



HAL
open science

Planification globale de trajectoires en milieux industriels : application à la maintenance et au démantèlement de centrales nucléaires

Erwin Schmitzberger

► **To cite this version:**

Erwin Schmitzberger. Planification globale de trajectoires en milieux industriels : application à la maintenance et au démantèlement de centrales nucléaires. Autre. Institut National Polytechnique de Lorraine, 2003. Français. NNT : 2003INPL081N . tel-01749714

HAL Id: tel-01749714

<https://hal.univ-lorraine.fr/tel-01749714>

Submitted on 29 Mar 2018

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.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

1360363600

03 INPL 081 N

(IN) 2003 SCHMITZBERGER E.



Ecole doctorale IAEM Lorraine
DFD Automatique et Production Automatisée
Institut National Polytechnique de Lorraine

THESE

Pour obtenir le grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Discipline : Automatique et Traitement du Signal

présentée
par
Erwin SCHMITZBERGER

Titre : Planification Globale de Trajectoires en Milieux Industriels : Application à la
Maintenance et au Démantèlement de Centrales Nucléaires

Directeur de thèse : Didier Wolf

JURY

Christian LAUGIER
Thierry SIMEON

Jean-Louis BOUCHET
Michel DUFAUT
René HUSSON
Noureddine ZERHOUNI

Service Commun de la Documentation
INPL
Nancy-Brabois

*« L'univers baignait alors dans l'ignorance,
et le savant le passait à la batée tel un prospecteur
accroupi au-dessus d'une rivière de montagne,
cherchant l'or de la connaissance parmi les
graviers de la déraison, le sable de l'incertitude et
les petits octopodes aquatiques poilus de la
superstition. »*

T. Pratchett.

Remerciements

Ce travail a été réalisé pendant plus de trois années passées à EDF R&D au sein du groupe T59 dans le cadre d'un contrat CIFRE avec le Centre de Recherche en Automatique de Nancy.

Mes premiers remerciements vont naturellement à Jean-Louis Bouchet, Michel Dufaut, René Husson, et Didier Wolf, mon directeur de thèse, qui m'ont permis de mener à bien ce travail en dégageant ses enjeux industriels et scientifiques.

Je remercie très sincèrement les membres de mon jury et en particulier mes rapporteurs Christian Laugier et Thierry Siméon d'avoir accepté de relire ce mémoire.

Je remercie les membres du projet européen MoLog, chercheurs du groupe RIA du LAAS de Toulouse, de l'université d'Utrecht et ingénieurs de la société CADCentre de Cambridge pour leur collaboration très enrichissante.

Enfin je n'oublie pas mes collègues de thèse du CRAN, du LAAS, de l'université Utrecht et les élèves ingénieurs passés par EDF R&D pour leur intérêt et leur soutien.

Mes derniers remerciements iront pour ma famille et ma future épouse Katia pour leur infaillible soutien.

Sommaire

Introduction générale	1
-----------------------	---

CHAPITRE I

Un enjeu pour la maintenance et le démantèlement de centrales nucléaires : la gestion intégrée
des simulations du mouvement

Identification des besoins fonctionnels d'EDF

Introduction	5
1. Contexte de la maintenance nucléaire : émergence de nouveaux besoins	6
1.1 Définitions des tâches et opérations	9
1.2 Etudes du planning des tâches	10
1.3 Etudes géométriques des tâches	11
1.4 Etudes dosimétriques des tâches	14
1.5 Prototypage géométrique d'une opération critique	17
1.6 Prototypage dynamique d'une opération critique	23
2. Vers un couplage des niveaux de modélisations	25
2.1 La disponibilité et la standardisation des données géométriques	25
2.2 L'intégration « ergonomique » des outils de modélisation cinématique et dynamique	27
Conclusion	31

CHAPITRE II

Caractérisation globale des contraintes géométriques s'exerçant sur la préparation d'un chantier nucléaire

Existence et limites algorithmiques d'une représentation finie des composantes connexes et classes d'homotopie

Introduction	33
1. Expressions implicites des contraintes géométriques	35
1.1 Définitions	35
1.2 Formulation des contraintes de non-collision et d'évitement de zones	39
1.3 Formulation des liaisons et déformations	40
2. Nature algébrique finie de l'espace des configurations d'un chantier	47
2.1. Une formulation canonique des contraintes	47
2.2 Les limites d'une décomposition semi-algébrique de l'espace des configurations libres	51
3. Planification de trajectoire et intérêt d'une capture des classes d'homotopie	56
3.1 Approches existantes	56
3.2 Intérêt d'une capture des classes d'homotopie	62
3.3 Existence d'une représentations des classes d'homotopie	65
3.4 Mise en œuvre d'une capture des classes d'homotopie	67
Conclusion	71

CHAPITRE III

Caractérisation des configurations libres d'une cellule au moyen d'un réseau probabiliste

Capture des composantes connexes et de leurs classes d'homotopie

Introduction	73
1 Détection de collisions et de distances	75
1.1 Détection de distance pour une configuration	75
1.1.1 Collision statique	75
1.1.2 Décompositions convexes hiérarchisées	78
1.2 Collision de courbes et surfaces paramétrées de configurations	83
1.2.1 Approches existantes pour les courbes	83
1.2.2 Détection de collision pour des arêtes et des triangles	84

2. Capture probabiliste des composantes connexes	88
2.1 Les méthodes à réseaux probabilistes	89
2.2 Une méthode probabiliste à recouvrement fini : MRP-visi	91
3 Capture probabiliste des classes d'homotopie	95
3.1 Les sous-graphes visibles depuis une configuration	95
3.2 Algorithme MRP_CH	96
3.3 Elimination de la redondance de la représentation du groupe d'homotopie	100
Conclusion	105

CHAPITRE IV

Mise en œuvre du démonstrateur MoLog

Validation des MRP et évolution de l'architecture logicielle

Introduction	107
1. Validation des méthodes à réseaux probabilistes pour des locaux industriels	108
1.1 Quelques exemples caractéristiques	109
1.1.1 Manipulation d'une turbo-pompe alimentaire	109
1.1.2 Changement d'un générateur de vapeur	111
1.2 Les premiers résultats	113
1.3 Evolution du détecteur de collision	115
2. Les évolutions de l'architecture du démonstrateur MoLog	118
2.1 Le cadre des tests des méthodes à réseaux probabilistes pour des environnements industriels	118
2.2 La gestion d'une série de problèmes standards de planification du mouvement	121
2.3 La gestion interactive des phases d'apprentissage, de recherche et d'optimisation du mouvement	123
2.4 La centralisation du calcul cinématique: Vers l'intégration d'un module de design et de calcul cinématique	125
3. Implémentation physique de l'architecture	127
Conclusion	130
Conclusion générale	133
Annexes 1-4	139
Références bibliographiques	161

Introduction générale

Cette thèse a pour objet la planification de trajectoires sans collisions d'engins articulés ou de robots dans un environnement géométriquement complexe : les centrales nucléaires d'EDF. La planification de trajectoires est un sujet récurrent en robotique ou en productique. L'intérêt du monde industriel s'est traduit chez les éditeurs de logiciels de CAO par la recherche d'algorithmes performants. Toutefois les moyens existants ne répondent pas aux attentes d'EDF qui cherche une solution intégrée, générique, à la mesure de la taille et de la complexité de ses installations et adaptée aux spécificités de son environnement technique et réglementaire.

L'enjeu est de mieux maîtriser la durée des interventions de maintenance. Quand la disponibilité ou la sûreté des sites de production risque d'être affectée, il est important de pouvoir évaluer la faisabilité d'interventions dans des zones exigües ou à l'intérieur des circuits, ou encore la faisabilité de déplacements dans des zones encombrées par les équipements d'intervention et par les dispositifs de radioprotection. A l'optimisation de la maintenance courante s'ajoute la mise au point de nouveaux chantiers, souvent très complexes, liés à la prolongation de la durée de vie des centrales nucléaires ou à l'anticipation de leur démantèlement. Certains de ces chantiers n'ont jamais été mis en œuvre notamment au niveau du changement de composants du circuit primaire ou d'intervention à l'intérieur de ceux-ci. C'est pourquoi on ne bénéficie pour leur réalisation d'aucun précédent et on ne peut s'appuyer pour leur organisation sur aucun retour d'expérience.

La détermination des trajectoires de objets déplacés ou démontés et des engins utilisés constitue l'un des problèmes majeurs à résoudre. De telles informations ont un impact important sur l'organisation globale d'un chantier et peuvent conduire :

- soit à modifier les infrastructures, à réviser les plans de stockage des équipements ou à adapter le placement des protections biologiques,
- soit à préparer des opérations à l'aide d'engins spéciaux comme les robots.

Le nombre potentiel d'évaluations nécessite le recours à des outils informatiques de gestion intégrée des contraintes géométriques s'exerçant sur un chantier, performants et interactifs.

Les services de R&D d'EDF mènent depuis plusieurs années des travaux dans ce domaine pour évaluer ou adapter les résultats de la recherche académique à ses problématiques propres, dont il faut retenir trois caractéristiques :

- le volume de données et la taille très importante des modèles CAO 3D des installations ;
- la variété des problèmes d'ingénierie
- les exigences de qualité sur les résultats.

A l'issue de travaux préliminaires ([FER96]), EDF a décidé d'approfondir le sujet en s'appuyant sur les techniques de réseaux probabilistes, les plus prometteuses.

Les progrès récents et conjoints des techniques de détection de collision (quelques synthèses à ce propos sont disponibles dans [LIN98,GEE99,JIM01]) et des méthodes de planification de trajectoire utilisant des réseaux probabilistes [OVE95,KAV96,AHU94] ont favorisé l'essor de techniques de plus en plus efficaces et génériques de planification de trajectoires qui couvrent une large classe de robots (robots manufacturier, robots mobiles, et plus récemment robots parallèles) et qui trouvent des applications en dehors du champ classique de la robotique comme l'étude d'assemblages de pièces, l'étude de mécanismes réactionnels en biochimie ou la génération de mouvement en réalité virtuelle. Ces applications laissaient augurer des perspectives intéressantes dans le contexte industriel mais il n'existe pas d'outil de planification de trajectoires directement utilisable pour la préparation d'un chantier de maintenance dans un environnement industriel.

Dans ce contexte industriel et scientifique, la division Recherche et Développement d'EDF, a participé au projet européen MoLog (Motion for Logistics) consacré à la planification de trajectoires au sein d'environnements industriels, dans le cadre duquel ce travail de thèse a été mené, au sein du groupe Simulation et Optimisation des Opérations de maintenance (T59). Notre travail, a été mené en lien étroit avec les partenaires du projet MoLog, à savoir le groupe Robotique et Intelligence Artificielle du LAAS-CNRS, l'Université d'Utrecht et l'entreprise Cadcenter, éditrice du logiciel PDMS utilisé pour gérer les maquettes numériques de centrales nucléaires.

Le premier chapitre de cette thèse analyse les besoins d'EDF, le second formalise les problèmes identifiés pour répondre à ces besoins, le troisième chapitre décrit les Méthodes à Réseaux Probabilistes (MRP) et propose de les compléter avec un nouvel algorithme MRP-CH et enfin le

quatrième chapitre décrit le travail effectué au sein du projet MoLog pour intégrer un outil de planification de trajectoire pour la préparation de chantiers et tester le comportement des MRPs dans ce contexte.

L'analyse du besoin, présentée dans le chapitre §I, a consisté tout d'abord à préciser les besoins d'EDF en termes d'intégration à ses outils informatiques d'ingénierie. Cette analyse a débouché sur la précision de différents niveaux de modélisation d'un chantier (§I.1). Ces niveaux correspondent à des outils logiciels distincts qui sont utilisés simultanément ou successivement lors de la préparation d'un chantier et souvent par des équipes d'ingénieries distinctes. Ces niveaux se complètent sans pouvoir se remplacer du fait des spécificités de chaque logiciel (§I.2). Les résultats des simulations effectuées à un niveau donné ont un impact sur les niveaux supérieurs. Nous avons donc formalisé les différentes interactions existant entre ces niveaux de modélisation (qui sont souvent à la source d'un ralentissement des procédures de simulation d'un chantier). Il ressort de manière générale que le fait de pouvoir générer ou optimiser automatiquement une trajectoire est une fonctionnalité transverse qui permet toujours un gain de temps quelque soit le niveau de modélisation et qui est souvent indispensable dans de nombreuses situations. En particulier, la simple recherche d'une trajectoire sans collision d'un objet ou d'un engin dans un environnement exigu conduit dans de nombreux cas à des situations non triviales ne pouvant pas être résolues même lorsque l'on dispose de fonctionnalités de définition rapide du mouvement et d'un détecteur de collision.

Pour analyser les situations évoquées à l'issue du chapitre précédent, nous présentons dans le chapitre §II quelques éléments de géométrie algébrique à partir d'exemples concrets tirés de situations rencontrées lors de la préparation d'un chantier (§II.1). Ce cadre formel est essentiel car :

- Il permet de mieux comprendre les difficultés algorithmiques auxquelles sont confrontés les outils de planification de trajectoires dans ces situations (§II.2). Ces difficultés sont inhérentes au problème et se traduisent pas une impossibilité pratique de représenter de manière exacte l'*espace des configurations libres*^{II.1.1} (1) des éléments mobiles du chantier.
- Il permet de mieux comprendre la diversité de nature de cet *espace des configurations*^{II.1.1} et en particulier de se ramener, quelque soit le niveau de modélisation géométrique, à un cas standard de planification de trajectoire dans une *cellule*^{II.3.2} (§II.3) de l'espace des configurations. Cette décomposition formelle

¹ Les *termes* ayant pour indice une référence de chapitre sont définis dans la partie correspondante, ceux ayant pour indice A1-A4 sont définis dans l'annexe correspondante.

présente l'avantage de ne pas restreindre l'ensemble des situations qui peuvent être traitées et n'empêche pas l'accès aux caractéristiques topologiques globales de l'espace des configurations libres, telles que ses composantes connexes et leurs classes d'homotopies que nous proposons d'utiliser lors de la recherche d'une trajectoire.

Le temps mis pour répondre à un problème de planification est susceptible de gêner fortement l'utilisateur. Pour apporter un solution à ce type de problème, nous proposons de réaliser un pré-calcul visant à caractériser globalement l'espace des configurations libres, ce qui nous paraît être la meilleure gestion possible du temps de calcul dans ce contexte. Cette approche permet, dans un second temps de générer plus rapidement des trajectoires suivant les exigences de l'utilisateur.

Nous abordons dans le chapitre §III la mise en œuvre algorithmique d'un pré-calcul. Pour ce faire, nous décrivons les techniques récentes de détections de collisions (§III.1) et les méthodes probabilistes de construction d'un réseau de trajectoires (§III.2) mais en se plaçant dans le cadre strict d'une cellule de configuration dont nous savons, à l'issue du chapitre précédent, qu'il permet de décomposer les situations complexes d'un chantier. Typiquement ce pré-calcul permet de représenter les *composantes connexes*^{A1} de l'espace des configurations libres au moyen d'un réseau de trajectoires. Néanmoins, dans certaines situations les trajectoires fournies par le réseau pré-calculé ne sont pas dans la bonne *classe d'homotopie*^{A1} par rapport à la trajectoire recherchée. Cette situation annule tout l'avantage du pré-calcul. **Nous proposons un nouvel algorithme MRP-CH dont l'objectif est de capturer les classes d'homotopie de l'ensemble des configurations libres d'une cellule (§III.3).**

Dans le chapitre §IV nous décrivons plus spécifiquement le travail d'intégration effectué dans le cadre du projet MoLog. (§IV.1). Ce travail a conduit à une validation expérimentale des Méthodes à Réseaux Probabilistes pour deux exemples critiques que nous avons choisis comme étant représentatifs des situations rencontrées durant la préparation d'un chantier. La complexité du problème de planification de trajectoire dans le contexte de la préparation d'un chantier a nécessité une réflexion approfondie sur l'architecture logicielle (§IV.2) permettant de garantir à l'utilisateur une bonne interactivité et une gestion souple des temps de calculs. L'implantation de cette architecture (§IV.3) a été réalisée par étapes en fonction de l'évolution des techniques et des interfaces des modules informatiques du projet MoLog.

CHAPITRE I

Un enjeu pour la maintenance et le démantèlement de centrales nucléaires : la gestion intégrée des simulations du mouvement

Identification des besoins fonctionnels d'EDF

La majeure partie des centres de production d'électricité et des services d'ingénierie d'EDF est impliquée dans la préparation des chantiers de maintenance de centrales électriques. Dans le cas des centrales nucléaires, la préparation de nouveaux chantiers de maintenance ou de démantèlement prend une place croissante qui nécessite de lourdes études sur l'organisation globale du chantier et requiert actuellement des études de faisabilité de nouveaux procédés. Nous consacrons la première partie de ce chapitre (§1.1) à une classification des types de modélisations d'un chantier et à l'apport potentiel d'outils de simulation du mouvement pour évaluer rapidement et précisément la faisabilité d'un nouveau chantier « simulé », voire pour en optimiser la sécurité, le coût et la durée.

Dans la deuxième partie de ce chapitre (§1.2), nous analyserons quelques problèmes caractéristiques rencontrés par les unités d'ingénieries et de recherche d'EDF pour référencer les conditions nécessaires pour que s'intègrent autour de la notion fédératrice de « maquette numérique d'un chantier », des outils de planification de trajectoires. Nous montrerons en particulier que les logiciels, la démarche et les contraintes du prototypage d'opérations en centrales nucléaires sont trop spécifiques pour bénéficier des outils actuels de génération automatique de mouvement et que ceux-ci doivent être adaptés et fortement intégrés pour que leur temps de calcul et leur ergonomie soit admissible dans un tel contexte d'utilisation.

1. Contexte de la maintenance nucléaire : émergence de nouveaux besoins

La maintenance des centrales nucléaires s'effectue généralement lors d'arrêts programmés. Dans le cas des centrales nucléaires, on parle d'un arrêt de tranche (*i.e.* arrêt d'un des bâtiments « réacteurs » d'une centrale à l'occasion du changement de combustible). La durée de l'arrêt a un impact direct sur la disponibilité d'une centrale pour la production d'électricité. L'optimisation de la durée d'un chantier est donc un enjeu stratégique pour EDF. Il convient en particulier d'identifier les *tâches*^{1.1.1} (1) d'un chantier et le chemin qui liste la chaîne de tâches nécessairement successives qui imposent la durée globale du chantier. C'est le chemin critique du chantier.

Si la plupart des chantiers programmés sont connus et bénéficient d'un riche retour d'expérience qui a permis de réduire leurs durées, une large proportion de centrales électriques et plus spécifiquement de centrales nucléaires arrivent à un stade de leur cycle de vie où certaines *opérations*^{1.1.1} de maintenance, de changement et de démantèlement de composants doivent être évaluées pour la première fois depuis leur construction. Ces **nouveaux chantiers** bénéficient de peu ou d'aucun retour d'expérience et l'évaluation de la faisabilité de chaque tâche, de sa dangerosité, de son coût, et de sa durée est extrêmement difficile et peut **invalidier toute tentative d'optimisation globale de l'organisation du chantier**.

EDF R&D a donc entamé une réflexion sur les méthodes et les moyens d'études de nouveaux chantiers visant, par le biais **d'une modélisation du chantier et l'utilisation de moyens de simulation**, à pallier au manque d'information issu du retour d'expérience.

Nous avons établi à partir de l'observation des études demandées par les unités d'ingénieries d'EDF au groupe T59 d'EDF R&D plusieurs types de modélisation qui correspondent chacun à la prise en compte de contraintes spécifiques. Nous proposons de distinguer cinq niveaux de modélisation correspondant chacun à un type d'étude permettant de répondre à l'évolution des besoins :

¹ Les *termes* ayant pour indice une référence de chapitre sont définis dans la partie correspondante, ceux ayant pour indice A1-A4 sont définis dans l'annexe correspondante.

Premier type de modélisation : Etudes du planning des tâches.

Ces études formalisent les tâches d'un chantier, les ressources associées (personnel, engins,...).et les contraintes logiques s'appliquant sur (ou entre) les tâches du chantier et relatives à la **disponibilité des ressources** Elles permettent alors, à partir des caractéristiques de chaque tâche (temps, coûts,...), d'optimiser l'organisation globale du chantier et d'évaluer sa durée et son coût.

Deuxième type de modélisation : Etudes géométriques des tâches.

Ces études utilisent des **modèles tridimensionnels** des **infrastructures** et des **objets manipulés** pour mettre en évidence des conflits géométriques sans *a priori* sur le personnel ou les engins de manutention alloués. La résolution de ces conflits conduit, par rapport à l'approche précédente, à créer de nouvelles tâches (par exemple pour modifier les infrastructures) et à introduire de nouvelles contraintes logiques entre les tâches (par exemple lorsque deux tâches ne peuvent s'effectuer simultanément dans un environnement donné), ce qui a un impact majeur sur l'évaluation globale de la faisabilité du chantier.

Troisième type de modélisation : Etudes dosimétriques des tâches.

Ces études permettent de définir et d'évaluer des **zones correspondant à des seuils de dangerosité** pour l'homme à partir de relevés et de simulation de rayonnement radioactif. Les outils de simulation permettent de définir les actions préventives nécessaires (décontamination, pose de protections, ...).

Quatrième type de modélisation : Prototypage géométrique d'une opération critique.

Pour certaines tâches critiques comme les manipulations de composants très volumineux, ou comme les interventions (usinage, soudage ou mesure) en environnement radioactif et / ou inaccessible, l'utilisation d'engin(s) spécialisé(s) dont les robots s'impose. La prise en compte du (ou des) modèles géométriques des engins permet de créer un prototype de l'opération associée à une telle tâche critique pour la valider en générant **des mouvements admissibles pour le ou les engin(s)** envisagés. Ceci permet de choisir, d'adapter, ou de placer au mieux les engins à utiliser.

Cinquième type de modélisation : Prototypage dynamique d'une opération critique.

La simulation des efforts (dus au poids, aux contacts statiques ou quasi-statiques ou si nécessaire aux phénomènes inertiels) permet de valider le dimensionnement des actionneurs d'un engin et le comportement de ses régulateurs lors d'un asservissement délicat (robot de soudage). Ce type d'étude est en particulier intimement lié au choix et à l'implantation du système de contrôle-commande d'un robot.

Les différents types d'études présentés ci-dessus doivent être inclus dans un processus intégré pour aboutir à une maîtrise des choix technologiques de la durée des opérations et de la sécurité des personnes des nouveaux chantiers. La figure (1.1) suivante présente de manière synthétique les différents types de modélisation et les enjeux associés à leur intégration.

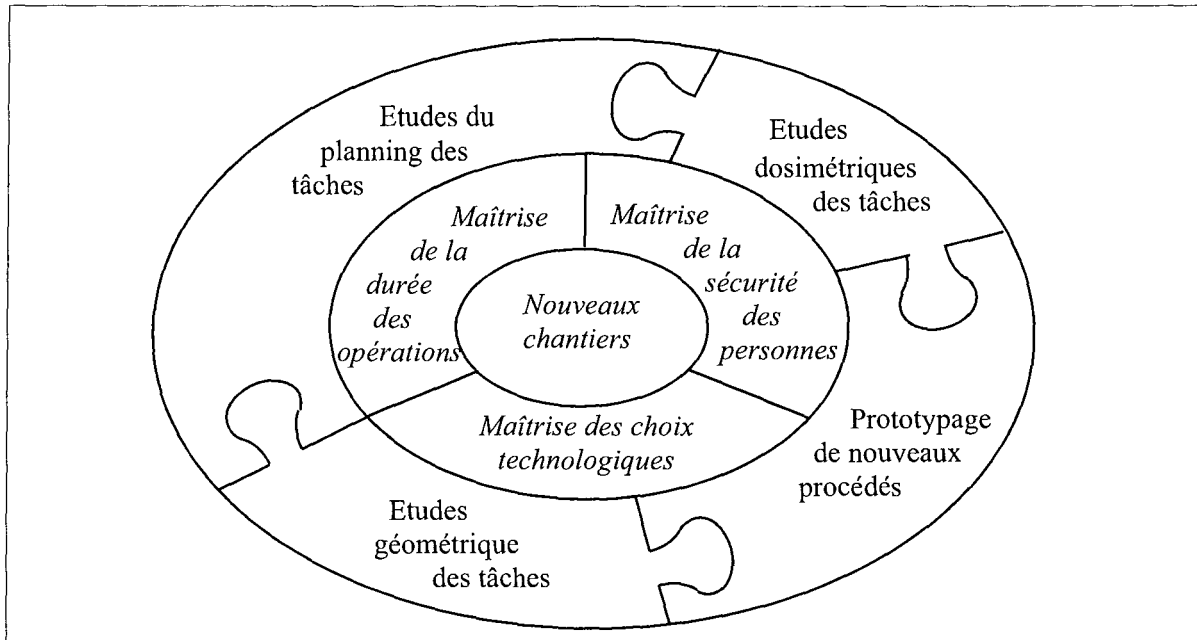


Figure 1.1: Apports potentiels du maquetage numérique pour la préparation de nouveaux chantiers de maintenance et de démantèlement

Nous allons détailler ces différents types de modélisation dans les paragraphes suivants en montrant d'une part l'opportunité de les **coupler** et d'autre part d'adapter ou, dans certains cas, de créer de nouveaux **outils de traitement des contraintes** pour évaluer de manière globale la préparation de nouveaux chantiers de maintenance ou de démantèlement.

1.1 Définitions des tâches et opérations

Pour structurer le déroulement d'un chantier, on peut caractériser un ensemble de **tâches** susceptibles de mobiliser des ressources (engins, personnel, équipements spécifiques,...) pendant leur exécution pour produire un résultat que nous appellerons une **opération** (opération de pose d'un échafaudage, opération de découpe, opération de colisage de déchets...).

Tâche	Ressources allouées
Identifiant	Personnel, engins mobilisés, temps d'exécution

Tableau 1.1 : Ressources allouées à chaque tâche

Les tâches, leurs ressources et leurs dépendances directes étant référencées, on peut définir la notion d'**état de chantier** comme la donnée, à un instant figé, de l'ensemble des paramètres qui permettrait « a un nouveau chef de chantier de reprendre la conduite du chantier et de la préparer jusqu'à son terme sans avoir suivi son déroulement ». Les tâches se redéfinissent alors comme l'ensemble des moyens permettant de modifier partiellement l'état du chantier.

En pratique, la définition d'un état de chantier peut être plus ou moins fine. Il s'agit dans un premier temps de référencer tout élément lié au chantier, de spécifier s'il est couramment extérieur ou intérieur au chantier et de lui associer des attributs généraux. Un nombre restreint de types de tâche peut être utilisé comme l'indique le tableau 1.2 ci-dessous (liste non exhaustive) :

Types de tâches	Action sur l'état du chantier
Introduction de l'objet A sur le chantier	$\emptyset \mapsto \text{objet A}$
Evacuation de l'objet A du chantier	$\text{objet A} \mapsto \emptyset$
Assemblage	$\text{objet A} + \text{objet B} \mapsto \text{objet AB}$
Désassemblage	$\text{objet AB} \mapsto \text{objet A} + \text{objet B}$

Tableau 1.2 : Exemple d'une structuration de l'état et des tâches d'un chantier

1.2 Etudes du planning des tâches

L'étude du planning des tâches correspond au référencement de toutes les tâches impossibles (ou possibles), ce qui permet de structurer le modèle. A titre d'exemple, une tâche d'évacuation d'un objet AB peut être impossible parce que celui-ci est trop volumineux, ce qui implique qu'il doit être préalablement désassemblé. De même certaines tâches de désassemblage peuvent être interdites, ce qui implique une contrainte sur l'ordre des tâches de désassemblage... Ce type de contrainte sera appelé par la suite **contrainte logique**.

Cette structuration de l'état et des tâches peut se poursuivre et prendre en compte la disponibilité des ressources allouées à chaque tâche, voire la modification des objets ou des ressources à l'issue d'une tâche. Cette structuration peut s'appuyer sur une méthode sous-jacente de formulation des contraintes (de type « *réseau de Pétri coloré* ») mais doit être adaptée aux besoins spécifiques du nucléaire.

Cependant, si l'utilisation de ce type de modèle permet de détecter quelques conflits sur l'organisation du chantier, il pourrait aussi donner lieu à des procédures d'**optimisation** qui visent à planifier le séquençement des tâches pour minimiser le temps global du chantier (et/ou le cumul dosimétrique individuel *cf.* 1.3). Les outils nécessaires d'optimisation combinatoire ne sont pas actuellement intégrés aux modèles. Cette situation s'explique par le fait que ce type d'approche suppose que les caractéristiques de chaque tâche soient connues (ressources identifiées, temps d'exécution, exposition dosimétrique prévue, coût,...) et n'interagissent entre elles que par le biais de la disponibilité de ressources communes et des contraintes logiques formulées.

Dans le cas de nouveaux chantiers ces hypothèses sont extrêmement difficiles à tenir et des approches complémentaires doivent être menées :

- pour déterminer quelles sont les tâches impossibles (d'assemblage / de désassemblage / d'évacuations)
- pour vérifier que des tâches « *a priori* indépendantes » n'interagissent pas compte tenu des limites d'encombrement global du chantier qui n'ont pas encore été prises en compte.

Cette approche complémentaire est présentée dans la partie suivante (§1.3) et correspond au diagramme (*Figure 1.2*) :

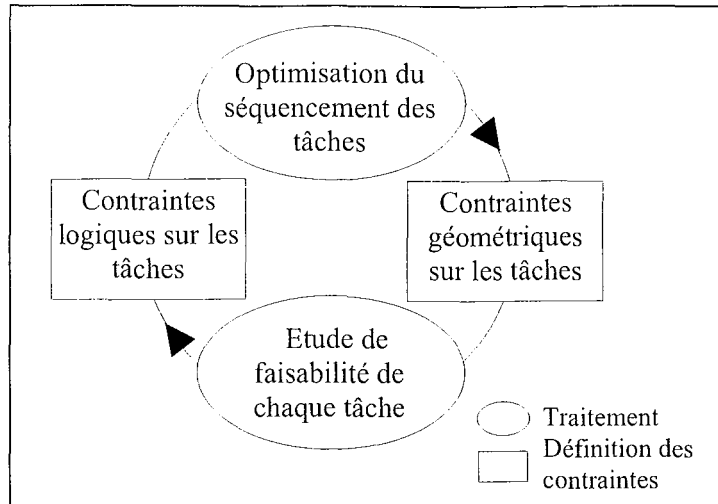


Figure 1.2 : Couplage du traitement des contraintes logiques et géométriques sur les tâches

La donnée d'un séquençement optimal des tâches vis-à-vis des contraintes logiques qui font l'objet des études de planning peut conduire à des conflits géométriques entre les tâches. L'optimisation globale du chantier ne peut être menée à bien rapidement que si les contraintes logiques et géométriques sur les tâches sont couplées.

1.3 Etudes géométriques des tâches

Des conflits de nature géométrique entre les tâches sont fréquents dans les locaux souvent confinés d'une centrale nucléaire. Ils se caractérisent par l'impossibilité d'effectuer une tâche du fait d'un environnement trop exigu ou trop encombré (par les autres objets). De nombreux chantiers ont été identifiés au cours du projet MoLog [NOT99] comme pouvant bénéficier rapidement d'outils génériques de gestion des contraintes géométriques sur les tâches. Nous citerons notamment les chantiers de :

- Décontamination du pressuriseur dans la planning d'un Remplacement des Générateurs de Vapeurs (RGV) lors de la deuxième Visite Décennale,
- Remplacement du couvercle de cuve (dans le planning d'un RGV),
- Démontage des Pompes Primaires (dans le planning d'un RGV),
- Colisage sur la dalle du Bâtiment Réacteur dans un arrêt de tranche simple et dépose des Groupes des MotoPompes Primaires,
- Manutention du Combustible (PMC) pendant un arrêt de tranche simple,

- Démontage et acheminement Echangeurs Economiseurs dans le Bâtiment des Auxiliaires Nucléaires (pour centrales type N4),
- Colisage tous travaux confondus du plancher machine (N4 ou 900 MW),
- Montage/Démontage et acheminement de la Turbo Pompe Alimentaire (TPA) en Salle des Machines N4 d'une centrale de type N4,
- Dépose de la volute de la Turbo Pompe Alimentaire (TPA) en Salle des Machines lors de la construction de l'une centrale de type N4.

