sont les deux manipulateurs considérés

Passer d'une prise R_i à une prise R_j pour le manipulateur M_1 (par exemple) consiste donc à trouver un p tel que $A_{2p-1}(i,j) = 1$; $A_{2p-1} = A(BA)^p$ où B est la

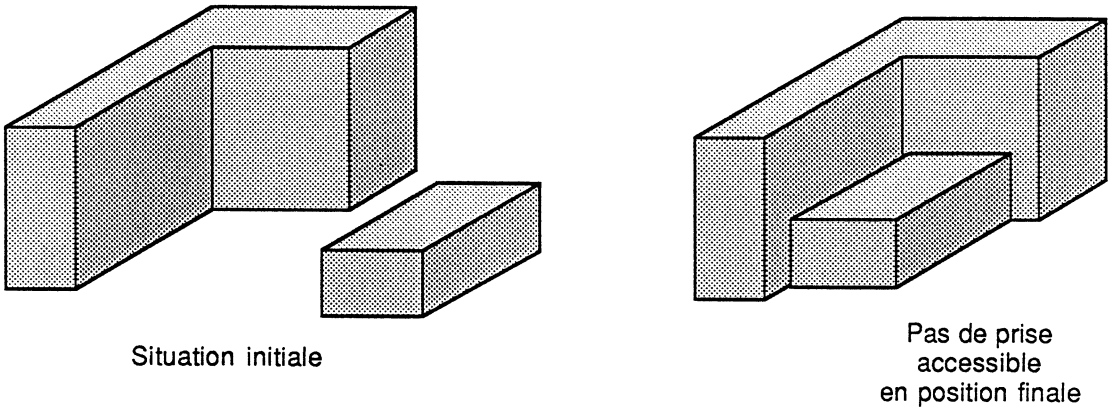


Figure 6.3: Environnement de lacher d'un objet trop contraint

transposée de A. L'intérêt de cette approche est double; d'une part elle est relativement rapide puisque qu'elle ne nécessite que des produits de matrices booléennes et d'autre part elle permet simplement de détecter une impossibilité de changement de prise (nous renvoyons le lecteur intéressé à [Dauchez 83]). Il faut cependant noter que la construction automatique de cette matrice A nous parait difficile.

Il est clair que ce type de méthode ne peut s'appliquer dans notre cas puisque les prises potentielles ne sont que partiellement déterminées à l'issue de la première phase. A l'heure actuelle, nous ne voyons pas clairement comment traiter ce type de problème.

Une situation voisine peut survenir lorsque les positions initiale ou finale de l'objet sont trop contraintes (cf. figure 6.3) et qu'aucune prise n'est accessible. La résolution de ce problème ne peut s'envisager que par le développement de gestes de saisie plus complexes incluant une phase de mouvements compliants. Il s'agira de dégager l'objet avant de le saisir ou bien de le mettre en position finale après l'avoir laché.

Interdépendance avec les mouvements fins

Considérons la situation décrite par la figure 6.4. Le contexte de manipulation va nous conduire à saisir le cylindre selon ses faces planes. Cette prise respecte les contraintes d'accessibilité imposées par les environnements de saisie et de lacher de l'objet. Cependant, elle ne permet pas d'effectuer l'assemblage.

Ce problème est dû à l'interdépendance entre saisie et mouvements fins. La prise doit être choisie en fonction de la trajectoire de montage. C'est un des problèmes de l'intégration des systèmes de niveau objet (cf. chapitre 7).

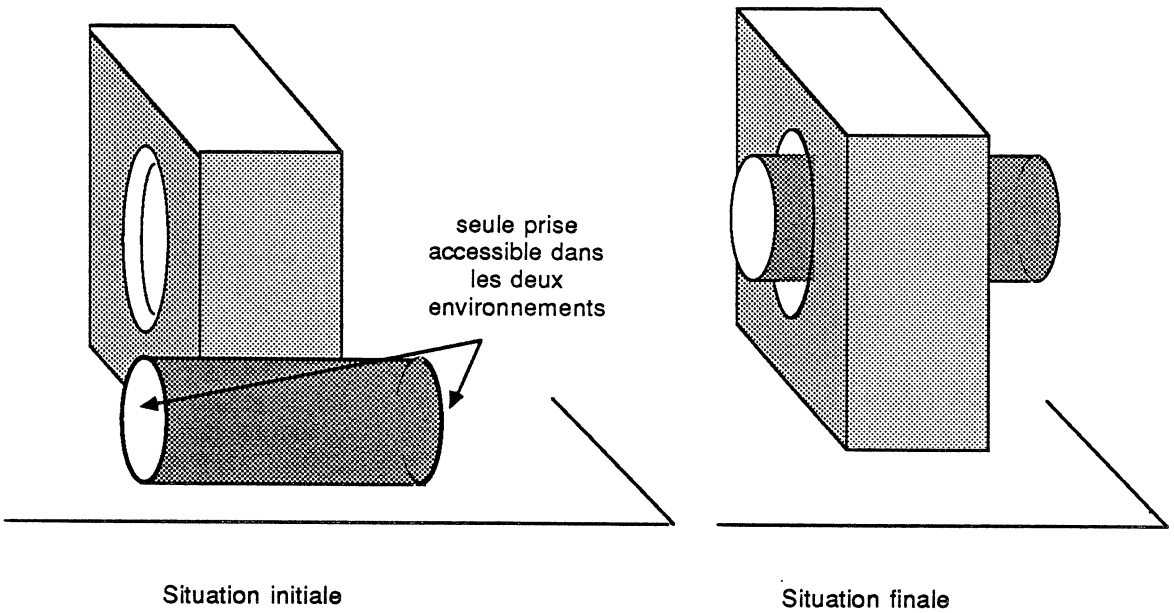


Figure 6.4: Prise incompatible avec le montage à réaliser

Compatibilité de niveau tâche

La compatibilité de niveau tâche nous semble beaucoup plus fondamentale que les deux problèmes précédents. Il s'agit, en effet, de choisir la prise et ses paramètres de manière à faciliter l'exécution de la tâche planifiée.

L'aspect plutôt qualitatif présenté dans [Lyons 85] est propre à une approche anthropomorphique de la saisie; par contre, l'approche présentée dans [Hanafusa 77b] nous semble extrêmement intéressante pour guider le processus de choix des prises. Des conseils extraits d'une description quantitative de la tâche à réaliser (direction et intensité des forces à appliquer, utilisation éventuelle de mouvements à compliance...) devrait nous permettre de choisir des prises satisfaisant au mieux les contraintes imposées par la tâche (cf. figure 6.5).

6.2.2 Saisie de structures

Nous avons, tout au long de cette étude, considéré la saisie d'objets élémentaires rigides. Il peut arriver, dans le contexte, de l'assemblage que l'on ait à manipuler des sous-assemblages non fixés (les objets sont uniquement posés) ou des sous-assemblages articulés (cf. figure 6.6).

Dans le cas du sous-assemblage non fixé, il est nécessaire, dans un premier temps, de déterminer l'objet dont la saisie va permettre de déplacer l'ensemble des objets (cf. §3.1.2). Cette information doit être extraite du modèle relationnel. Le choix de la prise et de ses paramètres pourra alors être effectué grâce à notre méthode. Si un tel objet n'existe pas, cela signifie que la saisie ne peut être effectuée sans le recours

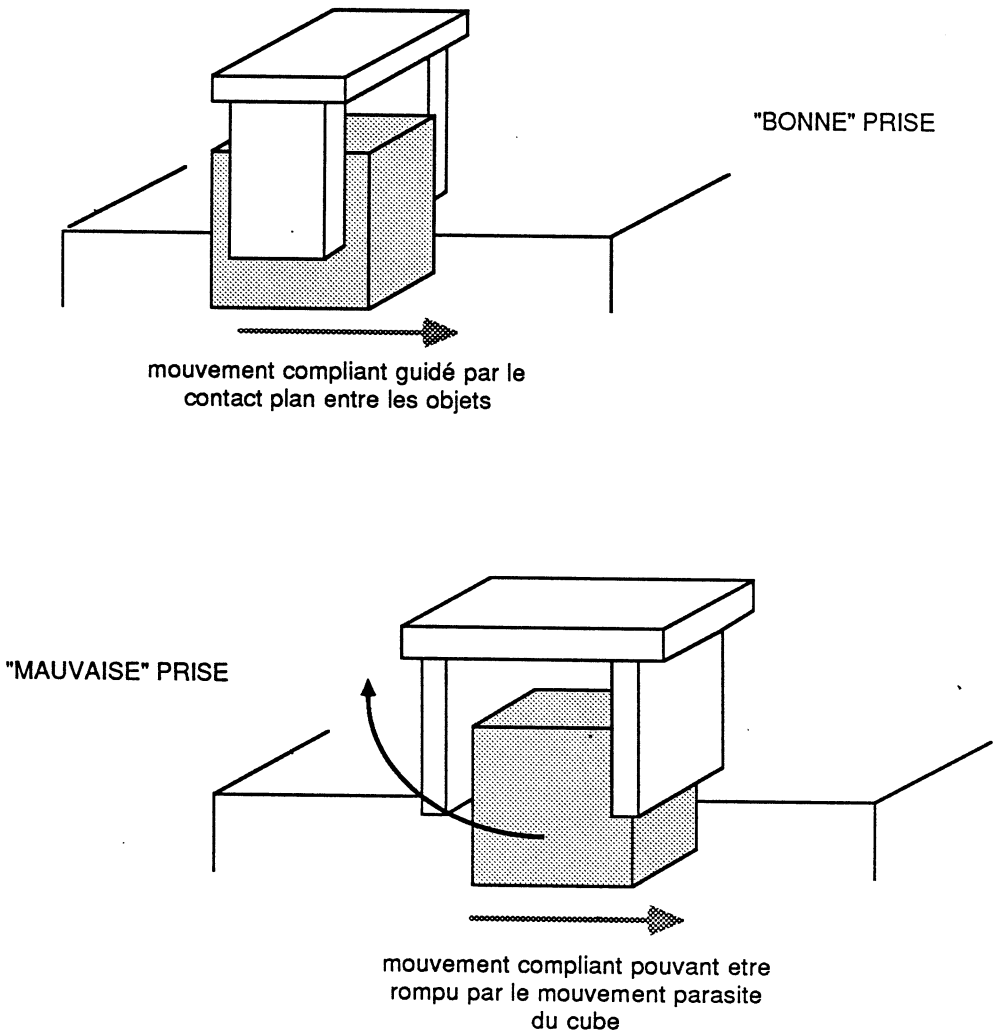


Figure 6.5: Choix de prise guidé par la tâche de manipulation

à un autre manipulateur ou à un artifice (cf. figure 6.7).

La saisie de structures articulées pourrait suivre un principe analogue: choix de l'objet à saisir puis choix de la prise sur cet objet.

6.2.3 Application de l'approche à la "machine intelligente"

L'utilisation exclusive d'un support de modélisation tel que SMGR est bien adaptée à la programmation automatique. Dans le contexte plus large de la machine intelligente opérant dans un univers non structuré, il convient de discuter ce choix dont la limitation principale réside dans l'impossibilité de prendre en compte des

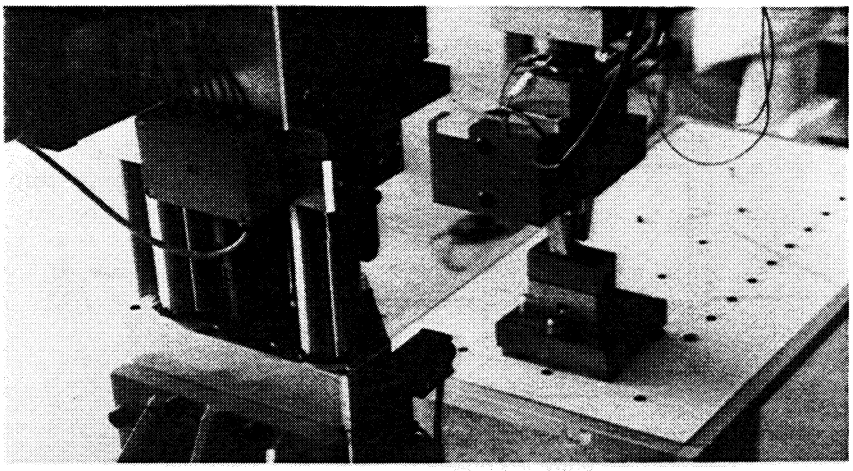


Figure 6.6: Manipulation de structures non rigides

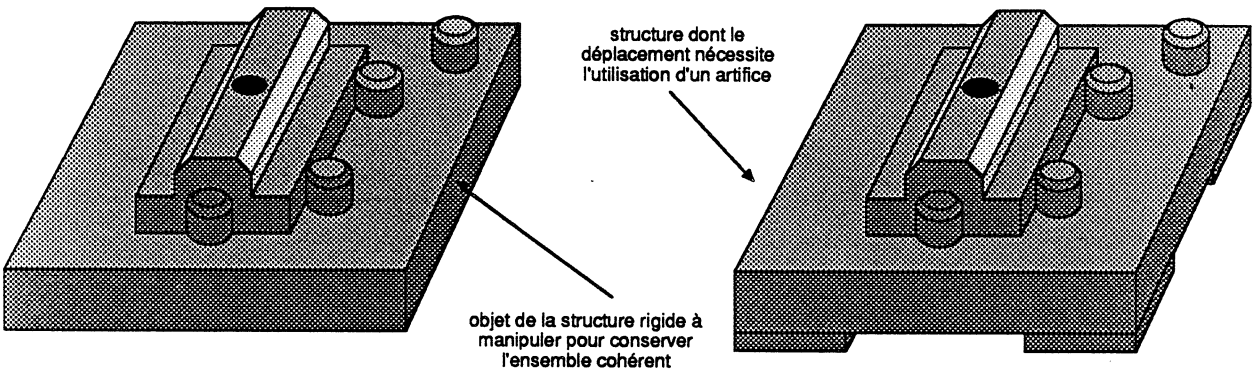


Figure 6.7: Choix de prise pour une structure non rigide

incidents (obstacles inattendus par exemple). D'autre part, l'étude d'accessibilité globale nécessite une modélisation simple de l'univers en terme d'espace occupé, dont la reconstruction est particulièrement lourde à partir d'un tel modèle.

Nous avons vu au chapitre 4 que, dans les travaux actuels, la perception était presque exclusivement réservée à un sous-problème de la saisie: la saisie en vrac. Ceci est dû, à notre avis, à la difficulté de construire des modèles de l'univers basés sur la notion d'objets à partir de données perceptives. La notion d'objet étant essentielle pour la réalisation d'une méthode de recherche de prises générale, en résulte trois types restrictifs d'utilisation de la vision:

- recherche d'indices visuels spécifiques dans l'image pour la saisie d'objets en vrac
- étude de l'accessibilité globale par construction de l'espace occupé (capteurs 3D) dans le contexte de la saisie en vrac
- reconnaissance/localisation d'objets modélisés par ailleurs

A l'heure actuelle, chacune des deux approches (modèles ou perception) présente des avantages qu'il ne convient pas de séparer par des applications distinctes mais de combiner dans une approche mixte à la fois plus générale et plus efficace. Nous proposons l'approche suivante.

La recherche de prise utilisera un modèle tel que SMGR alors que l'étude d'accessibilité globale puisera ses informations dans l'univers de manipulation grâce à des moyens de perception. Ceci n'exclut pas l'utilisation de la seule perception lorsque celle-ci permettra de construire des modèles d'objets (dans un avenir plus ou moins proche).

Le capteur de vision active tri-dimensionnelle développé par [Borianne 84] parait tout à fait adapté à notre méthode de calcul d'accessibilité globale. Il est basé sur l'émission d'un plan laser produit par un rayon laser équipé d'une lentille cylindrique. Ce plan coupe la scène selon un trait lumineux perçu par une caméra munie d'un filtre interférentiel (cf. figure 6.8).

La raison essentielle de ce choix réside dans la similitude la raison essentielle de ce choix réside dans la similitude existant entre les informations fournies par le capteur et les données manipulées tout au long de la seconde phase de notre méthode: notion d' "univers en tranches". Un autre avantage, plus pratique, du capteur est de pouvoir être monté sur un bras manipulateur et de constituer à ce titre une partie de l'outil de préhension.

Son utilisation entraine quelques modifications de l'algorithme de traitement de la seconde phase, l'idée de base étant de projeter le plan laser parallèlement au plan de préhension, le laser étant orienté dans la direction de mouvement de l'outil.

Les différentes courbes obtenues pour différentes positions du plan laser représenteront les obstacles susceptibles de s'opposer à l'outil de préhension dans son mouvement. Cette représentation partielle des obstacles liée à la direction de mouvement impliquera une nouvelle acquisition d'informations sensorielles en cas d' échec pendant le traitement d'un prise.

Si dans le cas de l' "approche modèle", le choix entre raisonnement sur les obstacles et raisonnement sur les solutions était délicat (cf. §5.4.2), il est dans le contexte présent tout à fait évident. En effet, chacune des courbes obtenues est décrite dans l'espace bidimensionnel du plan lumineux et l'union des obstacles représentés par

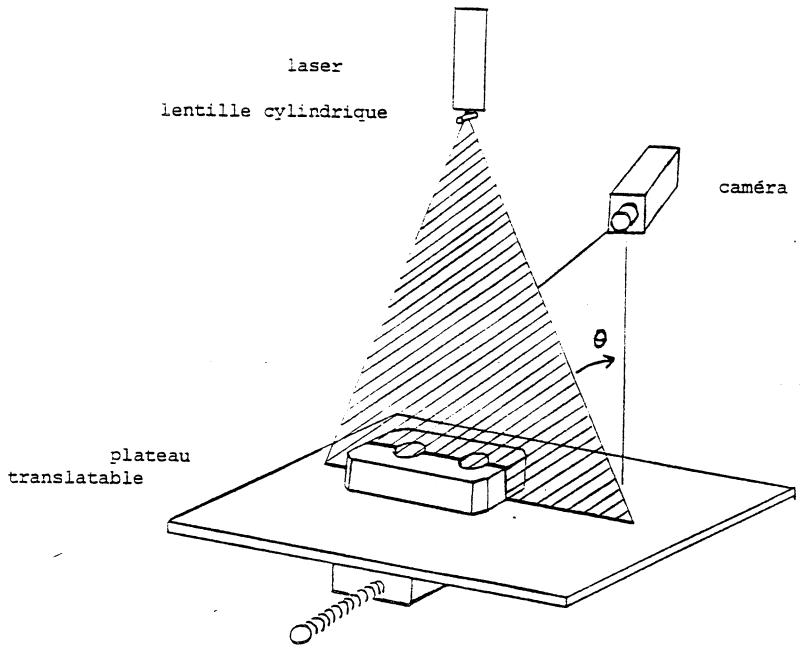


Figure 6.8: Capteur de vision active tri-dimensionnelle [Borianne 84]

ces courbes, avant segmentation, est une opération élémentaire. Chacune d'elles représente, en effet, la frontière "au-dessous" de laquelle l'outil ne doit pas aller. En faire l'union consiste à calculer une ligne de crête maximale (cf. figure 6.9). Cette technique est couramment utilisée pour l'élimination de lignes cachées dans le cas de dessins au trait de courbes paramétriques (Z = F[X,Y]) (cf. figure 6.10). La ligne de crête est d'autant plus aisément construite que la courbe est représentée sous la forme d'une matrice de points [Williamson 72].

L'algorithme résultant est présenté en Annexe C. Les performances d'un tel algorithme seront considérablement améliorées par la simplication algorithmique des coupes et des opérations ensemblistes.

Dans le cas de l'outil P, les performances de l'algorithme peuvent encore être améliorées puisque le grossissement des obstacles peut n'être effectué qu'une seule fois (chaque tranche produisant un grossissement identique). Il sera fait, dans ce cas, juste avant transformation des contraintes implicites de position pour l'outil en contraintes explicites.

Les limitations de la méthode sont alors liées à celles du capteur (zônes d'ombre, cavités...). Comme dans le cas de la saisie humaine, cette prise de décision sur la base de données visuelles partielles pourrait déboucher sur une meilleure intégration

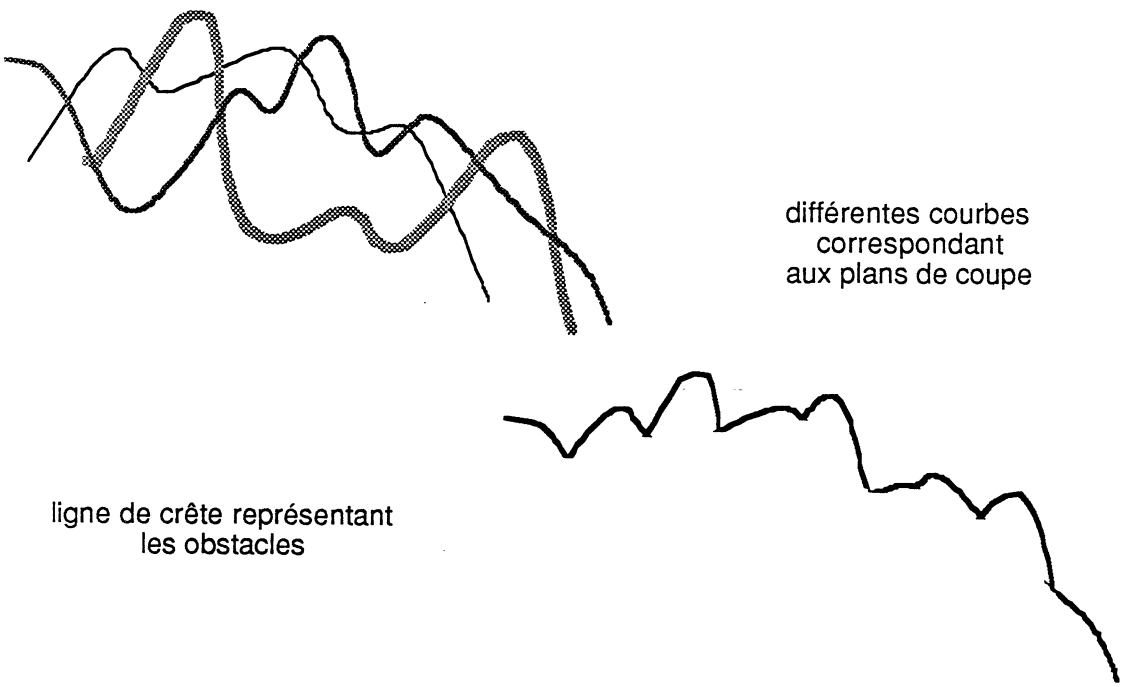


Figure 6.9: Recherche des obstacles par mise à jour d'une ligne de crête

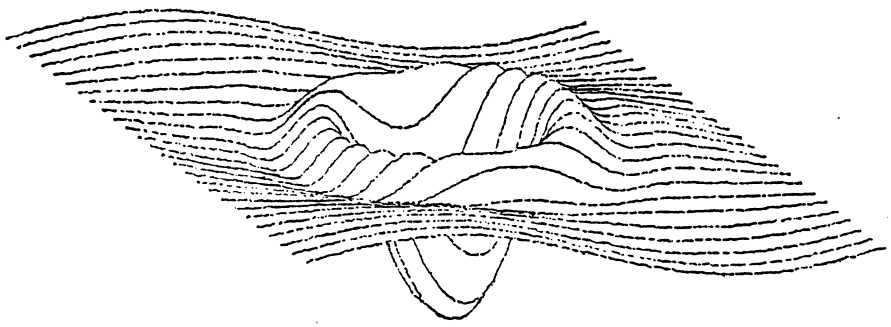


Figure 6.10: Visualisation de courbes paramèriques par l'algorithme de [Williamson 72]

du toucher à la saisie. Le domaine est complètement ouvert.

6.3 La préhension humaine

Un de nos soucis au cours de ce travail a été d'essayer de comprendre ce qu'est la saisie au niveau cognitif à partir d'observations du processus de saisie humaine. Il nous semble important de "légitimiser" les approches adoptées en Robotique sur la base de l'observation de la machine humaine. Si cette philosophie parait excessivement anthromorphique, elle repose cependant sur une longue évolution de l'homme tendant vers des solutions optimales [Napier 62].

Ce paragraphe n'a pas pour ambition d'analyser exhaustivement les recherches liées au mouvement dans le domaine de la neuro-physiologie et des sciences cognitives mais de présenter quelques idées clés qui nous semblent intéressantes par rapport à notre problème.

Une grande part des recherches en la matière, ou tout du moins dans l'ensemble des lectures que nous avons eu sur le sujet, s'attache plus à analyser l' exécution des programmes moteurs qu'à en chercher les sources. De nombreuses publications portent sur la capture d'un objet et font une place importante à l'analyse de l'interaction entre le geste et la vision.

Nous avons cherché dans ces travaux des élements de confirmation de certaines intuitions pouvant guider les développements ultérieurs de nos recherches en saisie. La première d'entre elles concerne ce qui nous parait être le caractère hiérarchisé de la saisie humaine; celui-ci est, nous semble-t-il, lié au rôle fondamental de l'opposition pauci-digitale confèrant à la main, à un certain niveau, un comportement analogue à celui d'une pince à deux mors. Cet état de fait nous a porté à penser que la saisie peut être décrite par un processus à trois niveaux:

- prise de décision sur la base d'indices de haut niveau (ces indices restant à préciser),
- commande de la pince pauci-digitale sur la base d'indices de plus bas niveau (ce qui correspondrait à notre travail actuel),
- et enfin ajustement tactilo-kinesthésique de la main sur l'objet saisi.

Cette approche hiérarchisée pourrait conduire au développement d'un système à trois niveaux intégrant nos travaux actuels et laissant une large part à l'utilisation de la perception.

Cette approche nous parait étroitement liée au rôle de la vision et nécessite de mieux cerner son influence sur la saisie tant dans sa phase de planification que dans celle de son exécution. Pour ce dernier point, il nous semblait, dans une première approche, que considérer la saisie comme un processus en boucle ouverte collait mal avec la réalité de la saisie humaine, la perception visuelle semblant jouer un rôle considérable dans le guidage de tout mouvement humain.

Différentes études dans le domaine de la physiologie nous ont apporté des réponses partielles relativement à nos deux préoccupations.
Notre intuition du guidage du mouvement par la vision n'était que partiellement exacte. En effet, de récentes études sur la motricité montrent clairement que la correction, sur la base d'indices visuels, d'un mouvement tel que le transport du bras vers l'objet à saisir est impossible. L'expérimentation a montré [Jeannerod 81] qu'il faut, en effet, plusieurs dixièmes de seconde pour qu'une information d'origine visuelle ait une influence sur un mouvement; les vitesses couramment utilisées pour effectuer les gestes de la vie courante (50cm/s pour le transport de la main) font donc rejeter l'hypothèse d'un fonctionnement en boucle fermée sur la vision. On qualifie ces mouvements de balistiques (à la manière du lancer d'un objet en un point de l'espace) [Jeannerod 81], [Paillard 78].

La vision intervient cependant à différents niveaux (cf. figure 6.11); son rôle prépondérant est de fournir les élements nécessaires à la localisation de l'objet et à l'évaluation de certains des indices de haut niveau précédemment cités (poids, taille, consistance, forme de l'objet par exemple); cette dernière information permet la mise en position anticipée (pendant la projection de la main) du poignet et des doigts. La vision est donc principalement un outil de base de la planification du mouvement plus qu'une aide à son exécution. Nous envisageons de ce fait d'étudier en profondeur le problème d'extraction, par la vision, d'indices morphologiques de "haut-niveau" dans une commande hiérarchisée de la saisie.

La vision est également utilisée dans la phase de réajustement terminal de la main mais de façon moindre que le toucher. Elle est enfin utile lorsque le geste est trop long (en durée) pour être complètement balistique. Dans ce cas, elle permettant un "rafraichissement" de l'image de l'environnement dans le cerveau [Jeannerod 81].
Cette approche balistique implique d'ailleurs des mécanismes de calibration très évolués [Berthoz 78] puisque la proprioception a un rôle fondamental dans le mouvement. Notons que ces problèmes de calibration sont tout à fait similaires à ceux que l'on rencontre dans la mise en œuvre de systèmes robotiques évolués.

L'élaboration des programmes moteurs i.e. la planification du mouvement n'est pas aussi clairement étudiée. Il est vrai qu'il est extrêmement difficile d'acquérir des certitudes en la matière, tout au plus des éléments de réponses reposant sur une base expérimentale.
Dans ce cadre de recherche, certains chercheurs vont puiser leur inspiration dans les modèles hiérarchisés (tâche -objet -outil -articulation) développés dans le domaine de la Robotique [Beaubaton 85a], [Beaubaton 85b] (certains éléments d'observation tels que le caractère rectiligne des trajectoires engendrées pour le transport de la main tendent à confirmer partiellement une telle approche; l'optimisation du mouvement est en effet liée à l'outil et pas à l'articulation comme on pourrait le penser). Il nous semblerait interessant de développer des collaborations dans ce domaine.

D'une manière plus générale, la confrontation des approches développées en In-

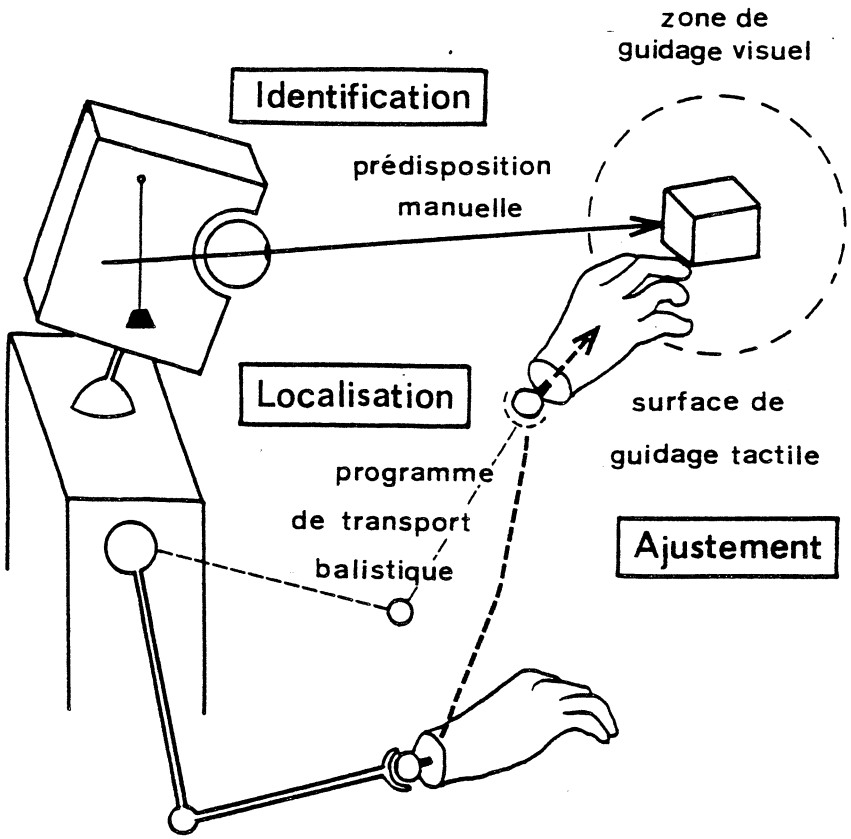


Figure 6.11: Coopération entre saisie et vision [Paillard 75]

telligence Artificielle et en Sciences Cognitives qui commencent à se faire, nous parait tout à fait enrichissante et novatrice pour les deux parties.

6.4 Raisonnement fonctionnel: vers une modèlisation 3d des objets

Notre méthode de recherche de prises pour un objet donné repose sur une analyse de sa capacité à produire des contacts d'un type donné basée sur un examen de sa morphologie. Le lien étroit existant entre forme et fonction est possible grâce à un niveau commun de raisonnement: les primitives fonctionnelles de stabilisation de l'objet (contacts potentiels) correspondent en effet à un examen local de propriétés morphologiques des objets manipulés. Ces propriétés sont directement accessibles dans SMGR ou sont aisément déductibles de notre modèle.

Un niveau de raisonnement fonctionnel supérieur ne peut plus se baser sur une modélisation du même type; un tel niveau de raisonnement peut être par exemple

celui de Lyons (cf. §4.2.4) reposant sur un examen morphologique global de l'objet; l'hypothèse qui sous-tend cette approche peut être schématisée comme suit:

$$\left.\begin{array}{c} structure\{objet\} \\ + \\ fonction\{objetsaisi\} \end{array}\right\} structure\{main\} \tag{6.2}$$

Elle nécessite l'évalution d'indices morphologiques tels que: un objet est-il plutôt plat, plutôt rond...? Aucun des modèles existant n'a été conçu pour fournir ce type de renseignements; à l'opposé, il est remarquable de constater avec quelle vigueur tous ces systèmes (ou plutôt leur concepteurs, nous y compris) fuient une réelle représentation tri-dimensionnelle. Nous qualifierons toutes les représentations manipulées par ces systèmes, d'informations 2,5D (sans connotation avec [Marr 79]). En effet, ne décrit-on pas un objet par ses surfaces ou par du 3D "disloqué" (CSG ou cellularisation de l'espace)? Le dimensionnement utilisé n'est-il pas lui-même intrinsèquement uni-dimensionnel? Quel intérêt, à ce niveau de raisonnement, de savoir qu'un parallèlépipède a des dimensions de $L = 1.1$ $l = 0.9$ $h = 0.95$ (resp. 30.0 5.0 0.1)? N'a-t-on pas besoin de savoir qu'il a tout les spécificités fonctionnelles d'un cube (resp. qu'il est long et plat)?
Il nous faut aller vers des modèles réellement tri-dimensionnels dans lequel les proportions et propriétés caractéristiques d'un volume soient représentées; c'est là le prix du développement d'un raisonnement fonctionnel pour la saisie.

Citons ici la représentation utilisée par le système ACRONYM qui est à notre sens un pas vers le "vrai" 3D; les cônes généralisés ont en effet l'immense avantage de représenter aisément certaines propriétés volumiques morphologiques (symètrie par exemple); cette représentation est malheureusement limitée à une classe d'objets restreinte. L'utilisation de modèles génériques a également l'avantage de décrire des familles d'objets sans considération de dimensionnement; un objet reste cependant caractérisé physiquement et non pas fonctionnellement interdisant par cela de considèrer certains objets présentant des variations morphologiques comme étant de la même famille.

Deux approches sont utilisées en ce qui concerne la modélisation dans un cadre fonctionnel. La première d'entre elles, utilisé par Brady dans son "Mechanic's mate" [Brady 84], consiste à spécialiser le modèle utilisé, c'est à dire, à ne conserver que certains indices morphologiques liés à un ensemble de fonctionnalités donné. C'est l'approche classiquement utilisée dans les systèmes de CFAO où les "features" sont des indices importants des objets en ce qui concerne leur fabrication mécanique.
La seconde approche consiste à avoir un modèle aussi général que possible et à y effectuer une association entre certaines caractéristiques physiques et des spécificités fonctionnelles: c'est l'approche retenue par Winston et al. [Winston 83] dans leur système ANALOGY utilisant les outils de modélisation de ACRONYM. Elle consiste à synthétiser une description physique des objets pertinente par rapport à leur spécification fonctionnelle à partir de la description en cônes généralisés. ANAL-

OGY permet d'apprendre ce modèle synthétique sur la base d'observations puis de reconnaitre visuellement un objet grâce à cette nouvelle représentation.

Il nous semble que dans le domaine bien défini qui est le nôtre, la première approche pourrait être utilisée avec succès. Le problème est alors de rechercher les indices morphologiques (ou autres) pertinents par rapport au domaine traité (saisie ou plus généralement assemblage). C'est évidemment une direction de recherche qu'il conviendrait de privilégier dans les années à venir, le but étant d'aller vers un raisonnement géométrique se basant sur une analyse fonctionnelle plus symbolique que celle que nous utilisons à l'heure actuelle: le raisonnement morphologique.