EDF dispose aujourd'hui d'outils CAO 3D pour aborder de telles études. Les plus caractéristiques sont RGV_4D et PDMS :

- Le remplacement des générateurs de vapeur (RGV) d'une centrale nucléaire est un exemple caractéristique de nouveau chantier de maintenance, dont la préparation a nécessité le développement d'un logiciel spécifique (RGV4D) pour caractériser les plans d'occupation associés à de nombreux états du chantier (voir figure ci dessous).
- Le logiciel PDMS de la société CADcenter est majoritairement utilisé à EDF pour visualiser les centrales nucléaires de type N4. Cependant, il ne permet pas de gérer le mouvement d'objets solides (et *a fortiori* de systèmes articulés *cf* 1.4).

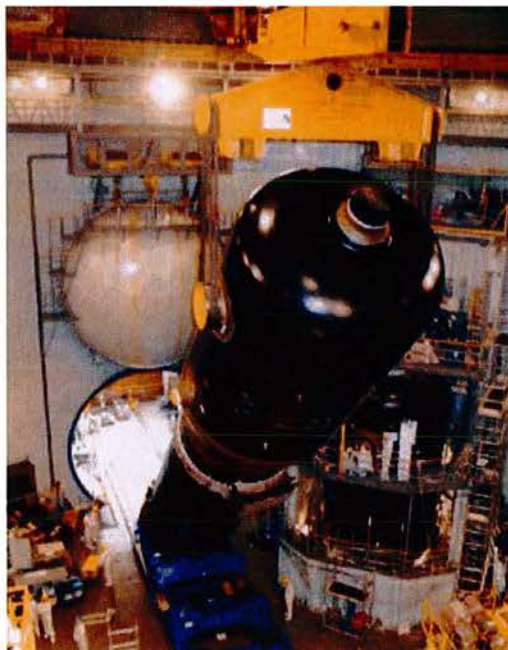


Figure 1.2 – Aperçu de la partie supérieure du bâtiment réacteur d'une centrale nucléaire pendant une opération de changement de générateur de vapeur

Dans PDMS, chaque étude de faisabilité des tâches de déplacements et d'assemblage d'objets doit être effectuée manuellement. Le déplacement d'un objet ne peut être testé que de manière discrète et requiert parfois un nombre important de positions intermédiaires sans garantir l'existence d'un mouvement continu sans collisions.

Dans ce contexte, l'intégration d'outils pour l'étude de faisabilité géométrique d'une tâche peut se révéler extrêmement précieux.

En effet, la détection rapide de tâches de déplacements impossibles permet de guider l'utilisateur dans le processus de création des tâches et de leurs actions. Si aucun mouvement (sans collision) ne peut être associé au déplacement d'un objet du fait d'un environnement statique trop exigü ou d'un agencement trop encombrant des autres objets, on doit créer une ou plusieurs tâches préalables pour dégager son parcours. Cela peut passer par le désassemblage et le déplacement de nouveaux objets et peut se traduire sur un chantier par des opérations lourdes d'ingénierie civile comme la découpe de tuyaux ou même de murs pour l'extraction d'objets volumineux et donc être préparé minutieusement. Le nombre potentiel d'études de faisabilité de tâches d'un chantier de démantèlement requiert un traitement informatique massif.

Une définition plus fine de l'état géométrique et des tâches du chantier est nécessaire pour traiter et détecter ces conflits. Une approximation assez réaliste consiste à considérer l'état géométrique du chantier comme la donnée de la position de tous les objets (potentiellement mobiles) de manière à ce qu'un **plan d'occupation 3D** puisse être associé à chaque état du chantier. Ces objets peuvent être des objets introduits au sein du chantier (équipements, colis,...) ou issus d'un démontage des infrastructures (pièces à changer,...). Ils se caractérisent par des attributs géométriques (Tableau 1.3)

Objets	Attributs
Identifiant	Forme et position dans l'espace

Tableau 1.3 : Attributs géométriques d'un objet

Cette approche n'introduit pas de contraintes géométriques relatives aux engins éventuellement alloués à la tâche mais permet de définir de nouvelles tâches (Tableau 1.4) et de mettre en évidence des conflits non triviaux sur les tâches de déplacement ou de modification des objets :

Tâches	Action sur l'état du chantier
Déplacement	objet A \mapsto objet A' (nouvel attribut position)
Modification	objet A \mapsto objet A' (nouvel attribut forme)

Tableau 1.4 : Types de tâches associés aux attributs géométriques

L'ensemble des attributs géométriques de tous les objets définit alors un état géométrique du chantier (approche retenue dans RGV_4D).

1.4 Etudes dosimétriques des tâches

La protection des intervenants contre l'irradiation et la contamination est une contrainte spécifique aux chantiers dans les centrales nucléaires. Ce type de contraintes peut se traduire de manière géométrique en introduisant la notion de **zone de travail** ou de zones dangereuses (plusieurs seuils) dans la maquette numérique du chantier.

Caractéristique de la tâche	Ressources allouées
Temps d'exécution, zone de travail	Personnel, engins mobilisés

Tableau 1.5 : Attributs d'une tâche

Sous l'impulsion de la démarche ALARA (« As Low As Reasonably Admissible ») des études préparatoires de radioprotection se mettent aujourd'hui en place à l'aide d'outils de simulation.

Le calcul du débit de dose dans un local à partir de sources radioactives identifiées et d'une modélisation géométrique de l'environnement (logiciel PANTHERE) , permet de définir des zones dangereuses et de les minimiser leur effet en aidant au choix des modes de protection appropriés. . A l'issue de ces études, les zones dangereuses identifiées peuvent être traitées comme des contraintes géométriques (zone interdite ou pénalisante).

La notion de **cumul dosimétrique individuel** ou cumul DED (Dose Equivalente Déduite) peut être introduite de manière explicite dans le modèle à partir d'une estimation d'exposition associée à chaque tâche. La prise en compte du cumul dosimétrique individuel durant la préparation d'un chantier permet d'éviter la démobilisation d'une partie du personnel du chantier lorsque leur quota réglementaire est atteint. C'est donc une contrainte essentielle qui doit faire partie intégrante du modèle des contraintes logiques du chantier et qui pourrait donner lieu à des procédures d'optimisation de l'affectation du personnel aux tâches en prenant comme critère le nombre de démobilisés (figure 1.3). **Mais le croisement entre les zones de travail des tâches, leur temps d'exécution, le personnel affecté et les zones dangereuses n'est pour l'instant pas pris en compte de manière directe.**

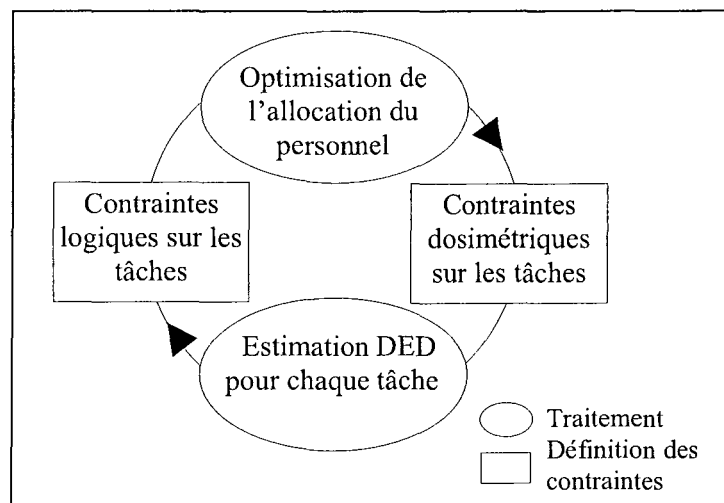


Figure 1.3 : Couplage des contraintes logiques et dosimétriques sur les tâches

De plus, les études dosimétriques visant à poser des protections biologiques (briques et matelas de plomb) ont un impact sur la géométrie du chantier et peuvent remettre en cause la faisabilité d'une ou de plusieurs tâches. Cet état de fait peut générer des situations difficiles. Si la prise en compte géométrique des protections gêne ou empêche le bon déroulement d'une ou de plusieurs tâches, ces étapes peuvent se répéter. Pour étudier l'impact géométrique des protections sur les tâches, les études devraient être couplées (figure 1.4).

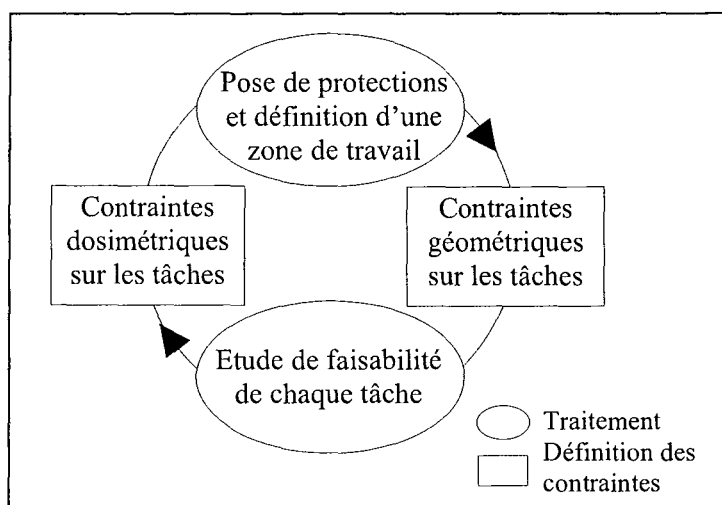


Figure 1.4 : Couplage des contraintes dosimétriques et géométriques sur les tâches

Lorsque aucune solution ne peut être trouvée (du fait de contraintes dosimétriques trop fortes), le recours à des procédés d'interventions téléopérées ou automatisées est nécessaire. La préparation de tels procédés introduit alors de nouvelles contraintes sur les tâches que nous abordons dans la section suivante.

1.5 Prototypage géométrique d'une opération critique

Au sein d'un chantier, les tâches les plus critiques concernent typiquement le changement de composants volumineux (générateurs de vapeurs, condenseurs, échangeurs EAS,...) ou les interventions (mesure, usinage, soudage,...) en milieux hostiles qui requièrent des procédés spécialisés (machine spéciales, robots téléopérés,...). La préparation de nouvelles tâches de ce type nécessite de lourdes études d'ingénieries pour le prototypage, l'évaluation et la comparaison de solutions techniques.

Opérations critiques	
Tâches	Ressources spécifiques allouées
Assemblage	Robot et outil de soudage, de vissage,..
Désassemblage	Robot et outil de découpe, de dévissage,..
Modification	Robot et outil de d'usinage, de mesure,..
<i>Déplacement à vide</i>	<i>Robot ou engin de manutention</i>
<i>Déplacement en charge</i>	<i>Robot ou engin de manutention</i>

Tableau 1.6 : Opérations définies par des ressources spécifiques associées aux tâches critiques

La notion de tâche est alors insuffisante car le mouvement des engins (et plus seulement celui des objets) doit être pris en compte de manière explicite. On doit alors recourir à une modélisation plus complète prenant en compte l'opération et non plus seulement la tâche.

Pour faire face à la multiplication des études liées au vieillissement des centrales tout en ayant le temps de comparer plusieurs options technologiques, il est important de réduire la durée des études.

Si la construction d'une maquette **réelle** des opérations critiques est possible, elle permettra d'optimiser puis de valider parfaitement les tâches les plus critiques. Mais elle est souvent extrêmement coûteuse et lente à mettre en place, tout en ne donnant qu'une approximation des contraintes d'environnement.

Le recours à une modélisation géométrique de l'opération peut répondre à ces contraintes sur la durée (et sur les coûts) des études préparatoires et permettre la comparaison et l'archivage pérenne des solutions évaluées.. Dans tous les cas, des problématiques similaires doivent être résolues avant la mise en place d'une maquette réelle : Comment choisir le robot ? Où placer le robot pour que la tâche puisse être effectuée ?

Nous allons présenter des exemples caractéristiques des besoins relatifs au prototypage d'opérations robotisées. Un robot redondant de type PA10 doit effectuer une opération de soudage suivant un cordon fixé à l'avance ou ressuer une surface détériorée de pale de turbine de centrale hydraulique.



Figure 1.5 : Opération robotisée de soudage sur le fond de la cuve réacteur



Figure 1.6 : Opération robotisée de ressuage pour la maintenance des augets d'une centrale hydraulique

Des exemples plus complexes de manipulations téléopérées ont été mis en place comme l'utilisation de deux bras de robots pour la découpe de déchets radioactifs (EDF/Groupe des laboratoires) ou l'utilisation d'un robot de type Titan pour effectuer une manipulation au sein de la partie inférieure d'un générateur de vapeur ou « bol GV » (EDF-R&D/MRI/T59).

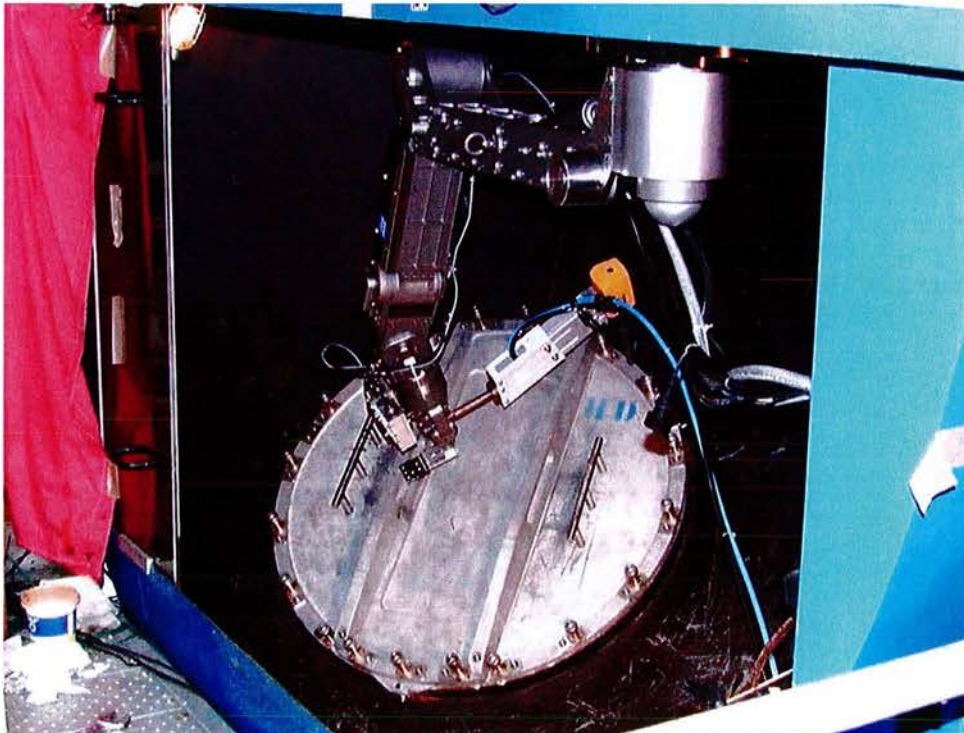


Figure 1.7 : Opération robotisée de manipulation d'une tige d'un générateur de vapeur de centrale nucléaire

La faisabilité d'une opération peut être liée :

- à la détermination de l'outillage ou du procédé approprié ;
- au placement fin de la base d'un robot dans son environnement (a),
- à la modification opportune d'un composant d'un engin ou d'un robot utilisé lors d'une nouvelle opération ((b) de la figure 1.8),
- au choix d'une position particulière de prise d'un outil ou d'un objet à manipuler par un robot (c),
- au choix d'une position particulière de pose d'un outil ou d'un objet à manipuler par un robot (d)

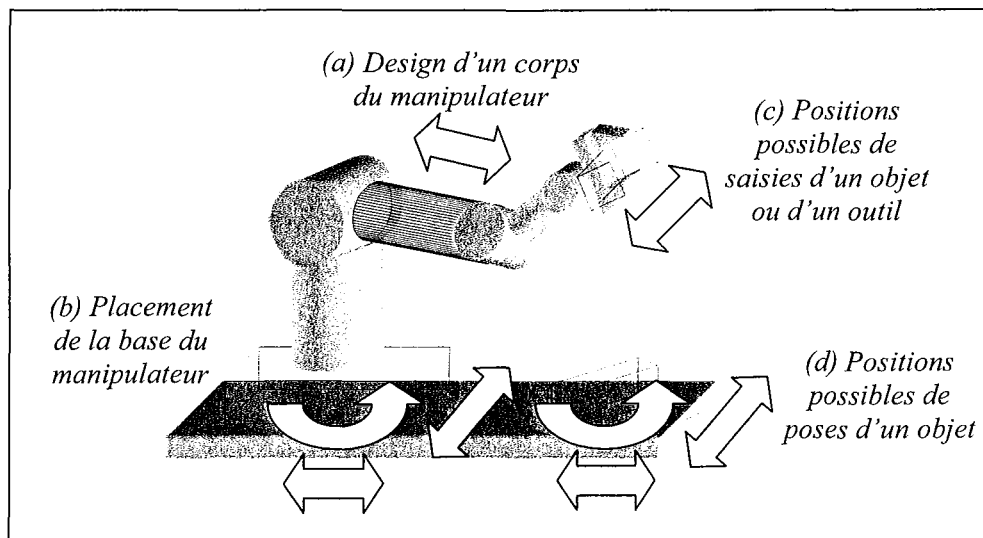


Figure 1.8 : Exemples de paramètres (« liaisons virtuelles ») à choisir pour le prototypage d'opérations

Notons que en dehors des opérations robotisées des contraintes géométriques complexes sur une opération peuvent apparaître. Si l'on revient à la tâche de changement d'un générateur de vapeur, évoquée dans la partie précédente, une opération critique a nécessité la coordination d'un pont roulant et d'un chariot roulant pour « coucher » le générateur de vapeur à remplacer.

Les logiciels de CAO 3D robotique (ROBCAD, ACT,...) offrent des outils de définition des contraintes géométriques (modèle géométrique direct ou inverse de mécanismes ou de robots, gestion des contacts,...) qui les rendent indispensables lors de la préparation d'opérations nécessitant un dimensionnement de machine spéciales ou l'utilisation de robots manipulateurs.

Cependant faute d'une exploration automatique des paramètres, la résolution de ces mêmes contraintes géométriques en jouant sur un paramétrage du design de l'engin (modification du porteur d'outil ou adaptation d'éléments mécaniques), de son implantation (base de l'engin dans son environnement) et de sa configuration articulaire reste un travail difficile, répétitif et non exhaustif lorsque l'on utilise les logiciels de CAO/CFAO actuellement disponibles. Par exemple, la fonction « Autoplace » de ROBCAD échantillonne en suivant une grille régulière différentes implantations de la base du robot en vérifiant à chaque fois la faisabilité de la tâche. Cette approche est très coûteuse en temps de calcul et elle est de plus non exhaustive (si la grille n'est pas assez fine ou si l'on souhaite jouer sur le design du robot) . Ces fonctionnalités concernent peu les clients traditionnels de la CFAO dont les besoins sont plus centrés sur le design de la pièce à traiter que celui de la machine ou du robot utilisé [WEN99]. De manière générale la préparation d'opérations critiques nécessite **l'intégration d'outils de prototypage d'une opération.**

La préparation d'une opération interagit sur l'ensemble du chantier. En effet, la mise en place d'une opération robotisée nécessite de nombreux équipements (armoires de commandes, poste de pilotage) et souvent la création de tâches préliminaires de modification des infrastructures et des tâches complémentaires comme l'introduction du robot dans son environnement. Ainsi l'opération PRODIP de nettoyage robotisé de l'intérieur d'un pressuriseur ou l'opération de pose de tape dans le bol d'un générateur de vapeur ont nécessité une étude pour l'introduction du robot à l'intérieur des structures.

Si l'on veut étudier l'impact d'une tâche robotisée sur les autres tâches les couplages suivants sont à considérer :

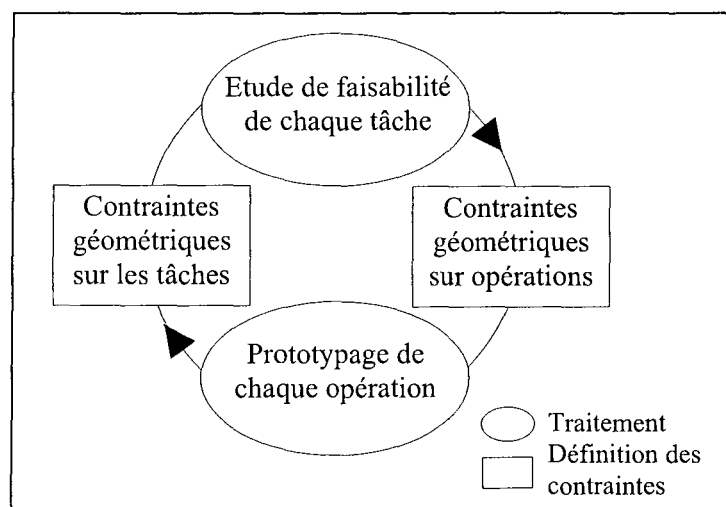


Figure 1.9 : Couplage des contraintes géométriques sur les tâches et les opérations

1.6 Prototypage dynamique d'une opération critique

Pour une opération critique mettant en œuvre un robot, un prototype réel de l'opération est souvent construit à l'issue de l'étude de faisabilité géométrique précédemment décrite, pour valider l'opération et le choix du robot avant de l'utiliser dans un environnement potentiellement hostile où le moindre problème technique peut avoir de lourdes conséquences. Pour des performances fixées (qualité de la soudure, de la découpe,...), ce prototype permet notamment d'évaluer la durée de l'opération. Ce temps est un attribut fondamental utilisé pour caractériser les tâches. Il a un impact sur le premier niveau de modélisation (*figure 1.10*). Il est directement lié aux trajectoires du robot.

La vérification de faisabilité sur la maquette physique utilise les trajectoires déterminées en CAO, mais il reste cependant à vérifier :

- que la trajectoire programmée du robot n'est pas dommageable pour les actionneurs du robot,
- que le système d'asservissement des actionneurs garantit le niveau escompté de performance en étant robuste vis-à-vis des perturbations dynamiques lorsque le modèle dynamique de l'engin n'est pas pris expressément en compte.

Il serait alors envisageable d'utiliser des outils de simulation physique intégrant le calcul des forces inertielles, des efforts de contacts et des efforts supportés par les actionneurs de l'engin (logiciels SDS ou ADAMS). Un raccourcissement des réglages à effectuer sur le prototype réel pourrait être escompté

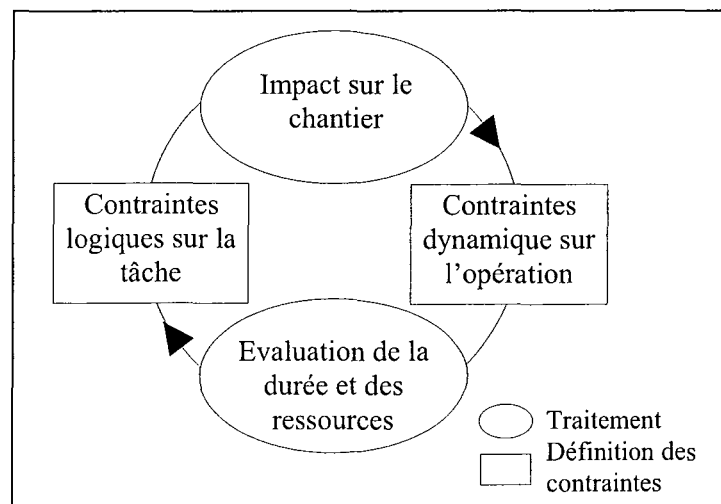


Figure 1.10 : Couplage des contraintes dynamique sur les opérations et logiques sur les tâches

Historiquement, les premiers robots industriels répétaient des mouvements appris. Puis, le mouvement, considéré comme une courbe paramétrée (en temps) dans l'espace des configurations du système (i.e. les coordonnées articulaires dans le cas d'un robot sériel de type bras manipulateur) a pu être obtenu en optimisant une ou plusieurs performances spécifiques souvent écrites sous forme d'une intégrale de coût à minimiser.

Des développements fructueux ont été effectués autour des principes variationnels directement inspirés de la mécanique classique de Lagrange ou d'Hamilton. Par exemple, la prise en compte de modèles dynamiques de robots a permis de générer des trajectoires offrant des compromis optimaux entre le temps d'exécution de l'opération et la minimisation des efforts supportés par les actionneurs du robot pour allonger leurs durées de vie.

En pratique, seule une version « locale » de ces principes, la commande optimale, fournit une méthode relativement générique qui, en s'appuyant sur des linéarisations exactes des systèmes mécaniques, déterminent des comportement réactifs optimaux. Mais les trajectoires analytiques ne sont pas disponibles pour planifier des mouvements globaux et savoir combien de temps dure le mouvement.

Cependant, compte tenu de l'amélioration des simulations des phénomènes mécaniques, le calcul formel peut être remplacé par un calcul *a posteriori* du critère à minimiser pour une trajectoire donnée et avoir recours à des techniques variationnelles simulées pour converger vers une trajectoire.

Au delà de la validation *a posteriori* des systèmes d'asservissements des actionneurs, ce type de simulation pourrait être utilisé directement pour la synthèse des régulateurs utilisés. Ce besoin se fait particulièrement ressentir lorsque des situations variées doivent être prises en charge par un même robot (succession d'une opération de soudage, d'un changement d'outil, d'un fraisage,...) et qu'elles conduisent à certains compromis sur les performances qui sont difficiles à formaliser sans recours aux simulations.

2. Vers un couplage des niveaux de modélisation

Les approches présentées dans les parties précédentes peuvent être vues globalement comme une démarche structurée de modélisation des différentes composantes d'un chantier. Les différents types de modélisation permettent d'ajuster le niveau de détail des contraintes (logiques, géométriques, dosimétriques ou dynamiques) au gré des besoins sur une échelle spatiale et temporelle de plus en plus réduite. Les résultats de chaque étude ont néanmoins un impact global sur l'organisation du chantier (*cf. figures de 1.1 à 1.9*) et les couplages d'outils logiciels sont nécessaires pour faciliter l'échange d'information technique.

Aujourd'hui, ces différents niveaux d'approches ont mis en œuvre avec de nombreux logiciels et par des équipes d'ingénieries distinctes, ce qui est particulièrement fastidieux.

Nous revenons dans ce paragraphe sur les points durs de cette intégration et nous décrivons les premières étapes pour les surmonter. Nous distinguons deux difficultés majeures qui sont :

- la disponibilité et la standardisation des données géométriques,
- l'intégration « ergonomique » des outils de modélisation et de planification du mouvement.

2.1 La disponibilité et la standardisation des données géométriques

La condition « *sine qua non* » pour préparer un chantier à l'aide de modèles géométriques est l'obtention des données caractérisant la forme et le positionnement des objets dans l'espace. Dans le cas des plans de centrales nucléaires, qui comprennent plusieurs centaines de milliers de *primitives géométriques* par bâtiment, ces conditions ne sont pas triviales et nécessitent l'utilisation de logiciels de modélisation CAO 3D disposant de fonctionnalités adaptées au métier (par exemple pour la génération d'une tuyauterie aux normes réglementaires,...).

Seule la dernière génération de centrales nucléaires (Palier technologique N4) a été conçue en CAO 3D à l'aide du logiciel PDMS. Dans ce cas, les données sont disponibles pour la préparation des opérations de maintenance.

Pour les centrales nucléaires plus anciennes (Paliers technologiques CP), il faut modéliser selon les besoins. La création de maquettes à partir des plans 2D existant n'est possible que pour des parties peu étendues de centrales. Il faut avoir recours à des procédés de type télémètre laser (procédé de mesure SOISIC assuré par EDF/CNEPE service topographie) dès que le nombre de primitives devient trop important, ou dès que la précision nécessite des données Telles Que Construites (TQC) de l'environnement, prenant en compte d'éventuelles modifications depuis la construction. Les mesures brutes issues de la télémétrie laser sont des nuages de points qui doivent être traités pour obtenir des primitives volumiques. L'utilisation de catalogues standards de composants permet alors de procéder à des identifications. Ces procédures demeurent assez complexes et le maquetage exhaustif du parc nucléaire n'est pas encore d'actualité. Or la majeure partie des opérations de maintenance et de démantèlement concerne ce type de centrales nucléaires. Il apparaît alors nécessaire de définir une échelle pertinente sur le niveau de détail des données des maquettes numériques en rapport avec l'étude, de manière à disposer rapidement des données.

Les études de simulation en CAO 3D réalisées aujourd'hui à EDF utilisent actuellement des logiciels distincts et partiellement redondants. SOISIC-ROBCAD-ACT-SW-PANTHERE disposent par exemple chacun d'un modèleur 3D, et ces différents modèleurs sont souvent incompatibles entre eux. Ces logiciels de CAO ne fournissent pas toujours de fichiers sous la même forme. En particulier, les objets peuvent être définis à l'aide d'une combinaison de primitives volumiques (« Constructive Solide Geometry ») ou à l'aide de leurs frontières (Boundary Representation ou B-Rep) comme les triangles ou les carreaux paramétriques. Aussi le transfert des données géométriques vers ces outils est une tâche laborieuse, qui pâtit souvent d'une perte d'information lors du transfert de données du fait par exemple d'une conversion des surfaces polynomiales en facettes.

Le format de PDMS est propriétaire dont la conversion pose des problèmes de format et d'organisation des objets géométriques.. Les infrastructures en béton, les tuyaux, les caillebotis sont par exemple regroupés en couches distinctes car ils concernent respectivement des métiers différents de la construction. Le premier travail du projet MoLog (cf. §IV), qui a impliqué tous les partenaires, a été de concevoir et de tester sur des exemples représentatifs des traducteurs pour passer à un format universel, compréhensible par les différents modules logiciels qui doivent traiter l'information géométrique et de définir un format pour les contraintes. Ce problème se retrouve de manière récurrente dès qu'un outil de traitement de la géométrie est utilisé et doit être pris en compte explicitement.

2.2 L'intégration « ergonomique » des outils de planification de mouvements

Les unités d'ingénierie d'EDF qui doivent préparer des prototypes virtuels pour des opérations de maintenance ou de démantèlement utilisent actuellement des logiciels distincts pour le design géométrique (CAD pour « Computer Aided Design ») et pour le design de la structure cinématique et des mouvements (CAM pour « Computer Aided Manufacturing »). Bien que les logiciels CAM offrent souvent des fonctions pour le design géométrique, ils ne sont pas conçus pour la représentation de grosses structures industrielles telles que les plans de centrales nucléaires. Les deux types de logiciels CAD & CAM doivent donc être utilisés dès que la modélisation d'engins articulés (robots, grues,...) doit être utilisée pour vérifier certains mouvements critiques.

De plus les logiciels CAM doivent bénéficier d'outils de génération automatique de mouvement (MP pour « Motion Planning ») lorsqu'il n'existe pas de solution directe. Les recherches en planification de mouvements ces vingt dernières années ont été assez abondantes mais leurs résultats, faute d'un besoin industriel massif, n'ont été que partiellement intégrés par les éditeurs de logiciels de CAD / CAM. EDF R&D a donc régulièrement contribué à l'intégration de nouvelles techniques algorithmiques [FER96] au sein de logiciels de type CAM (ACT, ROBCAD,..) qui ont permis de tester le choix et l'implantation de nouveaux procédés robotisés (Pose de tape) ou d'opérations complexes de déplacement de composants volumineux (Echangeur EAS). Le développement d'outils de planification de mouvements capables de traiter d'environnements complexes a constitué le cœur du projet MoLog et les tests des méthodes à réseaux probabilistes se sont révélés concluants.

Cependant, l'usage strict d'outils de planification de mouvements permet de tester rapidement la faisabilité d'un mouvement, mais ne fournit pas les moyens pour rectifier facilement le design géométrique (CAD) ou cinématique (CAM) pour rendre un mouvement faisable. C'est pourquoi nous devons focaliser notre attention sur de tels niveaux fonctionnels. Le design géométrique et cinématique habituellement fourni par les logiciels CAD et CAM doit être réalisé en interaction avec l'utilisation simultanée d'outils de planification de mouvements, et ce de manière à procéder à des modifications permettant de rendre un mouvement critique faisable et sûr.