Chapitre 7

Vers la non programmation des robots

7.1 Interêt de la programmation automatique

L'intérêt de la programmation automatique (ou plus immédiatement de la programmation de niveau objet) est multiple: fondamental, méthodologique et pratique.

- Les différents sous-problèmes de la programmation automatique permettent d'aborder de manière pragmatique des problèmes fondamentaux de l'Intelligence Artificielle (génération de plan d'actions, raisonnement géométrique, raisonnement fonctionnel, traitement de l'incertain, interaction perception - action...) en leur founissant un cadre expérimental réaliste.

- Sur le plan méthodologique, les recherches dans le cadre de la programmation automatique ont déjà eu un apport important en ce qui concerne la programmation de niveau effecteur (pour la prise en compte des incertitudes, essentiellement - [Dufay 83]). D'autre part, le développement de mécanismes de preuve de programmes en Robotique se matérialise déjà par une réflexion de fond sur les langages de manipulation. L'influence de telles recherches sur l'évolution des langages de programmation en Informatique nous laisse entrevoir l'apparition de nouveaux langages de programmation de robots et de nouvelles méthodologies de programmation, dans les années à venir.

- L'aboutissement de cette recherche se traduira, au niveau industriel, par une croissance de la productivité grâce à une meilleure utilisation des moyens matériels et humains: les phases d'écriture et de mise au point des programmes seront facilitées et écourtées; les programmes produits seront plus fiables; enfin, l'environnement matériel (essentiellement les capteurs) sera utilisé à bon escient. Si cet objectif ne peut être considéré dans un court terme, il est possible, dès à présent, d'envisager l'utilisation de certaines des méthodes développées comme outils d'aide à la programmation en-ligne (par exemple, la génération automatique de trajectoires pour un robot de peinture ou de soudure) ou comme outils de simulation (pour l'organisation spatiale lors la

mise en place de cellules robotisées ou dans le but d'analyser l'effet d'un choix de conception d'une pièce sur sa fabrication, par exemple).

7.2 Historique rapide

Dès l'apparition des premiers langages de programmation des robots manipulateurs, certains laboratoires se sont penchés sur le problème de l'automatisation de cette programmation. Les premières spécifications de systèmes de programmation de niveau objet sont apparues aux Etats-Unis vers les années 75: AL (Stanford) [Finkel 74], LAMA (MIT) [Lozano 76] et AUTOPASS (IBM) [Liebermann 77]. Elles ne donnèrent cependant lieu qu'à des implantations très partielles. En effet, face à la difficulté de concevoir d'amblée de tels systèmes, les recherches s'orientèrent naturellement vers la résolution de sous-problèmes de la programmation de niveau objet: la génération de trajectoires sans collisions et la prise en compte des incertitudes.

Forts de cette expérience, aujourd'hui certains laboratoires étudient à nouveau l'intégration d'un système de niveau objet. Dans le secteur de la recherche, citons le laboratoire d'Intelligence Artificielle du MIT (système TWAIN) et le LIFIA (système SHARP) dont l'objectif commun est la conception d'un système prototype dans un court terme. Certaines sociétés du secteur industriel poursuivent également des travaux dans cette direction sans qu'il soit généralement possible d'en connaitre l'état d'avancement (BOEING, IBM, ITMI, General Motors, Westinghouse, etc...).

On trouvera dans le §7.3 une brève description des recherches qui nous semblent les plus significatives des dix dernières années dans le cadre des sous-problèmes de la programmation de niveau objet (hormis la saisie automatique déjà étudiée). L'architecture du système TWAIN, représentative de l'école américaine, est étudiée plus en détail en §7.4.1. Les paragraphes suivants sont consacrés à la présentation des constituants du système SHARP, en cours d'implantation au LIFIA. Ils sont le fruit d'une réflexion collective sur l'automatisation de la programmation des robots.

7.3 La recherche sur les sous-problèmes de la programmation de niveau objet

De nombreux travaux ont porté sur les deux sous-problèmes de la programmation de niveau objet: génération de trajectoires sûres et prise en compte des incertitudes (cf. §2.1).

La recherche de trajectoires de transfert est généralement fondée sur une représentation explicite de l'espace libre. Si plusieurs approches ont été étudiées (construction des "freeways" de [Brooks 83] par exemple), c'est l'approche par espace des configurations qui est généralement utilisée pour planifier les mouvements d'un bras manipulateur. Les méthodes développées pour cette approche [Faverjon 84],

[Lozano 81], [Gouzenes 84] sont généralement liées à un type de manipulateur. Elles diffèrent essentiellement par les outils de représentation choisis pour la construction de l'espace des configurations (secteurs angulaires [Chien 84], octrees [Faverjon 84] ...).

Le problème de l'assemblage de deux pièces nécessite dans un premier temps d'engendrer une trajectoire nominale permettant de réaliser le montage, puis de transformer cette trajectoire pour prendre en compte les problèmes d'incertitudes (génération de mouvements fins).

Peu de chercheurs se sont intéressés à la génération des trajectoires nominales de montage; [Valade 85] propose une méthode fondée sur une analyse des concavités des objets montés, permettant d'infèrer une trajectoire de démontage par des mouvements en translation. Si certaines conditions de réversibilité sont vérifiées, la trajectoire de montage est assez aisément déductible de celle de démontage.

[Lozano 83] se sont intéressés à la génération de trajectoires réelles. La méthode développée permet d'engendrer une suite de mouvements à compliance à partir de l'étude de pré-images de la situation finale. L'ensemble des positions d'où peut provenir un objet se déplaçant à une certaine vitesse permettant d'atteindre une position donnée, constitue la pré-image de cette position (cf. figure 7.1). La construction de ces pré-images, utilisant la notion d'espace des configurations, se fait en tenant compte des problèmes de friction et en considèrant les incertitudes. Si cette approche est simple dans son principe, il n'en est pas de même en ce qui concerne sa mise en œuvre (cf. [Erdmann 85] où est analysée la calculabilité des pré-images).

Parallèlement à ces développements, d'autres recherches ont été entreprises sur le thème du raisonnement sur les incertitudes. [Taylor 76] a développé une méthode de propagation numérique des incertitudes permettant de déterminer les valeurs des paramètres de squelettes de programmes (un squelette est une macro correspondant à un type de tâche donné).

[Brooks 82b] a amélioré cette approche par l'utilisation de méthodes de calcul symbolique. Des contraintes d'applicabilité et de propagation sont associées à chaque squelette de programme sous la forme d'équations sur des paramètres simples (nombres réels). La manipulation de ces équations peut à la fois guider le choix des squelettes et permettre de vérifier la validité des choix (jeux de montage respectés). On peut regretter que cette approche ne manipule, non pas une position comme une entité, mais sous la forme de six paramètres. L'application de l'approche à l'univers tri-dimensionnel se traduit alors inévitablement par une inflation du nombre des équations. Elle semble se révèler mieux adaptée au contexte de la navigation des robots mobiles [Brooks 84].

Le Laboratoire d'Intelligence Artificielle de Edinburgh s'est davantage focalisé sur les problèmes de dialogue entre le système et l'utilisateur. Le travail effectué

Fig. 4. Pre-image of the goal for different $\mathbf{v}_i$.

Fig. 5. Backward chaining of pre-images.

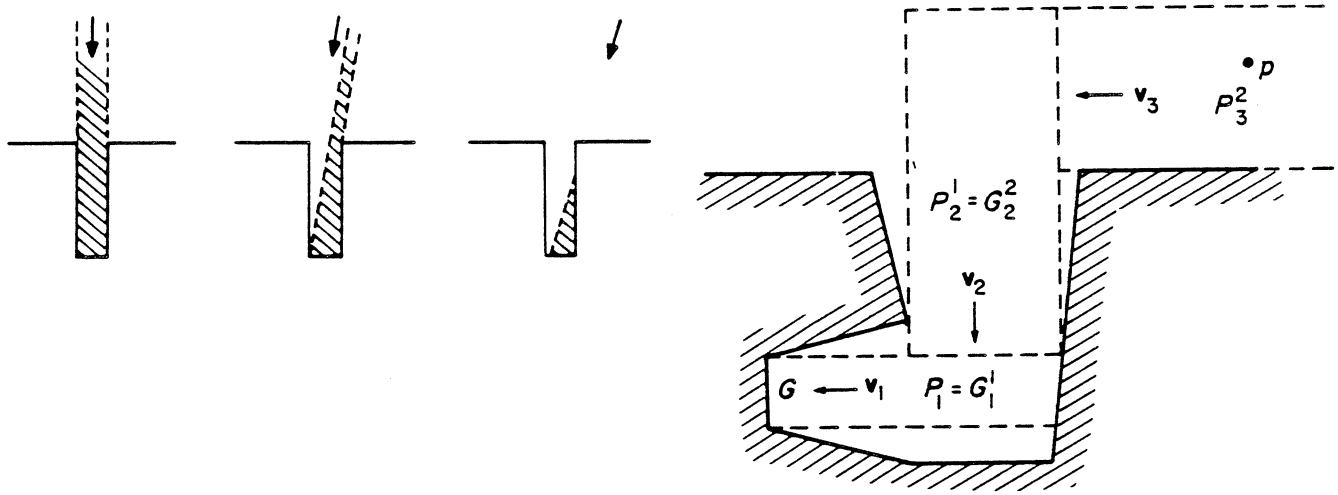


Figure 7.1: Notion de pré-image [Lozano-Pérez 83]

dans RAPT a porté essentiellement sur la définition et l'implantation d'un langage situé entre le niveau effecteur et le niveau objet [Popplestone 78]. Il repose sur une méthode symbolique permettant l'inférence d'une position, en terme cartésien, à partir de sa description en terme de relations entre entités géométriques.

7.4 Intégration d'un système de niveau objet

La difficulté principale de l'implantation d'un système de programmation de niveau objet réside dans le fort degré d'interdépendance entre les différents sous-problèmes. Cette interdépendance se manifeste au travers de contraintes d'accessibilité et de contraintes d'incertitudes.

En génération de plans d'actions, les différentes solutions adoptées pour résoudre les problèmes d'interdépendance sont de nature assez différentes [Cohen]. La résolution peut se faire au niveau du contrôle par utilisation de méthodes de backtracking ou de régression (par exemple RSTRIPS); on peut également chercher à décomposer le problème ou sous-contraindre les différentes solutions: ce sont les approches par hiérachisation et non linéarité (ABSTRIPS, NOAH, DCOMP par exemple). Enfin, on peut utiliser des techniques de "méta-planification" pour prendre en compte ces interdépendances au moment de la planification ou d'un éventuel backtracking (par exemple GARI [Descotte 81]).

La prise en compte des contraintes d'incertitude en programmation automatique est essentiellement quantitative; elle peut être faite par des méthodes proches de la régression (c'est l'approche de Brooks). Les contraintes d'accessibilité sont de

nature plus qualitative et semblent nécessiter des méthodes de "méta-planification". Malheureusement, les systèmes de niveau objet en cours de réalisation, ne savent quasiment pas traiter les interdépendances liées à l'accessibilité autrement que par un backtracking systèmatique (i.e. non guidé par une heuristique).

7.4.1 Le système TWAIN

Le système TWAIN (alias ATLAS) [Lozano 84,Lozano 85a], développé au MIT, doit permettre d'intégrer les travaux de Brooks, Lozano-Perez, Mason et Taylor (LAMA décrit dans [Lozano 76], plus brièvement dans [Lozano 77] était une ébauche de TWAIN). Il a pour but de synthétiser un programme d'assemblage et de spécifier l'allocation spatiale de l'espace de travail.

TWAIN prend en compte les interdépendances de deux manières différentes. Il utilise une planification hiérarchique à deux niveaux permettant de prendre aussi peu de décisions que possible avant d'avoir une première évaluation des interdépendances sous la forme de contraintes d'incertitude. Le niveau supérieur permet de transformer la spécification de niveau objet en squelettes de programmes (saisie, lacher, mouvement de transfert, insertion, mouvements fins). Exceptés les squelettes de mouvements fins qui sont calculés, les autres sont recherchés dans une librairie. Un graphe de dépendance entre les différents squelettes choisis est alors construit grâce à un raisonnement sur les positions des objets.

Ces contraintes sont utilisées au niveau inférieur pour l'affinage des squelettes. Les interdépendances y sont traitées grâce à la régression et au backtracking. On alterne à ce niveau les phases de planification et de propagation de contraintes. Grossièrement TWAIN opère de la façon suivante au niveau inférieur:

1. planification grossière de l'espace de travail (définition de zônes libres)
2. propagation de contraintes d'incertitudes
3. planification des opérations de montage
4. propagation de contraintes d'incertitudes
5. planification des opérations de saisie
6. propagation de contraintes d'incertitudes
7. planification détaillée de l'espace de travail
8. planification des opérations de transfert

Chacune de ces phases peut conduire à une meilleure spécification de la solution ou à sa remise en cause (retour à une phase de planification ou insertion de macros de perception). Seul, un retour arrière à partir de la phase (8) sera guidé par une analyse de l'échec (retour aux phases (1), (5) ou (7) selon le cas).

Deux artifices sont utilisés pour réduire les effets des interdépendances. D'une part, on engendre des solutions moins sensibles aux incertitudes (mouvements fins à compliance, prises stables). D'autre part, on s'occupe de la planification de l'espace de travail de façon à sous-contraindre les problèmes d'accessibilité et à atténuer ainsi les effets des interdépendances. Par exemple, la planification des opérations de saisie se fait sans considération de critères d'accessibilité globale et impose, de ce fait, des contraintes sur la planification détaillée de l'espace de travail (phase 7). Cette approche nous semble parfaitement légitime dans le cadre de la programmation automatique où l'environnement de manipulation doit être déterminé.

On peut regretter que TWAIN ne s'appuie sur aucun support de modélisation de l'univers relationnel et que la manipulation des incertitudes soit affaiblie par la représentation qui en est faite. On aimerait également connaitre le degré d'explicitation des squelettes de programme pour justifier l'approche hiérarchique adoptée.

7.4.2 Le système SHARP

Architecture du système SHARP

SHARP a pour objet d'intégrer les recherches menées au LIFIA dans le contexte de la programmation de niveau objet. Comme TWAIN, il n'existe actuellement qu'au travers d'une spécification encore grossière et de certains de ces constituants. Le but visé est, à partir d'une tâche spécifiée en terme de relations de montage et de jeux de montage, de produire un programme de manipulation réalisant cette tâche.

Le langage de spécification de la tâche pourra être LM-Géo [Mazer 83]. Comme RAPT, LM-Géo permet de transformer une spécification symbolique d'une situation géométrique (relations géométriques) en sa spécification numérique (positions dans l'espace cartésien). Contrairement à RAPT, LM-Géo utilise pour cela, une méthode de résolution numérique [Mazer 82]. Le programme de manipulation synthétisé pourra être écrit en LM. SHARP utilisera SMGR (ou un descendant) comme support de modélisation.

L'approche utilisée, au niveau de la structure de contrôle, est assez différente de celle de TWAIN. Nous pensons, en effet, que l'écriture d'un programme permettant de réaliser un assemblage et son adéquation à un univers réel de manipulation (imparfaitement modélisable) sont deux problémes conceptuellement très différents, qu'il convient donc de traiter séparemment, par des approches différentes.

Il s'agit donc de synthétiser un programme de manipulation avant d'en faire la preuve (au sens algorithmique classique du terme) et éventuellement la transformation. Les méthodes employées se plaçant toujours dans le cas le pire, on pourra ensuite optimiser le programme obtenu grâce à des méthodes d'apprentissage (cf. figure 7.2).

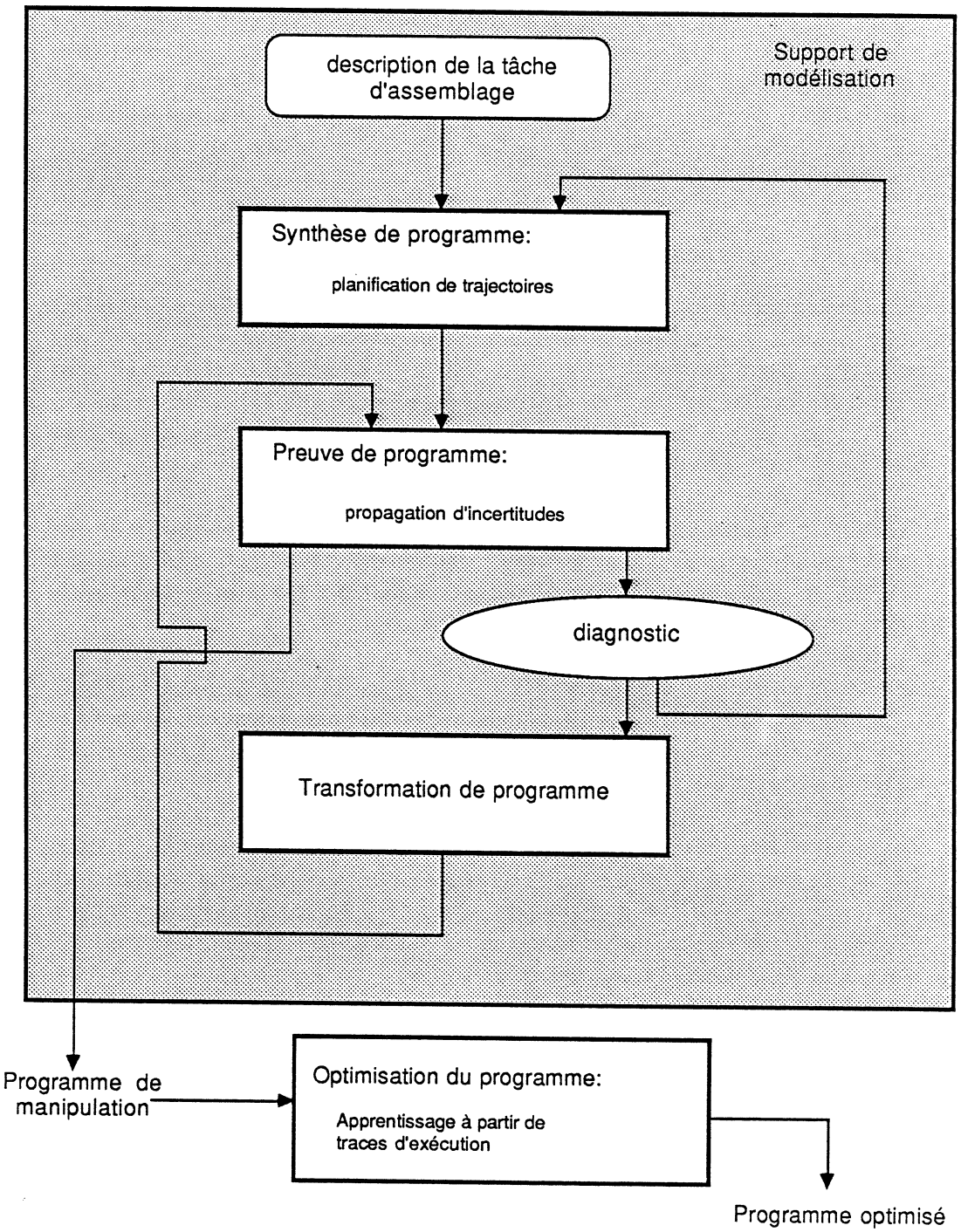


Figure 7.2: Architecture du système SHARP

Comme dans TWAIN, seules les interdépendances liées aux contraintes d'incertitude sont traitées en profondeur. Elles sont considérées à deux niveaux: dans la synthèse de programme par l'utilisation d'heuristiques (prises stables, mouvements fins à compliance) et dans la preuve de programme par l'utilisation de méthodes symboliques de propagation des incertitudes. Les interdépendances liées aux contraintes d'accessibilité sont prises en compte à la fois par la non linéarisation arbitraire des différentes phases de planification de trajectoires et par l'utilisation de backtracking en cas d'échec.

La synthèse de programme

Trois planificateurs composent le module de synthèse: les planificateurs des opérations de saisie (cf. chapitre 6), de transfert et de montage. Le séquencement des différentes opérations de planification n'est pas figé. L'approche consiste en effet à rechercher les problèmes les plus contraints et à les planifier d'abord. Ceci est possible puisque l' espace des solutions potentielles pour la saisie et le montage peut être calculé sans considération de critères globaux.

Une première phase (cf. figure 7.3) consistera donc à utiliser les méthodes de calculs d'accessibilité locale pour déterminer l'ensemble des prises potentielles et celui des stratégies de montage utilisables. On déterminera sur cette base le problème le plus contraint (généralement les opérations de montage).

La phase suivante sera fondée sur des critères d'accessibilité globale. La planification des trajectoires de transfert sera faite après celle des opérations de saisie et montage puisqu'elle est assez insensible aux incertitudes et qu'elle est relativement indépendante au niveau des contraintes d'accessibilité.

Les cas d'échec conduisent, dans le cas des trajectoires de transfert, à diagnostiquer les causes de l'échec à partir de l'espace des configurations: espace trop contraint ou problème dû à la charge utile. Dans ce dernier cas, il faudra replanifier les opérations de saisie; dans l'autre cas, il s'agira d'un échec du programme nécessitant une réorganisation de l'espace de travail.

* Les mouvements fins:

L'approche utilisée pour les mouvements fins est similaire à celle de la saisie. Elle consiste à opérer en deux étapes: l'une effectuant la prédiction de trajectoires potentielles par examen des contacts entre les faces des objets à assembler et l'autre (non encore traitée) effectuant la vérification contextuelle de ces trajectoires (outil compris).

Considérons deux objets en position montés; cette situation finale peut être caractérisée par les contacts mis en jeu.

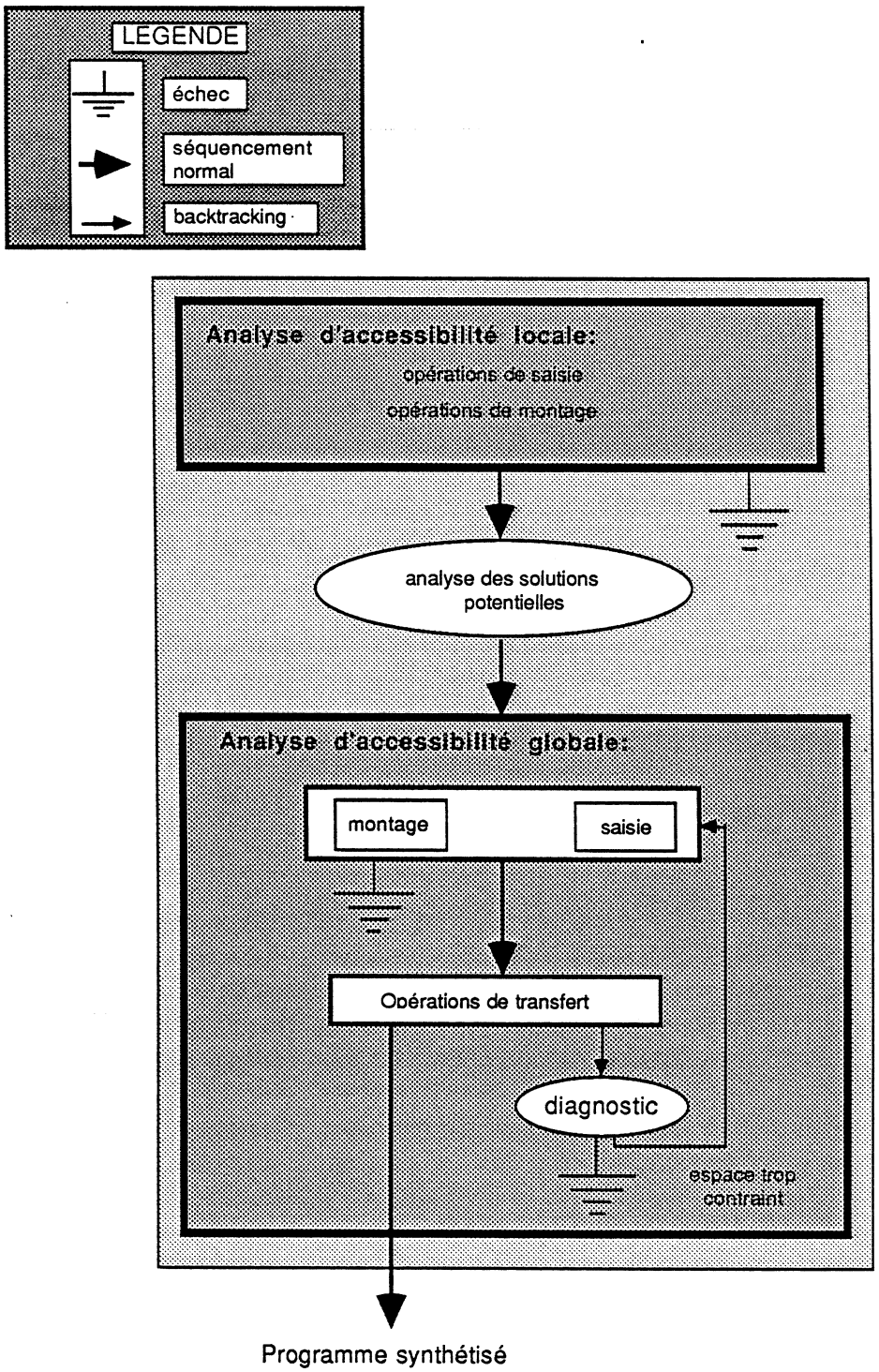


Figure 7.3: La synthèse de programme dans SHARP

La première phase a pour but d'engendrer une suite de trajectoires à compliance à partir de l'étude du démontage des deux objets concernés. Une représentation bien choisie permet de faciliter la résolution du problème. On utilise, en effet, une sphère unité (cf. figure 7.4) permettant de représenter les mouvements potentiels de deux objets dont deux faces sont en contact. La projection de cette sphère sur un plan fournit un domaine dont les frontières représentent les mouvements àcompliance respectant ce contact. Elle permet de propager aisément les contraintes liées aux différents contacts et donc de déterminer tous les mouvements possibles entre les deux objets compatibles avec ces contacts (intersection des domaines) (cf. figure 7.4). Un mouvement ainsi engendré conduit à une nouvelle situation de contact, qu'il faut traiter de manière analogue, ou au désassemblage des objets.

La trajectoire choisie par exploration du graphe ainsi construit est optimale au sens de sa longueur et de sa qualité. On combine pour cela, l'utilisation d'une fonction heuristique évaluant la distance du nœud au nœud solution, à celle d'un ensemble de conseils représentatifs d'une connaissance "experte" sur la qualité des trajectoires engendrées. On évitera, par exemple, la compliance dans un plan vertical ou sur une surface rugueuse.

Une première implantation de la méthode assurant l'optimalité de la trajectoire en longueur a déjà été réalisée. Le travail actuel [Laugier 86] porte d'une part sur la construction de la base de connaissances expertes et sur son exploitation, et d'autre part sur l'extension de la méthode aux mouvements en rotation (on se limite pour l'instant aux translations).

* Les mouvements de transfert:

Le problème posé est celui de la planification des trajectoires permettant au robot de transporter des objets dans un espace parsemé d'obstacles. Les trajectoires engendrées doivent à la fois satisfaire des contraintes physiques (accessibilité) et des contraintes économiques (optimalité de la solution). La difficulté principale vient de ce que les corps en mouvement appartiennent à un système articulé.

Les travaux du LIFIA dans ce domaine [Laugier 85a], [Germain 85] sont plus généraux que les précédentes approches dans le sens où ils sont orientés vers le développement d'outils de modélisation et de planification paramètrables à la fois par le système articulé et par l'environnement. Une approche similaire est présentée dans [Lozano 85b].
La planification des trajectoires de transfert est réalisée dans un contexte d'exécution connu a priori, aux incertitudes près. La méthode utilisée pour engendrer de telles trajectoires passe par une représentation "intelligente" (i.e. significative et concise) de l'ensemble des états possibles du système mobile, à la fois permis et atteignables. Elle repose sur un procédé récursif paramètrable permettant l'obtention d'un modèle surcontraint de l'espace des configurations du robot. Les trajectoires possibles sont de ce fait moins nombreuses que celles réellement valides, mais elles ont l'avantage

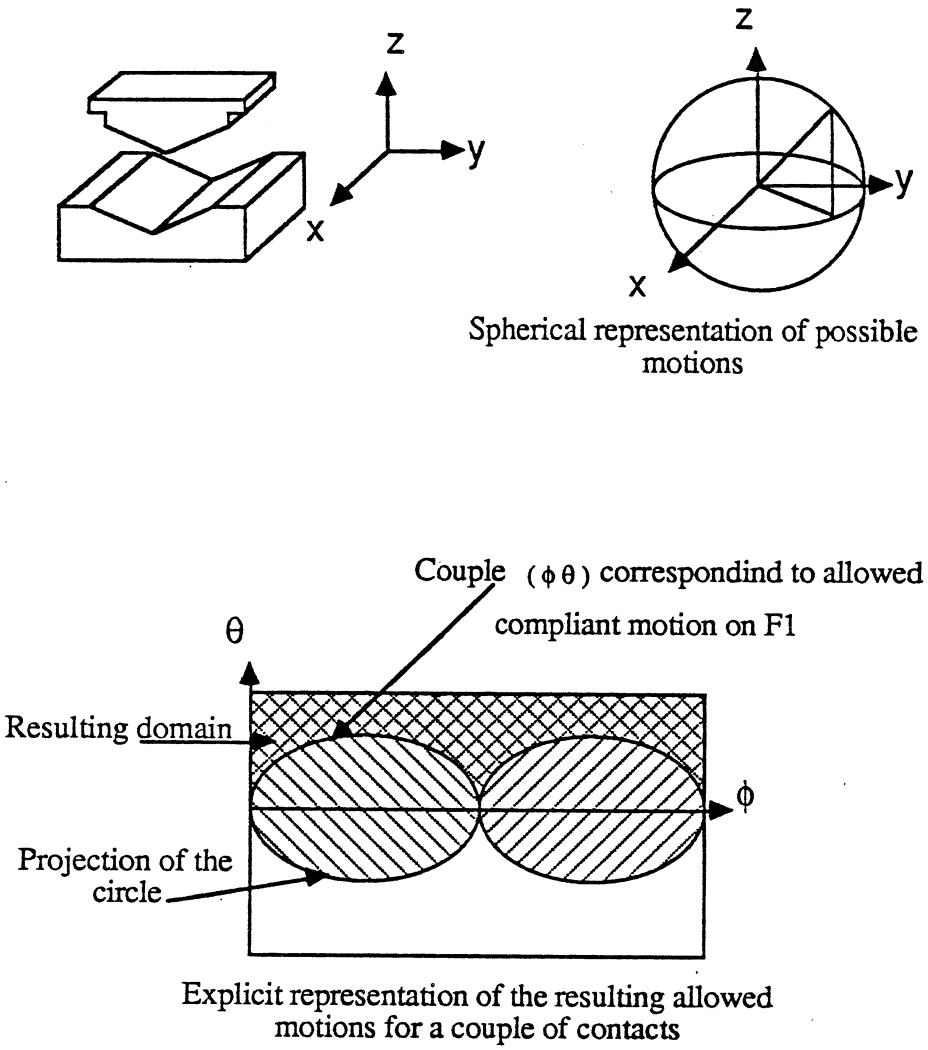


Figure 7.4: Recherche de stratégie de montage [Theveneau 85]

d'être parfaitement sûres. Ce procédé est fondé sur une discrétisation de l'espace articulaire du robot, et sur un mécanisme de propagation de contraintes géométriques inspiré de techniques de grossissement d'obstacles.

La principale difficulté provient de ce que le pas de discrétisation est difficile à maitriser: le comportement exponentiel de l'algorithme en fonction de la complexité des modèles traités peut être réduit de manière très significative par l'emploi judicieux de stratégies et d'heuristiques. Celles-ci permettent d'adapter le pas de discrétisation à partir d'une évaluation de la complexité locale de l'environnement.

Les travaux actuels portent pour une bonne part sur ce point, et des fonctions

de découpage paramètrables ont déjà éte expérimentées.

La preuve et la transformation de programme

Il s'agit d'utiliser des méthodes de preuve de programme (module de propagation des incertitudes) pour déterminer si un programme synthétisé à l'étape précédente s'exécutera correctement dans un univers réel de manipulation. Si ce n'est pas le cas, ce programme devra être transformé par modification de certaines instructions (module de planification de trajectoires) ou insertion de nouvelles instructions (module de transformation) [1] .

A l'heure actuelle, seule la phase de preuve de programme a été approfondie [Puget 85], [Pertin 86]. Elle repose sur le principe de modélisation des incertitudes présenté en §2.1.3.

La propagation des incertitudes à travers un programme peut se faire de deux manières:

- de façon descendante: en calculant les conditions d'incertitude en sortie d'une séquence d'instructions (postconditions) connaissant les conditions d'incertitude en entrée
- de façon ascendante: en calculant les conditions d'incertitude en entrée d'une séquence d'instructions (préconditions) connaissant les conditions d'incertitude en sortie.

Une première phase de propagation ascendante permettra de déterminer les paramètres pertinents par rapport au problème des incertitudes et par là de réduire les calculs symboliques. Connaitre l'incertitude sur la position d'un objet non manipulé mais constituant un obstacle est, par exemple, inutile.

Elle autorisera la recherche des points de programme à modifier pour la réduction des incertitudes, quand la propagation descendante détectera une impossibilité à l'exécution d'une instruction. Cette double propagation nécessite:

- d'une part de déterminer les équations définissant la propagation à travers les instructions du programme,
- d'autre part de mettre en œuvre les méthodes de calcul symbolique permettant la réalisation effective de cette propagation [Pertin 86].

Le premier point pose un problème essentiel au niveau du choix du langage intermédiaire dans lequel est synthétisé le programme. En effet, propager les contraintes d'incertitude à travers les langages dits de niveau effecteur semble relativement utopique à cause de l'utilisation conjointe, dans ces langages, de primitives

[1] Cette recherche est effectuée en étroite collaboration avec la société ITMI

de niveau effecteur et de primitives de niveau commande (déplacements d'axes, exploitation explicite de variables d'états, ...).

Ce langage intermédiaire n'a pas encore été clairement déterminé; cependant, nous pensons qu'il contiendra, outre les structures algorithmiques classiques (cf. LM), des instructions du type: saisie, lâcher, déplacement (absolu ou relatif) de l'effecteur, mouvements à compliance exprimés par des relations géométriques (cf. le langage LUCIFER développé au LAAS [Giraud 83]).

L'avantage de ces instructions à compliance, exprimées qualitativement, est de ne faire aucune référence au moyen de contrôle (compliance tactile, visuelle) et par là de donner aux programmes une relative indépendance par rapport à l'environnement de manipulation.

L'analyse, en cas d'échec, du programme synthétisé et sa transformation sont des problèmes difficiles qui n'ont pas encore été approfondis. On peut supposer en première approximation qu'on essaiera toujours de trouver une solution basée sur l'insertion d'instructions d'acquisition d'informations (utilisation de capteurs ou de mouvements "à la Mason") avant de tenter une replanification de certaines opérations. Les diagnostic et transformation pourront exploiter des connaissances sur les capteurs disponibles (type d'information acquise, précision de l'acquisition ...) par des mécanismes de type système expert (cf. [Dufay 84]).

L'optimisation de programme

Ce module repose sur de précédents travaux réalsés dans notre équipe concernant l'utilisation de méthodes d'apprentissage pour la synthèse de mouvements fins. Son intérêt réside dans la difficulté d'évaluer la répétabilité d'un manipulateur et les incertitudes; ceci conduit généralement à surcontraindre la solution.

La méthode développée par [Dufay 83] permet de synthétiser, en ligne, un programme de réalisation de mouvements fins adapté à un contexte réel de manipulation, grâce à un apprentissage à partir de traces d'exécution.