L'intégration des outils de génération et de résolution des contraintes associés aux niveaux de modélisation décrits ci-dessous permettra d'une part de remonter l'information nécessaire au niveau supérieur si celle-ci n'est pas disponible (ce qui est le cas lors de nouveaux chantiers) jusqu'à obtenir

des critères globaux d'évaluation de l'ensemble du chantier, et d'autre part de conserver les données de chantier sous une forme unique et cohérente.

Cette intégration passe par la mise à disposition d'une unique interface utilisateur (UI pour « User Interface »), d'une unique base de donnée et par le choix d'un format standard des données et fonctionnalités offertes par les outils (API pour « Application Programmable Interfaces »). Ceci requiert un important travail de **formalisation et de développement informatique** notamment au niveau de la traduction de formats d'échanges des données et des contraintes (logiques et géométriques) avant de considérer des outils de résolution automatique de ces mêmes contraintes. Cette approche intégrée peut se traduire par le diagramme suivant visant à évaluer la faisabilité et à optimiser un nouveau chantier :

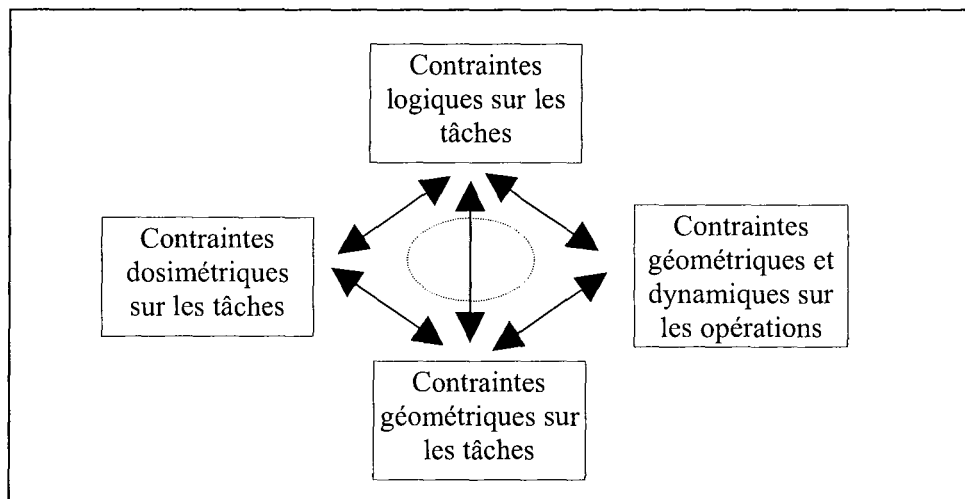


Figure 1.11 : Définition d'une interface adaptée au métier de l'ingénierie nucléaire

Nous avons distingué quatre niveaux fonctionnels dans le rapport MoLog E2.1. intitulé «*Testbed definition*» et nous avons affiné ce travail dans le document E2.3 intitulé «*MoLog Software Final Evaluation*» [SCH00a,SCH00b]. Chaque niveau fonctionnel est moins dépendant d'une diminution du temps de calcul des algorithmes qu'à des avancées en terme d'intégration et d'utilisation des nouvelles fonctionnalités du logiciel MoLog, qui ont été développées au cours du projet.

Le niveau 1 (figure 1.12) permet de réaliser des **tests de faisabilité de mouvement**. Il introduit la manière la plus simple d'utiliser les outils de planification de mouvement de MoLog (noté MP sur les figures suivantes). Si un logiciel CAD est utilisé pour effectuer des modifications algorithmiques du

design de la scène statique ou du design d'objets mobiles, chacune de ces modifications est susceptible de déboucher sur des procédures laborieuses pour pouvoir modifier les données dans les fichiers ou pour les traduire.

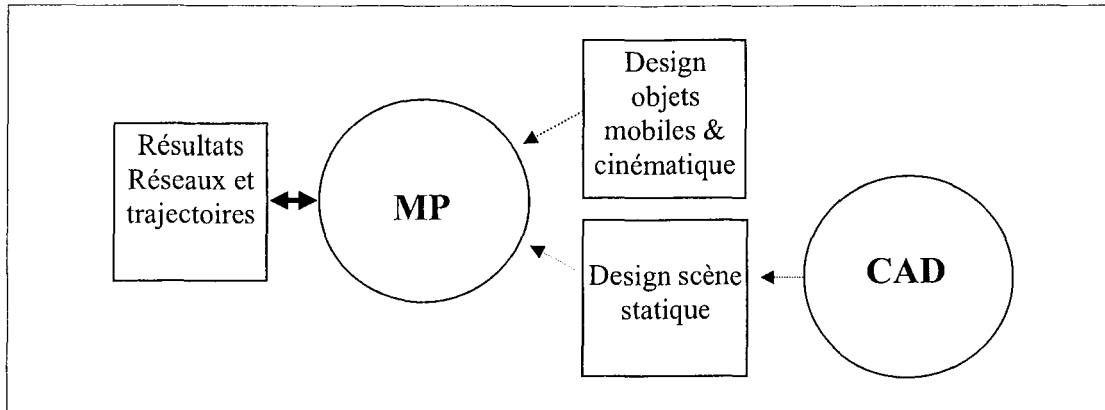


Figure 1.12 : Architecture des Tests de faisabilité de mouvements

Le deuxième niveau (figure 1.13) permet de réaliser plus facilement des **études d'impact géométrique**. Ce niveau nécessite la création dans le système CAD d'une banque de données qui puisse représenter les différents états de la scène statique, de façon à stocker et gérer chaque modification géométrique et à évaluer facilement ses impacts sur la faisabilité du mouvement étudié.

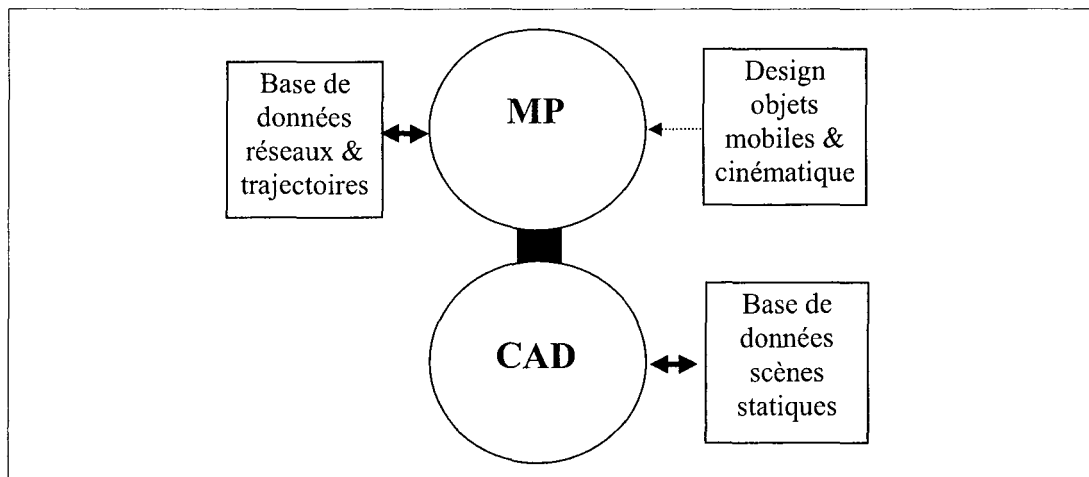


Figure 1.13 : Architecture pour les études d'impact géométrique

Le troisième niveau (figure 1.14) permet de réaliser les **tests de faisabilité d'une manipulation**. Par exemple, il permet de vérifier si une opération de saisie et de pose d'un objet par un robot «pick and place» est faisable. Néanmoins, les spécifications des modèles géométriques à considérer peuvent

constituer un travail laborieux jusqu'à ce que l'ensemble prédéfinis de saisies et de poses soit suffisamment important pour trouver une solution.

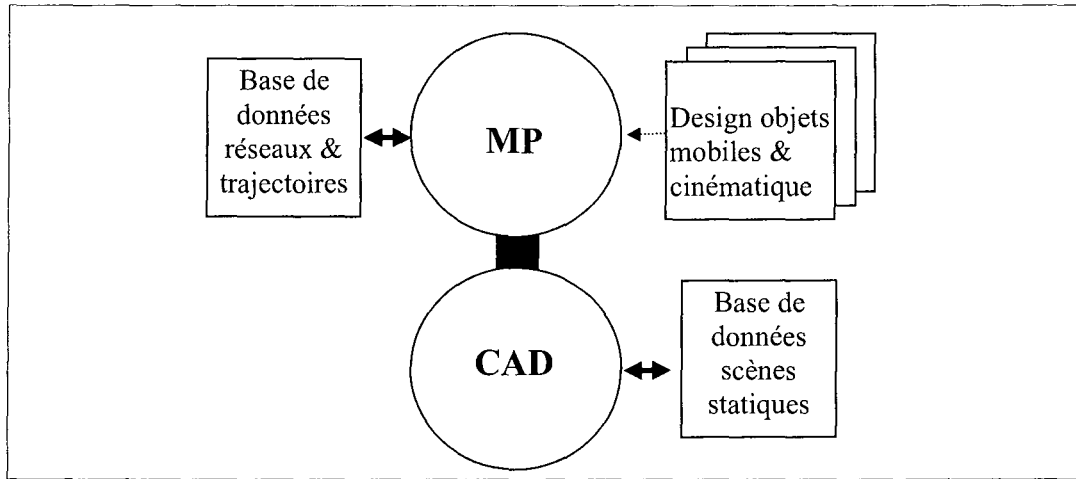


Figure 1.14 : Architecture pour les tests de faisabilité manuelle

Le quatrième niveau permet l'étude de l'impact du choix des liaisons virtuelles. Il utilise des fonctions CAM de design cinématique qui sont en dehors du cadre des objectifs du projet MoLog mais dont l'intégration procurerait un gain essentiel d'interactivité .

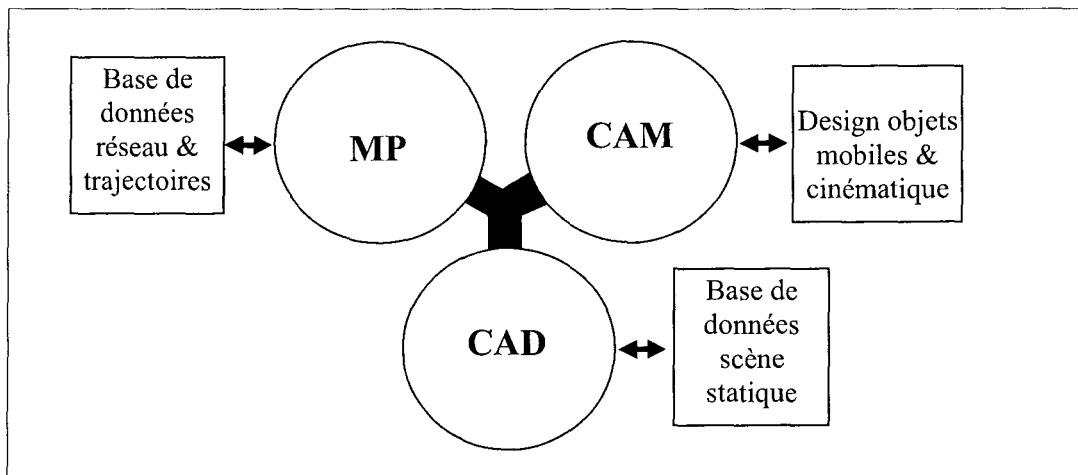


Figure 1.15 :: Architecture pour étude d'impact du choix des liaisons virtuelles

Les liens noirs de ce diagramme (figure 1.15) correspondent à des tâches d'intégration qui seront détaillées dans le chapitre §IV consacré à la mise en œuvre pratique et à l'évaluation d'une intégration d'outils de planification de trajectoires.

Conclusion

Les services d'ingénierie d'EDF et EDF R&D ont, entre autres, pour mission de fournir un appui aux directions opérationnelles dans le management des risques liés à l'exploitation, à la maintenance et au démantèlement des installations nucléaires. Ils contribuent à la maîtrise des coûts et de la sécurité via le développement de nouveaux procédés d'intervention.. Cette démarche ouvre la voie à la mise au point d' outils d'ingénierie plus performants que ceux qui sont aujourd'hui mis en œuvre.

Dans ce contexte, ce premier chapitre a permis d'identifier et de formaliser les besoins et plus particulièrement de préciser les apports potentiels d'outils de planification du mouvement. Ces besoins spécifiques à la préparation de chantiers nucléaires ne seront *a priori* pas disponibles sur le marché des logiciels de simulations sans une participation active d'EDF R&D. Nous avons précisé dans ce chapitre les étapes essentielles d'intégration logicielle des fonctionnalités que doivent remplir les outils de planification du mouvement.

Nous avons identifié cinq niveaux de modélisations nécessaire pour préparer un chantier il s'agit de :

- l'étude du planning des tâches,
- l'étude géométriques des tâches,
- l'étude dosimétrique des tâches,
- le prototypage géométrique d'une opération critique,
- le prototypage dynamique d'une opération critique.

Nous avons montré qu'il était nécessaire de les coupler et nous avons dégagé deux types de difficultés rencontrées pour ce faire:

- la disponibilité et la standardisation des données géométriques permettant d'étendre la base de données géométriques sur les centrales de production d'électricité,
- l'intégration des outils de modélisation du mouvement (CAM) et de planification automatique du mouvement (MP) dans un logiciel de design de bâtiments industriels (CAD) gérant la base de donnée géométrique des centrales électriques.

Dans la suite de ce mémoire , nous nous attacherons à traiter ce dernier point en considérant, dans les deux prochains chapitres, les adaptations algorithmiques de la planification du mouvement pour la préparation de chantiers nucléaires et, dans le dernier chapitre, les adaptations informatiques nécessaire à une intégration de tels outils.

CHAPITRE II

Caractérisation globale des contraintes géométriques s'exerçant sur la préparation d'un chantier nucléaire

Limites algorithmiques de la planification de trajectoire et intérêt d'une détection de l'homotopie de trajectoires

Dans ce chapitre, nous nous appuyerons tout d'abord sur quelques exemples caractéristiques pour formaliser les contraintes géométriques à l'aide d'une expression structurée notée G (§II.1). L'ensemble X_G des solutions à ces contraintes définit l'*espace des configurations libres*^{II.1.1} de manière suffisamment générale pour couvrir les opérations les plus complexes d'un chantier nucléaire comme celles incluant les coopérations ou les coordinations de plusieurs engins.

Le résultat fondamental que nous rappelons, est que, si une construction exacte de X_G est accessible *a priori* à un traitement informatique, elle est impossible à mettre en œuvre en pratique dans le cas d'une modélisation d'un chantier nucléaire (§II.2.1 & §II.2.2). Nous justifierons alors la décomposition d'espaces des configurations complexes en espaces standards de type $SO(d)^n \times [0,1]^m$ qui peuvent être paramétrés et sur lesquels on vérifie *a posteriori* et *localement* des contraintes géométriques complémentaires comme la détection de collision ou la fermeture de boucles de liaisons entre les objets (§II.2.3).

Une discussion pourra alors être menée sur les approches existantes de planification de trajectoire (§II.3.1) et leurs limites pour la préparation de chantiers. Une approche complémentaire

sera proposée (§II.3.2). Cette approche utilise la notion de pré-calcul global d'un réseau capturant les classes d'homotopie de l'espace des configurations libres. Cette notion n'a pas directement été exploitée, à notre connaissance, dans les algorithmes de planification de trajectoire. Elle permet d'avoir une vision plus exhaustive de l'ensemble des mouvements ce qui est particulièrement adapté aux préparations d'opérations nucléaires qui doivent envisager plusieurs solutions, parfois très différentes. Nous précisons les résultats présentés sur la nature de X_G pour montrer qu'une telle approche est possible et qu'elle est décomposable en utilisant la notion de recollement de cellules de configurations (§II.3.3).

1. Expressions implicites des contraintes géométriques

Nous présentons dans cette partie les contraintes géométriques sous une forme implicite sans les résoudre de manière explicite, c'est-à-dire sans chercher une paramétrisation explicite de l'ensemble des solutions. En effet, ce faisant, nous pouvons dégager une formulation commune couvrant tout le spectre nécessaire des contraintes que nous pouvons rencontrer comme :

- l'interdiction de contacts entre deux solides,
- les liaisons mécaniques usuelles,
- le maintien de contacts complexes de surfaces non planes entre deux solides (c'est le cas par exemple lors d'un usinage complexe),
- la déformation d'un objet.

Cette formulation nous permettra dans la partie suivante (§II.2) de rappeler la nature des solutions des contraintes géométriques présentées. L'utilisation de connecteurs booléens ET (\wedge), OU (\vee) et NON (\neg) associés aux expressions implicites des contraintes, nous permettra de formaliser les différentes interactions de plusieurs solides sans être cantonné à un unique graphe de liaisons. Cette situation apparaît par exemple, lorsque l'on autorise l'occurrence de contacts durant une opération. C'est le cas en particulier lors d'une opération de manipulation d'un ou plusieurs objets saisis (ou poussés) par un ou plusieurs engins.

1.1 Définitions

Nous allons définir les contraintes géométriques $G(x)$. Elles s'exercent sur le positionnement d'un nombre fini d'objets dans l'espace et expriment le fait que ces objets ne peuvent entrer en collision les uns avec les autres, et qu'ils sont soumis à des liaisons de contact, permanents ou

temporaires, voire à des articulations entre les corps rigides des engins ou des robots dans le cas de la modélisation géométrique d'une opération.

Définition 2.1 : Etat géométrique. L'espace des états géométriques X est un ensemble *suffisant* (que l'on supposera fini) de variables réelles qui permet de placer tous les objets dans l'espace et de déterminer leur formes si celles ci peuvent évoluer.

La définition d'un repère quelconque dans l'espace \mathbf{R}^3 nécessite les trois coordonnées spatiales de quatre points distincts (l'origine du repère et 3 points non coplanaires). La donnée de ce repère permet de placer un solide dans l'espace. Ainsi, le placement de N solides peut toujours être effectué à l'aide de $d=12.N$ variables réelles regroupées en un vecteur $x \in X = \mathbf{R}^d$. Cette définition n'est pas minimale (6 variables permettent déjà de placer un solide dans l'espace) mais elle a le mérite de ne pas introduire trop tôt un choix particulier de paramétrisation. Elle peut de plus se généraliser aux objets déformables (câbles, ombilics d'alimentation d'un robot,...) qui peuvent nécessiter plus de quatre points distincts pour être placés dans l'espace avec une forme donnée.

En pratique, dans le cas de la préparation d'un chantier, la définition de l'espace X peut difficilement se faire *a priori*. Elle est susceptible d'évoluer au cours des études géométriques. Si l'on s'aperçoit qu'un nouvel objet est nécessaire (issu d'une découpe ou introduit dans le chantier), il faut considérer le nouvel espace $X \leftarrow X \otimes X_{objet}$ où X_{objet} permet de placer ce nouvel objet (\otimes désigne le produit cartésien d'ensembles). En fait, l'espace géométrique est (re)défini de telle manière que toutes les contraintes géométriques puissent s'exprimer sur x et seulement sur x .

Définition 2.2 : Contraintes géométriques $G(x)$. Nous appelons contraintes géométriques $G(x)$ l'ensemble des équations ou inéquations de variable $x \in X$ définissant les interactions entre les différents objets dont le placement est défini par $x \in X$.

Nous pouvons néanmoins évoquer dès à présent une sous classe de contraintes. Les *contraintes algébriques strictes* notées $g(x)$ comprennent les contraintes de rigidité ou les lois de déformations, les liaisons permanentes entre les solides comme les articulations d'un robot, les liaisons temporaires comme les contacts résultant d'un usinage ou d'une saisie d'un objet. Ces contraintes définissent l'espace des configurations.

Définition 2.3 : Espace des configurations. Les solutions des contraintes algébriques strictes notées $g(x)$ définissent un sous-ensemble de $X_g \subset X = \mathbf{R}^d$ appelé espace des configurations.

Les contraintes semi-algébriques strictes comprennent principalement les contraintes de non collisions des surfaces des solides mais peuvent s'étendre à d'autres contraintes comme par exemple l'interdiction pour un objet de pénétrer dans une zone dangereuse. La prise en compte simultanée de ces deux types de contraintes définit $G(x)$.

Définition 2.4 : Espace des configurations libres. Les solutions de toutes les contraintes X_G définissent un sous-ensemble de $X = \mathbf{R}^d$ appelé espace des configurations libres.

La définition des formes géométriques utilisée dans la plupart des logiciels actuels de modelage est une combinaison booléenne d'inéquations de type $\varphi = [P_i(px, py, pz) > 0]$ où les $P_i(px, py, pz)$ sont des polynômes à coefficients réels (sphères, cylindres,...) et les $[px, py, pz]^T$ sont les coordonnées cartésiennes d'un point exprimées dans un repère de référence. Les opérations élémentaires (une succession d'unions, d'intersections, de différences ou de déformations semi-algébriques,...) sur ces primitives géométriques peuvent être formulées de la même manière. On parle dans ce cas de Géométrie Constructive des Solides (en anglais CSG).

La définition de la position d'un repère ou d'un solide dans l'espace ($d = 3$) ou dans le plan ($d = 2$) est donnée par un élément du groupe Spécial Euclidien $SE(d)$. Une représentation matricielle de cette position $g \in SE(d)$ est donnée par la matrice suivante dite matrice homogène construite de telle manière que la loi de groupe se ramène au produit matriciel:

$$g = \begin{pmatrix} & & & t \\ \dots & R & & \vdots \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ avec } t \in \mathbf{R}^d \text{ et } R \in SO(d) \text{ le groupe spécial orthogonal.}$$

Pour $d=3$, on notera $p_1 = [px_1, py_1, pz_1, 1]^T$ et $p_2 = [px_2, py_2, pz_2, 1]^T$ les coordonnées homogènes d'un point p quelconque exprimées dans les repères $r=1$ ou $r=2$. Ces coordonnées sont liées par la relation matricielle :

$$p_1 = g_{1,2} \cdot p_2 \quad (1)$$

Dans le cas d'une opération, impliquant l'utilisation d'engin(s), plusieurs solides doivent être pris en compte et la formalisation présentée ci-dessus doit être généralisée. Nous allons ainsi généraliser la définition des liaisons à N solides de manière à former un graphe orienté dont les sommets représentent les solides et les arcs les différentes liaisons. On considère un ensemble ordonné de N solides $\{S_i\}_{i=1..N}$, potentiellement mobiles et liés chacun à un repère. L'ensemble des objets statiques, (*i.e.* « l'environnement »), est quant à lui muni d'un repère que l'on appellera absolu et qui sera considéré comme un solide noté S_0 . On notera $g_{i,j}$ la matrice homogène qui permet de positionner le repère du solide S_i par rapport à celui de $S_{j \neq i}$.

La relation matricielle (1) et les propriétés de groupes de $SE(3)$ induisent pour N repères les règles (2) :

$$\forall r_1, r_2, r_3 \in \{0..N\} \quad g_{r_1, r_2} \cdot g_{r_2, r_3} = g_{r_1, r_3}, \quad (2)$$

$$g_{r_1, r_1} = Id \quad \text{et} \quad (g_{r_1, r_2})^{-1} = g_{r_2, r_1}$$

Ainsi, N éléments de $SE(3)$ notés $g_{r_1(l), r_2(l)}$ pour $l=1..N$ suffisent à placer N repères dans l'espace si les couples orientés $(r_1(l), r_2(l)) \in \{0, \dots, N\}^2$ correspondent aux arcs d'un arbre couvrant dont les nœuds sont les repères. Les règles (2) permettent de passer d'un arbre à l'autre.

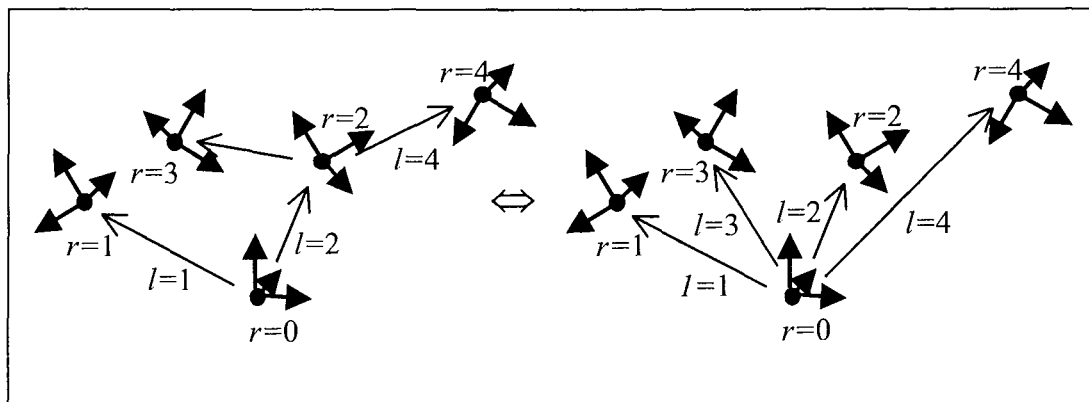


Figure 2.1 : Exemples d'arbres de référence

La figure ci-dessus présente deux exemples de conventions qui peuvent être choisis comme conventions pour le placement de $N=4$ repères dans l'espace. Par défaut, la convention de droite sera choisie.

1.2 Formulation des contraintes de non-collision et d'évitement de zones

Les contraintes de non-collision sont, du fait de leur nombre, les contraintes géométriques dont le traitement est le plus problématique. Pour illustrer la formalisation d'une contrainte de non-collision, nous considérons, avant de généraliser, un premier exemple simple comprenant deux solides dans le plan. Le premier solide S_1 est constitué d'un carré de côté 4 dont on a extrudé un disque de rayon 1 en son centre. S_1 peut être représenté par une fonction logique sur les coordonnées $p_1 = (px_1, py_1)^T$ d'un point du plan dans un repère lié au solide S_1 . Cette fonction logique doit prendre la valeur booléenne VRAI (ou 1) lorsque le point p_1 est dans le solide et FAUX (ou 0) sinon. L'ensemble des points pour lesquels φ_1 est vérifiée définit le solide S_1 dans son repère. On définit de la même manière un triangle S_2 .

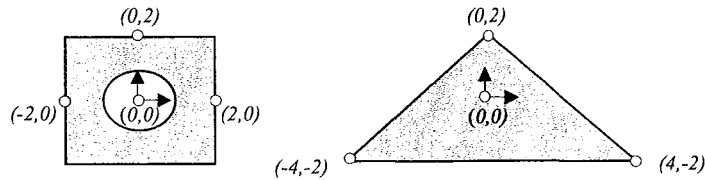


Figure 2.2 :Exemples élémentaire de CSG

$$\varphi_1(px, py) = (px - 2 \leq 0) \wedge (px + 2 \geq 0) \wedge (py - 2 \leq 0) \wedge (py + 2 \geq 0) \wedge (px^2 + py^2 - 1 \geq 0)$$

$$\varphi_2(px, py) = (px - py + 2 \geq 0) \wedge (px + py - 2 \leq 0) \wedge (py + 2 \geq 0)$$

Les coordonnées $(px, py)^T$ apparaissant dans ces expressions ne sont pas définies par rapport aux mêmes repères. Pour un même point p du plan, on distinguera par la suite le vecteur $p_1 = (px_1, py_1)^T$ de ses coordonnées dans le repère lié à S_1 du vecteur $p_2 = (px_2, py_2)^T$ de ses coordonnées dans le repère lié à S_2 . Ces vecteurs sont liés par le changement de repère $p_1 = g_{1,2} \cdot p_2$ où $g_{1,2}$ est une matrice homogène représentant un changement de repère en dimension deux ($SO(2) \times \mathbb{T}^2$) ..

Pour un positionnement relatif de deux solides donné par $g_{1,2}$, la détection de collision entre ces deux solides revient alors à vérifier l'expression suivante :

$$\left[\exists p_1, \varphi_1(p_1) \wedge \varphi_2(g_{2,1} \cdot p_1) \right] \quad \text{qui est vraie en cas de collision}$$

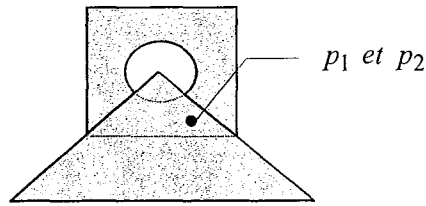


Figure 2.3 : collision détectée (p_1 et p_2 coïncident)

De manière équivalente la garantie de non collision revient à vérifier :

$$\left[\forall p_1, \neg \varphi_1(p_1) \vee \neg \varphi_2(g_{2,1} \cdot p_1) \right] \quad \text{qui est vraie en cas de non-collision}$$

Pour généraliser au cas de N solides ne pouvant entrer en collision, on doit exprimer l'ensemble des variables et des matrices homogènes dans un repère commun. L'expression d'une collision entre au moins deux solides est alors pour $i, j \in \{1, \dots, N\}$:

$$\left[\exists p_0, \bigvee_{i \neq j} \left[\varphi_i(g_{i,0} \cdot p_0) \wedge \varphi_j(g_{j,0} \cdot p_0) \right] \right]$$

et l'expression d'une « non-collision » s'en déduit :

$$\left[\forall p_0, \bigwedge_{i \neq j} \left[\neg \varphi_i(g_{i,0} \cdot p_0) \vee \neg \varphi_j(g_{j,0} \cdot p_0) \right] \right]$$

Nous noterons que des expressions similaires peuvent être utilisées lorsque que l'on souhaite détecter si un ou plusieurs solides rentrent dans une zone définie elle aussi à l'aide d'un système d'inéquations φ dans le repère absolu. Ceci permet de formaliser des contraintes d'évitement de zones dangereuses ou de référencer la rentrée d'objets dans de telles zones.

1.3 Formulation des liaisons et déformations

Nous allons maintenant rappeler que les liaisons entre des paires de solides, comme des articulations ou le maintien de contacts peuvent s'exprimer elles aussi à l'aide d'équations

polynomiales et de quantificateurs. Ces formules doivent être vérifiées continuellement lorsque la position relative des repères évolue. Elles définissent des liaisons mécaniques entre les deux solides.

Pour ce faire nous considérerons qu'un point p doit appartenir simultanément à deux surfaces de deux solides distincts. Chaque surface est définie par un système d'équations polynomiales dans le repère du solide correspondant que nous noterons $\varphi_1(p_1)$ et $\varphi_2(p_2)$ ou encore $\varphi_2(g_{2,1} \cdot p_1)$. La contrainte géométrique s'exprime par la formule :

$$\Psi = [\forall p_1, \varphi_1(p_1) \wedge \varphi_2(g_{2,1} \cdot p_1)]$$

Si nous considérons les cas où φ_1 et φ_2 décrivent chacun les équations implicites d'un plan, d'une droite ou simplement d'un point, les différentes combinaisons possibles permettent de retrouver la plupart des liaisons usuelles.

<i>Liaisons $\Psi = [\forall p_1, \varphi_1(p_1) \wedge \varphi_2(g_{2,1} \cdot p_1)]$</i>			
	φ_1	φ_2	<i>dim.</i>
<i>Libre</i>	X	X	6
<i>Ponctuelle</i>	<i>point</i>	<i>plan</i>	5
<i>Linéique rectiligne</i>	<i>droite</i>	<i>plan</i>	4
<i>Linéique annulaire</i>	<i>point</i>	<i>droite</i>	4
<i>Rotule</i>	<i>point</i>	<i>point</i>	3
<i>Appui plan</i>	<i>plan</i>	<i>plan</i>	3
<i>Pivot glissant</i>	<i>droite</i>	<i>droite</i>	2

Tableau 2.2 : Liaisons usuelles

Les deux exemples les plus caractéristiques de liaisons concernent les liaisons glissières et les liaisons pivots. Elles s'obtiennent en combinant deux des liaisons du tableau précédent.