Dans notre cas, le programme étant synthétisé par ailleurs, nous utiliserons uniquement les méthodes d'inférence de grammaire sousjacentes à cette approche, pour l'optimisation du programme.

7.4.3 TWAIN et SHARP

La comparaison entre TWAIN et SHARP ne sera effectivement intéressante que lorsqu'elle reposera sur une implantation réelle des deux systèmes. A l'heure actuelle, il faut reconnaitre que, si les deux approches sont différentes et si les problèmes traités ne sont pas tout à fait identiques, les faiblesses des structures de contrôle sont tout à fait les mêmes. Elles concernent les points suivants:

- la prise en compte des interdépendances liées aux contraintes d'accessibilité par des méthodes de type "méta-planification" permettant des backtracking "intelligents",
- la transformation de programmes.

Notons cependant que l'expérience précédemment acquise au LIFIA dans le contexte des mouvements fins [Dufay 83] peut guider notre approche du second problème.

De plus, l'approche par synthèse/preuve de programme adoptée dans SHARP nous parait conceptuellement plus propre que l'approche de TWAIN. Elle permet en outre d'avoir des programmes reflètant le problème de manipulation lui-même (en terme de relations d'assemblage à réaliser) relativement indépendemment des problèmes d'incertitude liés au contexte de manipulation. Des programmes de niveau intermédiaire pourront ainsi être adaptés à des cadences de production variables ou à des modications de la disponibilité des capteurs sans que la manipulation elle-même soit remise en cause. Ce gain de flexibilité nous semble important dans le contexte de l'automatisation.

D'autre part, une telle approche autorisera plus facilement l'extraction de résultats fondamentaux et méthodologiques, et d'outils de programmation en-ligne (primitives de synthèse de programmes ou de vérification de programmes écrits par ailleurs).

Conclusion

La programmation de niveau objet des robots d'assemblage repose pour une assez large part sur l'automatisation du raisonnement géométrique et spatial, et nécessite le développement de modèles de représentation adéquats. Notre contribution s'est faite sur deux points essentiels:

- Nous avons tout d'abord défini les besoins inhérents à la modélisation de l'univers de manipulation, univers à la fois géométrique, physique, évolutif et incertain. En l'absence de modèles répondant à ces différents besoins, aussi bien dans le domaine de la Robotique que dans celui de la CFAO, nous avons conçu le système SMGR: Système de Modélisation Géométrique et Relationnelle. Celui-ci permet une représentation multiple des solides et fournit une structure de données propre à supporter la définition de relations entre objets. Un tel modèle a été implanté et validé au travers de son utilisation pour la simulation graphique de tâches de manipulation. Une représentation des incertitudes de positionnement des objets, devant être intégrée à SMGR, est également proposée.

- Nous nous sommes ensuite intéressés au problème de l'automatisation du raisonnement géométrique et spatial dans le cas particulier des opérations de saisie. Nous avons été amenés à préciser le concept de prise sous-tendant notre approche. Une méthode opérant en deux phases a été développée. La phase de raisonnement géométrique destinée à engendrer un ensemble de prises potentielles, est fondée sur la notion de contact entre l'outil et l'objet. La phase de raisonnement spatial permettant de vérifier l'exécutabilité de ces prises est effectuée dans un espace bi-dimensionnel grâce à l'utilisation d'une représentation particulière de l'outil. Malgré certains points faibles, notre approche présente une certaine généralité, donc originalité, par rapport au type d'outil considéré et aux représentations utilisées. Un sous-ensemble de l'approche a été implanté.

Enfin nous décrivons partiellement le système SHARP, système prototype de programmation de niveau objet, auquel doit s'intégrer ce travail.

Les recherches dans le domaine de la programmation automatique nous ont permis de comprendre et de maitriser, en partie, un certain nombre de problèmes inhérents au comportement actif du manipulateur. Les hypothèses faites dans le cadre de ces recherches sont cependant assez restrictives: on se place dans un univers

physique clos, totalement modélisable, complètement contrôlé par le système de programmation (aux alèas du monde physique près) et uniquement modifié par le manipulateur lui-même. En outre, les outils perceptifs ne sont utilisés qu'à des fins d'aide à l'exécution des actions du manipulateur. De telles hypothèses conviennent bien à un environnement structuré et peuvent donc permettre le développement et la mise au point de programmes fiables de commande de robots dans le contexte de l'automatisation de la production.

L'utilisation des robots dans un univers non structuré (robotique en milieu extrême, robotique de surveillance et de maintenance, robotique militaire) nécessite le développement de nouvelles approches.

Le robot de maintenance nous semble à l'heure actuelle un sujet de choix pour développer de telles recherches. Il est à la fois riche en problèmes fondamentaux et en applications (maintenance d'atelier, surveillance de centrale nucléaire...).

Cette orientation des recherches en Robotique est extrêmement importante puisqu'elle constitue à notre sens le réel premier pas vers la machine intelligente autonome. C'est de plus un sujet de choix pour étudier l'intéraction planification - action - perception et faire converger les recherches de notre laboratoire en Robotique (mobile et de manipulation), en Vision et en Intelligence Artificielle.

Les problèmes très différents posés par un tel robot impliquent une remise en cause partielle des méthodes développées dans le contexte de la programmation automatique.

Il est alors essentiel de ne plus considérer la perception, uniquement comme une partie de l'action, mais également comme un outil de base de la planification. Dans ce contexte, la vision a une importance fondamentale puisqu'elle doit nous permettre de construire des modèles [Demazeau 86a] utilisables à la fois pour la planification des actions et pour le contrôle de leur exécution [Lux 85].

Le problème de la saisie nous parait particulièrement bien adapté à ce type d'étude (cf. §6.3). C'est pourquoi nous nous intéressons actuellement à un type de collaboration entre la perception, l'action et la planification dans le contexte particulier de la vision tri-dimensionnelle et de la saisie automatique. Cette coopération a déjà éte évoquée à un niveau de raisonnement plutôt quantitatif (cf. §6.2.3). Pour être réellement enrichissante, elle doit passer par la définition de modèles de représentation adéquats, de nature plutôt symbolique, et par le développement de modèles de raisonnement qualitatifs sur ces représentations (cf. §6.4).

De tels travaux doivent compléter, de façon significative, nos précédents travaux sur la saisie et la modélisation.

Bibliographie

[Ahuja 80] N.Ahuja, R.T.Chien, N.Bridwell: *"Interference detection and collision avoidance among three dimensional objects"*, 1st Annual National Conference on Artificial Intelligence, Stanford University, Août 1980

[Arbab 85] F.Arbab, J.M.Wing: *"Geometric reasoning: a new paradigm for processing geometric information"*, IFIP W.G.5.2 Working Conference on Design Theory for CAD, Tokyo, 1-3 Octobre 1985/ Technical Report no TR-85-333, Computer Science Department, University of Southern California, Los Angeles, Juillet 1985

[Ayache 84] N.Ayache et al.: *"Manipulation automatique de pièces industrielles en vrac planaire"*, 1er Colloque IMAGE (CESTA), Biarritz, Mai 1984

[Ballard] D.H.Ballard, C.M.Brown: *"Computer vision"*, pp264/307, Prentice Hall.

[Baker 85] B.S.Baker, S.Fortune, E.Grosse: *"Stable prehension with a multifingered hand"*, IEEE International Conference on Robotics and Automation, St Louis, Mars 1985

[Bancon 82] G.Bancon, B.Huber: *"Depression and dual grippers with their possible applications"*, 12th International Symposium on Industrial Robots, Paris, Juin 1982

[Beaubaton 85a] D.Beaubaton: *"Le contrôle de la main: planification du geste et assistance sensori-motrice"*, Colloque UNICEF, Paris, 1985

[Beaubaton 85b] D.Beaubaton: *"Plans et programmes moteurs: quelques principes communs dans le contrôle des systèmes artificiels et biologiques"*, à paraitre dans "Automatique, Intelligence Artificielle et Psychologie" édité par C.Bonnet, J.M.Hoc et G.Tiberghien, Mardaga (Bruxelles)

[Berthoz 78] A.Berthoz: *"Rôle de la proprioception dans le contrôle de la posture et du geste"* dans "Du contrôle moteur à l'organisation du geste", édité par H.Hecaen et M.Jeannerod, Masson, 1978

[Binford 79] T.O.Binford: *"Computer integrated assembly systems"*, Artificial Intelligence Laboratory, Stanford, 1979

[Birk 81] J.R.Birk et al.: *"An orienting robot for feeding workpieces stored in bins"*, IEEE Transactions on System, Man and Cybernetics, vol. SMC-11, Février 1981

[Boissonnat 81] J.D.Boissonnat, F.Germain: *"A new approach to the problem of acquiring randomly oriented workpieces out of a bin"*, Proceedings IJCAI-81, Vancouver, Août 1981

[Boissonnat 82] J.D.Boissonnat: *"Stable matching between a hand structure and an object silhouette"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-4, no 6, Novembre 1982

[Boissonnat 85] J.D.Boissonnat: *"An automatic solid modeler for Robotics applications"*, 3rd International Symposium of Robotics Research, Gouvieux (France), Octobre 1985

[Borianne 84] P.L.Borianne: *"Contribution à la vision par ordinateur tridimensionnelle"*, Thèse de 3ème cycle, INPG, LIFIA, Avril 1984

[Boulle 83] P.Boulle: *"Une nouvelle approche des problèmes liés à la description de scènes tridimensionnelles"*, Rapport de Recherche IMAG no 359, Mars 1983

[Borrel 83] P.Borrel et al.: *"The robotics facilities in the CAD/CAM CATIA system"*, édité par Brooks, IFS Publications, 1983

[Boyse 79] J.W.Boyse: *"Interference detection among solids and surfaces"*, CACM, vol.22, nb.1, Janvier 1979

[Boyse 82] J.W.Boyse, J.E.Gilchrist: *"GMSolid: Interactive modelling for design and analysis of solids"*, IEEE Computer Graphics and its applications, Mars 1982

[Brady 84] M.Brady et al.: *"The mechanic's mate"*, European Conference on Artificial Intelligence, Pisa, 1984

[Braid 73] I.C.Braid: *"Designing with volumes"*, Cantab Press, Cambridge, 1973

[Brooks 79] R.A.Brooks, R.Greiner, T.O.Binford: *"The ACRONYM model-based vision system"*, 6th IJCAI, Tokyo, Août 1979

[Brooks 82a] R.A.Brooks: *"Solving the find-path problem by good representation of free space"*, 2nd AAAI conference, Carnegie Mellon university, Août 1982

[Brooks 82b] R.A.Brooks: *"Symbolic error analysis and robot planning"*, International Journal of Robotics Research, vol 1, no 4, Hiver 1982

[Brooks 83] R.A.Brooks: *"Planning collision free motions for pick and place operations"*, International Journal of Robotics Research, vol 2, no 4, Hiver 1983/ 1st International Symposium of Robotics Research, Bretton Woods, Août 1983

[Brooks 84] R.A.Brooks: *"Aspects of mobile robot visual map making"*, 2nd International Symposium of Robotics Research, Kyoto, Août 1984

[Brost 85] R.C.Brost: *"Planning robot grasping motions in the presence of uncertainty"*, Technical Report CMU-RI-TR-85-12, Carneggie Mellon University, The Robotics Institute, Pittsburgh, Juillet 1985

[Cadoz 84] C.Cadoz, A.Luciani, J.L.Florens: *"Responsive input devices and sound synthesis by simulation of instrumental mechanisms: the Cordis system"*, Computer Music Journal, vol 8, no 3, MIT Press, 1984

[Cameron 81] S.Cameron: *"Bridget: a geometric body modeller"*, Department of Artificial Intelligence, University of Edinburgh, DAI Draft Working Paper, Juillet 1981

[Chien 84] R.T.Chien Ling Zhang, Bo Zhang: *"Planning collision-free paths for robotic arm among obstacles"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol PAMI-6, no 1, Janvier 1984

[Cohen] P.R.Cohen, E.A.Feigenbaum: *"The handbook of Artificial Intelligence"*, Tome 3, pp 513/563, William Kaufmann Inc.

[Comba 68] P.G.Comba: *"A procedure for detecting intersections of three dimensional objects"*, JACM, vol.15, nb.3, pp. 354-366, Juillet 1968

[Crowley 85] J.L.Crowley: *"The construction and maintenance of a composite surface model"*, SPIE, Cannes, Novembre 1985

[Dauchez 83] P.Dauchez: *"Etude de la commande de deux robots manipulateurs lors de tâches coordonnées"*, Thèse de 3ème cycle, Université des Sciences et Techniques du Languedoc, Laboratoire d'Automatique et de Microélectronique de Montpellier, Juin 1983

[Demazeau 86a] Y.Demazeau: *"Indices d'images et indices de scène pour la vision par ordinateur. Application à l'inférence de formes à partir de contours et de la stéréoscopie simple"*, Thèse de Docteur de l'INPG, LIFIA/ENSIMAG, Grenoble, Septembre 1986

[Denavit 55] J.Denavit, R.S.Hartenberg: *"A kinematic notation for lower-pair mechanisms based on matrices"*, Journal of Applied Mechanics, vol 22, Trans. ASME, vol 77, series E, Juin 1955

[Descotte 81] Y. Descotte: *"Représentation et exploitation de connaissances "expertes" en génération de plans d'actions. Application à la conception de gammes d'usinage"*, Thèse de 3ème cycle, Institut National Polytechnique de Grenoble, ENSIMAG, Décembre 1981

[Dreyfus 72] H.L.Dreyfus: *"What computers can't do - The limits of Artificial Intelligence"* Harper and Row Publications, Inc., New York 1972/ "Intelligence Artificielle: Mythes et Limites" édition en langue française, Flammarion, 1984

[Dufay 83] B.Dufay: *"Apprentissage par induction en Robotique: Application à la synthèse de programmes de montage"*, Thèse de 3ème cycle, INPG, Juin 1983

[Dufay 84] B.Dufay, J.C.Latombe: *"An approach to automatic robot programming based on inductive learning"*, Rapport de recherche IMAG no 433, Février 1984/ International Journal of Robotics Research, 1984

[Erdmann 85] M.Erdmann: *"Using backprojections for fine motions planning with uncertainty"*, IEEE International Conference on Robotics and Automation, St Louis, Mars 1985

[Espiau 80] B.Espiau, J.Y.Catros: *"Use of optical reflectance sensors in robotics applications"*, IEEE Transactions on Systems, Man on Cybernetics, vol.SMC-10, no.12, Décembre 80

[Fahlmann 74] S.E.Fahlmann: *"A planning system for robot construction tasks"*, Artificial Intelligence Journal, vol 5, 1974

[Faverjon 84] B.Faverjon: *"Obstacle avoidance using an octree in the configuration space of a manipulator"*, IEEE International Conference on Robotics and Automation, Atlanta, Mars 1984

[Fearing 84] R.S.Fearing: *"Simplified grasping and manipulation with dextrous robot hands"*, AI-Memo-809, Artificial Intelligence Lab., M.I.T., Cambridge, Novembre 1984

[Fikes 71] R.E.Fikes, N.J.Nilsson: *"STRIPS: A new approach to the application of theorem proving to problem solving"*, Artificial Intelligence Journal, no 2 (3/4), 1971

[Fikes 72] R.E.Fikes: *"Monitored execution of plans produced by STRIPS"*, Information Processing 71, North Holland, 1972

[Finkel 74] R.A.Finkel et al.: *"AL, a programming system for automation"*, Memo AIM 243, Artificial Intelligence Laboratory, Stanford, Novembre 1974

[Freeman 71] P.Freeman, A.Newell: *"A model for functional reasoning in design"*, 2nd International Conference on Artificial Intelligence, Londres, Septembre 1971

[Germain 85] F.Germain: *"Génération statique de trajectoires: un algorithme paramètrable par le contexte"*, 5ème Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, Grenoble, Novembre 1985

[Giraud 83] M.Giraud: *"Generalized active compliance for part-mating with assembly robots"*, 1st International Symposium of Robotics Research, Bretton Woods, 1983

[Godard] Godard, Ribeira: *"Le vagabond des limbes"*, Dargaud Editeur

[Gouzenes 84] L.Gouzenes: *"Strategies for solving collision-free trajectories problems for mobile and manipulator robots"*, International Journal of Robotics Reserch, vol 3, no 4, Hiver 1984

[Guinot 85] J.C.Guinot: *"Modelisation and simulation of force-position control for a manipulator-gripper"*, 3rd International Symposium of Robotics Research, Gouvieux (France), Septembre 1985

[Hanafusa 77a] H.Hanafusa, H.Asada: *"Stable prehension by a robot hand with elastic fingers"*, 7th International Symposium on Industrial Robots Tokyo, Octobre 1977/ publié dans "Robot Motion: Planning and Control", édité par M.Brady et al., M.I.T. Press, 1982

[Hanafusa 77b] H.Hanafusa, H.Asada: *"A robot hand with elastic fingers and its application to assembly process"*, IFAC Symposium on Information and Control Problems in Manufacturing Technology, Tokyo, 1977/ publié dans "Robot Motion: Planning and Control", édité par M.Brady et al., M.I.T. Press, 1982

[Henderson 84a] M.R.Henderson: *"Feature recognition in geometric modelling"*, CAM-I's 13th Annual Meeting and Technical Conference "Man or Machine - a choice of intelligence", Clearwater Beach (Floride), Novembre 1984

[Henderson 84b] M.R.Henderson, D.C.Anderson: *"Computer recognition and extraction of form features: a CAD/CAM link"*, Computers in Industry, 5, pp 329/339, 1984

[High Tech 85] D.V.Edson: *"Giving robot hands a human touch"*, High Technology, Septembre 1985

[Hillyard 82] R.Hillyard: *"The BUILD group of solid modelers"*, IEEE, Computer Graphics and its applications, Mars 1982

[Himeno 83] S.Himeno, H.Tsumura: *"The locomotive and control mechanism of the human finger and its application to robotics"*, International Conference on Advanced Robotics, Tokyo, Septembre 1983