<i>Liaisons $\Psi \equiv \Psi_1 \wedge \Psi_2$</i>			
	Ψ_1	Ψ_2	<i>dim</i>
<i>Glissière</i>	<i>Appui plan</i>	<i>Pivot glissant (droites dans plans)</i>	<i>1</i>
<i>Pivot</i>	<i>Pivot glissant</i>	<i>Rotule (points dans droites)</i>	<i>1</i>

Tableau 2.3 : Liaisons simples

Les contraintes des liaisons rotoïdes ou prismatiques (ou plus généralement les vissages) définissent une variété de dimension 1 qui peut être paramétrée (dans l'annexe 2 nous présentons une paramétrisation en $K_1 \cdot \exp(V \cdot q) \cdot K_2$ avec $q \in \mathbb{R}$, $V \in T_{Id}SE(3)$ et $K_{i=1,2} \in T_{Id}SE(3)$). La mise en série de $n=2$ à 6 de ces liaisons permet de paramétrer toutes les liaisons usuelles du tableau 2.2 à l'aide du paramètre $q \in \mathbb{R}$. Néanmoins, cette représentation est limitée à une carte et n'est donc que partielle. Pour s'en convaincre, nous pouvons considérer l'exemple critique de paramétrisation suivant :

Dans l'exemple ci-dessous (figure 2.4), le mouvement planaire du solide paramétré en $q=(x,y,\theta)$ doit franchir $\theta=0$ pour ne pas créer artificiellement deux composantes connexes. Si l'on cantonne θ à $[0,2\pi]$, une représentation erroné des composantes connexes et classes d'homotopies de l'espace des configuration libre est possible.

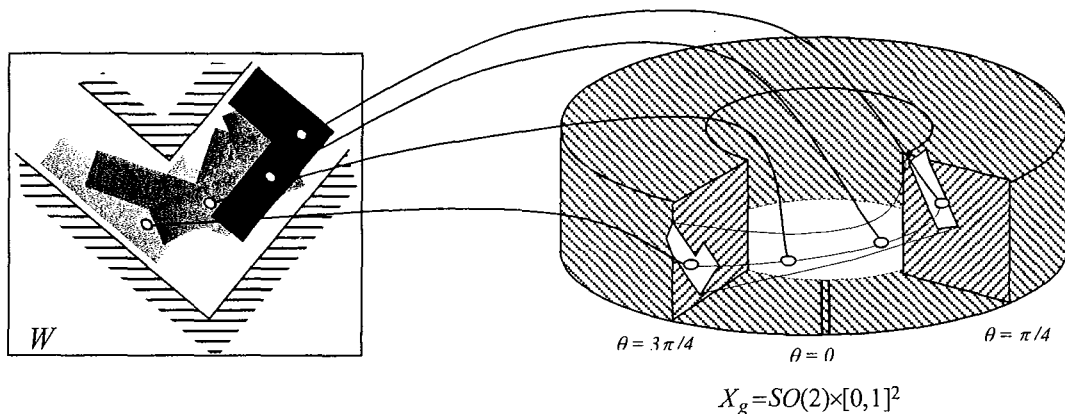


Figure 2.4 : Cas critique de paramétrisation par $\theta \in [0,2\pi]$: perte de connectivité

La formulation implicite des contraintes est donc plus exacte, elle est aussi plus générale. Elle devient particulièrement intéressante lorsqu'il s'agit de modéliser le maintien de contacts entre deux solides dont les surfaces sont elles-mêmes décrites à l'aide d'équations polynomiales des trois variables spatiales.

$\Psi = [\forall p_1, \varphi_1(p_1) \wedge \varphi_2(g_{2,1} \cdot p_1)]$		
	φ_1	φ_2
<i>Ponctuelle surfacique</i>	<i>point</i>	<i>surface</i>
<i>Linéique gauche</i>	<i>courbe gauche</i>	<i>surface</i>
<i>Linéique annulaire gauche</i>	<i>point</i>	<i>courbe gauche</i>
<i>Appui surfacique</i>	<i>surface</i>	<i>surface</i>
<i>Pivot glissant gauche</i>	<i>courbe gauche</i>	<i>courbe gauche</i>

Tableau 2.4 : Liaisons de contact complexes

Nous présentons ci-dessous un exemple d'une contrainte de maintien d'un contact ponctuel le long d'une courbe gauche obtenue comme l'intersection d'une surface cylindrique et d'un plan. Les positions d'un outil sur une surface à traiter doivent être précisément étudiées pour mettre en exergue les contraintes liées au procédés ou à la géométrie de la tâche.

Exemple 1 : Tâche de modification d'un objet et usinage de l'intérieur d'un tuyau

Les services d'ingénieries d'EDF sont amenés dans le cadre d'études de faisabilité de réparation par l'intérieur des conduites du circuit primaire d'une centrale nucléaire à considérer les mouvements d'un outil (de soudage, fraisage,..) en contact ponctuel avec l'intérieur d'un tuyau.

L'outil est soumis à une contrainte de contact ponctuel le long d'un cordon suivant l'intérieur du tuyau. Ce cordon se définit comme un portion de l'intersection du tuyau (cylindrique) de rayon r et d'un plan défini par trois constantes $k_{i=1,2,3}$:

$$\varphi(px_1, py_1, pz_1) = \left[\begin{array}{l} px_1^2 + py_1^2 - r^2 = 0 \\ px_1 + k_1 py_1 + k_2 pz_1 + k_3 = 0 \end{array} \right]$$

Les positions de l'outil qui respectent ce contact se traduisent par $\varphi(p_1) = \varphi(g_{1,2} \cdot p_2)$ où p_2 désigne les coordonnées de l'extrémité de l'outil dans son repère (c'est une constante). Cette expression se traduit directement par un système d'équations polynomiales.

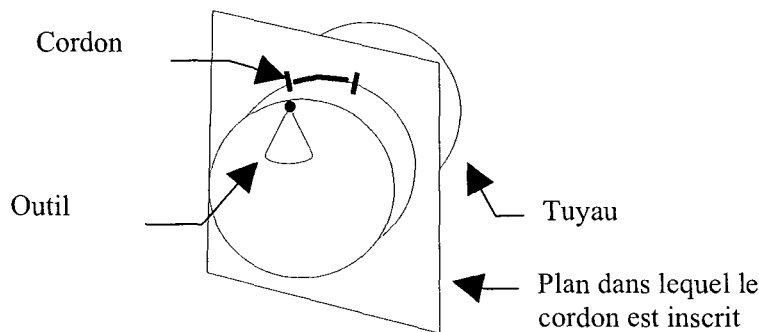


Figure 2.5 : contrainte de contact ponctuel sur un cordon cylindrique

La considération des liaisons entre deux solides fournit le formalisme nécessaire à la définition de contraintes sur les **tâches de déplacement ou de modification** d'un unique solide mobile dans son environnement.

Exemple 2 : Déformations élémentaires d'un corps cylindrique

Les problèmes de planification de trajectoire d'objets déformables ont déjà été abordés par exemple dans [ANS00]. Dans le contexte d'un chantier nucléaire, ils concernent principalement les câbles flexibles (présents en grand nombre dans le fond d'une cuve de réacteurs nucléaires, les câbles de grues ou les ombilics d'alimentation de robots mobile).

Considérons l'exemple élémentaire suivant :

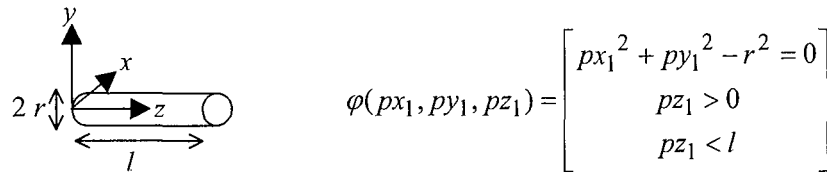


Figure 2.6: : Déformations longitudinales et radiales d'une portion cylindrique

La transformation $\varphi_1(\lambda.px_1, \lambda.py_1, a.pz_1)$ décrit un allongement du câble lorsque a varie et une variation du rayon du cylindre lorsque λ varie. Néanmoins, les phénomènes physiques mis en jeux au cours d'une déformation peuvent être complexes et insuffisamment déterministes pour être simulés. Il est, par exemple, difficile de prédire le comportement de l'ombilic d'alimentation d'un robot mobile. Les contraintes de déformation présentent en revanche un intérêt indéniable pour le prototypage d'opération car ils permettent d'introduire un paramétrage du design d'un engin. Les liaisons virtuelles (cf. *figure 1.8* du chapitre §I) s'expriment suivant le même formalisme que les déformations et liaisons réelles présentées ci-dessus mais modélisent des paramètres destinés à être figés au cours d'un mouvement.

Exemple 3 : Description des liaisons d'une charge sur le sol et d'un transpalette

On représente ci-contre un graphe de liaisons désignant le châssis d'un véhicule transpalette (2) muni de fourches (3) destiné à transporter un palette (1) posée sur le sol (0).

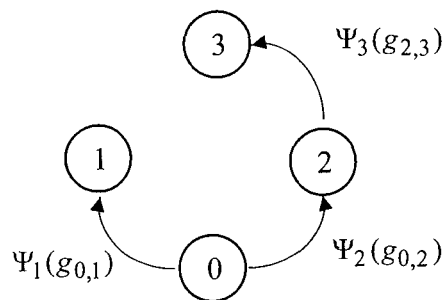


Figure 2.7: graphe de liaison

L'expression $\Psi = \Psi_1 \wedge \Psi_2 \wedge \Psi_3$ rassemble l'ensemble des contraintes associées au graphe de liaison. Celui-ci peut former des cycles. Un graphe des liaisons ne suffit pas toujours pour décrire une situation. L'utilisation de l'opérateur booléen \vee peut être mis à profit pour décrire une famille de graphes de liaisons (avec ou sans cycles).

Exemple 4 : Gestion de la manipulation d'une charge par un transpalette

Nous considérons un transpalette dont l'agencement des composants solides et le maintien au contact du sol est vérifié par l'expression Ψ_4 (correspondant à $\Psi_4 = \Psi_2 \wedge \Psi_3$ de l'exemple précédent) et une palette dont les contacts sur le sol ou sur les fourches du transpalette sont respectivement vérifiés par les formules Ψ_1 et Ψ_5 . La formule $\Psi_4 \wedge (\Psi_1 \vee \Psi_5)$ permet de décrire, à l'aide d'une unique formule, la situation où le transpalette transporte sa palette qui peut être représentée à l'aide d'un premier graphe de liaison arborescent, la situation où la palette est posée sur le sol qui correspond à un deuxième graphe arborescent et la situation où la palette est simultanément en contact avec le sol et les fourches du transpalette (le connecteur OU est non exclusif). Cette dernière situation correspond à $\Psi_4 \wedge \Psi_1 \wedge \Psi_5$ que nous appellerons une transition et qui correspond à un graphe de liaison comprenant un cycle.

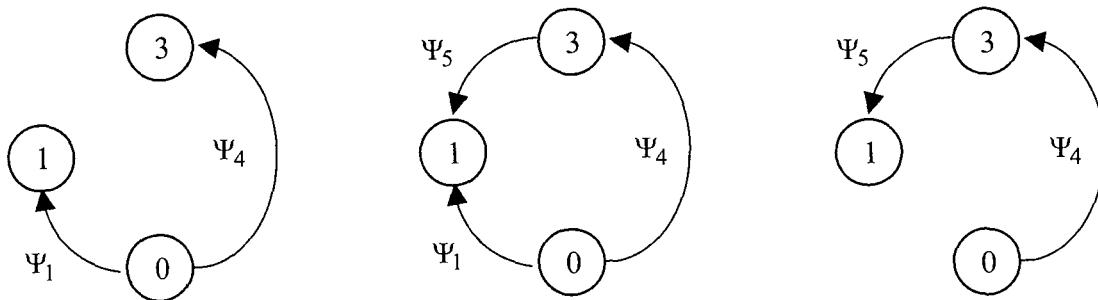


Figure 2.8: Développements de l'expression $\Psi_4 \wedge (\Psi_1 \vee \Psi_5)$

2. Nature algébrique finie de l'espace des configurations d'un chantier

Dans le paragraphe précédent, pour ne pas limiter de manière prématurée la diversité potentielle des situations rencontrées durant l'évolution d'un chantier, nous avons présenté une formulation générale des contraintes géométriques sous une forme implicite, et ce à partir de la définition des inéquations définissant les formes des objets. Ces définitions peuvent être replacées dans le cadre théorique de l'algèbre de Tarski. Ce cadre a fait l'objet d'abondantes recherches ces vingt dernières années pour la formulation de très nombreux problèmes liés à la CAO dont on peut trouver des synthèses dans [DEY99]. Nous verrons que cette approche ne pourra pas donner lieu directement à la conception d'algorithmes. Son intérêt réside essentiellement dans l'analyse de la nature des difficultés algorithmiques que l'on rencontre lorsque l'on cherche à construire ces solutions de manière exhaustive ou de manière équivalente, lorsque l'on cherche à caractériser les données d'un problème susceptible d'être résolu par des moyens informatiques dans le cas de la préparation d'un chantier.

2.1. Une formulation canonique des contraintes géométriques

Les définitions suivantes précisent les notions d'ensemble semi-algébriques réels.

Définition 2.5 : Ensemble semi-algébrique. *Un ensemble semi-algébrique de \mathbb{R}^d est un sous-ensemble de \mathbb{R}^d vérifiant une combinaison booléenne d'équations et d'inéquations polynomiales. L'ensemble des ensembles semi-algébriques de \mathbb{R}^d est stable par unions ou intersections finies, produit cartésien, et passage au complémentaire.*

Exemple 1 : Les primitives géométriques utilisées en CAO sont représentées par une ou plusieurs inéquations polynomiales et forment des ensembles semi-algébriques de \mathbb{R}^3 . La résultante est une combinaison d'inéquations des polynômes des trois variables spatiales séparées par les opérateurs booléens, il s'agit donc toujours d'un ensemble semi-algébrique de \mathbb{R}^3 . Nous avons réservé la lettre φ pour les primitives géométriques de \mathbb{R}^3 et leurs résultantes.

Exemple 2 : Nous pouvons replacer la définition d'une matrice homogène dans le cadre fixé par la définition précédente. Rassemblons préalablement les 12 paramètres réels d'une matrice homogène, à l'aide de l'isomorphisme de notation suivant :

$$coef : g = \left(\begin{array}{ccc|c} x_1 & x_2 & x_3 & x_{10} \\ x_4 & x_5 & x_6 & x_{11} \\ x_7 & x_8 & x_9 & x_{12} \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \mapsto x = (x_1 \dots x_{12})^T \in \mathbb{R}^{12}$$

Les contraintes $[R^T R - Id = [0]]$ sur la définition de la rotation fournissent 6 équations polynomiales dont les variables sont les 9 coefficients de la matrice de rotation. La conservation de l'orientation des repères nécessite une inéquation supplémentaire que l'on désignera de la même manière par $[\det(R) > 0]$ pour lever l'ambiguïté $\det(R) = \pm 1$. Si l'on note

$$\alpha(x) = [R^T R - Id = [0]] \wedge [\det(R) > 0]$$

$$X = \mathbb{R}^{12}.$$

L'ensembles X_α des $x \in X = \mathbb{R}^{12}$ vérifiant le système polynomial α présenté ci-dessus, est un ensemble semi-algébrique de \mathbb{R}^{12} (il s'agit d'une sous-variété de dimension 6).

Définition 2.6: Fonction semi-algébrique. Soient deux ensembles semi-algébriques $X_1 \subset \mathbb{R}^{n_1}$ et $X_2 \subset \mathbb{R}^{n_2}$, une fonction $f : X_1 \rightarrow X_2$ est semi-algébrique si son graphe $\{(x_1, x_2) \in X_1 \times X_2 / x_2 = f(x_1)\}$ est un ensemble semi-algébrique.

En particulier, l'image et l'image réciproque d'un ensemble semi-algébrique sont des ensembles semi-algébriques. La composée de fonctions semi-algébriques est semi-algébrique.

Exemple : Les déformations présentées dans la partie précédente sont des fonctions semi-algébriques.

Nous montrons dans cette partie comment les expressions mises en évidence dans la partie précédente (§II.1) peuvent correspondre à une unique formule de Tarski.

Définition 2.7: Formules de Tarski. Soient $\mathbb{F}[x_1, \dots, x_d]$ l'ensemble des polynômes à d variables et à coefficients réels, les opérateurs booléens ET (noté \wedge), OU (noté \vee) et l'opérateur de négation (noté \neg), on pose les règles suivantes :

- 1 - si $P \in \mathbb{F}[x_1, \dots, x_d]$ alors $[P = 0]$ et $[P > 0]$ sont des formules de Tarski
- 2 - si Φ et Ψ sont des formules, alors $\Phi \wedge \Psi$, $\Phi \vee \Psi$ et $\neg \Phi$ sont des formules de Tarski
- 3 - si Φ est une formule de Tarski et x_i une variable réelle alors $[\forall x_i, \Phi]$ $[\exists x_i, \Phi]$ sont des formules de Tarski.

L'ensemble des formules obtenues au moyens de 1 et 2 sont des formules sans quantificateur qui définissent directement un ensemble semi-algébrique. Nous les désignerons par la suite au moyen des minuscules grecques (notamment α , β et φ) en réservant les majuscules aux formules avec quantificateurs. Pour $X = \mathbb{F}^d$, on note $X_\alpha = \{x \in X / \alpha(x)\}$ et $X_\beta = \{x \in X / \beta(x)\}$ les ensembles semi-algébriques correspondant. On a alors :

$$X_{\alpha \wedge \beta} = X_\alpha \cap X_\beta$$

$$X_{\alpha \vee \beta} = X_\alpha \cup X_\beta$$

Exemple 1: pour $d=2$, $\varphi(x) = [x_1^2 + x_2^2 - 1 = 0]$ force un point du plan \mathbb{F}^2 à se situer sur un cercle unitaire. On peut considérer φ comme une expression logique vraie ou fausse suivant le choix de $x = (x_1, x_2)$.

Exemple 2 : L'expression $\Phi(x) = [(\exists x_1), (\forall x_2), [x_2^2 - x_1 < 0]]$ est fausse.

Nous avons vu dans la partie précédente que toutes les contraintes formalisant les interactions de deux solides s'expriment à partir de systèmes d'équations (ou d'inéquations) définissant dans les cas les plus complexes une surface polynomiale de \mathbb{E}^3 dans un repère donné d'indice i . Ce système, noté $\varphi_i(p_i)$ s'exprime de manière équivalente par $\varphi_i(g_{i,j}p_j)$. Ceci permet de fixer dans un repère donné les coordonnées d'un même point p de \mathbb{E}^3 , dans le cas d'une expression de type $\varphi_i(g_{i,j}p_j) \wedge \varphi_j(p_j)$.

Cette expression peut être considérée comme une formule des variables réelles, $(g_{i,j})_k$, px_j , py_j et pz_j à laquelle on adjoint les contraintes de définition sur les coefficients réels $(g_{i,j})_k$ en tant que coefficients de la matrice homogène $g_{i,j}$ (cf. exemple 2 du §II.2.1). Cette manipulation se généralise au produit de type $\varphi_i(g_{i,k1} \cdot g_{k1,k2} \dots g_{kn-1,kn} \cdot p_{kn})$. Or nous avons rappelé (§II.1.1) que le placement de N solides dans l'espace pouvait toujours, (i.e. indépendamment des contraintes) être fourni par N matrices homogènes si celles-ci sont associées à un arbre de référence. Par ce moyen, étant donné un arbre de référence, toutes les contraintes (y compris celles correspondant à des liaisons qui ne sont pas sur l'arbre de liaisons de référence) peuvent être ramenées à une formule de Tarski avec quantificateurs dont les variables regroupent les coefficients des N matrices homogènes de référence et les coordonnées des points utilisés (il suffit pour cela de considérer que deux repères-noeuds quelconques de l'arbre peuvent être liés par une chaîne de l'arbre $\varphi_i(g_{i,k1} \cdot g_{k1,k2} \dots g_{kn-1,kn} \cdot p_{kn})$).

Il convient en pratique de choisir un arbre pour lequel les formules de Tarski se simplifient (et éventuellement se résolvent formellement cf §III), mais ce n'est pas l'objet de ce chapitre. Notre propos est de montrer que, sous des hypothèses relativement faibles comme la nature polynomiale des surfaces des objets, les formules de Tarski avec quantificateurs couvrent le spectre des contraintes géométriques $G(x)$ que l'on peut rencontrer lors de la préparation d'un chantier. Les liaisons articulées usuelles (liaisons rotoïdes, prismatiques,...), le maintien de contacts entre deux solides (ponctuels, linéiques, surfaciques) lors d'une tâche d'usinage mais aussi les liaisons virtuelles dont le choix a un impact sur le prototypage d'une l'opération (placement de la base fixe d'un robot manipulateur dans son environnement, position de saisie d'un objet ou d'un outil,...) sont des formules de Tarski. On pourra les mettre sous une forme canonique en regroupant les quantificateurs consécutifs de même type (en prenant garde à l'ordre).

$$G(x) = (Q^1 x^1) \dots (Q^\omega x^\omega) \alpha(x)$$

où $Q^i \in \{\exists, \forall\}$, x^i est un vecteur de dimension n_i extrait de $x = [x_1, \dots, x_d]^T$.

L'expression des contraintes sous une forme implicite permet de définir de manière très générale un espace tangent en x sous recours à une quelconque *paramétrisation*^{A2}.

Définition 2.8: Espace tangent d'une variété algébrique. Si X_α est une sous-variété algébrique réelle définie par le système $\bigwedge_{j \in J} [P_j = 0]$ avec $P_j \in \mathbb{F}[x_1, \dots, x_d]$, l'espace tangent en x se définit par.

$$T_x X_\alpha = \bigcap_{j=1}^J \left\{ (v_1, \dots, v_d) \in \mathbb{F}^d, \sum_{i=1}^d v_i \cdot \frac{\partial P_j}{\partial x_i} = 0 \right\}$$

Nous noterons que cette définition met en évidence la structure d'espace vectoriel de $T_x X_\alpha$ (les réels $(v_1, \dots, v_d) \in \mathbb{F}^d$ sont localement des solutions de systèmes linéaires) mais qu'elle ne fournit pas de *repère naturel*^{A2} pour l'expression des vecteurs tangents correspondants. Ceci est une conséquence de la définition implicite des contraintes géométriques dont les solutions ne bénéficient pas de paramétrisation explicite sur laquelle la définition d'une base en $x \in X_\alpha$ pourrait s'appuyer comme dans le cas du passage d'un modèle géométrique à un modèle cinématiques en robotique. Néanmoins, cette définition permet de trouver la dimension de la variété considérée (*i.e.* le nombre de variables libres ou le degré de liberté).

Exemple : Si l'on reprend l'exemple d'une matrice homogène, et que l'on se réfère à la définition de $T_x X_\alpha$ et aux équations polynomiales α , on peut aisément vérifier que les coefficients $(v_1, \dots, v_{12}) \in \mathbb{F}^{12}$ sont soumis aux équations linéaires $v_1 = v_5 = v_9 = 0$, $v_4 = -v_2$, $v_3 = -v_7$ et $v_6 = -v_8$ qui définissent un sous-espace vectoriel de dimension 6 pour tout $x \in X_\alpha$ (*i.e.* $x \in X_\alpha$ une sous-variété³ algébrique de \mathbb{F}^{12}).

2.2 Les limites d'une décomposition semi-algébrique de l'espace des configurations libres

Pour décider si la formule générale présentée ci-dessus est vraie pour un x donné, il faudrait trouver une expression équivalente qui ne contienne pas de quantificateur (\forall ou \exists) de manière à être

évaluée directement en testant le respect des (in)équations sur chacun des polynômes du système. Ce problème d'élimination des quantificateurs a été formulé et résolu par Alfred Tarski [TAR51].

Théorème 2.1 : Tarski-Seidenberg (Version 1). *Si Φ est une formule de Tarski, l'ensemble des variables libres qui satisfait Φ est un sous-ensemble semi-algébrique de \mathbb{R}^d .*

En d'autres termes, ce théorème montre qu'il est toujours possible de passer d'une formule avec quantificateurs à une formule équivalente sans quantificateurs. Notons $P \subset \mathbb{R}[x_1, \dots, x_d]$ l'ensemble fini des polynômes obtenus.

Définition 2.9 : Décomposition semi-algébrique. *On appelle décomposition semi-algébrique d'un ensemble de polynômes P une collection finie de sous-ensembles algébriques connexes, disjoints, pour lesquels le signe ($>0, <0, 0$) de chaque polynôme de P est constant.*

Pour un placement arbitraire soumis aux contraintes $G(x)$, il suffit ainsi de déterminer à quelle partie d'une décomposition semi-algébrique il appartient pour décider s'il est admissible ou non. On dit alors que le problème de recherche d'un point $x \in X$ admissible est décidable pour signifier qu'un traitement informatique est possible.

Les quantificateurs des formules peuvent donc en théorie être éliminés et la formule sans quantificateur obtenue définit une partition **finie** de l'ensemble des solutions X_G pour laquelle chaque partie de \mathbb{R}^N vérifie (ou ne vérifie pas) dans son intégralité, les contraintes géométriques définies.

Exemple : Si l'on reprend l'exemple d'une collision et que l'on considère le cas où le placement relatif des deux objets est figé (à défaut d'une représentation graphique de \mathbb{R}^9 lorsque les objets bougent) la variable est réduite à la seule considération d'un point $x \in \mathbb{R}^3$. Nous pouvons représenter la décomposition semi-algébrique.

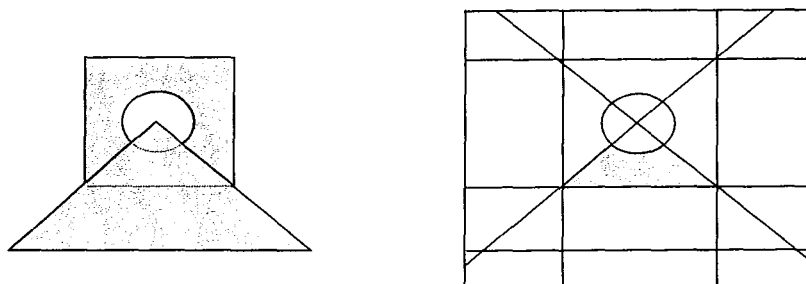


Figure 2.9: Décomposition semi-algébrique pour une collision statique

L'existence des décompositions semi-algébriques met en évidence la nature algébrique finie de l'espace des configurations d'un chantier. Nous abordons ci-dessous la mise en œuvre algorithmique d'une décomposition semi-algébrique et ses limites.

Un algorithme général de décomposition appelé décomposition cylindrique algébriques (en anglais CAD) a été formulé par Collins [COL75]. **Il constitue en fait le cœur de nombreux problèmes de traitement informatique des contraintes géométriques.** Il s'appuie directement sur la version suivante du théorème de Tarski-Seidenberg :

Théorème 2.2 : Tarski-Seidenberg (Version 2). Si S est un ensemble semi-algébrique et $proj(\cdot)$ la projection qui associe à tout point $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ le point $proj(x) = (x_1, \dots, x_{n-1}) \in \mathbb{R}^{n-1}$, alors $proj(S)$ est un ensemble semi-algébrique. (ce qui n'est pas le cas des sous-ensembles algébriques réels).

L'algorithme de Collins est une application directe du théorème ci-dessus (en considérant que l'image réciproque d'un ensemble semi-algébrique par une fonction semi-algébrique est semi-algébrique), il consiste à scinder l'espace à l'aide des racines des expressions polynomiales en augmentant progressivement la dimension pour obtenir une décomposition semi-algébrique (en fait une décomposition cellulaire). A chaque cellule, est associée un unique échantillon.

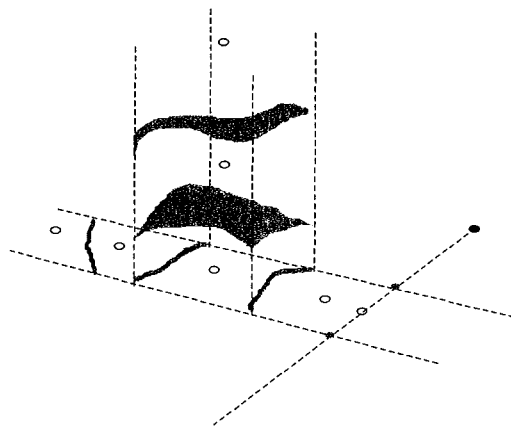


Figure 2.10 : Principe d'une CAD et de l'échantillonnage cylindrique correspondant

Cet algorithme est à la base de la plupart des études de complexité des algorithmes de planification de trajectoire [MIS97] et notamment du fameux problème du « déménagement d'un piano » [SCH83a&b]. On peut trouver de nombreux travaux sur la recherche d'une borne supérieure pour le nombre de telles cellules [COL75,GRI88, BAS94]. Celle-ci dépend d'après Collins de façon polynomiale du nombre et du degré des contraintes semi-algébriques lorsque le nombre de degrés de liberté ($\sum n_i$) est fixé. La dépendance vis-à-vis du nombre de degrés de liberté est quant à elle doublement exponentielle en $L^3(m.d)^{2^{O(\sum n_i)}}$. Des travaux plus récents ont permis de limiter cette borne à $(L \log L \log \log L)(m)^{\prod (n_i+1)}(d)^{\prod O(n_i)}$ [BAS94] où les variables sont le nombre (m) et le degré maximal des polynômes (d) ainsi que la dimension (n_i) des variables regroupées suivant la forme standard :

$$G(x) = (Q^1 x^1) \dots (Q^\omega x^\omega) \alpha(x)$$

où $Q^i \in \{\exists, \forall\}$, x^i est un vecteur de dimension n_i extrait de $x = [x_1, \dots, x_d]^T$ et L le nombre de bits nécessaires pour stocker les coefficients des polynômes de la formule de Tarski $\alpha(x)$.

La recherche d'une borne inférieure pour le nombre de cellules a été effectuée dans de nombreux cas particuliers en se ramenant à des classes de complexité standards ($L \subseteq P \subseteq NP \subseteq PSPACE$) dont une synthèse est présentée dans [LAT91]. Ces résultats tendent à montrer que le nombre de cellules dépend hormis quelques cas particuliers de manière au moins exponentielle du nombre de degrés de liberté.

La grande généralité des décompositions semi-algébriques permet de formaliser le traitement de l'ensemble des contraintes géométriques, d'évaluer leur coût algorithmique ainsi que la nature de l'espace des configurations libres. Mais leur emploi reste très limité en pratique si l'on cherche à reconstruire de manière exacte l'espace des configurations libres. Même avec une approximation des objets visant à réduire l'ordre et le nombre des polynômes ou une considération exclusive des cellules de dimension 0 ou 1, le nombre de cellules est rapidement réductible. Nous invitons le lecteur à consulter [LAT91] pour avoir une vision historique et exhaustive jusqu'au début des années 90 des approches visant à construire l'espace des configurations libres à partir de décomposition semi-algébriques.

Les approches ultérieures se sont placées dans l'espace des configurations en écartant ainsi tout traitement exhaustif et exact des contraintes semi-algébriques dues aux obstacles au profit de procédures de détection de collision effectuée *a posteriori* et de manière plus localisée. Nous verrons que les décompositions semi-algébriques, présentées dans cette partie pour nous éclairer sur les difficultés intrinsèques de l'étude des espaces des configurations libres pour la préparation d'un chantier, sont aussi intéressantes pour structurer l'espace des configurations (*cf.* notamment §II.3.3).

Si l'on écarte les contraintes semi-algébriques de non-collision, l'analyse de l'espace des configurations X_g n'en demeure pas moins difficile. Les décompositions semi-algébriques de X_g génèrent alors des ensembles semi-algébriques dont les configurations ne peuvent pas toujours être paramétrées. Ces problèmes sont toujours abondamment traités en robotique notamment pour les robots d'usinage et les robots parallèles [CHA98]. De nombreuses recherches utilisant par exemple le calcul formel de bases de Gröbner associées aux systèmes polynomiaux ont permis de résoudre des problèmes de classification de mécanismes complexes comme les robots parallèles [FAU95] mais l'utilisation pratique de telles approches est limitée à des ensembles de solutions de dimension zéro (solutions réelles finies). Des configurations ponctuelles peuvent être obtenues mais la définition de chemin dans l'espace des configurations libres est confrontée à des problèmes de franchissement de singularités lors du passage d'une variété semi-algébrique à une autre.