[Hirose 77] S.Hirose, Y.Umetani: *"The development of soft gripper for the versatile robot hand"*, 7th International Symposium on Industrial Robots, Tokyo, Octobre 1977

[Holzmann 85] W.Holzmann, J.M.McCarthy: *"Computing the friction forces associated with a three fingered grasp"*, IEEE International Conference on Robotics and Automation, St Louis, Mars 1985

[Iberall 84] T.Iberall, D.Lyons: *"Towards perceptual robotics"*, COINS Technical Report 84-17, Laboratory for Perceptual Robotics, University of Massachusetts, Amherst, Août 1984

[Iberall 85] T.Iberall, G.Bingham, M.A.Arbib: *Opposition space as a structuring concept for the analysis of skilled hand movements"*, COINS Technical Report 85-19, Laboratory for Perceptual Robotics, University of Massachusetts, Amherst, Juillet 1985

[Ignazi 78] G.Ignazi, B.Mollard: *"Organes et sytèmes de préhension: étude bibliographique"*, Rapport IRIA GE-020, Projet SPARTACUS (Aide aux handicapés physiques des membres supérieurs), Janvier 1978

[Ikeuchi 84] K.Ikeuchi et al.: *"Determining grasp points using photometric stereo and the PRISM binocular stereo system"*, AI-Memo-772, Artificial Intelligence Lab., M.I.T., Cambridge, Août 1984

[IMI 85] P.Jorrand et al.: *"IMI - L'Institut de la Machine Intelligente: Objectifs généraux, Travaux scientifiques et Moyens nécessaires"*, INPG, Janvier 1985

[Ingrand 84a] F.Ingrand: *"Inférence d'une description physique à partir d'une description fonctionnelle: application à la conception automatique d'outillage de tenue de pièces sur machine-outil"*, Rapport de DEA, LIFIA, INPG, Juillet 1984

[Ingrand 84b] F.Ingrand, J.C.Latombe: *"Functional reasoning for automatic fixture design"*, CAM-I's 13th Annual Meeting and Technical Conference "Man or Machine - a choice of Intelligence", Clearwater Beach, Florida, Novembre 1984

[ISRR 85] Actes du 3rd International Symposium of Robotics Research, Gouvieux (France), Octobre 1985

[Jacobsen 82] S.C.Jacobsen et al.: *"Development of the UTAH artificial arm"*, IEEE Transactions on Biomedical Engineering, vol BME-29, no 4, Avril 1982

[Jacobsen 84a] S.C.Jacobsen et al.: *"The UTAH/MIT dextrous hand: work in progress"*, International Journal of Robotics Research, vol 3, no 4, Hiver 1984

[Jacobsen 84b] S.C.Jacobsen et al.: *"The version I of the UTAH/MIT dextrous hand"*, 2nd International Symposium of Robotics Research, Kyoto, Août 84/ publié dans "Robotics Research 2", édité par H.Hanafusa et H.Inoue, MIT Press, 1985

[Jared 83] G.E.M.Jared: *"Shape features in geometric modelling"*, General Motors Symposium on "Solid modelling by computers: from theory to applications", Détroit, Septembre 1983/ publié par Plenum Press, 1984

[Jeannerod 81] M.Jeannerod: *"Le contrôle de l'œil sur le geste"*, La Recherche, no 120, Mars 1981

[Jeannerod 83] M.Jeannerod: *"Le cerveau-machine - Physiologie de la volonté"*, Collection Le Temps des Sciences, Editions Fayard, 1983

[Jourob 84] *"La main articulée de Hitachi"*, Le Journal de la Robotique, no 2, Décembre 1984

[Kapur 85] D.Kapur et al.: *"Reasoning about three dimensional space"*, IEEE International Conference on Robtics and Automation, St Louis, Mars 1985

[Kelley 84] R.B.Kelley et al.: *"General methods to enable robots with vision to acquire, orient and transport workpieces"*, NSF Final report, University of Rhode Island, Décembre 1984

[Khatib 80] O.Khatib: *"Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles"*, Thèse de Docteur-Ingénieur, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, Décembre 1980

[Lagoude 85] Y.Lagoude, J.P.Tsang: *"A plan representation structure for expert planning systems"*, Symposium on Computer Aided Process Planning, Miami Beach, Florida, Novembre 1985

[Latombe 77] J.C.Latombe: *"Une application de l'Intelligence Artificielle à la Conception Assistée par Ordinateur (TROPIC)"*, Thèse d'Etat, Institut National Polytechnique de Grenoble, ENSIMAG, Novembre 1977

[Latombe 81a] J.C.Latombe: *"Langages de programmation pour la commande de robots de manipulation"*, 3èmes Journées Scientifiques et Techniques de la Production Automatisée (ADEPA), Toulouse, Juin 1981

[Latombe 81b] J.C.Latombe, E.Mazer: *"LM 1981: a high-level programming language for controlling assembly robots"*, 11th International Symposium on Industrial Robots, Tokyo, Octobre 1981

[Latombe 84] J.C.Latombe et al.: *"The LM robot programming system"*, 2nd International Symposium on Robotics Research, Kyoto, Août 1984

[Laugier 81] C.Laugier: *"A program for automatic grasping of objects with a robot arm"*, 11th International Symposium on Industrial Robots, Tokyo, Octobre 1981

[Laugier 82a] C.Laugier, B.Dufay: *"Geometrical reasoning in automatic grasping and contact analysis"*, PROLAMAT 82, Leningrad, May 1982

[Laugier 82b] C.Laugier: *"LISP-3D: logiciel graphique pour la manipulation et la visualisation de scènes tridimensionnelles"*, Rapport de Recherche IMAG, no 328, Septembre 1982

[Laugier 83a] C.Laugier, J.Pertin: *"Automatic grasping: a case study in accessibility analysis"*, International Conference on Advanced Software in Robotics, Liège, Mai 1983/ publié dans "Advanced Software in Robotics", édité par A.Danthine et M.Géradin, North Holland, 1984

[Laugier 83b] C.Laugier: *"Influence du raisonnement géométrique dans le choix d'une prise d'objet"*, Rapport de Recherche IMAG no 414, Décembre 83

[Laugier 84] C.Laugier, J.Pertin-Troccaz: *"Graphic simulation as a tool for debugging robot control programs"*, 1st International Symposium on Design and Synthesis, Tokyo, Juillet 1984

[Laugier 85a] C.Laugier, F.Germain: *"An adaptative collision-free trajectory planner"*, '85 International Conference on Advanced Robotics, Tokyo, Septembre 1985

[Laugier 85b] C.Laugier, J.Pertin-Troccaz: *"S.H.A.R.P.: A system for automatic programming of manipulation robots"*, 3rd International Symposium of Robotics Research, Gouvieux (France), Octobre 1985

[Laugier 86] C.Laugier, P.Theveneau: *"Planning sensor-based motions for part-mating using geometric reasonning"*, ECAI'86, Brighton, Juillet 1986

[Liebermann 77] L.I.Liebermann, M.A.Wesley: *"AUTOPASS: an automatic programming system for computer controlled mechanical assembly"*, IBM Journal of Research and Development, Juillet 1977

[Lorenz 70] K.Lorenz: *"Trois essais sur le comportement animal et humain"*, Edition du Seuil, 1970

[Lozano 76] T.Lozano-Perez: *"The design of a mechanical assembly system"*, AI-TR-397, M.I.T., Artificial Intelligence Laboratory, Cambridge, Décembre 1976

[Lozano 77] T.Lozano-Perez, P.H.Winston: *"LAMA: a language for automatic mechanical assembly"*, 5th International Joint Conference on Artificial Intelligence, MIT, Cambridge, Août 1977

[Lozano 79] T.Lozano-Perez, M.A.Wesley: *"An algorithm for planning collision free paths among polyhedral obstacles"*, CACM, Vol.22, nb.1, Octobre 1979

[Lozano 80] T.Lozano-Perez: *"Spatial planning: a configuration space approach"*, AI Memo no. 605, Artificial Intelligence Laboratory, M.I.T., Cambridge, Décembre 1980

[Lozano 81] T.Lozano-Perez: *"Automatic planning of manipulator transfer movements"*, IEEE Transactions on System, Man and Cybernetics, SMC-11, 10, 1981/ publié dans "Robot Motion: Planning and Control", édité par M.Brady et al., MIT Press, 1982

[Lozano 83] T.Lozano-Perez, M.T.Mason, R.H.Taylor: *"Automatic synthesis of fine-motions strategies for robots"*, 1st International Symposium of Robotics Research, Bretton Woods, Août 1983

[Lozano 84] T.Lozano-Perez, R.A.Brooks: *"An approach to automatic robot programming"*, General Motors Symposium on "Solid modelling by computers: from theory to applications", Detroit, Septembre 1983/ publié par Plenum Press, 1984

[Lozano 85a] T.Lozano-Perez, R.A.Brooks: *"An approach to automatic robot programming"*, AI Memo 842, Artificial Intelligence Laboratory, M.I.T., Avril 1985

[Lozano 85b] T.Lozano-Perez: *"Motion planning for simple manipulator robots"*, 3rd International Symposium of Robotics Research, Gouvieux (France), Octobre 1985

[Lyons 85] D.M.Lyons: *"A simple set of grasps for a dextrous hand"*, IEEE International Conference on Robotics and Automation, St Louis, Mars 1985

[Lux 85] A.Lux, J.L.Gordillo: *"Synthetizing vision programs from robot task specifications"*, 3rd International Symposium of Robotics Research, Gouvieux (France), Octobre 1985

[Marr 79] D.Marr: *"Visual information processing: the structure and creation of visual information"*, 6th International Joint Conference on Artificial Intelligence, Tokyo, 1979

[Mason 82] M.T.Mason: *"Manipulator grasping and pushing operations"*, AI-TR-690, Artificial Intelligence Lab., M.I.T., Cambridge, Juin 1982

[Mason 84] M.T.Mason: *"Mechanics of pushing"*, 2nd International Symposium of Robotics Research, Kyoto, Août 1984

[Mason 85] M.T.Mason: *"The mechanics of manipulation"*, IEEE International Conference on Robotics and Automation, St Louis, Mars 1985

[Maruyama 72] K.Maruyama: *"A procedure to determine intersections between polyedral objects"*, Int. Journal of Computer and Information Sciences", vol.1, nb.3, 1972

[Mazer 81] E.Mazer: *"Réalisation d'un support expérimental de recherche pour le projet robotique PANDORE. Définition et implantation du langage LM"*, Thèse de 3ème cycle, Institut National Polytechnique de Grenoble, Janvier 1981

[Mazer 82] E.Mazer: *"An algorithm for computing relative positions between two objects from symbolical specifications"*, IMAG, Rapport de Recherche no 297, 1982

[Mazer 83] E.Mazer: *"Geometric programming of assembly robots (LM-Géo)"*, International Symposium on Advanced software in Robotics, Liège, Mai 1983

[Meagher 81] D.Meagher: *"Geometric modelling using octree encoding"*, Computer Graphics and Image Processing, no 19, 1982

[Miribel 84] J.F.Miribel: *"Conception et implantation d'un système de programmation de robots"*, Thèse de 3ème cycle, LIFIA/IMAG, Grenoble ,France, Octobre 1984

[Mundy 85] J.L.Mundy: *"Reasoning about 3-d space with algebraic deduction"*, 3rd International Symposium of Robotics Research, Gouvieux (France), Octobre 1985

[Napier 56] J.R.Napier: *"The prehensile movements of the human hand"*, Journal of Bone and Joint surgery, vol 38B, no 4, Novembre 1956

[Napier 62] J.R.Napier: *"The evolution of the hand"*, Scientific American, vol 207, no 6, pp 56/62, 1962

[Nitzan 76] D.Nitzan, C.A.Rosen: *"Programmable industrial auotmation"*, IEEE Transactions on Computers, vol C-25, no 12, Décembre 1976

[Okada 77] T.Okada: *"On a versatile finger system"*, 7th International Symposium on Industrial Robots, Tokyo, Octobre 1977

[Okada 79] T.Okada: *"Object handling system for manual industry"*, IEEE Transactions on Systems, Man and Cybernetics, vol SMC-9, no 2, Février 1979

[Paillard 75] J.Paillard, D.Beaubaton: *"Problèmes posés par les contrôles moteurs ipsilatéraux après déconnexion hémisphérique chez le singe"*, dans "Les syndrômes de disconnexion calleuse chez l'Homme", édité par B.Scott et F.Michel, Lyon, 1975

[Paillard 78] J.Paillard, D.Beaubaton: *"De la coordination visuo-motrice à l'organisation du geste"*, dans "Du contrôle moteur à l'organisation du geste", édité par H.Hecaen, M.Jeannerod, Masson, 1978

[Pertin 84] J.Pertin-Troccaz: *"S.M.G.R.: un système de modélisation géométrique et relationnelle pour la Robotique"*, Rapport de recherche IMAG no 422, Juin 1984

[Pertin 85] J.Pertin-Troccaz: *"S.M.G.R.: a geometric and relational modeller for robotics"*, International Conference on Advanced Robotics, Tokyo, Septembre 1985

[Pertin 86] J.Pertin-Troccaz, P.Puget: *"Contrôle dans le système de programmation automatique SHARP: gestion des interdépendances liées aux contraintes d'accessibilité et d'incertitudes"*, Rapport de Recherche LIFIA no 50, Juin 1986

[Popplestone 78] R.J.Popplestone, A.P.Ambler, I.Bellos: *"RAPT: a language for describing assemblies"*, The Industrial Robot, Septembre 1978

[Puget 85] P.Puget: *"Problèmes de prise en compte d'incertitudes en Robotique d'assemblage"*, Rapport de DEA, LIFIA/IMAG Laboratory, Juin 1985

[Requicha 80] A.A.G.Requicha: *"Representations for rigids solids: theorys, methods and systems"*, ACM Computing Surveys, vol. 12, no 4, Décembre 1980

[Requicha 83] A.A.G. Requicha: *"Toward a theory of geometric tolerancing"*, International Journal of Robotics Research, vol 2, no 4, Décembre 1983

[Requicha 79] A.A.G.Requicha, H.B.Voelcker: *"Geometric modelling of mechanical parts and machining processes"*, COMPCONTROL'79 , Sopron, Hungary, Novembre 1979

[Requicha 82] A.A.G.Requicha, H.B. VOELCKER: *"Solid modelling: a historical summary and contemporary assessment"*, IEEE, Computer Graphics and its applications, Mars 1982

[Richard 68] *"Intelligence"*, Encyclopædia Universalis (1968), vol 8

[Rodet 84] M.Rodet et al.: *"FORMES: composition and scheduling of processes"*, Computer Music Journal, vol 8, no 3, MIT Press, 1984

[Romiti 81] A.Romiti et al.: *"Picking from a bin through tactile sensing"*, 11th Int. Symposium on Industrial Robots, Tokyo, Octobre 1981

[Romulus] "ROMULUS: User's Manual", Shape Data Ltd

[Salisbury 82] J.K.Salisbury, J.J.Craig: *"Articulated hands: force control and kinematics issues"*, International Journal of Robotics Research, vol 1, no 1, Printemps 1982

[Seroussi 82] S.Seroussi: *"Des gestes qu'on voudrait copies conformes"* dans "La science des robots", Science et Vie, Hors Série, 1982

[Shimano 84] B.E.Shimano, C.C.Geschke, C.H.Splalding: *"VAL-II: a new robot control system for automatic manufacturing"*, IEEE International Conference on Robotics, Atlanta, Mars 1984

[Taylor 76] R.H.Taylor: *"Synthesis of manipulator control programs from task-level specifications"*, AIM 228, Artificial Intelligence Laboratory, Stanford, Juillet 1976

[Taylor 82] R.H.Taylor, P.D.Summers, J.M.Meyer: *"AML: a manufacturing language"*, The International Journal of Robotics Research, Vol1, No3, 1982

[Theveneau 85] P.Theveneau: *"Une méthode de génération de mouvements fins basée sur une analyse des contacts"*, Rapport de DEA, LIFIA/IMAG Laboratory, Juin 1985

[Tilove 83] R.B.Tilove: *"Extending solid modelling systems for mechanism design and kinematic simulation"*, IEEE, Computer Graphics and its applications, mai/juin 1983

[Todd 81] D.J.Todd: *"A method for grasping randomly oriented objects using touch sensing"*, Artificial Intelligence Lab., Queen Mary College, Univ. of London, Juin 1981

[Udupa 77] S.M.Udupa: *"Collision detection and avoidance in computer controlled manipulators"*, Proceedings IJCAI 77, Cambridge, 1977

[Valade 85] J.M.Valade: *"Raisonnement géométrique et synthèse de trajectoire d'assemblage"*, Thèse de Docteur-Ingénieur, Université Paul Sabatier, Laboratoire d'Automatique et d'Analyse des Systèmes, Toulouse, Janvier 1985

[Voelcker 77] H.B.Voelcker, A.A.G.Requicha: *"Geometric modelling of mechanical parts and processes"*, Computer Journal, Décembre 1977

[Voelcker 78] H.B.Voelcker et al.: *"The PADL 1.0/2 system for defining and displaying solid objects"*, ACM Computer Graphics, vol 12, no 3, Août 1978

[Volz 83] R.A.Volz et al.: *"CAD, robot programming and ADA"*, NATO Advanced studies in Robotics and Artificial Intelligence, El Ciocco (Italie), Juin 1983/ publié dans "Robotics and Artificial Intelligence", édité par M.Brady et al., Springer Verlag, 1984

[Wesley 80] M.A.Wesley et al.: *"A geometric modeling system for automated assembly"*, IBM Journal of Research and Development, vol 24, no 1, Janvier 1980

[Williamson 72] H.Williamson: *"Algorithm 420, hidden line plotting program"*, Communications of the ACM, Vol 15, no 2, Février 1972

[Wingham 77] M.Wingham: *"Planning how to grasp objects in a cluttered environment"*, M. Phil. Thesis, University of Edinburgh, Scotland, 1977

[Winston 83] P.H.Winston et al.: *"Learning physical descriptions from functional definitions, examples and precedents"*, 1st International Symposium of Robotics Research, Bretton Woods, Août 1983