3. Planification de trajectoire et intérêt d'une capture des classes d'homotopie

Nous avons décrit la nature des espaces des configurations et des espaces des configurations libres dans un cadre suffisamment général pour couvrir les différents niveaux de modélisations géométriques d'un chantier de maintenance et mieux identifier les difficultés d'une représentation exacte des configurations libres. Nous abordons dans cette partie la possibilité de planifier ou optimiser automatiquement un mouvement. Cette fonctionnalité intervient lorsque la tâche que doit effectuer l'engin, le robot ou l'intervenant peut être accomplie de « diverses manières » et qu'il n'est pas trivial de trouver la manière adéquate. Si l'on se réfère aux parties précédentes du chapitre §I (tableau 1.6), les opérations de déplacements entrent directement dans cette catégorie, en particulier lorsque l'environnement est complexe ou lorsque le transport d'un objet est difficile. Les autres opérations sont aussi concernées dès lors qu'un choix judicieux sur l'implantation de l'engin doit être effectué ou dès lors que le pilotage de degrés de libertés redondants par rapport aux exigences de la tâche pose problème.

3.1 Approches existantes

Nous précisons tout d'abord quelques définitions sur la formulation d'un problème de planification de trajectoire dans le cadre général que nous avons précédemment posé.

Définition 2.10 : Planification de trajectoire sous contraintes géométriques. *La donnée d'un couple de configurations libres $(x_0, x_1) \in X_G^2$ définit un problème de planification qui est résolu si l'on trouve un chemin $\sigma \subset X_G$ paramétré en $t \in [0,1]$ avec $\sigma(0) = x_0$ et $\sigma(1) = x_1$ ou, inversement, si l'on sait qu'un tel chemin ne peut être trouvé.*

Des contraintes peuvent être définies sur le chemin lui-même. Plus globalement, ces contraintes d'origine physique ou liées à une optimisation peuvent s'exprimer localement et pour tout $x \in X_G$, comme une restriction sur les valeurs des vecteurs tangents de $T_x X_G$. L'étude de ces contraintes ne font pas l'objet de ce mémoire mais, le cas échéant, nous supposons disposer d'une intégration de ces contraintes sous la forme suivante :

Définition 2.11 : Méthode locale. *La donnée pour tout couple de configurations $(x_0, x_1) \in X_g^2$ d'un chemin $\sigma \subset X_g$ paramétré en $t \in [0,1]$ avec $\sigma(0) = x_0$ et $\sigma(1) = x_1$ définit une méthode locale.*

La possibilité, pour une méthode locale donnée d'approcher d'aussi près que l'on veut une trajectoire solution est une condition à remplir pour ne pas restreindre l'ensemble des solutions recherchées [LAU98]. Cette condition est liée à la commandabilité en temps petit du système considéré [LAF91]. Dans le cas de la préparation d'un chantier, la contrainte la plus récurrente est celle qui consiste (du fait de contraintes réglementaires de sécurité) à n'utiliser qu'un actionneur à la fois pour les grues, ponts roulants ou les ponts polaires utilisés dans les centrales nucléaires (on qualifiera par la suite ce type de mouvement de « Manhattan »). Cette contrainte ne présente pas de difficulté particulière quant à la condition que nous venons d'évoquer.

De nombreuses approches algorithmiques pour la génération ou l'optimisation d'un mouvement ont été explorées ces trois dernières décennies. Les algorithmes *complets*, c'est-à-dire capables de détecter en un temps fini si aucune trajectoire sans collision n'existe ou de fournir la trajectoire dans le cas contraire se sont avérés lents voire hors de portée des calculateurs actuels dès que l'environnement est complexe.

La plupart des approches récentes se ramènent à des explorations des configurations libres à l'aide de trajectoires ou de réseaux de trajectoires dont la construction est orientée vers la résolution d'un problème de planification donné. Elles écartent dans un premier temps les contraintes de non-collision et les contraintes de bouclage de liaisons d'une résolution directe, pour se placer dans le cadre d'un espace des configurations qui se réduit à un pavé de \mathbf{R}^n noté C . Les contraintes peuvent être vérifiées a posteriori sur des configurations $q \in C$ ou des chemins de configurations $q(t \in [0,1])$ dans ce cadre structuré (d'un point de vue topologique ou différentiel) pour explorer l'espace des configurations libres $C_{libre} \subset C$. Nous décrivons ci-dessous quelques stratégies apparues ces dix dernières années.

Les premières approches locales efficaces de planification comme RPP [BAR91] utilisant

une descente de *potentiels* calculé localement couplée à des mouvements browniens par échapper aux *minima locaux*. Il fut le premier à utiliser avec succès une approche probabiliste. Dans [AHU94], des phases d'expansion et d'exploration sont alternées. L'algorithme du fil d'Ariane alterne une phase d'exploration à l'aide d'un réseau arborescent partant de l'origine et une phase de recherche qui tente de relier le réseau au but. L'algorithme génétique utilisé pendant l'exploration nécessite néanmoins un réglage adapté des paramètres pour être efficace.

Les *Réseaux à apprentissage probabilistes* [KAV96, OV95], fondées sur une exploration stochastique des chemins de l'espace des configurations libres à l'aide d'une méthode locale, se sont révélées particulièrement efficaces et souples d'utilisation. Elles semblent éviter certains écueils des stratégies d'exploration trop rigides qui conduisent souvent à des problèmes locaux d'inflation des calculs dans le cas des réseaux réguliers ou de minima locaux dans le cas de l'utilisation de potentiels. Dans sa version standard, la construction du réseau s'achève lorsque les configurations initiale et finale du problème de planification sont dans la même composante connexe du réseau. Cette phase d'apprentissage est alors complétée par une recherche de la trajectoire dans le réseau par un algorithme de type A* puis une variation locale du chemin permettant un lissage de celui-ci. Le réseau doit potentiellement être complété pour répondre à un nouveau problème (*i.e.* une modification des configurations initiales ou finale, des contraintes géométriques ou des contraintes sur les chemins). C'est pourquoi nous qualifierons cette approche de locale. Néanmoins, la construction du réseau à l'aide de domaine de visibilité, MRP-Visi [SIM00], permet de définir une phase d'apprentissage indépendante de tout problème particulier de planification. MRP-Visi n'ajoute que des nœuds qui ne peuvent pas être directement reliés au réseau ou qui permettent de lier deux composantes non-connexes de celui-ci. Cette approche permet de réduire très fortement la taille du graphe et elle fournit un critère d'arrêt à la phase d'apprentissage qui est le nombre de nœuds successifs qui ne peuvent être ajoutés au graphe. Nous exploiterons dans les prochains chapitres cette propriété.

Les bonnes performances pratiques constatées des méthodes à réseaux probabilistes par rapport aux méthodes à réseaux déterministes comme les méthodes de rétractions [LAT91,ODU85] restent mal comprises. Leur analyse [HSU97&98] et l'étude des réseaux construits montre néanmoins une certaine difficulté à relier deux composantes connexes reliées dans C_{libre} par un *passage étroit*. Pour pallier ce problème, certains auteurs misent sur une augmentation de la probabilité d'échantillonner des configurations libres sur les bords [BOO99] ou le squelette [WIL99] de C_{libre} . Des procédures de dilatation de l'espace libre [HSU98, FER96] augmentent la « largeur des rétrécissements » de C_{libre} en autorisant une interpénétration limitée des

objets.

Un avantage déterminant des Méthodes à Réseaux Probabilistes a été de prendre en compte des *contraintes non-géométriques* comme certaines contraintes non holonomes (commandables en temps petit [SVE98]) ou des contraintes dynamiques [LAV01]. Elles sont complètes en probabilité (*i.e.* trouvent en un temps fini - mais inconnu - la trajectoire solution si celle-ci existe). La donnée d'un réseau de trajectoires *optimales* limitée au réseau permet de se rapprocher de la trajectoire optimale avec la densité de points échantillonnés à l'aide de procédure de programmation dynamique (de type Dijkstra ou A*).

De nombreuses techniques fondées sur une construction probabiliste du réseau peuvent être utilisées pour converger plus rapidement pour la résolution d'un unique problème de planification. Le réseau peut diffuser à partir des germes constitués des configurations initiale et finale. Nous parlerons dans ce cas de *réseaux à diffusion probabiliste*. Dans [KUF00], on génère un chemin qui part de l'origine et tente de rejoindre une configuration intermédiaire générée aléatoirement. Le réseau arborescent développé se rapproche du but à chaque itération. Cependant, le temps mis pour résoudre un problème peut varier fortement suivant la situation ; des *blocages* temporaires peuvent apparaître lors de l'expansion du réseau de manière analogue aux problèmes des minima locaux. Le réseau peut se développer autour d'un chemin qui varie pour satisfaire les contraintes. Nous parlerons dans ce cas de *réseaux à variation probabiliste*. Dans [FER96], les contraintes géométriques sont introduites progressivement sur un chemin qui varie localement jusqu'à les satisfaire. Dans [BOH00], les contraintes de non-collision ne sont appliquées que sur un chemin d'un graphe généré aléatoirement et pas sur le réseau complet. Ce chemin est trouvé par un algorithme de type A* pour des conditions fixées. Les nœuds et arcs du chemin (et seulement) sont testés et retirés du graphe s'ils sont non-admissibles. D'autres nœuds et arcs sont ajoutés au graphes et la procédure recommence jusqu'à ce qu'une solution soit trouvée.

Dans le cas des chantiers, l'utilisateur procède à des requêtes multiples de planification de trajectoire en fonction de l'évolution de la préparation du chantier (*cf §I*). Le logiciel est donc amené à réitérer ses calculs. Dans ces circonstances, il peut être intéressant d'investir dans une phase d'apprentissage global pré-calculé des trajectoires, qui permettrait ensuite de diminuer les temps de calcul de chacune des requêtes de planification de trajectoire postérieures. Nous chercherons de manière privilégiée, une méthode qui réalise une exploration la plus exhaustive possible de l'ensemble des trajectoires de l'espace des configurations, qui puisse être effectuée dans des temps raisonnables (au maximum une nuit de calcul) et qui garantisse ensuite une réponse rapide à chaque requête de

planification de trajectoires supplémentaire sollicitée lors les phases d'études de tâches ou d'opérations d'un chantier nucléaire.

Une des difficultés majeures de la préparation de chantier est, qu'à l'instar de l'étude des mouvements autorisant les contacts [JI01] ou des manipulations [ALA95] dont il peut être un des avatars dans le cadre du prototypage d'une opération d'usinage ou de manipulation, la prise en compte des liaisons temporaires (manipulation, coopération) ou permanentes (robot parallèle ou coordination d'engins) crée des boucles dans le graphe de liaisons. A chaque graphe cycliques correspond à un système polynomial complexe.

En particulier, l'analyse d'un problème de manipulation d'un objet par un robot [LAT91,ALA95] permet de décomposer formellement les contraintes de manière à définir des ensembles de solutions géométriques distincts comme celui où l'objet est saisi par le robot, celui où l'objet est posé sur le sol. Cette étape permet de se ramener à l'étude d'un nombre fini de sous-systèmes de contraintes géométriques qui définissent chacun des variétés de configurations. Le sous espace des configurations du robot et de l'objet correspondant à la situation où l'objet est simultanément posé sur le sol et saisi par le robot ne définit pas forcément une unique variété. Il est confronté aux problèmes de paramétrisation que nous avons évoqués ci-dessous et nécessite un traitement spécifique.

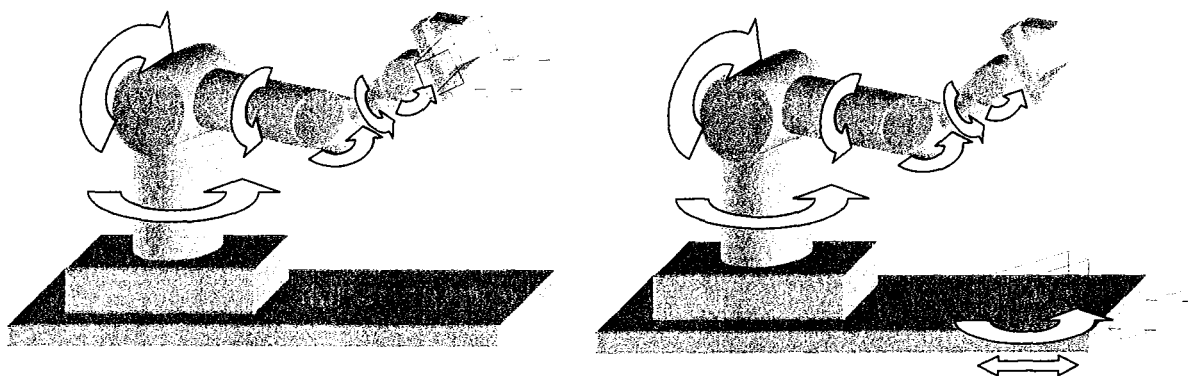


Figure 2.11: : La variété « Saisies possibles de l'objet » à gauche et la variété « Placements possibles de l'objet » à droite

L'approche standard en robotique pour générer des configurations respectant de telles contraintes consiste à ramener l'étude d'un graphe cyclique ayant n degrés de libertés (*i.e.* $n = \dim(TX_g)$) à celle d'un graphe acyclique ayant n' degrés de libertés auquel on adjoint des contraintes géométriques résiduelles de fermeture de boucle. On se ramène alors au cas standard $X_g \approx \text{SO}(d)^n \times [0,1]^m$ avec $d=2,3$ et $m, n \in \mathbb{N}$ qui est une variété paramétrable au moyen de cartes de

coordonnées $q \in C \subset \mathbf{R}^n$ où l'ensemble des coordonnées C est un pavé de \mathbb{I}^n (cf annexe 2). Dans [FER96], les cycles sont rompus ($n' > n$) puis on déplace localement la configuration jusqu'à ce qu'elle vérifie les contraintes écartées. Dans [COR02], un graphe acyclique connexe ayant n degrés de libertés est extrait de telle manière que les chaînes écartées correspondent à des modèles géométriques inversibles (avec un nombre fini de solutions). Les configurations du graphes acyclique ont alors une probabilité non nulle d'être complétées par les chaînes. Des développements sur le choix et l'existence d'un tel sous-graphe de liaisons et de nombreux exemples ont été établis. Dans tous les cas, s'il est possible de générer des configurations qui vérifient des contraintes géométriques résiduelles statiques (collision statique et inversibilité des chaînes écartées), la génération de chemin doit aussi vérifier des contraintes résiduelles dynamiques (collision dynamique et détection de singularités sur les chemins). Des méthodes de planification de trajectoires de manipulations ont été proposées dans le contexte des méthodes à réseau probabiliste [FER97,AHU99,COR02].

Des méthodes de planification de trajectoires de manipulations existent donc mais leur adaptation nécessite dans le cadre de la préparation d'un chantier une procédure propre à chaque opération. S'il est possible de disposer, pour les engins comme les robots parallèles d'une bibliothèque de fonctions adéquates, en ce qui concerne les problèmes de manipulation ou de coordination d'engins la diversité des situations rend l'utilisation d'un unique outil de planification difficile.

En résumé, nous proposons deux axes pour l'adaptation des méthodes existantes de planification de trajectoires au cas de la préparation d'un chantier :

"Réduire le temps de calcul de planification au moyen d'un pré-calcul global de manière à ce qu'ils soient adaptés à la préparation de chantiers "

"Définir une démarche permettant de se ramener à une situation standard pour l'outil de planification de trajectoires pour la plus large partie des situations apparaissant lors de la préparation de chantiers nucléaires"

3.2 Intérêt d'une capture des classes d'homotopie

La faisabilité d'une tâche ou d'une opération soumise aux contraintes géométriques G revient à tester la connexité des points initiaux et finaux dans X_G . Nous venons par ailleurs de préciser qu'il était important d'avoir une « vision assez exhaustive » des principaux types de mouvements que l'on peut utiliser pour accomplir une tâche, avant, si nécessaire, de préciser de nouvelles contraintes géométriques ou de modéliser l'opération associée. Nous allons tout d'abord montrer à l'aide de quelques exemples simples que la capture des *classes d'homotopies*^{A1} est une condition à remplir pour atteindre cet objectif.

Exemple 1 : Minimisation de la longueur d'un chemin

Dans le premier exemple élémentaire ci-dessous nous considérons le mouvement d'un disque entre une configuration initiale (*en gris clair*) et finale (*en noir*).

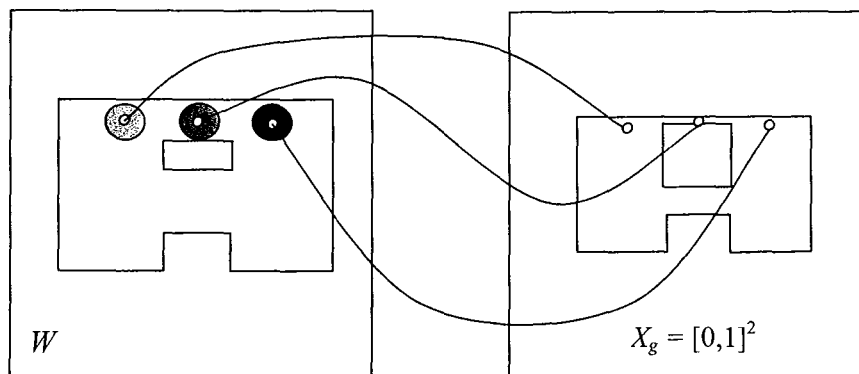


Figure 2.12: : Intérêt d'un pré-calcul des deux classes d'homotopie d'un disque devant passer d'une cavité à l'autre

On peut associer à ce mouvement un *chemin*^{A1} dans l'espace des configurations $[0,1]^2$. Le chemin le plus facile à trouver est le passage le plus large (*par le bas*). Si l'on dispose d'un réseau pré-calculé ne capturant que ce premier chemin et que l'on cherche *a posteriori* le plus court chemin (*par le haut*), tout est à recommencer. En revanche, un pré-calcul exhaustif des classes d'homotopie au moyen d'un réseau G force la capture d'un second chemin dans le passage le plus étroit. A l'issue de ce pré-calcul, l'ajout d'une contrainte de *minimisation de la longueur du chemin* est rapide puisque limitée à une variation locale du chemin dans une unique classe d'homotopie. (cf. [GRI97] pour une justification

plus théorique de la rapidité de la minimisation de la longueur d'un chemin dans une classe d'homotopie donnée).

Exemple 2 : Ajout de contrainte géométriques

Dans le deuxième exemple élémentaire, présenté ci-dessous, l'objectif est de démontrer la faisabilité d'une tâche de déplacement d'un fût dans un passage étroit (dans un plan vertical). L'espace des configurations est dans ce cas $SO(2) \times [0,1]^2$.

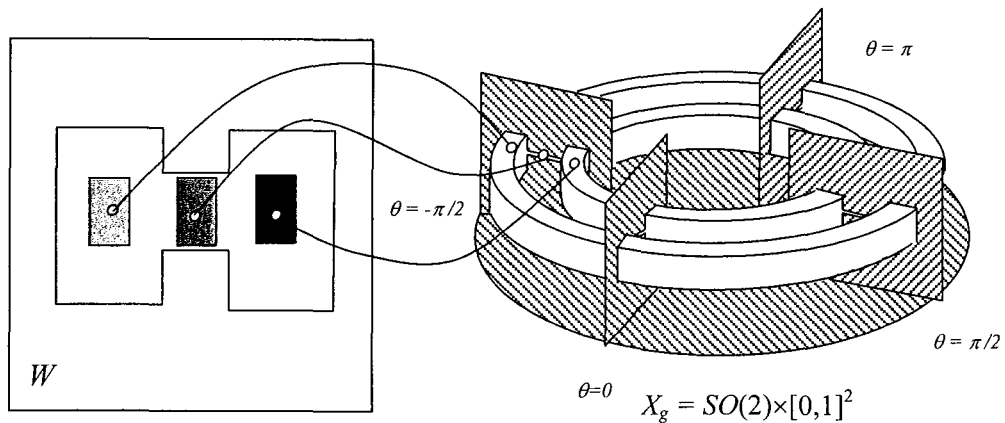


Figure 2.13: : Intérêt d'un pré-calcul des quatre classes d'homotopie d'un fût devant passer d'une cavité à l'autre

Si l'on considère **a posteriori** que le fût doit **rester vertical** durant le mouvement (parce qu'il contient un liquide pouvant se renverser...), le chemin fourni par un planificateur de type réseau probabiliste renvoie typiquement un mouvement où le fût passe horizontalement à travers le rétrécissement (la probabilité de trouver ce type de chemin entre les deux points à relier est beaucoup plus grande). Il ne permettra pas de répondre affirmativement quant à la faisabilité géométrique de cette tâche de déplacement. En revanche, si l'on force le réseau G à capturer tous les chemins passant par les différents rétrécissements ($\theta=0, \theta = \pi / 2, \theta=0, \theta=0$), cette situation ne conduira pas à une réinitialisation complète du problème.

Plus généralement, les contraintes introduites a posteriori peuvent être de diverses natures, elles concernent:

- les contraintes d'optimisation et les approches variationnelles de la dynamique des solides [ZEF97,BOB85],
- les contrainte géométriques semi-algébrique (collision) dans le cadre, par exemple des méthodes de réactualisation rapide par bande élastique [JAO97,BRO99],

- les contraintes géométriques algébrique (nouvelle liaisons) dans le cadre, par exemple, d'une approche en deux temps d'un problème de planification de trajectoire de manipulation consistant à considérer d'abord le mouvement d'un objet en déplacement libre avant [FER97],
- les contraintes non-géométriques, cette approche a par exemple été utilisée dans le cas de la planification d'un véhicule soumis à des contraintes cinématiques non intégrables [LAU94].

Dans tous ces cas, les variations locales du chemin sont problématiques s'il y a un changement de classe d'homotopie. Si l'on sait qu'une trajectoire solution doit être homotope à une trajectoire connue alors on peut rejeter rapidement toute trajectoire non-homotope. On peut aussi favoriser la progression de trajectoire des méthodes à diffusion probabiliste en utilisant la « distance estimée au but en passant par une classe d'homotopie donnée » au-lieu des heuristiques classiques fondées sur des distances au but. Le calcul d'un réseau capturant les classes d'homotopie peut être fait sur un problème simplifié (objet sans engin...) et néanmoins guider la recherche de problème plus complexe suivant ces mêmes principes. Un pré-calcul des classes d'homotopie présente donc un intérêt important pour répondre aux deux problèmes liés à une adaptation directe des méthodes de planification de trajectoires existantes. Il permet de :

- **répertorier** tous les chemins pour prototypage d'une tâche de déplacement d'un objet ($X_G \subset SE(3)$ ou \mathbb{R}^3). Dans le cas d'une préparation d'un chantier, le choix des engins, les critères d'optimisation de trajectoires, les lieux de stockage du matériel ou le placement de protections biologiques (dont on doit éviter la collision) sont des contraintes qui sont impossibles à fixer a priori (cf. §I). Il est nécessaire d'avoir une vision exhaustive des déplacements possible d'un objet avant de pouvoir définir les contraintes liées à sa manipulation ou à l'optimisation de celle-ci.
- **anticiper** certains écueils des approches de planification locale comme les minima de potentiels, les blocages ou les passages étroits des échantillonnages probabilistes. Le fait de répertorier toutes les classes d'homotopie d'un problème ou d'un sous-problème permet à l'utilisateur de sélectionner une classe donnée avant tout traitement additionnel ce qui contribue donc à une meilleure maîtrise du temps de réponse des outils de planification de trajectoire.

3.3 Existence d'une représentations des classes d'homotopie

Si l'on cherche à référencer les classes d'homotopies de deux points connexes de X_G , la donnée d'une décomposition semi-algébrique telle que définie précédemment se révèle insuffisante. Il faut tout d'abord imposer aux sous-ensembles algébriques d'être des cellules pour préciser leur nature topologique. La structure de CW -complexe^{III.3.2&A3} peut alors être utilisée. Les propriétés de CW -complexe ont déjà été introduites dans le cadre des méthodes de rétraction [ODU85] pour des problèmes de planification de mouvement de primitives simples (disque) dans le plan. Ils permettent, d'un point de vue théorique, de mettre en évidence l'existence de représentations finies de la connexité et de l'homotopie. Nous l'utiliserons par la suite dans un autre contexte en ne construisant que le CW -complexe d'espaces des configurations (et pas ceux, inaccessibles en pratique, associés aux espaces des configurations libres). Nous illustrerons cette démarche en retrouvant les groupes fondamentaux d'homotopie des espaces de configurations des liaisons usuelles et nous expliquerons dans la partie suivante la démarche que nous proposons pour organiser l'étude d'un espace des configurations libres à partir d'une construction explicite du CW -complexe de l'espace des configurations correspondant.

Définition 2.10 : Cellule. Pour $n \in \mathbb{N} - \{0\}$, une *cellule* de dimension n (ou encore n -cellule) est un espace topologique homéomorphe à une boule de dimension n B_n . Une *cellule ouverte* de dimension n (ou encore n -cellule ouverte) est un espace topologique homéomorphe à $B_n - S_{n-1}$.

Les décompositions obtenues sont appelées dans ce cas des décompositions cellulaires. La structure de CW -complexe fini peut alors être construite.

Définition 2.11 : CW -complexe fini. Un espace topologique X muni d'une famille $(X^{(n)})_{n \in \mathbb{N}}$ de sous-espaces est un CW -complexe si, avec la convention que $X^{(-1)} = \emptyset$, $X^{(n)}$ est obtenu par *recollement de cellules*² de dimension n à partir de $X^{(n-1)}$.

La définition d'un recollement de cellules et quelques-unes des propriétés des CW -complexes sont présentées dans l'appendice de ce chapitre. Les exemples présentés ci-dessous mettent en évidence la structure de CW -complexe fini d'un cercle et de la surface d'un tore.

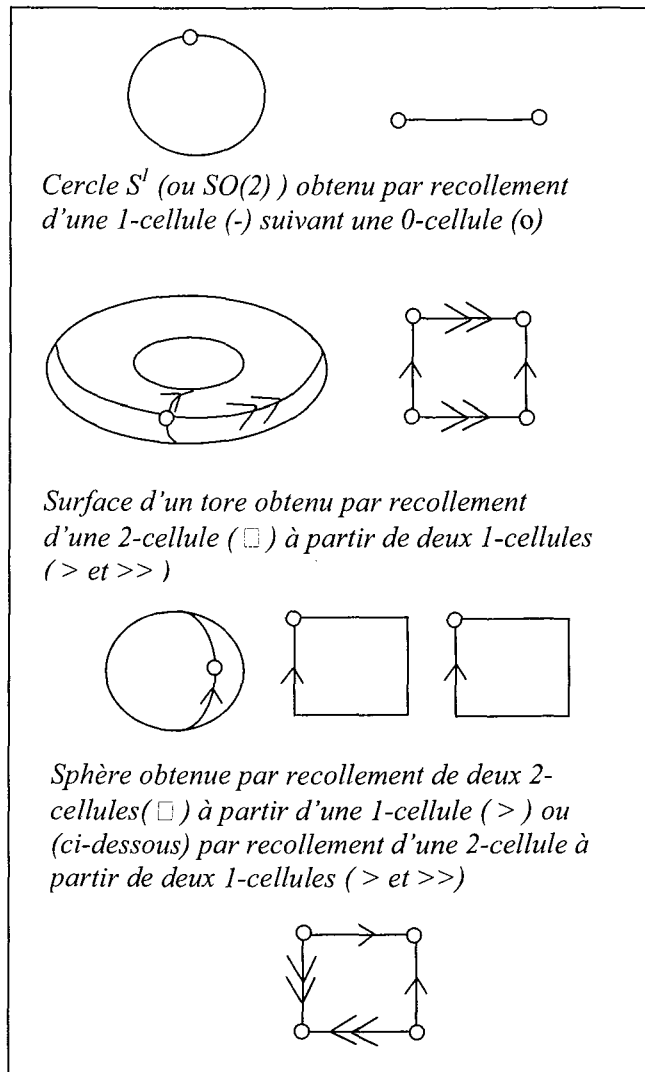


Figure 2.14 : Exemples de recollements de cellules

Les propriétés essentielles que l'on peut déduire de cette structure, dans le cadre de ce mémoire, est qu'il est possible de définir puis de construire :

- **une représentation finie des composantes connexes** (au sens de la *topologie faible*^{A3}) en s'appuyant sur le recollement de $X_G^{(n)}$ suivant $X_G^{(n-1)}$. Le graphe dont les nœuds représentent les cellules de $X_G^{(n)}$ et dont les arcs représentent les relations de recollement suivant les cellules de $X_G^{(n-1)}$ est appelé graphe de connexité. Sa construction à partir de la structure de *CW-complexe* est immédiate.
- **une (re)présentation finie du groupe fondamental d'homotopie**, en utilisant le fait que les groupes fondamentaux d'homotopie de X_G et de son 2-squelette $X_G^{(2)}$, qui est un sous-*CW-complexe* (fini) de X_G , sont isomorphes :

$$\pi_1(X_G, x) \approx \pi_1(X_G^{(2)}, x)$$

Ce groupe est isomorphe à un *groupe libre de type fini*^{A4} (typiquement Z ou $Z/2Z$ pour les liaisons usuelles cf tableaux 2.7 et 2.8), construit à partir des relations d'homotopies associées aux 2-cellules du CW-complexe (cf. appendice). Etant donné un lacet, sa classe d'homotopie en tant qu'élément d'un groupe libre de type fini peut alors être décomposée sur un ensemble générateur (fini) du groupe. Si l'élément considéré est l'identité, le lacet est réductible.

Néanmoins, du fait du nombre de cellules nécessaires dans le cas des milieux industriels complexes, la « construction » effective de telles représentations ne peut être mise en œuvre dire pour l'espace des configurations libres.

3.4 Mise en œuvre d'une capture des classes d'homotopie

Nous abordons dans cette partie la possibilité de représenter les caractéristiques globales des configurations libres qui ont un intérêt direct pour les problèmes de planification et d'optimisation de trajectoires. Le fait de pouvoir savoir, pour toute paire de configurations, si celles-ci sont connexes et quelles sont leurs différentes classes d'homotopie est fondamental pour savoir si une trajectoire solution existe et organiser la recherche d'une trajectoire optimale en un temps limité. L'objectif est d'avoir accès à ces caractéristiques sans passer par une construction exacte des configurations libres.

Pour ce faire, la structure de CW-complexe peut être construite de manière explicite pour certains espace des configurations (i.e. sans présence d'obstacles) comme celui d'un solide en contact planaire $X_g \approx SE(2)$, celui d'un solide se déplaçant librement dans l'espace $X_g \approx SE(3)$ ou plus généralement un espace des configurations de type $X_g \approx SO(d)^{n_1} \times [0,1]^{n_2}$ correspondant à un arbre de liaisons. Dans chacun de ces cas, la structure de CW-complexe peut être mis en évidence et les cellules composant les CW-complexes peuvent être définies et paramétrées (cf. annexe 2) de manière explicite. L'exemple ci-dessous nous permet d'illustrer la nature du groupe fondamental d'homotopie de $SO(3)$. Nous en déduirons celles des liaisons usuelles.

Exemples de construction du groupe fondamental d'homotopie d'espace des configurations

Si l'on représente un élément de $SO(3)$ par un vecteur de \mathbf{R}^3 représentant un axe de rotation, de longueur de $-\pi$ à π pour l'angle de la rotation, $SO(3)$ est représenté par l'intérieur d'une sphère de \mathbf{R}^3 . Les points de la sphère symétriques par rapport au centre sont confondus (par recollement). De cette manière tout chemin traversant la surface de la sphère réapparaît du côté opposé par rapport au centre de la sphère. Une boucle simple traversant la surface sphérique (a) ne peut être déformée en un point, alors qu'une double boucle (b) peut l'être, comme le montre les étapes (c,d,e) de la figure suivante:

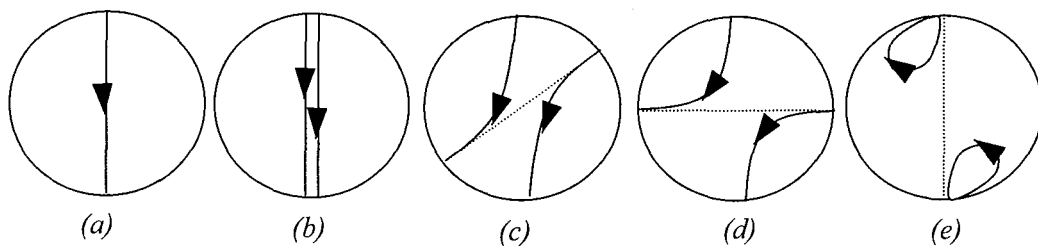


Figure 2.15: Mise en évidence de $\pi_1(SO(3)) \approx \mathbf{Z}/2\mathbf{Z}$

En répétant ce procédé on montre que pour $n \in \mathbb{N}$, $2n$ boucles sont équivalentes par déformation à un point et $2n+1$ boucles à une unique boucle. Le groupe fondamental d'homotopie de $SO(3)$ est isomorphe à $\mathbb{Z}/2\mathbb{Z}$. Le tableau ci-dessous fournit quelques groupes d'homotopie d'ensembles simples. On peut utiliser la relation $\pi_k(M \times N) \approx \pi_k(M) \times \pi_k(N)$ ($k=1$ pour le groupe fondamental d'homotopie) pour en construire d'autres.

	π_1	π_2	π_3	π_4
\mathbf{R}^n	0	0	0	0
$SO(2) \approx S^1$	\mathbf{Z}	0	0	0
$SO(3)$	$\mathbf{Z}/2\mathbf{Z}$	0	\mathbf{Z}	$\mathbf{Z}/2\mathbf{Z}$
S^2	0	\mathbf{Z}	\mathbf{Z}	$\mathbf{Z}/2\mathbf{Z}$
S^3	0	0	\mathbf{Z}	$\mathbf{Z}/2\mathbf{Z}$

Tableau 2.7 : Quelques groupes d'homotopies

On obtient alors facilement les groupes d'homotopie des espaces de configuration des liaisons usuelles.

<i>Liaisons bilatérales</i>	X_g	π_1
<i>Libre</i>	$\mathbf{R}^3 \times SO(3)$	$Z/2Z$
<i>Ponctuelle plan ou surfacique</i>	$\mathbf{R}^2 \times SO(3)$	$Z/2Z$
<i>Linéique rectiligne ou curviligne</i>	$\mathbf{R}^2 \times SO(2) \times SO(2)$	$Z \times Z$
<i>Linéique annulaire</i>	$\mathbf{R} \times SO(3)$	$Z/2Z$
<i>Rotule</i>	$SO(3)$	$Z/2Z$
<i>Appui plan</i>	$\mathbf{R}^2 \times SO(2)$	Z
<i>Pivot glissant</i>	$\mathbf{R} \times SO(2)$	Z
<i>Glissière</i>	\mathbf{R}	Id
<i>Pivot</i>	$SO(2)$	Z

Tableau 2.8: Espaces des configurations usuels pour une liaison bilatérale et nature de leurs groupes d'homotopie

L'objectif est de représenter les composantes connexes et groupes d'homotopies de l'espace des configurations libres (et pas seulement de l'espace des configurations). Nous proposons dans cette partie une simplification de la construction des groupes fondamentaux d'homotopie des espaces de configuration libre en se ramenant à une cellule de l'espace des configurations.

Nous pouvons nous appuyer sur les propriétés des CW-complexes (cf. annexe 3) pour proposer la démarche suivante qui permet de construire une représentation de la connexité et du (ou des) groupes d'homotopie des espaces de configurations libres d'espace de type $X_g \approx SO(d)^{n_1} \times [0,1]^{n_2}$:

- On extrait (cf. 2.3) une cellule paramétrée de l'espace des configurations. Cette démarche peut être menée à son terme pour des espaces de configuration de type $X_g \approx SO(d)^{n_1} \times [0,1]^{n_2}$. Notons C l'ensemble des paramètres de la cellule considérée.

- On introduit les obstacles. Notons C_{libre} le sous-ensemble de C des coordonnées correspondant à des configurations sans collisions (que l'on ne construit pas).
- on capture les composantes connexes de C_{libre} au moyen d'un 1-complexe simplicial $K^{(1)}$ (cf. annexe 3)
- pour chaque composante connexe on cherche le groupe d'homotopie en complétant le 1-complexe simplicial $K^{(1)}$ en un 2-complexe simplicial $K^{(2)}$ jusqu'à ce que

$$\pi_1(C_{libre}, q) = \pi_1(K^{(1)}, q)$$
- on recolle chaque cellule de manière à construire globalement les composantes connexes et les groupes d'homotopies (on s'appuie alors conjointement sur les propriétés de CW-complexe et le *théorème Van Kampen* (cf. [PAU02]) qui garantissent que cette démarche n'est pas vaine)

La construction effective des complexes $K^{(1)}$ et $K^{(2)}$ peut en théorie correspondre à une triangulation de Delaunay. Mais, du fait des limites algorithmes dues à la dimension, cette procédure est inapplicable et nous lui consacrerons en fait le prochain chapitre §III en montrant que l'on peut tirer parti de méthode à réseau probabiliste MRP-Visi [SIM00].

Conclusion

Nous avons vu dans ce chapitre §II que toutes les contraintes géométriques caractéristiques d'un chantier nucléaire pouvaient se mettre sous une forme implicite dont on a rappelé qu'elle correspondait à une formule de Tarski. Ce faisant, nous avons pu rappeler qu'un échantillonnage fini, où chaque échantillon est associé à une cellule, suffisait pour représenter de manière exacte l'espace des configurations libres X_G que l'on peut associer à une ou plusieurs tâches ou opérations figurant une phase critique d'un chantier. Ceci nous a permis de rappeler certains résultats fondamentaux :

- le nombre de cellules d'une décomposition algébrique de l'espace des configurations libres X_G rend leur utilisation directe impossible (§II.2.2), en pratique il convient de se placer dans une unique variété de l'espace des configurations et d'explorer l'espace des configurations libres à l'aide de sous-ensembles de faible dimension (la dépendance en la dimension du temps mis pour résoudre un problème de détection de collision étant extrêmement défavorable),
- une décomposition cellulaire de l'espace des configurations X_g peut conduire à des cellules sous une forme implicite qui ne peuvent être paramétrées dans le cas de systèmes de contraintes géométriques formant une ou plusieurs boucles dans le graphe des liaisons (§II.2.2). Dans ce cas il n'existe pas de méthodes génériques pour générer des configurations et *a fortiori* des chemins de configurations.

Les captures, lors d'un pré-calcul, des propriétés topologiques globales de X_G (au premier rang desquelles on trouve la connexité et les classes d'homotopie) ont un intérêt pratique direct pour la préparation rapide de tâches et d'opérations d'un chantier (§II.3.2). Compte tenu du nombre restreint d'études à propos de la capture des classes d'homotopie, nous avons rappelé que celle-ci, au même titre que la capture de la connexité, n'est pas vaine. En effet :

- il existe une représentation finie de la connexité entre les cellules à l'aide d'un graphe de connexité (§II.3.3).
- Il existe une représentation finie de l'homotopie au moyen de 2-CW-complexes (§II.3.3),

Mais, du fait des problèmes de paramétrisation, nous nous sommes restreint au cas des sous-systèmes de contraintes définissant chacun une cellule de X_g qui peut être paramétrée par un pavé de \mathbb{R}^n , à l'instar du cas standard d'un mécanisme à liaisons acycliques dont les coordonnées articulaires sont bornées. Dans l'optique d'un pré-calcul d'une représentation de la connexité et des classes d'homotopie de X_G , nous avons alors proposé une démarche s'appuyant sur la construction des groupes d'homotopies associés à chaque cellule et sur les propriétés de CW -complexe pour obtenir des résultats globaux sur la caractérisation de la connexité et de l'homotopie de l'espace des configurations libres (§III.3.3) dans le cas d'espaces de configurations de type $X_g \approx SO(d)^{n_1} \times [0,1]^{n_2}$ décomposables directement en cellules paramétrées. Cette restriction sur le type des espaces de configuration couvre néanmoins les premières étapes de la préparation d'un chantier comme l'étude géométriques des tâches et elle inclut en particulier le cas d'un ou plusieurs solides en déplacement libre ou planaire mais aussi les situations où le graphe des liaisons est acyclique. Cette formulation canonique permet ainsi d'envisager la mise en place d'un outil générique (qui ne dépend pas de la nature topologique de l'espace des configurations) de planification de trajectoires sans collision, outil qui est capable de faire face à la grande diversité des contraintes géométriques rencontrées durant la préparation d'un chantier.

Pour mener la démarche proposée à son terme, nous devons préciser la méthode de construction des complexes simpliciaux utilisés dans §II.3.4 sachant que les résultats présentés dans ce chapitre permettent de décomposer formellement les problèmes mais que la complexité algorithmique intrinsèque du problème global demeure. Nous abordons ce point dans le chapitre suivant en considérant les techniques récentes de détection de collision et de construction de réseaux probabilistes dans le cadre que nous venons de définir.

CHAPITRE III

Caractérisation des configurations libres d'une cellule au moyen d'un réseau probabiliste

Capture des composantes connexes et de leurs classe d'homotopies

Pour des études de faisabilité de tâches ou d'opérations d'un chantier, le précalcul d'une structure représentant de manière exhaustive les composantes connexes et les classes d'homotopies de l'espace des configurations libres est une condition nécessaire à une planification interactive du mouvement étant donné le caractère répétitif des études et l'évolution des contraintes. Nous avons montré, dans le chapitre précédent (§II), qu'il était possible de décomposer des espaces des configurations de type $SO(d)^m \times [0,1]^n$ en cellules paramétrables et, pour chaque cellule C , de représenter les caractéristiques topologiques recherchées pour C_{libre} .

Nous présentons dans ce chapitre (plus précisément dans §III.3) une nouvelle méthode que nous appelons MRP-CH de construction d'une telle représentation. Elle s'inscrit dans la démarche des Méthodes à Réseaux Probabilistes (présentée dans §III.2) qui permettent de capturer les composantes connexes de C_{libre} mais en l'étendant à la capture des classes d'homotopie.

L'efficacité des méthodes à réseaux probabilistes repose dans une large mesure sur l'existence de procédures efficaces de détection de collision pour une configuration (dimension 0) ou un chemin de configurations (dimension 1). La notion d'homotopie utilisée par MRP-CH requiert quant à elle une nouvelle procédure de détection de collision pour une surface (dimension 2) de configurations.

Pour mener à bien la construction d'un réseau construit par les Méthodes à Réseaux Probabilistes capturant la connexité ou par MRP-CH, nous devons préalablement considérer les procédures de détection de collision (§III.1) qui constituent la clef de voûte de ces algorithmes.

1 Détection de collisions et de distances

La plupart des logiciels de CAO ou de CFAO actuellement utilisés dans le cadre de la préparation de chantiers n'offrent que des outils de détection de collision pour une configuration donnée de solides dans l'espace. Qui plus est, leur calcul reste lent et ne peut être utilisé pour la simulation interactive de phénomènes physiques (déformation d'un pneu au contact d'un sol irrégulier, déformation d'un organe pour la simulation d'une opération chirurgicale,...) ou pour la planification de trajectoires sans collision dans des environnements industriels complexes. De plus, la détection de collision au cours d'un mouvement, ou plus généralement pour un ensemble « continu » de configurations ne peut se réduire directement au test d'un nombre fini de configurations.

Nous consacrons donc cette partie à une revue des techniques efficaces de détection de collision pour une configuration donnée puis pour un ensemble continu de configurations. Ces techniques conditionnent très largement le succès des méthodes de planification de trajectoires. Elles doivent donc être préalablement implantées dans les logiciels de CAO ou de CFAO qui veulent intégrer des outils de planification de trajectoires.

1.1 Détection de distance pour une configuration

1.1.1 Collision statique

Le problème de détection de collision de N solides mobiles doit *a priori* être décomposé en considérant toutes les paires de solides mobiles les uns par rapport aux autres et susceptibles d'entrer en collision. Nous allons traiter le cas d'une paire de solides en mouvement l'un par rapport à l'autre, avant de généraliser. Pour ce faire, nous définirons dans une première partie une distance entre deux

solides à un instant donné (*i.e.* une configuration donnée) et donnerons quelques méthodes de calcul pour en obtenir plus rapidement une sous-estimation en bénéficiant des techniques de hiérarchisation.

A notre connaissance (une synthèse est présentée dans [LIN98]), toutes les procédures de détections de collisions se ramènent *in fine* aux tests de deux primitives convexes. La nature des primitives utilisées varie. Elle peut être volumique (CSG, polyèdre) ou surfacique (triangle, carreaux paramétrés : NURBS, B-Spline,..) et plus ou moins structurée (B-rep' ou « soupe de triangles »). Les algorithmes de détection de collision pour deux primitives convexes utilisent des procédures itératives qui peuvent s'arrêter de plusieurs façons distinctes :

- lorsqu'une condition suffisante de non collision (ou de collision) est mise en évidence (*figure 3.1*),
- lorsque l'estimation de la distance $\tilde{d}_{i,j}$ entre les deux primitives (ou de la pénétration) est jugée suffisamment bonne (*figure 3.1*).
- lorsque l'on cherche la distance exacte $d_{i,j}$ entre les deux primitives (*figure 3.2*)

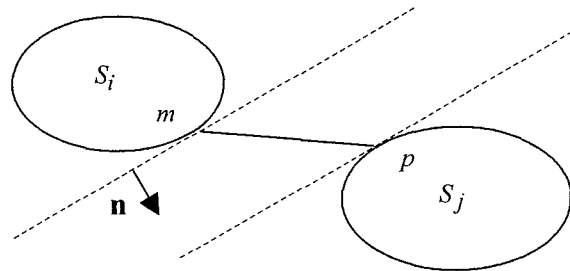


Figure 3.1: Condition suffisante de non pénétration de deux solides convexes

$$\exists n, \|n\| = 1 / \inf_{\substack{m \in S_i \\ p \in S_j}} n^T (p - m) > 0$$

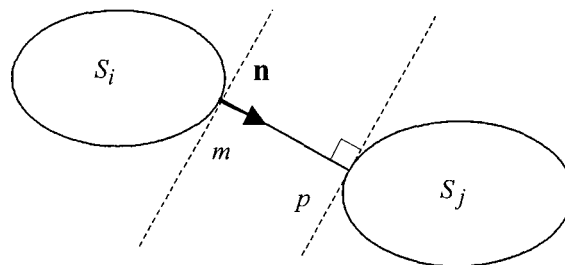


Figure 3.2: Distance entre deux solides convexes

$$d_{i,j} = \sup_{\|n\|=1} \inf_{\substack{m \in S_i \\ p \in S_j}} n^T (p - m)$$

Dans le premier cas la détection de collision statique proprement dite est plus rapide alors que dans le second, une bonne approximation de la distance peut être exploitée, comme nous le verrons dans la prochaine partie pour la détection de collision pour un ensemble continu de configurations. Il existe deux procédures majoritairement utilisées pour la détection de collision statique de deux polyèdres convexes :

- dans [LIN91] et pour deux primitives polyédriques, les auteurs utilisent les *zones de Voronoï* associées aux éléments de la surface du polyèdre (sommet, arête ou face) pour guider le déplacement de deux points situés respectivement sur les surfaces des solides jusqu'à ce qu'ils convergent vers le couple de points les plus proches.
- dans [GIL88] décrivant l'algorithme GJK, les auteurs considèrent deux polyèdres convexes définis respectivement par l'enveloppe convexe de deux familles de points $\{m_i\}_{i=1..M}$ et $\{p_j\}_{j=1..P}$. Le problème de détection de collision est alors formulé dans \mathbb{F}^6 pour un polyèdre E de $M \times P$ sommets $T = \{E_{i,j} = m_i - p_j\}$ (somme de Minkowski). Il y a collision si l'origine de \mathbb{F}^6 appartient au polyèdre E . Le problème se ramène alors à la résolution d'un système linéaire en extrayant un simplexe de dimension au plus 6 pour lequel on peut tester la position de l'origine. Le simplexe est réactualisé de manière itérative jusqu'à converger vers un unique point du polyèdre E qui correspond au couple de points les plus proches.

Ces deux procédures peuvent s'arrêter suivant les trois critères présentés ci-dessus. Néanmoins, leur utilisation pratique n'est possible que sous deux conditions :

- lorsque les algorithmes sont initialisés (par simplexe de \mathbb{F}^6 pour GJK ou une paire de primitives pour LIN) de telle manière que quelques itérations suffisent pour vérifier la condition d'arrêt choisie. C'est le cas lorsque l'on teste la collision de configurations voisines (on parle de condition de « cohérence temporelle »).
- Lorsque le problème comprend peu de solides (convexes) car toutes les paires de solides (convexes) doivent être considérées séparément.

Pour pallier ces deux conditions trop strictes, nous présentons dans la partie suivante les techniques de hiérarchisations par volumes englobants qui complètent idéalement ces deux procédures de détections de collision pour deux polyèdres convexes.

1.1.2 Décompositions convexes hiérarchisées

Nous avons formulé dans le chapitre précédent (§II.1.2) le problème de détection de collision de deux solides pour une configuration donnée. L'évaluation de la valeur logique de la *formule de Tarski*^{II.2} associée à une détection de collision se ramène, après une décomposition semi-algébrique, à vérifier un nombre fini de cellules qui sont chacune caractérisée par une valeur logique constante de la formule de Tarski de détection de collision. Le nombre de cellule dépend du nombre et de l'ordre des polynômes définissant chaque solide (§II.2.2). De manière évidente, plus le nombre de cellules est important, plus la borne supérieure du temps mis pour décider si les deux objets sont en collision est importante. Nous verrons néanmoins dans ce paragraphe qu'une hiérarchisation des volumes permet de diminuer radicalement le temps mis pour détecter l'éventuelle non-collision de deux solides « suffisamment éloignés » pour que la simple considération de leurs enveloppes grossières suffise pour conclure.

En effet, la plupart des procédures de détection de collision existantes pour une configuration donnée s'appuient sur la recherche de conditions suffisantes de non-collision (ou de collision) à l'aide de volumes englobants (ou de volumes englobés) définis par un nombre restreint de polynômes d'ordres limités. Si les volumes englobants de deux solides ne sont pas en collision, c'est *a fortiori* le cas des solides. De nombreuses classes de volumes englobants ont ainsi été testées intensivement ces dernières années dont les « Axe Aligned Bounding Boxes » AABB [COH95]. Ces dernières ont été généralisées en volumes englobants ayant un nombre discret d'orientations possibles de leur faces « k-DOP » [KLO98]. Les « Oriented Bounding Boxes » OBB [GOT96] et ceux utilisant des sphères [HUB95] ou des polyèdres dilatés [LAR00]. La détection de collision ou de distance de deux volume englobants de même type est extrêmement rapide et les procédures plus générales décrites dans le cas de polyèdres quelconques (§III.1.1.1) se réduisent à un nombre très limité d'opérations (comme le « separating axis theorem » utilisé comme condition suffisante de non-collision de deux volumes OBB dans [GOT96]).

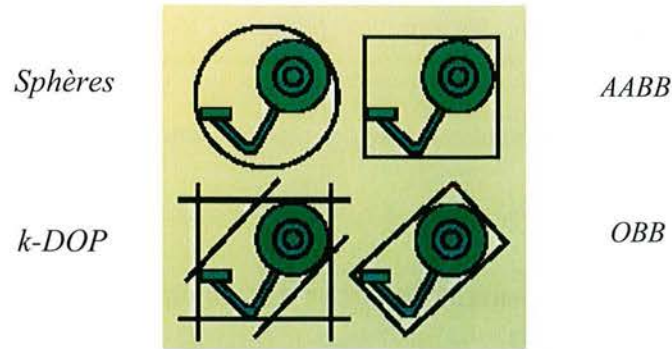


Figure 3.3: Exemple de volumes englobants

A chacune de ces classes correspondent des algorithmes qui construisent une hiérarchie de volumes englobants (cf. par exemple [GOT96]). Il existe deux stratégies de construction d'arbres (généralement binaire) de volumes englobants :

- soit en groupant deux primitives ou deux volumes englobants de manière itérative jusqu'à obtenir un volume englobant global,
- soit, inversement, en divisant en deux un volume englobant en partant d'un volume englobant global et en affinant à chaque étape l'approximation des primitives de chaque partie au moyen d'un nouveau volume englobant.

Disposant d'une hiérarchisation et d'une procédure de détection de l'interférence de deux volumes englobant ou de deux primitives, les tests de collisions de deux arbres de hiérarchisation correspondent à la procédure suivante

```

Algorithme d'interférence de 2 arbres hiérarchiques (booléen)
I2AH(A,B)
Si  $\neg$ interférence(A,B) retourner faux
Sinon Si A est une feuille
    Si B est une feuille
        Pour toute paire de primitives ( $P_a \in A, P_b \in B$ )
            Si interférence( $P_a, P_b$ ) retourner vrai
    Sinon
        Pour tous fils  $F_b$  de B
            I2AH( $F_b, A$ )
Sinon
    Pour tous fils  $F_a$  de A
        I2AH( $B, F_a$ )
Retourner faux
    
```

A l'issue de cette procédure, les paires de primitives identifiées sont traitées au moyens des procédures présentées dans §II.1.1.1. Ces procédures ont une complexité moyenne en temps bien inférieure à celle que l'on peut escompter au vue des résultats dans le pire des cas (§II.2).

Les **calculs de distances peuvent bénéficier de techniques similaires** si l'on recherche une estimation par valeurs inférieures $\tilde{d}_{i,j}$. Pour ce faire, la paire de volume englobant correspondant à la plus petite distance doit être prioritairement développée (*i.e.* par passages aux volumes fils d'un des deux volumes alternativement). L'arrêt de la recherche de distance s'effectue par exemple lorsque celle-ci est jugée suffisamment grande ou que sa valeur n'évolue plus suffisamment pour être significative (cf. deuxième critère d'arrêt de §I.1.1.1). Nous reviendrons sur ce point important en le précisant dans la partie §I.2.2.

Les constructions de hiérarchie sont lancées une seule fois pour chaque solide (et se « déplacent avec lui ») dans le cas des OBB et des Spheres ou doivent être réactualisées à chaque déplacement d'un solide dans le cas des cubes réguliers comme les octree, des AABB ou des k-DOP. Dans ce dernier cas, la hiérarchie peut grouper des solides voisins et limiter le test systématique de $N(N-1)/2$ paires possibles concernant N solides mobiles, ce qui compense le coût d'une réactualisation des volumes englobants. Les performances obtenues pour les différents types de volumes englobants présentent une grande diversité suivant le type d'environnement et le nombre d'objets mobiles.

Nous résumons dans le tableaux présenté ci-dessous quelques outils de détection de collision développés au moyen des tests de collision et des techniques de hiérarchisation présentées.

I-Collide (95)	Liste d'AABB contenant des polyèdres convexes, paires traitées par marche incrémentale guidée par zone de Voronoi extérieure
V-Clip (97) :	Liste de polyèdres convexes traités par marche incrémentale guidée par zone de Voronoi extérieure ou intérieure
Q-collide (96) :	Liste d'AABB contenant des polyèdres convexes plan de séparation entre polyèdres convexes
Solid (98) :	Arbre d'AABB avec des primitives CSG pour feuilles, paires traitées par GJK
QuickCD (98) :	Arbre de k-DOP avec des triangles pour feuilles (d'où non-convexes)
V-collide (97) :	Arbre d'OBB avec des triangles pour feuilles (d'où non-convexes)
PQP (99) :	Arbre d'OBB dilatées

Tableau 3.1 : Outils de détection de collision utilisant des hiérarchisation convexes

Les spécificités des maquettes numériques de centrales nucléaires caractérisées par une forte complexité de l'environnement statique font qu'aucun outil de détection de collision présenté dans le tableau 3.1 ne peut être utilisé tel quel. Un nouveau détecteur de collision KCD [GEE00] a donc été développé dans le cadre du projet MoLog pour faire face à la complexité des environnements de centrales nucléaires. Nous en décrivons les grandes lignes dans cette partie. Notre rôle s'est borné à le tester.

KCD permet, pour un état géométrique donné, de répondre à deux types de requêtes. La première indique si aucun objet est en collision (booléen) la seconde fournit une estimation des distances entre les agrégats « rigides » d'objets mobiles les uns par rapport aux autres (dont la scène statique). Il utilise différentes techniques issues de l'expérience acquise par le LAAS une utilisation intensive des détecteurs de collisions I-Collide [COH95], V-Collide [HUD97] pour la planification de mouvement dans Move3D [SIM01].

KCD exploite la complémentarité des techniques de hiérarchisation AABB et OBB et l'efficacité de l'algorithme GJK pour le calcul précis de distance. KCD construit lors de son initialisation une hiérarchie de boîtes englobantes à axes alignés AABB (pour Axis Aligned Bounding Box) associée à l'agrégat des objets statiques de la scène. Les objets rigides mobiles (constituant les corps d'engins articulés ou d'éventuels objets déplaçables) sont enveloppés chacun d'une unique boîte du type AABB. Ces boîtes, contrairement à celles des objets statiques de la scène, sont réactualisées (en dimension et en position) suivant le placement des objets mobiles à chaque nouvelle requête. Le test des paires de boîtes AABB-AABB en collision permet une première sélection rapide. Le traitement de la requête fait ensuite appel à des hiérarchisations plus fine basée sur des arbres de boîtes englobantes OBB (pour oriented bounding box). Ces arbres sont construits durant l'initialisation de KCD dans chaque repère d'objet rigide mobile et dans le repère global suivant la nature de l'objet considéré (statique ou mobile). La position (mais pas la dimension puisque les boîtes peuvent être orientées) des OBBs des objets mobiles est réactualisée et les paires de boîtes OBBs en collisions peuvent toutes être identifiées durant cette étape. Enfin, après ces deux étapes, les primitives contenues dans chacune des paires d'OBB sont traitées par l'algorithme GJK amélioré [CAM97, BER99]. Il permet de renvoyer une réponse booléenne ou une liste de distances suivant la nature de la requête.

1.2 Collision de courbes et de surfaces paramétrées

Nous abordons dans ce paragraphe les procédures de détection de collision pour un ensemble de configurations de dimension 1 (une courbe) puis 2 (une surface). Ces procédures conditionnent l'efficacité d'une exploration des configurations par un réseau de trajectoires que nous aborderons dans les paragraphes suivants.

1.2.1 Approches existantes pour les courbes

De nombreuses techniques de détection de collision de solides en mouvement ont été développées ces dernières années pour répondre aux besoins divers de la CAO et de la simulation. Une synthèse est présentée dans [JIM01].

Une première approche consiste à considérer l'interférence des volumes 4D de chaque solide représenté dans un espace spatio-temporel [CAM90]. Cette approche peut s'intégrer dans le cadre théorique général du chapitre §II lorsque les chemins sont définis à l'aide de contraintes semi-algébriques. Les opérations booléennes définissant les formes de chaque solide définissent alors des extrusions en 4D. Certains auteurs (par exemple [FOI93]) considèrent uniquement la projection de ce volume 4D en 3D donnant le volume balayé. Dans ce cas toutes les collisions sont détectées mais certaines détections inexactes. Une solution consiste à considérer toutes les paires. Le principal désavantage de ce type d'approche est de fixer le type de mouvement de chaque solide pour créer et mettre en évidence des propriétés géométriques 4D. Dans [RED02] le principe des axes de séparation est étendu à deux OBB ayant un mouvement relatif de vissage et permet de bénéficier des techniques de hiérarchisations.

Une seconde approche s'appuie sur les techniques de détection de collision statique présentées dans la partie précédente (*cf.* §I.1). Les paires de points de chaque solide, lorsqu'elles sont initialisées au voisinage de la situation satisfaisant le critère d'arrêt peuvent répondre rapidement. Ainsi pour des échantillons pris successivement sur la courbe, ce phénomène est pleinement exploité, on parle de cohérence temporelle. Cette technique n'a pas besoin de hiérarchisation des solides.

Une troisième approche exploite le calcul des distances entre les solides à une configuration donnée pour définir une zone sans collision autour de cette configuration. Dans le cadre de la planification de trajectoire par pré-calcul probabiliste d'un réseau, nous optons pour ce type d'approche de détection de collision car :

- elle peut bénéficier de l'accélération due à la hiérarchisation dans le cas d'un échantillonnage de configuration « distantes les uns des autres » comme c'est le cas lors d'un échantillonnage probabiliste,
- elle peut avoir la faculté de rejeter rapidement une courbe en collision car elle n'est pas astreinte à chercher précisément la première occurrence d'une collision sur un chemin,
- elle s'adapte assez facilement à la détection de collision de courbe quelconque (paramétrée) et surtout, dans le cas qui nous intéresse, elle peut facilement se généraliser à la détection de face ce qui n'est pas le cas des deux autres approches.

1.2.2 Détection de collision pour des arêtes et des triangles

Ce chapitre se place dans le cas de configurations réduites à une cellule paramétrée et d'un système sans autres contraintes que les contraintes géométriques, les configurations ponctuelles, les chemins ou les surfaces de configurations peuvent être construits à partir de *simplexes*^{A4} dans l'espace affine des paramètres. (point, arête et triangle respectivement). Ce choix n'est pas arbitraire, les simplexes permettront de préciser la nature topologique des réseaux à construire en utilisant les propriétés des complexes simpliciaux (*cf. §II.3 et annexe 4*). Nous détaillons donc dans ce paragraphe les procédures de détections de collision pour les arêtes et les triangles qui seront utilisées dans les parties suivantes §III.2 et §III.3.

Remarque : Si le système comprend d'autres contraintes que les contraintes géométriques, nous considérons qu'elles sont appliquées dans un second temps en déformant localement les trajectoires soumises aux seules contraintes géométriques (*cf. II.3.1*). Dans ce contexte le choix des simplexes n'est nullement restrictif par rapport aux systèmes considérés.

Nous allons considérer les variations de la distance entre deux solides mobiles l'un par rapport à l'autre puis déterminer (estimer) le rayon d'une boule sans collision autour d'une configuration

comme dans [FAV87] et enfin utiliser une série de telles boules pour recouvrir une arête et conclure quant à l'existence ou non d'une collision sur l'arête. Nous généralisons cette approche aux triangles de configurations. Revenons sur le schéma précédemment utilisé dans le cas d'une détection de collision statique en considérant cette fois-ci les variations liées au mouvement relatif des deux solides :

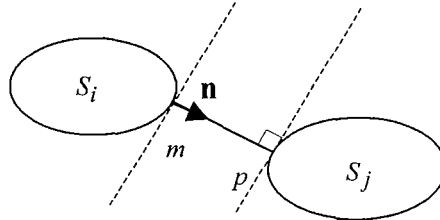


Figure 3.4: Distance entre deux solides convexes

$$d_{i,j} = \sup_{\|n\|=1} \inf_{\substack{m \in S_i \\ p \in S_j}} n^T (p - m)$$

Les points p et m parcourent respectivement les bords des solides S_i et S_j lorsque ceux-ci se déplacent. Le vecteur unitaire n est quant à lui normal aux deux surfaces en p et m . L'expression scalaire $n^T (p - m)$ qui vérifie (3.1) peut être dérivée et exprimée dans le référentiel i en donnant $\dot{n}_i^T (p_i - m_i) + n_i^T (\dot{p}_i - \dot{m}_i)$. Le premier terme est nul par définition. Pour le second terme, on décompose les vitesses en vitesse du point coïncidant et vitesse relative au solide. $\dot{p}_i = \dot{p}'_i + \dot{p}''_i$ et $\dot{m}_i = \dot{m}'_i + \dot{m}''_i$ les vitesses relatives étant tangentes aux surfaces des solides et donc normales à n , les termes en \dot{p}''_i et \dot{m}''_i disparaissent ainsi que \dot{m}'_i qui est nulle dans le référentiel i . Il reste donc :

$$\dot{d}_i = n_i \dot{p}'_i \quad (3.1)$$

Nous allons voir maintenant comment exprimer la vitesse du point coïncidant \dot{p}'_i en fonction de la configuration du système et de sa première dérivée.