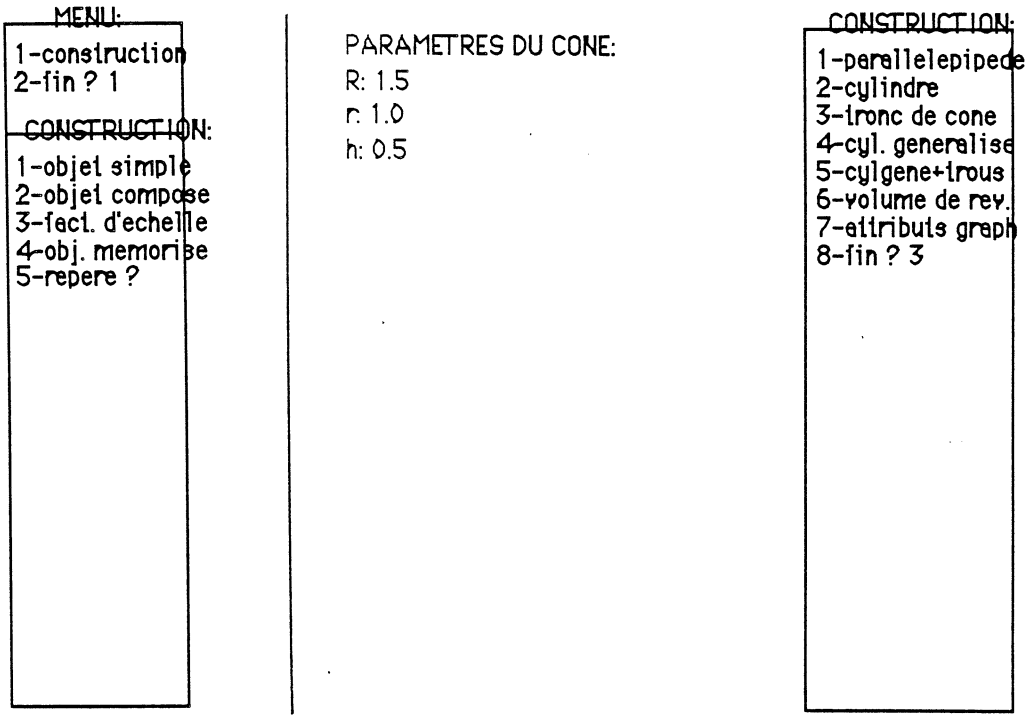
[Wolter 84] J.D.Wolter, R.A.Volz, A.C.Woo: *"Automatic generation of gripping positions"*, RSD-TR-2-84, Center for Robotics and Integrated Manufacturing, Robot Systems Division, Ann Arbor (Michigan), Février 1984

[Yavorski] B.Yavorski, A.Detlaf: *"Aide-Mémoire de Physique"*, Editions de Moscou, 1975

Annexe A

SMGR: Exemples

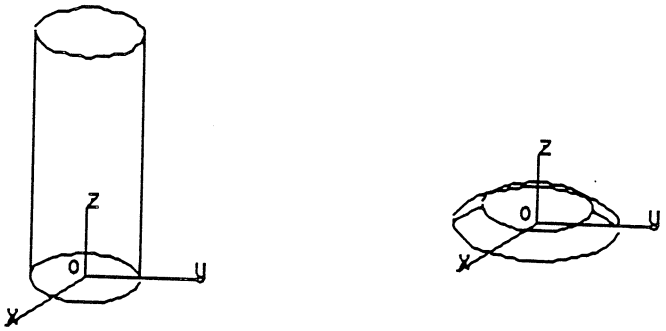
Le module d'entrée de SMGR est, on peut le voir ci-dessous, fondé sur l'utilisation de menus hiérarchisés. Son mode d'utilisation privilégié se fait grace à un terminal graphique muni d'une tablette à digitaliser. Dans cet annexe, nous décrivons brievement ce module.



A.1 Les objets elémentaires

Chaque objet élémentaire créé dans SMGR possède un repère cartésien qui lui est propre (repère de construction). L'opération de construction consiste donc à:

1. définir des composants élémentaires,



2. associer à chacun un repère cartésien (si le repère de construction ne convient pas),

3. combiner ces objets à l'aide d'opérateurs géométriques (collage, juxtaposition, coupe par un plan).

Nous ne décrirons ici que les deux premiers points, le troisième consistant uniquement pour l'utilisateur à positionner ses objets et à invoquer l'opérateur concerné. Nous reportons le lecteur interessé à [Pertin 84].

A.1.1 Les composants élémentaires

La définition de composants élémentaires peut se faire à l'aide de trois primitives:

- La création de volumes élémentaires (parallèlépipède, cône, tronc de cône, cylindre)

- Le balayage d'une surface le long d'un axe; il est caractérisé par la surface de base et la règle de balayage. La surface de base est circulaire, rectangulaire ou polygonale (obtenue à la tablette à digitaliser); elle se trouve dans le plan xOy du composant créé. Le balayage est rectiligne ou circulaire et est caractérisé par la distance de translation ou le rayon et l'angle de giration; il se fait autour ou le long de l'axe z.

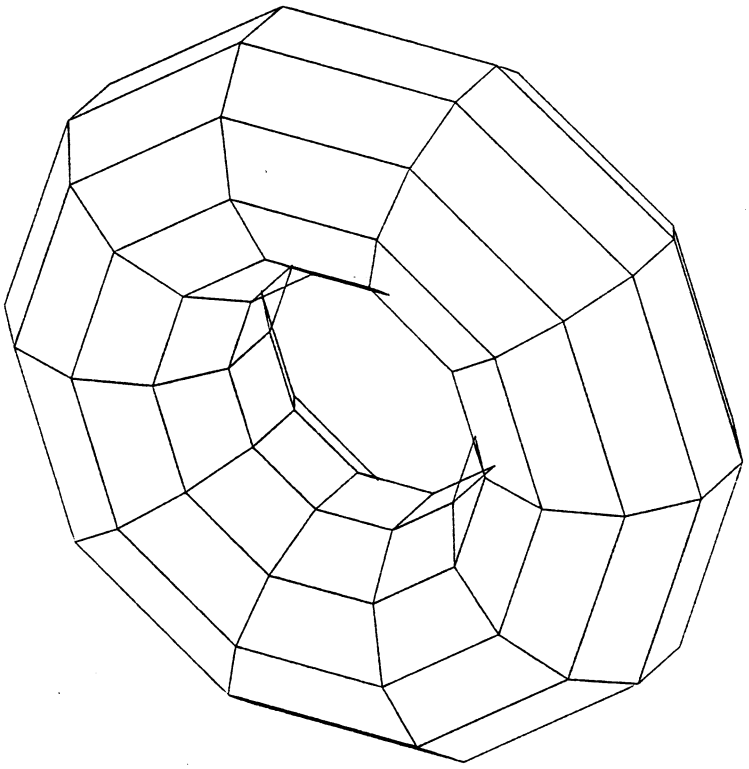
- La révolution d'une génératrice autour d'un axe. La génératrice est une suite de segments de droites et d'arcs de cercles; elle se trouve dans le plan yOz du composant créé. La révolution se fait autour de l'axe z et peut être comprise entre 0 et $2 \star \pi$. Le volume est fermé par deux surfaces situées dans un plan parallèle à xOy et éventuellement deux surfaces supplémentaires jointives de long de l'axe de z (lorsque l'angle de rotation est inférieur à $2 \star \pi$).

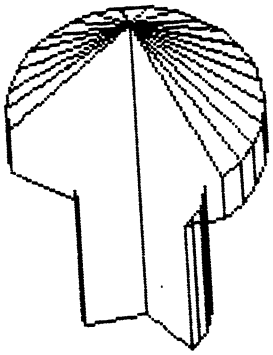
A.1.2 Définition du repère d'un objet élémentaire

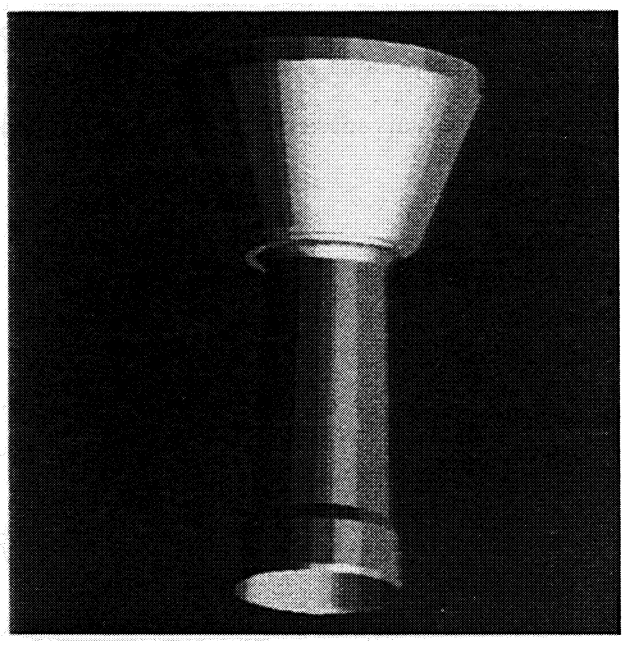
Des informations de position et d'orientation sont nécessaires pour effectuer le positionnement relatif du repère définitif $\Re_i$ par rapport au repère de construction $\Re_{i_0}$.

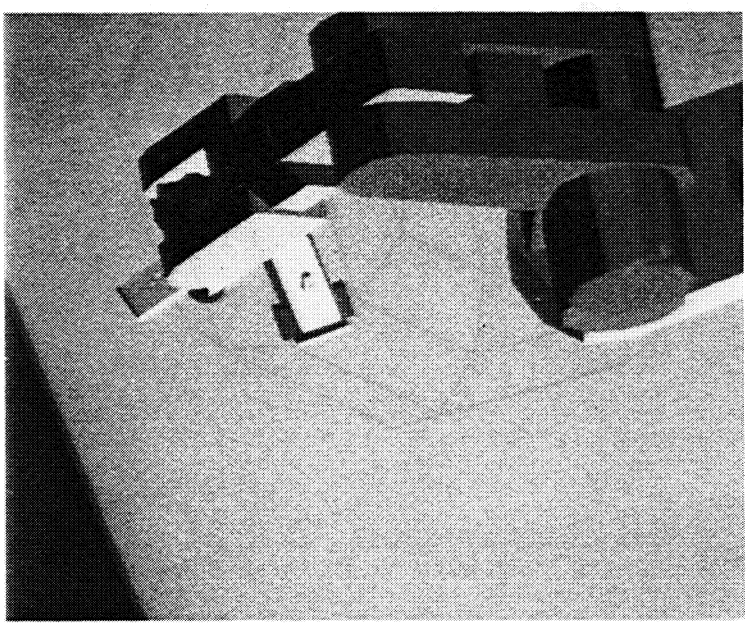
- La position du repère $\Re_i$ par rapport à $\Re_{i_0}$ est donnée par les coordonnées de O_i dans $\Re_{i_0}$.
- L'orientation de $\Re_i$ par rapport à $\Re_{i_0}$ est obtenue au moyen d'une permutation sur les axes de $\Re_{i_0}$, conservant l'orientation directe du repère (soit 24 permutations possibles).
  Chaque permutation possible est caractérisée par un triplet (a, b, c) dont les éléments prennent leur valeur dans $\{-x, x, -y, y, -z, z\}$. Le premier (resp. deuxième, troisième) élément du triplet indique quel axe du repère $\Re_i$ est superposable avec l'axe x_{i_0} (resp. y_{i_0}, z_{i_0}) de $\Re_{i_0}$.

Les figures suivantes montrent différents types d'objets créés au moyen de ces primitives.









A.2 Les structures articulées

A.2.1 Mode opératoire

Notre logiciel permet de modéliser les chaînes cinématiques ouvertes. Pour cela, l'utilisateur peut positionner deux corps C_i et C_{i+1} (précédemment créés) l'un par rapport à l'autre. Le mode opératoire utilisé comporte deux étapes:

1. La définition du type de liaison utilisé: prismatique, rotoïde, rigide (transformation constante entre les deux corps),...

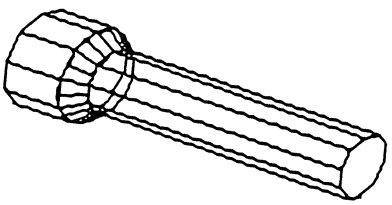
2. La description de la liaison: le positionnement relatif des deux corps est réalisé par une caractérisation de la position et de l'orientation du repère $\Re_{i+1}$ par rapport au repère $\Re_i$. Ce positionnement relatif permet d'initialiser les valeurs des quatre paramètres de Denavit-Hartenberg (cf. Annexe B) qui seront automatiquement mises à jour lorsque les valeurs des degrés de liberté changeront ou d'une matrice constante (pour une liaison rigide). Il est effectué de la même manière que le positionnement du repère d'un objet élémentaire (cf. §B.1.2).

A.2.2 Exemple

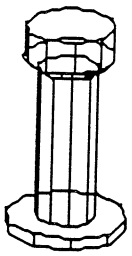
Nous reproduisons ici la trace d'exécution commentée de la modélisation d'un manipulateur à six degrés de liberté inspiré du robot PUMA d'UNIMATION. Cette trace a été produite grace au logiciel "Tek alike" de Mesa Graphics permettant d'émuler un terminal graphique Tektronics sur un Mac Intosh Apple. Nous avons utilisé la version FranzLisp/UNIX de SMGR implantée sur un DEC-VAX750.

- SOCLE
  Objet simple:
  - cylindre
    - * R=2.0 h=0.5
    - * nil nil

- PIED:
  Objet composé:
  - cylindre
    - * R=1.0 h=7.0
    - * (0.0 0.0 8.0) (z -x -y)
  - tronc de cône
    - * R=1.5 r=1.0 h=0.5
    - * (0.0 0.0 -0.5) (z x y)
  - cylindre



* R=1.5 h=1.5
* (0.0 0.0 0.5) (z -x -y)

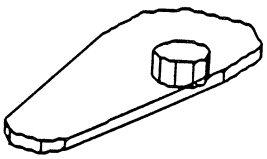


Degré de liberté:

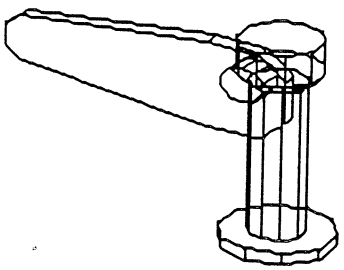
- rotoïde
- (0.0 0.0 8.5) (z -x -y)

- BRAS:
  Objet composé:

  - cylindre
    * R=0.8 h=1.0
    * (0.0 -7.5 1.5) (y -x z)



  - cylindre généralisé
    * origine[1] (1.0 2.0)
    * contour[2] (0.0 1.5 0.0 2.5 0.5 3.0 7.5 4.0 9.5 4.0 10.5 3.0 10.5 1.0 9.5 0.0 7.5 0.0 0.5 1.0)
    * d=1.5
    * nil (-y -x -z)



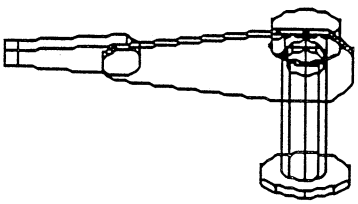
Degré de liberté:

- rotoïde
- (7.5 0.0 3.0) nil

- AVANT BRAS:
  Objet composé:

  - cylindre généralisé
    * origine (0.0 1.0)
    * contour (0.5 0.5 0.5 1.5 5.0 2.0 5.5 1.5 5.5 0.5 5.5 0.0)
    * d=1.0
    * (0.0 0.0 0.5) (-y -x -z)

  - parallèlépipède
    * L=1.0 l=1.0 h=0.5
    * nil (z y -x)



Degré de liberté:

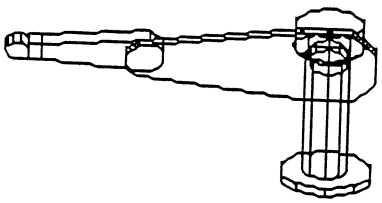
- rotoïde
- (5.0 0.0 -1.0) nil

[1] acquis à la tablette à digitaliser

[2] idem

- POIGNET:
  Objet simple:
  - cylindre
    * R=0.5 h=1.0
    * (0.0 0.0 0.5) (y z x)

  Degré de liberté:
  - rotoïde
  - (0.5 0.0 0.0) (z -y x)



- MAIN:
  Objet composé:
  - cylindre
    * R=0.1 h=0.6
    * (0.0 0.0 -0.6) (x -z y)
  - parallèlépipède
    * L=1.0 l=0.5 h=0.2
    * (0.0 0.0 -0.4) (x -z y)

  Degré de liberté:
  - rotoïde
  - (0.0 0.0 1.8) (x z -y)

- MORS 1:
  Objet simple:
  - cylindre généralisé
    * origine (0.0 0.0)
    * contour (0.0 0.0 0.0 0.1 0.5 0.2 0.5 0.0)
    * d=0.4
    * (0.0 0.0 0.2) (z y -x)