Les éléments $g_{i,j}$ de $SE(3)$ permettant de placer deux solides l'un par rapport à l'autre peuvent être vus comme des transformations entre deux repères $g_{o,i}$ et g_o , en suivant la relation de groupe $g_{i,j} = g_{i,o} \cdot g_{o,j} = g_{o,i}^{-1} \cdot g_{o,j}$. Cette relation de groupe permet de définir de manière matricielle les torseurs cinématiques (cf. annexe 2) par $\dot{g}_{o,i} = g_{o,i} V_{o,i}$ (convention gauche) avec $\dot{g} \in T_g SE(3), V_{o,i} \in T_{id} SE(3)$.

Le torseur $V_{o,i}$ en tant qu'élément de l'espace vectoriel $T_{id}SE(3)$ peut être décomposé sur une base (de matrices) $\{u_k\}_{k=1,6}$ de $T_{id}SE(3)$ par $V_{o,i} = \sum_{k=1}^{k=6} \alpha_{o,i}^k \cdot u_k^{o,i}$. La position d'un point peut être exprimée par rapport à un repère quelconque à l'aide de la relation $p_i = g_{i,o} p_o$. La vitesse de p , en considérant i comme un référentiel immobile, est donnée par $\dot{p}_i = \dot{g}_{i,o} p_o + g_{i,o} \dot{p}_o$ et ne doit pas être confondue avec la vitesse de p par rapport à un référentiel absolu, exprimée dans le repère i qui est $g_{i,o} \dot{p}_o$. Si le point p est fixe par rapport au repère mobile i , alors $\dot{p}_o = \dot{g}_{o,i} p_i$. La dérivée d'une chaîne de repères $g_{i,j} = g_{i,i+1} \cdot g_{i+1,i+2} \dots g_{j-1,j}$ donne la loi de composition des torseurs $V_{i,j} = \sum_{k=i+1}^{k=j} g_{i,k} \cdot V_{k-1,k} \cdot g_{k,j}$. Si p est fixe dans le repère j on obtient alors

$$\dot{p}_i = \dot{g}_{i,j} p_j = g_{i,j} V_{i,j} p_j = \sum_{k=i+1}^{k=j} g_{i,k} \cdot V_{k-1,k} \cdot g_{k,j} \cdot p_j.$$

Si $V_{k-1,k} = U_{k-1,k} \cdot \alpha_k$ où les matrices $U_{k-1,k}$ sont des torseurs constants et les variables α_k sont des scalaires, on a alors $g_{k-1,k}(t) = g_{k-1,k}(0) \exp(U_{k-1,k} \int_0^t \alpha_k dt)$. On posera pour simplifier les notations $q_k = \int_0^t \alpha_k dt$ où q_k est par définition la $k^{\text{ième}}$ configuration du système et $\alpha_k = \dot{q}_k$ sa dérivée temporelle.

$$\dot{p}_i = \sum_{k=i+1}^{k=j} g_{i,k}(q) \cdot U_{k-1,k} \cdot \dot{q}_k \cdot g_{k,j}(q) \cdot p_j = \sum_{k=i}^{k=j} A_{i,j}^k(q) \cdot \dot{q}_k \cdot p_j = J_{i,j}(q) \cdot \dot{q} \quad (3.3)$$

L'expression $A_{i,j}^k(q) = g_{i,k}(q) \cdot U_{k-1,k} \cdot g_{k,j}(q)$ permet d'évaluer l'impact de la variation de la $k^{\text{ième}}$ configuration sur la liaison (i,j)

Notre objectif est de déterminer les variations sans collision de configuration dans le cas d'un rapprochement des deux solides (dérivée négative de la distance). En appliquant (3.2) et (3.3) on obtient finalement la forme différentielle suivante:

$$d(d_{i,j}) = \sum_{k=1}^n (n_i(q)^T A_{i,j}^k(q) p_j) dq_k \quad (3.4)$$

Pour ce faire, on cherche à majorer le plus finement possible le coefficient scalaire de chaque composante de la vitesse articulaire dans (3.3) en

$$0 > \dot{d}_{i,j} \geq \sum_{k=1}^n M^k \cdot \dot{q}_k.$$

En intégrant cette relation et en se plaçant dans le cas limite de décroissance de la distance on obtient $d_{i,j} = \sum_{k=1}^n M^k |\Delta q_k|$. Si l'on impose $\|\dot{q}\|=1$, on peut déterminer le rayon d'une boule de configurations (pour la distance euclidienne) centrée sur q dans laquelle il n'y a pas de configurations en collision.

Si l'on astreint q à se situer sur une arête définie par $\forall s \in [0,1] l_s(q_1, q_2) = q_1 + s \cdot (q_2 - q_1)$ ou un triangle défini par $\forall (s_1, s_2) \in [0,1]^2 h_{s_1, s_2}(q_1, q_2, q_3) = q_1 + s_1 \cdot (q_2 - q_1) + s_2 \cdot (q_3 - q_2)$, des procédures de recouvrement déterministe par des boules sans collision peuvent être mises en œuvre (figure 3.5) en utilisant par exemple une dichotomie (a) ou, dans le cas d'un triangle, en divisant l'arête la « moins recouverte » en deux de manière itérative (b) jusqu'à ce que l'ensemble du triangle soit recouvert.

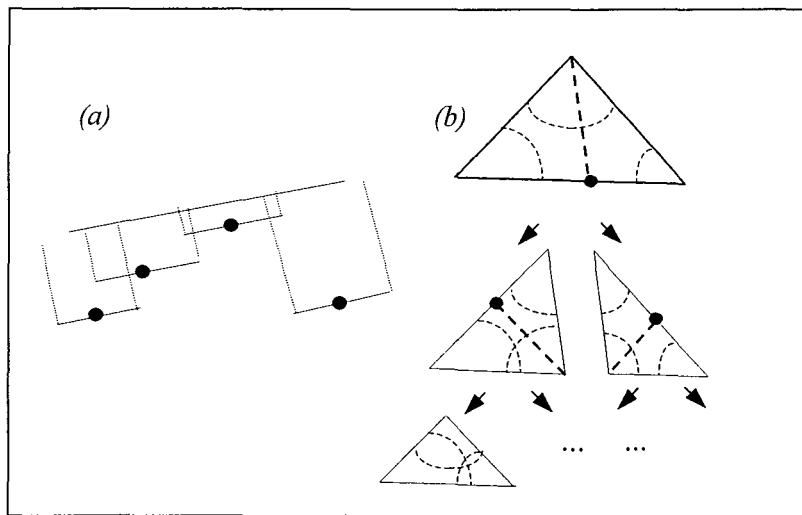


Figure 3.5: Procédures déterministes de recouvrement pour une arête et un triangle orientées vers une détection rapide de collision

Nous avons implanté une procédure originale de détection de collision pour un triangle (figure 3.5b) pour résoudre le problème de capture des classes d'homotopie abordé dans §III.3.2. Elle utilise la procédure récursive de division d'un triangle présentée ci-dessus. Les distances entre solides sont estimées (par valeur inférieure) à partir d'une hiérarchisation convexe (cf. §III.1.1.2) à ceci-près que les paires de volumes englobant prioritairement développées ne sont pas les plus proches mais celles qui induisent le plus petit rayon de la boule sans collision de la configuration concernée.

2. Capture probabiliste des composantes connexes

Nous abordons dans ce paragraphe les procédures d'exploration des configurations par un réseau de trajectoires sans collisions et plus particulièrement la capacité de ce réseau à capturer la connectivité des configurations libres d'une cellule. L'objectif est, à l'issue du pré-calcul de ce réseau, de pouvoir associer rapidement au moins une trajectoire du réseaux à tout couple de configurations connexes.

Nous considérons, à l'issue des paragraphes précédents que l'on dispose d'une fonction suivante qui permet de simplifier les notations :

Définition : La fonction libre. *Il s'agit d'une fonction booléenne définie sur l'ensemble des parties compactes de C c'est à dire :*

$$\text{libre}: M \subset C \mapsto \in \{\text{vrai}; \text{faux}\}.$$

Elle indique si toutes les configurations de la partie considérée sont sans collisions.

Exemple : $\text{libre}: q \in C \mapsto \text{libre}(q) \in \{\text{vrai}; \text{faux}\}$ détermine si la configuration q correspond à une collision. De manière équivalente, elle teste l'appartenance d'une configuration à l'un des deux ensembles $C_{\text{libre}} = \{q \in C / \text{libre}(q)\}$ ou $C_{\neg \text{libre}} = \{q \in C / \neg \text{libre}(q)\}$.

Nous considérons aussi que l'on dispose d'une fonction capable de donner une distance $d_{\text{libre}}(q)$ (par exemple la distance euclidienne) à la configuration la plus proche qui correspond à une collision (Il s'agit en fait d'une sous-estimation). Une représentation constructive (d'un point de vue topologique) des configurations libres d'une cellule peut alors être envisagée. Nous remarquerons que $C_{\text{libre}}^\varepsilon = \{q \in C / d_{\text{libre}}(q) \geq \varepsilon\}$ est un compact dont, par définition, tout recouvrement à l'aide de boules admet un sous-recouvrement fini. Nous pourrions ainsi obtenir une représentation finie de $C_{\text{libre}}^\varepsilon$ avec une approximation ε aussi fine que l'on veut. Cette représentation finie est une condition nécessaire

pour la mise en place de traitements algorithmiques finis dans C . D'autres types de topologies peuvent être choisies [SVE98] mais on impose qu'elles soient équivalentes à une topologie métrique définie sur C .

2.1 Les méthodes à apprentissage probabiliste

Nous introduisons aussi ci-dessous la fonction **Echantillon_libre (C)** qui est à l'origine du terme "probabiliste" associé aux méthodes que nous présentons. Son premier intérêt pratique (du point de vue de son implantation) est qu'elle permet de s'affranchir d'une définition contraignante d'une grille d'échantillonnage de C et introduit une certaine souplesse dans la procédure d'exploration des configurations libres. La configuration q est générée de manière aléatoire de la manière suivante :

```

Echantillon_libre ( C )
a ← vrai;

Tant_que (a);{
    Tirer q suivant une loi uniforme sur C;
    a = ¬ libre(q) ;
};

```

Disposant d'un ensemble d'échantillons généré aléatoirement $S = \{qs_1, \dots, qs_{ns}\}$, on peut construire un graphe $G = (S, A)$ dont les arcs A sont les paires d'échantillons de S qui peuvent être reliés par une arête sans collision. c'est-à-dire les arcs (qs_i, qs_j) qui vérifient $libre(l_s(qs_i, qs_j))$ avec :

$$\forall (q_1, q_2) \in C^2, \forall s \in [0,1], l_s(q_1, q_2) = q_1 + s.(q_2 - q_1)$$

pour $s = 0, l_0(q_1, q_2) = q_1$ et pour $s = 1, l_1(q_1, q_2) = q_2$

Si l'objectif du graphe est de conduire à une partition de S en **composantes connexes**, le test de toutes les paires n'est pas nécessaire. Pour construire un tel graphe, la méthode à réseau probabiliste (MRP) standard [OVE95, KAV96] considère tout d'abord que chaque élément de S est une composante connexe à part entière puis fusionne les composantes en ajoutant une unique arête à chaque fusion. De cette manière, pour N nœuds, $M=N-ncc$ arêtes permettent de faire une partition de S en ncc composantes connexes. Le graphe $G=(S,A)$ permet de représenter toutes les composantes connexes à l'aide de ncc arbres.

Mais les graphes composés d'arbres (les forêts) n'associent qu'un seul chemin entre deux sommets connexes ce qui ne correspond pas aux besoins évoqué dans le chapitre précédent (§II.3.1) où l'on a évoqué le besoin d'une représentation exhaustive des classes d'homotopie. Inversement, si l'on autorise le graphe à former des cycles, alors l'immense majorité des cycles produits n'apporte aucune information et les calculs sont fortement ralentis.

Pour faciliter la présentation des méthodes à réseau probabiliste, nous utilisons par la suite la fonction élémentaire **Premier_connecté** (l, q, S) qui teste si une configuration q donnée peut être reliée à une des configurations d'un ensemble $S = \{qs_1, \dots, qs_{n_s}\}$ par un chemin défini par l .

Premier_connecté (l, q, S)

$a \leftarrow \text{vrai}; b \leftarrow \text{vrai};$

Tant_que ($a \wedge \neg b$);{

$a \leftarrow$ (on peut choisir ^(*) $qs \in S$ qui n'a pas été testé);

Si (a);{ $b \leftarrow \text{libre}(l(q, qs))$ };

};

Si ($a \wedge b$);{Retourner $\{qs\}$ };Sinon {Retourner \emptyset };

(*) Par exemple par ordre croissant d'une C-distance à q pour MRP-basique

L'algorithme MRP-basique permet de construire un graphe $G=(S,A)$ composés d'arbres en ne considérant, pour chaque nouveau sommets qu'un voisinage $B(q,dmax)$ dans lequel la recherche d'une connexion par un arc sera recherchée.

MRP-basique ($l,C,NB_nœuds, dmax$)

$S \leftarrow \emptyset; A \leftarrow \emptyset; nb_nœuds \leftarrow 0;$

Tant_que $nb_nœuds < NB_nœuds$;{

$q \leftarrow \text{Echantillon_libre}(C); S \leftarrow S \cup \{q\}; nb_nœuds = nb_nœuds + 1;$

Pour chaque composante connexe $CC \subset S$;{

$V_{cc} \leftarrow CC \cap B(q,dmax);$

$qc = \text{Premier_connecté}(l, q, V_{cc});$

$A \leftarrow A \cup \{(q, qc)\};$

};

}

Retourner $G=(S,A)$

$NB_nœuds$ est le nombre maximal de nœuds libres générés

$B(q,dmax)$ est une boule pour une distance définie sur C ,

$dmax$ est une distance au delà de laquelle la recherche d'une connexion n'est pas effectuée,

Nous présentons ci-dessous un échantillonnage tiré de [BOO99] qui augmente la probabilité d'un tirage proche d'une configuration en collision. Cette procédure peut remplacer facilement le tirage uniforme Echantillon_libre (C) précédemment évoqué dans MRP-basique. On parlera dans ce cas de MRP-Gaussien. Les paramètres d_moy et d_var permettent de calculer une distance définie sur C suivant une loi gaussienne de moyenne et variance donnée.

```

Echantillon_Gaussien ( $d\_moy, d\_var, C$ )
réussi ← faux;

Tant que ( $\neg$  réussi) {
  Tirer  $q_1$  suivant une loi uniforme sur C ;
  Tirer  $d$  suivant une loi gaussienne ( $d\_moy, d\_var$ ) ;
  Tirer  $q_2$  tel que  $d(q_1, q_2) = d$  ;
   $a \leftarrow libre(q_1)$ ;  $b \leftarrow libre(q_2)$ ;
  Si ( $a \wedge \neg b$ ); { Retourner  $q_1$ ; réussi ← vrai};
  Si ( $\neg a \wedge b$ ); { Retourner  $q_2$ ; réussi ← vrai};
};

```

2.2 Une méthode probabiliste à recouvrement fini : MRP-visi

Les méthodes MRP-basique ou MRP-Gaussien arrêtent de construire le réseau lorsque deux configurations données sont reliées, ou par défaut, après qu'un certain nombre de nœuds aient été générés. On ne peut pas garantir que toutes les composantes connexes ont été ainsi capturées ni avoir une idée précise du travail effectué à l'issue de l'algorithme. Néanmoins, la méthode présentée ci-dessous s'appuie sur l'existence de recouvrement **fini** à l'aide de domaines de visibilité [SIM00]. Elle permet d'estimer cette couverture.

Définition: Un domaine de visibilité associé à une configuration q_0 est l'ensemble des configurations qui vérifient :

$$Visi_l(q_0) = \{q \in C / libre(l(q_0, q))\}$$

Les domaines de visibilité fournissent une méthode pour effectuer des recouvrements finis (figure 3.6) de manière plus concise avec $S_{gard} = \{qs_1, qs_2, \dots, qs_N\}$ un ensemble fini et la condition :

$$\forall i, j \in \{1, \dots, N\}, i \neq j, \neg libre(l(qs_i, qs_j))$$

Les configurations q_0 sont appelées des gardiens. Par définition, un gardien n'appartient à aucun domaine de visibilité d'un autre gardien. L'ensemble S représente alors un ensemble de gardiens qui définit un recouvrement par les domaines de visibilité associés.

$$C_{libre}^\epsilon \subset \bigcup_{q \in S_{gard}} Visi_l(q)$$

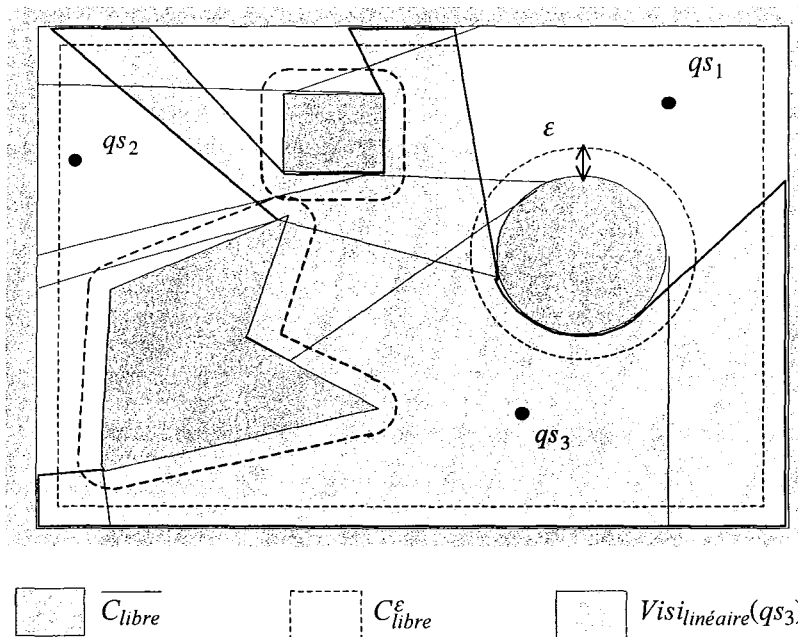


Figure 3.6: Recouvrements à l'aide de domaines de visibilité

Le premier avantage de MRP-Visi par rapport à MRP-basique, MRP-gaussien (ou plus généralement les différents variantes de MRP qui modifient la distribution de l'échantillonnage) est que le nombre de gardiens utilisés est fini et que, par conséquent, il s'arrête naturellement lorsqu'il ne peut plus ajouter de gardiens contrairement à MRP-basique qui s'arrête quand deux configurations données sont connexes ou quand un nombre donné de configurations ont été testées. **Par conséquent il est beaucoup plus adapté aux procédures de caractérisation globales de l'espace des configurations libres.**

Remarque : Le nombre de gardiens peut être minimisé en ne retenant, pour un échantillonnage S donné, que les gardiens qui ont les plus grands domaines de visibilité. Pour ce faire, il suffit de considérer que le degré des sommets (leur nombre d'arcs incidents) d'un graphe $G=(S,A)$ construit en tentant toutes les connections possibles donne une estimation du volume du domaine de visibilité de ce sommet. Le sommet de plus grand degré est alors celui qui a priori le plus grand domaine de visibilité

(pour un échantillonnage uniforme). Ce sommet devient un gardien. Un nouveau graphe $G=(S,A)$ peut être construit en dehors des domaines de visibilité des gardiens déjà sélectionnés et ainsi de suite.

L'existence d'un point qa appartenant à deux domaines distincts $Visi_l(qs_i)$ et $Visi_l(qs_j)$ peut être détectée dès que qa vérifie $libre(l(qa,qs_i)) \wedge libre(l(qa,qs_j))$. Les approches de construction d'un graphe $G=(S,A)$ pour représenter la connectivité peuvent s'appliquer sans restrictions (figure 3.7).

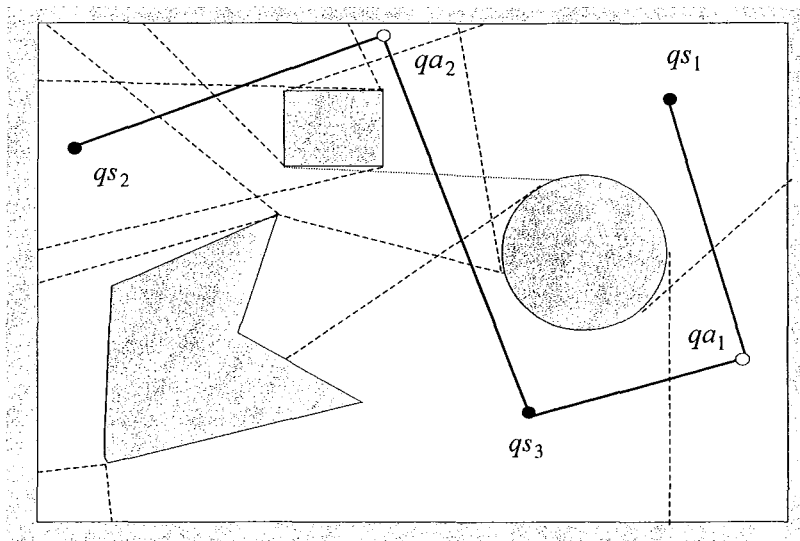


Figure 3.7: Une composante connexe

Une arête du graphe est donnée par la configuration que l'on notera qa qui permet de relier deux sommets correspondant respectivement aux configurations qs_i et qs_j . L'ensemble des arêtes se définit alors par une liste ordonnée de configurations $QA=\{qa_1,qa_2,\dots,qa_M\}$ associée à la liste ordonnée correspondantes des couples de sommets reliés $A=\{(i,j_1),\dots,(i_M,j_M)\}$.

L'algorithme MRP-Visi [SIM00] défini ci-dessous permet simultanément de construire un recouvrement par des domaines de visibilité en construisant un ensemble de gardiens et de connecter les gardiens.

MRP-Visi (NB_essais) $S \leftarrow \emptyset; DA \leftarrow \emptyset; nb_essais \leftarrow 0;$ Tant que ($nb_essais < NB_essais$);{ $q \leftarrow \text{Echantillon_libre}(C); \text{stock} \leftarrow \emptyset;$ Pour chaque composante connexe $CC \subset S$; $\text{stock} \leftarrow \text{stock} \cup \text{Premier_connecté}(l, q, CC);$

};

Si ($\text{stock} = \emptyset$); { $S \leftarrow S \cup \{q\}; nb_essais \leftarrow 0;$

};

Si ($\text{Card}(\text{stock}) > 1$); {Pour tout $qs \in \text{stock}$; { $DA \leftarrow DA \cup \{(q, qs)\};$

Actualiser les composantes connexes de S;

};

};

Sinon ;{

 $nb_essais \leftarrow nb_essais + 1;$

};

}

Retourner (S,DA)

 NB_essais est le nombre maximal d'essais consécutifs autorisé

3. Capture probabiliste des classes d'homotopie

Nous abordons dans ce paragraphe les procédures d'exploration des configurations par un réseau de trajectoires sans collisions et plus particulièrement la capacité de ce réseau à capturer les classes d'homotopie de chaque composante connexe des configurations libres d'une cellule. L'objectif est, à l'issu du pré-calcul de ce réseau, de pouvoir associer rapidement toutes les classes d'homotopie de trajectoire à tout couple de configurations connexes.

Par définition une approche directe pour construire un graphe homotope consisterait à n'ajouter au graphe G que des arêtes que l'on ne peut pas déformer continûment dans C_{libre} en des chemins existant déjà dans le graphe G . En d'autres termes, l'ajout d'arêtes dans le graphe ne doit pas créer de cycles **réductibles** (c'est-à-dire des cycles qui peuvent être déformés continûment dans C_{libre} en un point). Mais une telle approche n'est pas adaptée à un traitement algorithmique. En effet, le nombre potentiel de cycles à tester augmente de manière exponentielle avec le nombre d'arêtes du graphe. De plus, le fait de tester si un cycle est réductible ou non n'est pas un problème trivial. Il peut conduire à des minima locaux si l'on tente de réduire la longueur du cycle par déformation.

3.1 Les sous-graphes visibles depuis une configuration

Pour s'affranchir des difficultés évoquées ci-dessus, nous proposons de considérer exclusivement la réductibilité des cycles élémentaires qui sont le bord d'une face locale. Nous définissons les **faces locales** comme des simplexes de dimension deux de l'espace des configurations C :

$$\forall (q_1, q_2, q_3) \in C^3, \forall (s_1, s_2) \in [0,1]^2, h_{s_1, s_2}(q_1, q_2, q_3) = q_1 + s_1 \cdot (q_2 - q_1) + s_2 \cdot (q_3 - q_2)$$

La procédure de détection de collision s'appliquant à des faces locales utilise le recouvrement de la face avec des boules de la base topologique induite B_h . Nous pouvons étendre la définition de la fonction *libre* à la détection de collision de face locales. La fonction logique $libre(h(q_1, q_2, q_3))$ n'est pas vérifiée dès qu'une configuration de la face locale correspond à une collision.

Dans notre cas, le fait qu'une face locale soit sans collision est une condition suffisante pour statuer sur la réductibilité du cycle correspondant.

Des faces locales sans collisions peuvent alors être vues comme le moyen de détecter l'inclusion d'un chemin local dans un domaine de visibilité. La notion de sous-graphe visible depuis un sommet donné est une extension de l'emploi de ce moyen à un graphe. Les sous-graphes visibles permettent de faciliter la présentation du traitement de la réductibilité des cycles. Etant donné un graphe $G = (S, A)$ et une configuration qv , nous pouvons définir le sous-graphe $G_v = (S_v, A_v)$ associé au point de vue qv par :

- une sous-liste de gardiens visibles depuis la configuration qv :

$$S_v = \{q \in V \mid libre(l(qv, q))\}$$

- une sous-liste d'arêtes visibles depuis la configuration qv , c'est-à-dire formant des cycles élémentaires réductibles :

$$A_v = \{qa_k \in A \mid (i, j) = IA_k, libre(h(qv, qs_i, qa)) \wedge libre(h(qv, qs_j, qa))\}$$

Pour compléter l'analogie amorcée par la définition des sous-graphes visibles, nous dirons que le graphe G respecte une propriété selon le point de vue de la configuration qv si G_v présente cette propriété.

3.2 Algorithme MRP_CH

Nous utilisons un échantillonnage aléatoire C_{libre}^E , pour tester si un graphe G est simplement connexe selon un point de vue quelconque de C_{libre}^E . Pour chaque échantillon, le sous-graphe visible

associé est construit par application directe de la définition des graphes visibles. Si ce sous-graphe n'est pas connexe, le principe de MRP-CH est d'ajouter un nombre minimal d'arêtes à G pour qu'il soit au moins simplement connexe du point de vue considéré. L'algorithme MRP-CH que nous avons élaboré est décrit ci-dessous :

Algorithme MRP_CH
 $S = \emptyset, A = \emptyset$ (1)
Tant que (le nombre d'essais infructueux (2) < limite),{
On construit $G_v = (S_v, A_v)$ pour un échantillon aléatoire $qv \in C_{libre}^\varepsilon$, (3)
Si $S_v = \emptyset$;{
Alors on ajoute qv à la liste S des sommets de G ;}
Sinon ;{
Si G_v n'est pas connexe,{
Alors on ajoute un nombre minimal d'arêtes dans les listes
 A et A_v pour rendre G_v simplement connexe
}
}
Extension(RL) (1)
}

(1) Nous décrivons dans §III.3.3 une extension de l'algorithme qui nécessite l'initialisation $RL = \emptyset$ et qui sera notée **Extension(RL)**.
(2) Les essais infructueux correspondent aux essais qui ne permettent pas d'ajouter de nouveaux sommets, de nouvelles arêtes (voire par la suite de nouveaux cycles).
(3) La partition du graphe G en composantes connexes a déjà été traitée dans la partie §3.1 et $G_v = (S_v, A_v)$ n'est construit que dans une seule composante connexe de G .

L'ensemble $\left\{ \mathcal{G} \in C_{libre}^\varepsilon \mid \exists qs_i \neq qs_j \in S, q \in Visi(qs_i) \cap Visi(qs_j) \right\}$ est compact et peut être recouvert par un nombre fini de boules de la topologie induite. Ceci permet, comme dans le cas du recouvrement de C_{libre}^ε , d'établir que l'ensemble des intersections des domaines de visibilité (appelée la nervure) peut être recouverte à l'aide d'un nombre fini de domaines de visibilité (pour ε fixé). Une reformulation de ce résultat à l'aide des graphes visibles revient à dire que **le nombre d'arêtes que l'on ajoute à un arbre pour le rendre au moins simplement connexe d'un point de vue quelconque de C_{libre}^ε est fini**. De cette manière, les composantes connexes du graphe tendent à l'être selon un point de vue quelconque de C_{libre}^ε avec une probabilité qui tend vers un en un temps fini. (figure 3.8).

On peut remarquer que l'ajout d'arcs au graphe dépend au plus linéairement du nombre d'arcs courants du graphe.

Nous pouvons considérer l'exemple suivant de construction du graphe par MRP-CH :

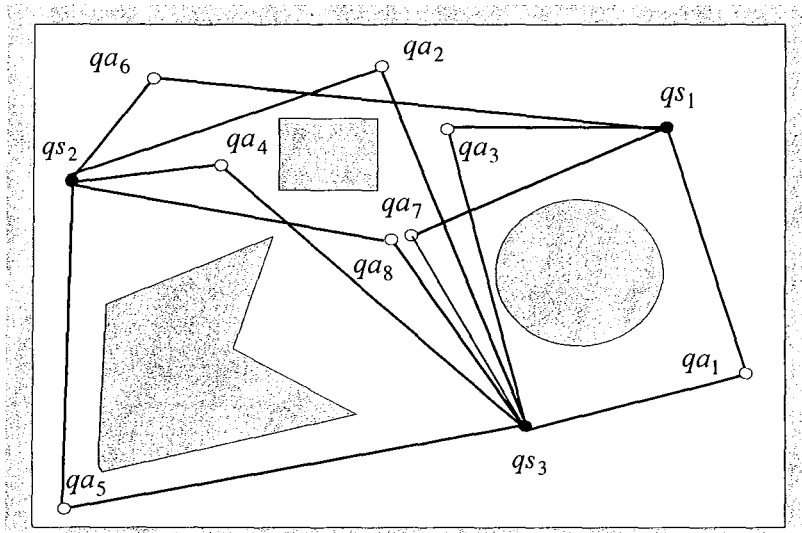


Figure 3.8: Graphe connexe selon un point de vue quelconque

Les sommets qs_1 , qs_2 , qs_3 et les arcs qa_1 qa_2 sont construits par MRP-Visi. A ce stade on lance MRP-CH. Une nouvelle configuration qa_3 est échantillonnée aléatoirement. qa_3 voit qs_1 et qs_3 mais ne peut pas voir l'arc les liant défini par qa_1 , il voit donc un graphe non connexe et il est donc ajouté au graphe. Une nouvelle configuration qa_4 est échantillonnée aléatoirement. qa_4 voit qs_2 et qs_3 mais ne peut pas voir l'arc les liant défini par qa_2 , il voit donc un graphe non connexe et il est donc ajouté au graphe. Les nœuds qa_5 , ..., qa_8 sont ajoutés au graphe de manière similaire. A ce stade, tout nouvel échantillon voit un sous-graphe connexe (éventuellement réduit à un unique sommet).

Nous supposons maintenant que ce processus a été mené à son terme. Si une configuration qui définit un point de vue n'est plus figée mais décrit un chemin quelconque de C_{libre}^E (éventuellement un lacet), cette configuration « voit » **sans interruption au moins une chaîne (respectivement un cycle) de G** . En fait, plusieurs chaînes du graphe peuvent être associées de cette manière car des cycles réductibles ont pu apparaître lors de l'ajout de nouvelles arêtes pendant le déroulement de HPPR. Par exemple le cycle $(qs_2, qa_4, qs_3, qa_8, qs_2)$ créé par MRP_CH de la figure 3.8 est réductible. C'est pourquoi nous appellerons le graphe construit à l'issue de MRP_CH un **graphe homotope redondant**. Il capture les cycles non réductibles mais il n'est pas homotope à C_{libre}^E .