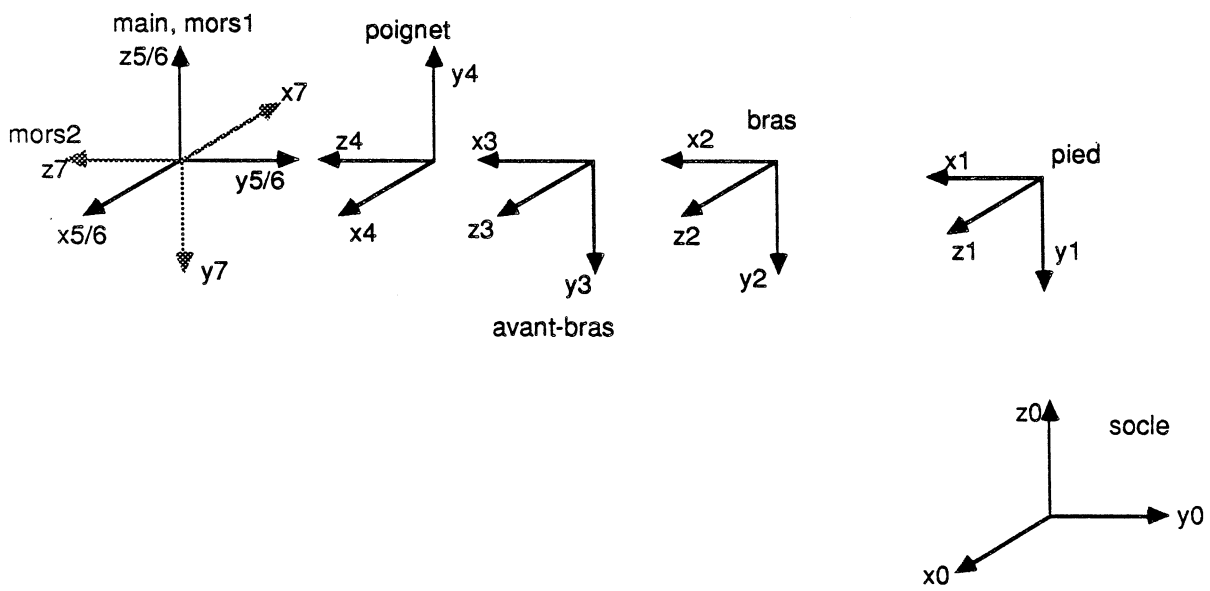
  Degré de liberté:
  - bi-prismatique
  - nil nil

- MORS 2:
  Objet simple: cf. MORS 1
  Degré de liberté:
  - bi-prismatique

```
- nil (-x z y)
```

Le modèle cinématique engendré est le suivant (les propriétés "liberté" et "composition" représentent respectivement la valeur des degrés de liberté et des matrices complémentaires (cf. Annexe B) pour chacun des élements de la structure articulée):

```
(putprop 'puma
         '((f_rotoide (4.71239 0.0 4.71239 8.5))
           (f_rotoide (0.0 7.5 0.0 3.0))
           (f_rotoide (0.0 5.0 0.0 -1.0))
           (f_rotoide (4.71239 0.5 0.0 0.0))
           (f_rotoide (4.71239 0.0 0.0 1.8))
           (f_biprismatique (0.0 0.0 0.0 0.0))
           (f_biprismatique (3.14159 0.0 1.57079 0.0)))
         'liberte)
(putprop 'puma
         '(nil nil nil
           (0.0 1.0 0.0 0.0 -1.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0)
           nil nil nil)
         'composition)
```



Annexe B

Description analytique des articulations dans SMGR

Considèrons une structure mécanique articulée possèdant n corps. La liaison entre le corps C_i et le corps C_{i+1} peut être caractérisé par une matrice de passage en coordonnées homogènes. Le problème est de trouver une définition simple des liaisons permettant d'extraire ces matrices. Nous considèrons pour cela uniquement le cas des liaisons rotoïdes et prismatiques qui sont les liaisons de base utilisées pour la construction des autres types de liaisons articulaires (cf. §3.1.2).

B.1 Paramètres de Denavit-Hartenberg

Considèrons deux corps C_i et C_{i+1} représentés respectivement par les repères $\Re_i$ et $\Re_{i+1}$. Les paramètres de Denavit-Hartenberg [Denavit 55] permettent de représenter une liaison entre les deux corps C_i et C_{i+1} par quatre valeurs: α_i, a_i, θ_i et r_i. La i^{eme} coordonnée généralisée q_i de la structure articulée s'identifiera à θ_i dans le cas d'une liaison rotoïde ou à r_i dans le cas d'une liaison prismatique.

La figure B.1 explicite la signification de ces quatre paramètres.

Ce formalisme permet d'exprimer la matrice de passage homogène $T_{i,i+1}$ de $\Re_i$ à $\Re_{i+1}$:

$$T_{i,i+1} = \begin{vmatrix} \cos\theta_i & -\sin\theta_i \star \cos\alpha_i & \sin\theta_i \star \sin\alpha_i & a_i \star \cos\theta_i \\ \sin\theta_i & \cos\theta_i \star \cos\alpha_i & -\cos\theta_i \star \sin\alpha_i & a_i \star \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & r_i \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Cette matrice permet d'exprimer la relation suivante:

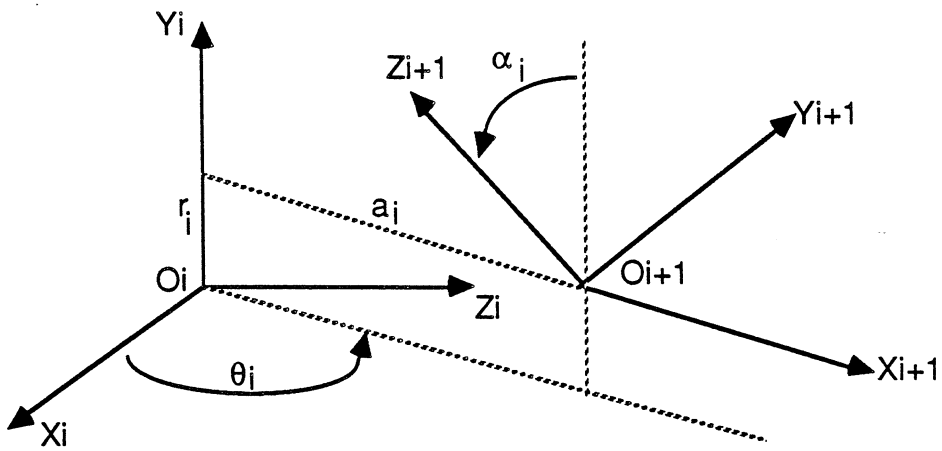


Figure B.1: Paramètres de Denavit-Hartenberg

$$\begin{vmatrix} x_i \\ y_i \\ z_i \\ 1 \end{vmatrix} = T_{i,i+1} * \begin{vmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+1} \\ 1 \end{vmatrix}$$

Il est clair que ce formalisme ne permet d'exprimer que des transformations d'axes z_i (rotation autour de cet axe ou translation le long de cet axe). Cette restriction pose des problèmes au niveau de la définition des repères associés aux différents corps du fait que le positionnement de l'axe z_i du corps C_i est conditionné par la liaison avec le corps C_{i+1}. Elle peut cependant être supprimée par l'adjonction de repères supplémentaires judicieusement choisis (cf. paragraphe suivant).

B.2 Représentation des liaisons entre corps

Soit P_i la projection du point O_{i+1}, origine de $\Re_{i+1}$, sur le plan $x_iO_iy_i$ de $\Re_i$ (cf. figure B.2). Les transformations permises (d'axe z_i) sont telles que x_{i+1} reste dans un plan parallèle à $x_iO_iy_i$ ce qui équivaut à dire que le vecteur $\overrightarrow{O_iP_i}$ est colinéaire à l'axe x_{i+1}. De telles transformations sont aisément exprimables grâce aux paramètres de Denavit-Hartenberg.

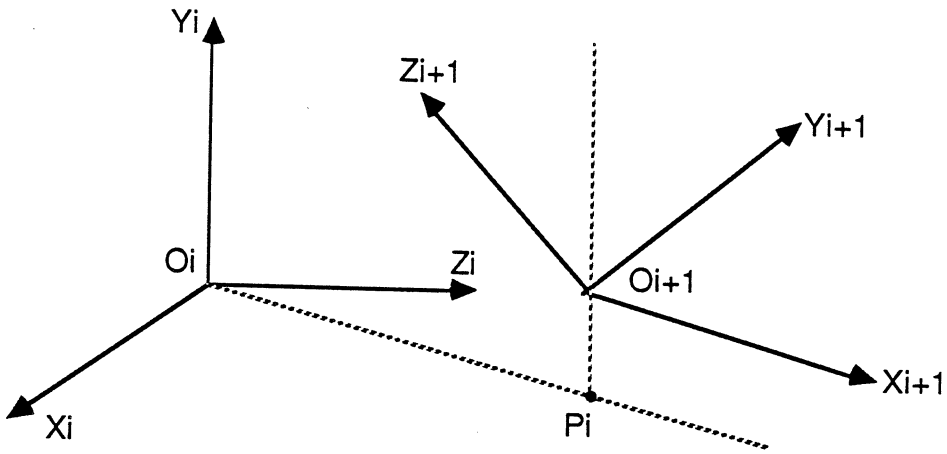


Figure B.2: Utilisation d'un repére intermédiaire (a)

Par contre, ce formalisme n'est pas utilisable directement pour les autres cas ($\overrightarrow{O_iP_i}$ colinéaire à y_{i+1} ou z_{i+1})[1]. La solution adoptée pour traiter les autres situations consiste à définir un repère intermédiaire $\Re'_i$ défini de telle façon que l'on puisse passer de $\Re_i$ à $\Re'_i$ et de $\Re'_{i+1}$ par des tranformations exprimables à l'aide des paramètres de Denavit-Hartenberg. Le choix de ce repère (cf. figure B.3) est dépendant du cas envisagé ($\overrightarrow{O_iP_i}$ colinéaire à y_{i+1} ou à z_{i+1}).

De cette façon, si:

$$\Re_i \overset{T_{i,i'}}{\Longrightarrow} \Re'_i \overset{T_{i',i+1}}{\Longrightarrow} \Re_{i+1}$$

alors:

$$T_{i,i+1} = T_{i,i'} * T_{i',i+1}$$

$T_{i,i'}$ et $T_{i',i+1}$ sont des matrices de passage en coordonnées homogènes; $T_{i,i'}$ est la matrice dans laquelle est injectée la i^{eme} coordonnée homogène et $T_{i',i+1}$ est une matrice constante.

L'introduction de ce repère intermédiare est réalisée automatiquement par le système lors de la définition de la liaison et est complètement transparente à l'utilisateur.

[1]nous imposons que le passage d'un repère à son suivant se fasse par des rotations de $k \star \frac{\pi}{2}$

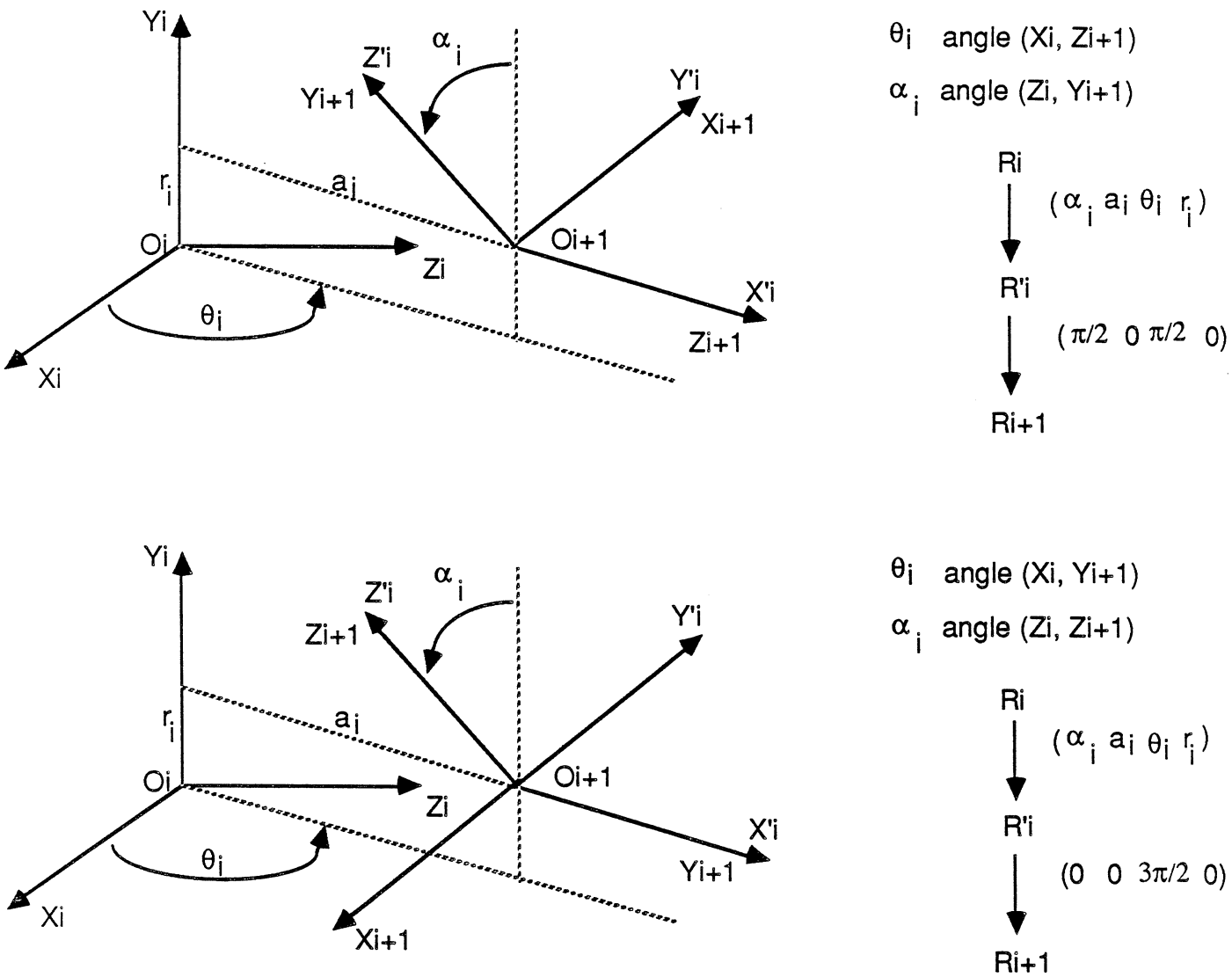


Figure B.3: Utilisation d'un repère intermédiaire (b)

Annexe C

Algorithme d'analyse d'accessibilité globale

Nous présentons ici deux versions de l'algorithme de calcul d'accessibilité globale.

L'algorithme 1 correspond au processus de calcul que nous avons décrit au chapitre 5.

L'algorithme 2 est une proposition de modification de celui-ci dans le cadre d'une implantation utilisant un capteur de vision tri-dimensionnelle telle que nous la discutons au chapitre 6.

Algorithme 1

choisir Y_P heuristiquement s'il n'est pas déjà déterminé par la prise considérée

répéter

choisir heuristiquement Z_P

déterminer l'ensemble des contraintes initiales de position imposées par la prise traitée

répéter

si cela n'a pas déjà été fait
alors
discrétiser l'espace balayé par la partie de l'outil considérée grâce à un ensemble de coupes par un plan

répéter

effectuer la propagation des contraintes de la position de l'outil à la tranche d'outil considérée

déterminer l'espace des configurations par grossissement des obstacles inversement à la tranche d'outil considérée

en déduire un nouvel ensemble de contraintes explicites de position pour la tranche d'outil par différence (au sens ensembliste)

effectuer la propagation des contraintes de position déterminées au pas précédent à la position de l'outil

jusqu'à avoir traité l'ensemble des tranches considéré ou n'avoir pas de positions valides

jusqu'à avoir traité l'ensemble des parties de l'outil ou n'avoir pas de solutions valides

jusqu'à avoir trouvé une ou plusieurs positions valides ou se trouver dans un cas d'échec

si c'est possible

alors choisir une position valide aussi près que possible du centre de gravité de l'objet, et en déduire une trajectoire pour l'outil

sinon choisir une autre prise et réitérer le processus

Algorithme 2

choisir Y_P heuristiquement s'il n'est pas déjà déterminé par la prise considérée

répéter

choisir heuristiquement Z_P

déterminer l'ensemble des contraintes initiales de position imposées par la prise traitée

discrétiser l'espace balayé par l'outil grâce à un ensemble de coupes par un plan laser

répéter

répéter

déterminer l'espace des configurations par grossissement des obstacles inversement à la tranche d'outil considérée

effectuer la propagation des contraintes implicites de position pour la tranche à la position de l'outil

calculer un nouvel ensemble de contraintes implicites de position pour l'outil par union (au sens ensembliste) avec l'ensemble des contraintes précédemment déterminées

jusqu'à avoir traiter l'ensemble des tranches considéré

jusqu'à avoir traiter l'ensemble des parties de l'outil

transformer les contraintes implicites de position de l'outil en contraintes explicites par différence avec l'ensemble de contraintes initiales

jusqu'à avoir trouver une ou plusieurs positions valides ou se trouver dans un cas d'échec

si c'est possible

alors choisir une position valide aussi près que possible du centre de gravité de l'objet, et en déduire une trajectoire pour l'outil

sinon choisir une autre prise et réitérer le processus

INPG département des études doctorales

AUTORISATION de SOUTENANCE

VU les dispositions de l'article 15 titre III de l'arrêté du 5 juillet 1984 relatif aux études doctorales

VU les rapports de présentation de Messieurs
. G. GIRALT, Directeur de recherche
. F. MARTINEZ, Professeur

Madame PERTIN Jocelyne épouse **TROCCAZ**

est autorisée à présenter une thèse en soutenance en vue de l'obtention du diplôme de DOCTEUR de L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, spécialité "Informatique".

Fait à Grenoble, le 25 février 1986

D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble

P.O. le Vice-Président,

Thèse de Doctorat de l'Institut National Polytechnique de Grenoble

Auteur: Jocelyne PERTIN épouse TROCCAZ

Etablissement: Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle

Titre: Modélisation du raisonnement géométrique pour la programmation des robots

Résumé: Cette thèse est relative à la programmation automatique des robots. La première partie porte sur l'analyse des besoins en modélisation géométrique pour la Robotique et la définition d'un modèle approprié (représentations multiples, univers dynamique, représentation de l'incertain). La seconde partie présente une méthode originale de raisonnement géométrique pour l'automatisation de la préhension dans le contexte d'une tâche d'assemblage. Finalement, nous proposons dans une troisième partie de la thèse, une architecture de système pour la programmation automatique des robots de manipulation.

Mots clés: Robotique, Intelligence Artificielle, modélisation géométrique, incertitudes en Robotique, raisonnement géométrique, préhension, espace des configurations, programmation automatique des robots.