Nous allons expliquer dans la partie suivante, la manière utilisée pour détecter les cycles réductibles apparus au cours de MRP-CH, de manière à réduire, voire éliminer, la redondance de représentation des chemins et des lacets. Ce complément de l'algorithme permet de répondre à des problèmes de réductibilité de lacets quelconques et de créer réellement un graphe homotope pour chaque composante connexe. Dans son état actuel l'algorithme garantit simplement de capturer (en un temps fini) au moins un représentant de chaque classe d'homotopie mais pas de construire le groupe d'homotopie car certains cycles sont réductibles et ne représentent donc pas une classe d'homotopie.

3.3 Elimination de la redondance de la représentation du groupe d'homotopie

Les cycles réductibles doivent être éliminés pour minimiser le nombre de solutions homotopes d'un même problème de planification de trajectoire. Nous allons utiliser les propriétés de groupes associés aux cycles de manière à minimiser le nombre de tests de réductibilités des cycles du graphe. En particulier, les compositions des représentants de cycles déjà testés ne doivent plus être testés.

Notre objectif est de montrer que tous les tests de réductibilité des cycles peuvent se ramener à des combinaisons de cycles élémentaires et que par conséquent, les tests des cycles élémentaires suffisent à caractériser l'ensemble des cycles réductibles. Les domaines de visibilité et les graphes visibles permettent d'organiser la création et les tests des cycles élémentaires. Nous allons revenir à la représentation discrète des chemins et des cycles et des relations de connexité et d'homotopie.

Un théorème dû à Leray et décrit dans [DEY99] dans le cas d'une *triangulation*^{A4} de Delaunay stipule qu'un recouvrement à l'aide d'ensembles simplement connexes dont les intersections sont soit vides soit *contractiles*^{A4} permet de capturer l'homotopie. Malheureusement, dans le cas de recouvrement à l'aide de domaine de visibilité, ce théorème ne peut s'appliquer. Il suffit pour s'en convaincre de considérer une boule en 3D vue par deux gardiens. L'intersection des deux domaines est alors homéomorphe à un tore et n'est pas contractile.

Pour forcer MRP-CH à capturer de manière concise toutes les classes d'homotopies nous proposons de détecter les cycles du graphe entièrement inclus dans un domaine de visibilité.

De manière évidente, la classe des cycles réductibles forme un sous-groupe correspondant à un unique élément qui est l'identité du groupe d'homotopie. Cette structure de sous-groupe (commutatifs dans ce cas) peut être mise à profit pour détecter et stocker sans redondance l'ensemble des cycles réductibles. Dans notre procédure, le nombre de cycles est fini. Nous pouvons stocker l'ensemble des cycles réductibles de manière concise à l'aide d'un générateur fini RL du sous-groupe des cycles réductibles que l'on note :

$$RL = \{c_1, c_2, \dots, c_{nrl}\}$$

Le groupe généré, $GEN(RL)$, est obtenu par compositions des éléments de RL . En pratique $GEN(RL)$ n'est pas construit. On utilise le fait que l'on peut tester si un cycle arbitraire c appartient

ou non à $GEN(RL)$. Ce test, appelé l'algorithme de division élémentaire, réduit c à l'aide d'éléments de RL jusqu'à ce qu'il ne reste plus qu'un seul point ou qu'aucune autre réduction ne soit possible. Dans ce cas, le résidu des divisions successives est ajouté à la liste des cycles RL .

Extension(t)

SI $\exists c \in (V_t, E_t)$ et $c \notin GEN(RL)$,

Ajouter le résidu de c dans la liste
des cycles réductibles RL

FIN

Remarque : Pour représenter chaque cycle de manière unique avant de tester toute division, on doit fixer un arbre couvrant extrait de G , les arcs orientés restant correspondent au générateur du groupe, et tout cycle s'exprime comme une combinaison de ces éléments (cf ANNEXE 4).

Nous avons ainsi obtenu une procédure pour détecter plus rapidement les cycles réductibles. Le problème est de savoir si tous les cycles réductibles peuvent être détectés de cette manière.

Lorsque l'on dispose, à l'issue de MRP-CH d'un graphe qui est connexe pour un point de vue quelconque (i.e. visiblement connexe), nous pouvons parcourir un chemin quelconque (éventuellement un lacet) de C_{libre}^e tout en lui associant au moins une chaîne (ou respectivement un cycle) du graphe.

Un lacet réductible est par définition le bord d'une surface continue de C_{libre} . Cette surface peut être approximée aussi près que l'on veut au moyen d'un maillage constitué de faces locales de C_{libre} . Disposant d'un graphe visiblement connexe, nous pouvons associer un cycle de G à chaque face locale du maillage. Si l'approximation de la surface S est de plus en plus fine, les faces utilisées tendent chacune vers un point. Les cycles associés tendent à être visibles de ce point et peuvent alors être détectés comme réductibles. De manière plus synthétique, si G est visiblement connexe dans C_{libre} , alors un lacet réductible peut soit être entièrement inclus dans un domaine de visibilité, soit être obtenu en composant (les classes d'équivalence) des cycles réductibles de G .

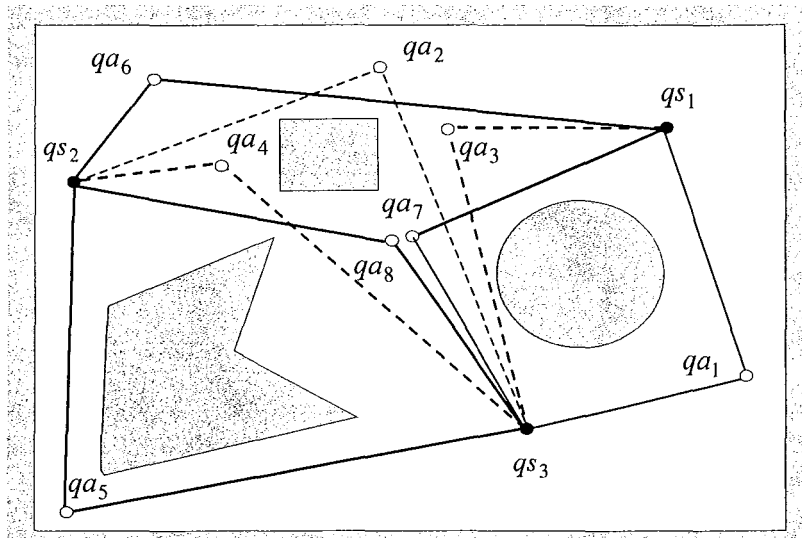


Figure 3.9: *Elimination d'un arc de chaque cycle réductible détecté : Obtention d'un graphe homotope*

La donnée d'un générateur RL des cycles réductibles permet, en éliminant des arcs distincts pour chaque représentant d'un cycle réductible de définir un graphe non redondant (figure 3.9) en terme de représentation des classes d'homotopie. Le résultat d'une telle opération n'est pas unique, il est possible par exemple de minimiser la longueur de tous les arcs du graphe recherché.

Pour tester la validité de l'approche prospective de capture des classes d'homotopie, nous avons développé sous Matlab 6.0 dans des environnements polygonaux simples en 2D un démonstrateur élémentaire incluant les fonctionnalités suivantes :

- le calcul de distance entre les solides (comprenant 2 niveaux de hiérarchisation convexe à l'aide de sphères),
- le calcul de liaisons arborescentes (rotoïde / prismatique / vissage),
- les algorithmes MRP-Visi et MRP-Gaussiens avec une procédure originale de minimisation du nombre de gardiens pour MRP-Visi (cf remarque du §II.2.2),
- les algorithmes MRP-CH avec ou sans phase de réduction des cycles redondants (Algorithme « Extension » présenté ci-dessus),
- la visualisation des mouvements de chaque classe d'homotopie reliant deux configurations données.

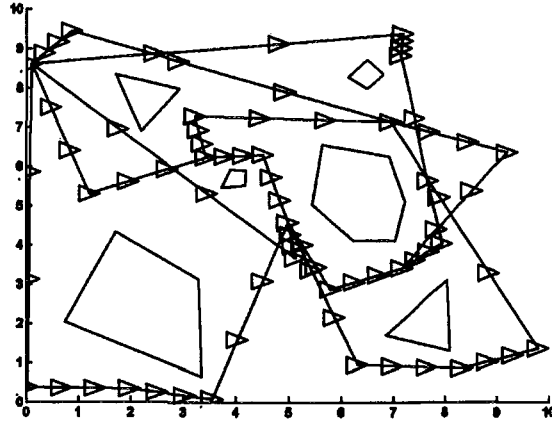


Figure 3.10: Résultats de MRP-CH pour un solide triangulaire en translation $[0,1]^2$ et 6 obstacles polygonaux

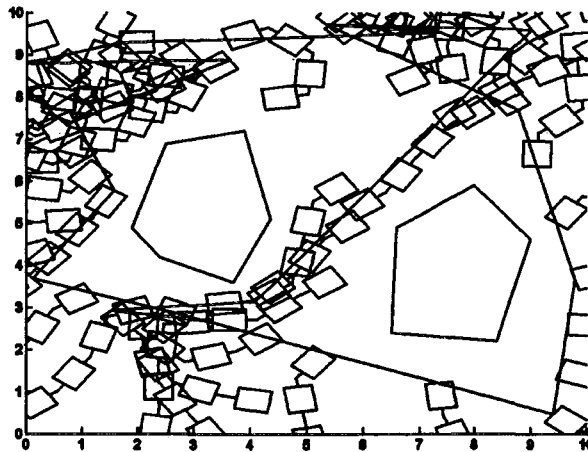


Figure 3.11: Résultats de MRP-CH pour une cellule de $SO(2) \times [0,1]^5$ représentant une chaîne articulée de 4 solides

Ce démonstrateur élémentaire du point de vue graphique a permis de valider la détection de collision de faces (décrite dans §III.1) y compris pour des systèmes articulés (figure 3.11) et de montrer en pratique la convergence de MRP-CH ainsi que l'élimination des cycles réductibles.

Nous présentons ci-dessous le nombre d'arcs ajoutés pour l'exemple de la figure 3.10 en fonction du nombre d'échantillons infructueux successifs autorisés (critère d'arrêt de l'algorithme MRP-CH). Ce nombre tend à être constant. Les essais correspondent à une moyenne sur 10 lancements de MRP-CH avec quatre critères d'arrêts différents (10,50,100,200).

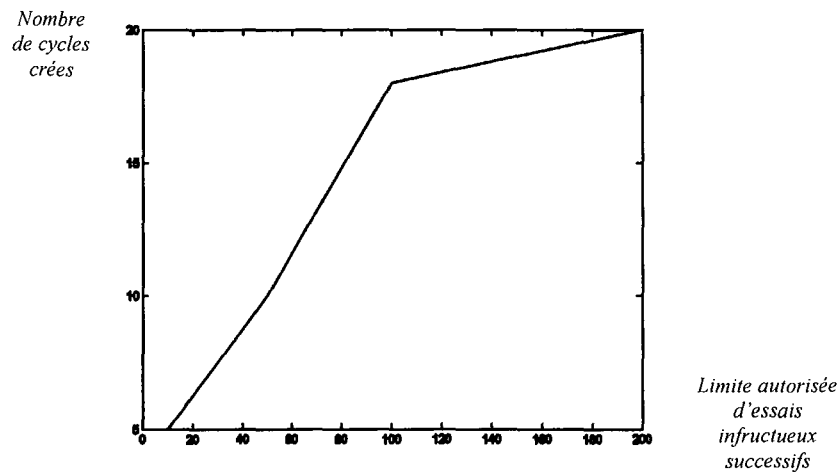


Figure 3.12: Confirmation expérimentale du nombre fini de cycle additionnels créés par MRP-CH

Dans tous les cas, tous les cycles réductibles ont été détectés. Tous les cycles non réductibles (*i.e.* les représentants de classe d'homotopie) ont été capturés à partir d'un critère d'arrêt fixé à 100.

Conclusion

Nous avons proposé dans ce chapitre un nouvel algorithme, MRP-CH, qui étend la méthode à réseau probabiliste MRP-Visi. Il permet en effet de capturer les classes d'homotopie d'une cellule paramétrée de configurations en construisant un réseau probabiliste. L'algorithme MRP-Visi associe un ou plusieurs sommets (chaque configuration est vue par au moins un gardien) à chaque configuration et une unique chaîne à chaque paire connexes de configurations. Le résultat de l'algorithme MRP-CH dans sa version réduite est un graphe pour lequel on peut canoniquement associer une ou plusieurs chaînes homotopes à chaque chemin de configuration. L'algorithme MRP-CH dans sa version étendue fournit une unique chaîne homotope à chaque chemin de configuration.

Cet algorithme s'appuie sur une procédure de détection de collision pour un triangle défini dans l'espace (affine) des paramètres de la cellule considérée. Nous avons proposé une approche déterministe de recouvrement du triangle au moyen de boules visant avant tout à détecter rapidement une collision sur le triangle. Cette procédure utilise en entrée une estimation conservatrice des distances entre chaque paire de solides. Le calcul de ces distances peut bénéficier des techniques de décompositions convexes hiérarchisées (§III.1.1.2) en s'arrêtant dès que l'estimation des distances est jugée suffisamment bonne. En particulier aucune distance nulle entre deux solides ne doit être renvoyée si ceux-ci ne sont pas en collision.

L'approche de MRP-CH s'inscrit dans le cadre général que nous avons fixé dans le chapitre précédent (celui d'une décomposition cellulaire explicite d'un espace des configurations de type $SO(d)^n \times [0,1]^m$). Cet algorithme permet de compléter le pré-calcul nécessaire d'un réseau en évitant les situations critiques où l'utilisateur désire obtenir ou modifier rapidement (en introduisant de nouvelle contrainte cf. §II.3) une trajectoire dans une classe d'homotopie de son choix.

Les perspectives d'évolution de MRP-CH sont avant tout la mise en œuvre générale d'une procédure de fusion des groupes d'homotopies par recollement de cellules de $SO(d)^n \times [0,1]^m$. MRP-CH, à l'instar de MRP-Visi, peut être vu comme une procédure de représentation concise du mouvement. Dans cette optique de « compression de la représentation du mouvement », il peut être intéressant de mettre en œuvre des procédures de minimisation du nombre de gardiens recouvrant C_{libre} et de minimisation du nombre de témoins de connexité.

CHAPITRE IV

Mise en œuvre du démonstrateur MoLog

Validation des MRP et évolution de l'architecture logicielle

Le projet MoLog a pour objectif général de démontrer que des méthodes de planification de mouvement peuvent être mises en œuvre pour des environnements complexes comme les maquettes numériques d'installations industrielles (§III.1). Les méthodes à réseaux probabilistes ont été choisies comme candidates potentielles. Elles fédèrent actuellement une large partie de la communauté robotique et trouvent des applications scientifiques et industrielles croissantes en dehors du champ d'application traditionnel de la robotique mais elles doivent être adaptées pour répondre aux besoins de la préparation d'un chantier.

Mon rôle au sein d'EDF R&D et dans le cadre du projet Molog a été de définir le protocole de test de ces méthodes en fonction des besoins et d'interagir, sur la base de tests complets effectués régulièrement (tous les 6 mois), avec les différents partenaires du projet Molog pour trouver des solutions d'intégration les plus satisfaisantes pour l'utilisateur notamment au niveau de la gestion du temps de calcul. L'objectif de ce travail est de statuer quant à la possibilité d'obtenir à court terme une intégration logicielle d'outils efficaces de planification de trajectoire dans le contexte particulier de la préparation d'un chantier (§III.2). Le critère de validation est qu'une animation fidèle du déroulement d'une séquence critique d'un chantier doit pouvoir être construite rapidement sans que l'utilisateur ne soit arrêté par des difficultés de nature géométrique ou par la nécessité d'une connaissance des méthodes de planifications de trajectoires.

1. Validation des méthodes à réseaux probabilistes pour des locaux industriels

Le test et l'intégration de plusieurs techniques de construction d'un réseau probabiliste de chemins a été mis en œuvre dans le projet MoLog. Les méthodes de planifications de trajectoires (*cf. §III.2*) disponibles dans le logiciel *Move3D [SIM01]* au début du projet étaient :

- MRP-Basique,
- MRP-Visi fondée sur les domaines de visibilité, qui permet entre autre de réduire la taille du réseau,
- MRP-Gaussien fondée sur la modification de la distribution aléatoire d'échantillonnage, qui permet d'augmenter la probabilité de générer des chemins élémentaires dans les « passages étroits ».

Le premier objectif de ces tests est d'évaluer les performances globales de ces différents algorithmes disponibles pour démontrer la pertinence de l'approche MRP pour la résolution de tâches et d'opérations dans des environnements industriels complexes de centrales nucléaires.

Les évaluations des performances ont consisté à évaluer le *temps global mis pour résoudre un unique problème de planification* donné. Nous précisons tout d'abord la nature de ces problèmes.

1.1 Quelques exemples caractéristiques

La première tâche qui m'a été confiée a été de trouver des exemples industriels significatifs dont le maquettage numérique est disponible (essentiellement pour des centrales nucléaires de type N4) et de les décomposer en problèmes élémentaires de planification du mouvement. Nous avons utilisé pour cela divers types d'environnements caractéristiques qui sont présentés et décrits ci-dessous.

1.1.1 Manipulation d'une turbo-pompe alimentaire

Le premier exemple industriel choisi consiste à manipuler les différents éléments (notamment le capot qui est la partie la plus encombrante) d'une des deux Turbo-pompes alimentaires (TPA) situées dans la salle des machines d'une centrale nucléaire (image ci-dessous). Cet exemple a été choisi car une étude a été menée sans outil de planification dans *PDMS* (le logiciel de CAO utilisé pour le maquettage numérique des centrales nucléaires) en discrétisant "à la main" le mouvement, en utilisant le détecteur de collision de *PDMS*. Ce travail a demandé plusieurs semaines.



Figure 4.1 – Aperçu d'une salle des machines de centrale nucléaire et zone de maintenance difficile

La scène statique modélisée dans *PDMS* (review format, *.vit) correspond à 5 Mo de primitives géométrique *PDMS* (type *CSG*). Les images suivantes sont extraites de *PDMS*. Elles définissent les positions initiales et finales à relier à l'aide d'un mouvement sans collision si celui-ci existe.

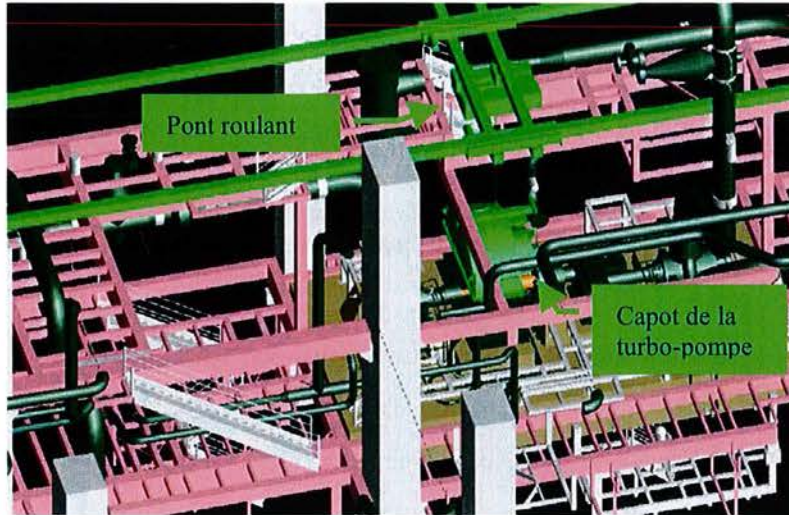


Figure 4.2 – Configuration initiale d'un pont roulant transportant un capot de Turbo-pompe alimentaire « TPA »

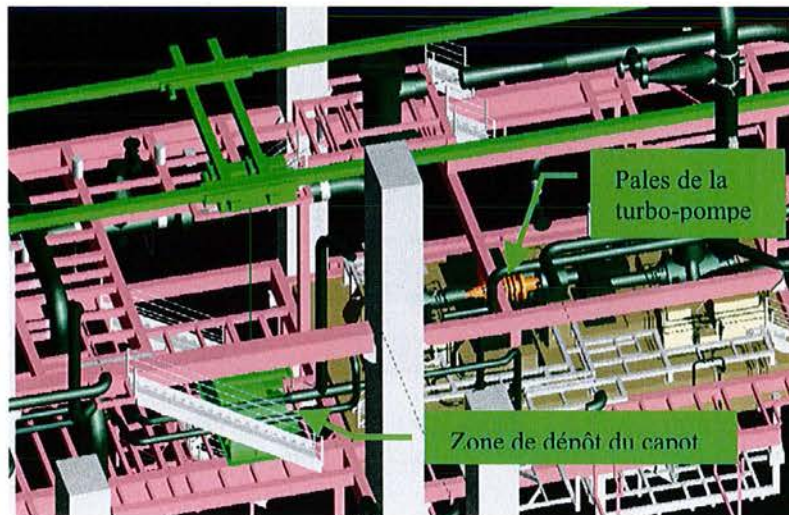


Figure 4.3 – Configuration finale que l'on cherche à atteindre par un mouvement sans collision

1.1.2 Changement d'un générateur de vapeur

L'exemple du changement de générateur de vapeur (GV) a été choisi car il représente un exemple de graphe de liaison cyclique. Outre son intérêt pratique, il permet de montrer que les méthodes à réseaux probabilistes peuvent traiter certaines situations de manipulation ou de coordination d'engins.

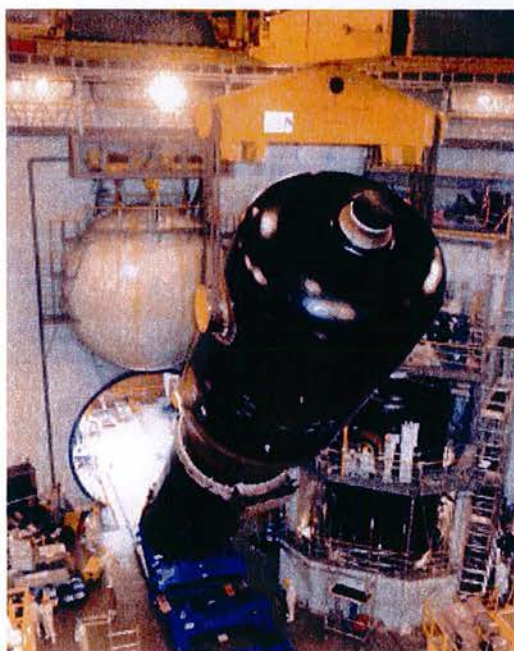
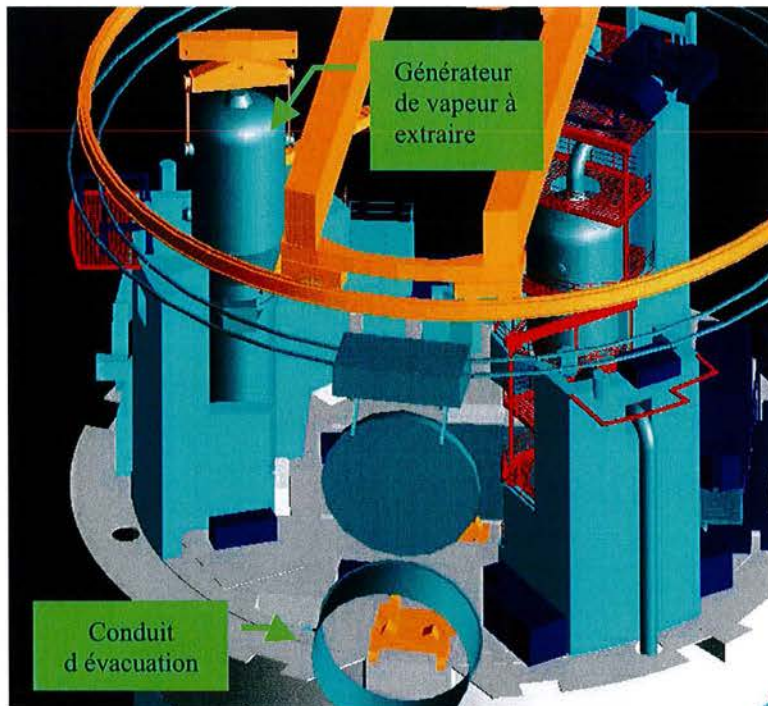
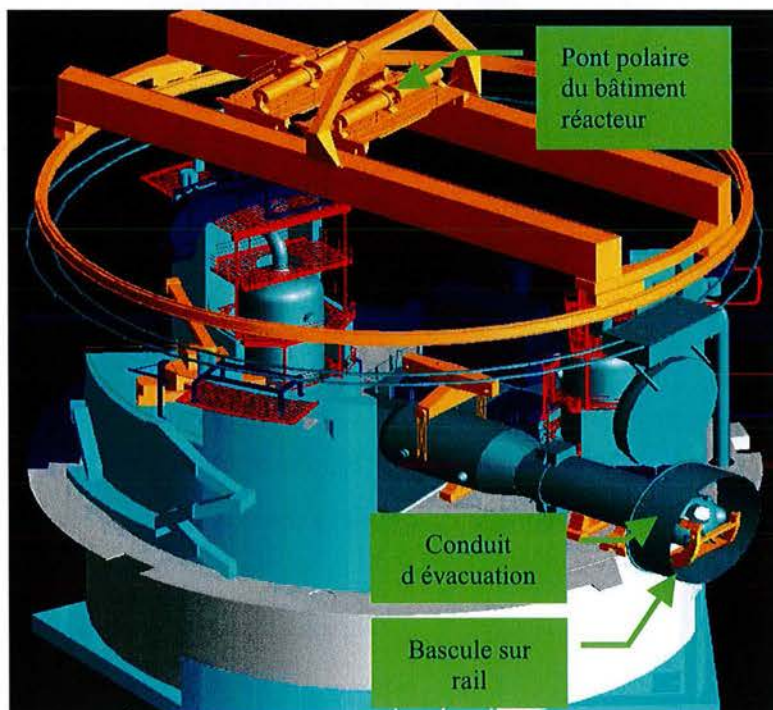


Figure 4.4 – Aperçu de la partie supérieure du bâtiment réacteur d'une centrale nucléaire pendant une opération de changement de générateur de vapeur

La scène statique modélisée dans *PDMS* correspond à 3.1 Mo de primitives géométrique *PDMS*. Les images suivantes sont extraites de *PDMS*. Elles définissent les positions initiales et finales à relier à l'aide d'un mouvement sans collision si celui-ci existe.



Figures 4.5 – Configurations initiales de l'opération de changement de générateur de vapeur



Figures 4.6 – Configuration finale que l'on cherche à atteindre par un mouvement sans collision et respectant la phase de coordination du pont polaire et du chariot sur rail

1.2 Les premiers résultats

Les premiers exemples ont été traités dans le logiciel *Move3D*. En particulier, les définitions du problème de planification et des liaisons des engins ont été définies au moyen de l'interface utilisateur de *Move3D* et de fichiers de définitions écrits manuellement pour décrire les liaisons (*.p3d). Seule la scène statique et les corps rigides constituant les engins ont été traduits depuis le format *PDMS* (*.vit) dans le format de *Move3D* (*.p3d). Ce travail de traduction a impliqué le LAAS et CADCENTRE (format propriétaire *.vit) et ne comprend que les formes des objets mais pas leurs liaisons (hormis les liaisons rigides).

La trajectoire produite par *Move3D* comprend une liste ordonnée de vecteurs réels de configurations (*.lpc). Elle est interprétée dans *PDMS* à l'aide d'une macro spécialement écrite (manuellement par CADCENTRE) à cet effet pour placer des objets liés.

Après une phase de réglage du traducteur, j'ai réalisé au sein d'EDF R&D des tests sur les temps de calcul de planification de trajectoire sur deux versions consécutives du planificateur *Move3D* (version 1.2 et 2.0). Nous présentons ci-dessous quelques-uns des tests obtenus avec *Move3D2.0* testés sur Silicon Graphics Workstation (Octane, Irix 6.5).

Comparaison de différentes méthodes pour la scène "Turbopompe TPA" et pour un problème simple de planification de mouvements avec le détecteur de collisions *V-Collide*

Nom des algorithmes	Temps CPU		Nœuds libres dans le graphe	Nombre de nœuds testés
	s	ms/nd		
MRP	1,606	5,317	302	1,757
MRP-Visi	130	--	47	562
MRP-Gaussien	134	--	84	--

Tableau 4.1 – Comportement des MRP pour l'exemple TPA

Les résultats montrent que ce type de problème est largement à la portée de l'approche à réseau probabiliste classique MRP-basique utilisant le détecteur de collision *V-Collide* (cf. §III.1). Ils laissent entrevoir la possibilité de préparer l'ensemble des opérations relatives au démontage et au déplacement d'une turbo-pompe dans un environnement encombré en une seule journée (au lieu d'une semaine).

Mais, si l'on compte le temps mis pour écrire les différents fichiers (*.p3d) et la macro d'interprétation des fichiers de trajectoire (*.lpc) pour une visualisation du résultat dans *PDMS*, le gain de temps escompté est malheureusement perdu. Les premiers tests ont donc mis en lumière un besoin d'intégration nécessaire à la valorisation de ces résultats.

Les résultats obtenus montrent par ailleurs un progrès significatif (division du temps de calcul nécessaire pour connecter deux points donnés par un facteur 10) des méthodes MRP-Visi et MRP-gaussien par rapport à la méthode MRP classique. La méthode MRP-Visi présente l'avantage de ne pas avoir de paramètres à fournir (ceux de la gaussienne) et limite l'expansion du réseau et ainsi la place nécessaire à son stockage.

Néanmoins le détecteur de collision *V-collide* présente quelques problèmes notamment durant la phase d'optimisation (minimisation de la longueur de la trajectoire type Manhattan) qui rapproche le mouvement des obstacles. Des collisions inopportunes ont été constatées. Une solution consiste à affiner le pas d'échantillonnage de la trajectoire utilisé par *V-collide*. Mais cette procédure est laborieuse et peut conduire à un ralentissement rédhibitoire des calculs.

Ce résultat a été identifié comme critique dès le début du projet MoLog. Pour pallier ce problème de détection dynamique peu robuste ou trop lente au voisinage des obstacles, les partenaires du projet MoLog (LAAS, UTRECHT) ont développé un nouveau détecteur de collision *KCD* (cf. chapitre §III).

1.3 Evolution du détecteur de collision

Le rejet rapide des configurations (collision dite statique) et des chemins de configuration (collision dite dynamique) est l'élément critique de la planification de trajectoire et en particulier des méthodes à réseaux probabilistes qui consacrent, d'après leurs concepteurs, plus de 90% de leurs temps de calcul à cette tâche. Le test des différentes techniques actuelles de détections de collisions basées sur une hiérarchisation des objets (décomposition en *primitives convexe* et regroupement hiérarchisé à l'aide de *volumes englobants* de diverses natures), puis la conception et l'intégration d'un nouveau détecteur de collision *KCD* ont donc été un élément central du projet MoLog.

Nous avons réalisé des essais sur le logiciel *MoLog* en utilisant la méthode MRP-Visi (§III.2). Nous avons utilisé deux types de détecteurs de collision, *KCD* et *V-Collide* (§III.1). Ils ont été testés sur une station de travail Silicon Graphics (Octane de 2Gb de Mémoire, OS Irix 6.5).

Le tableau suivant donne les différentes caractéristiques du réseau généré ainsi que le nombre et le type d'appel au détecteur de collision (configuration isolée ou méthode locale) pour les deux détecteurs *V-collide* et *KCD*. Le générateur aléatoire utilisé dans le planificateur de mouvements *Molog* a toujours été initialisé avec la même graine (seed). Les différents essais étaient donc rigoureusement identiques. Le calcul du réseau cesse quand un chemin a été trouvé entre les configurations initiales et finales. Nous présentons aussi le temps d'initialisation du détecteur de collisions qui correspond à la hiérarchisation convexe décrite dans §III.1 .

<i>Détecteur de collision</i>	V-collide	KCD
<i>Algorithme & stratégie d'échantillonnage</i>	MRP-Visi	MRP-Visi
Calcul du réseau	397 s	227 s
Nœuds dans le graphe	57	52
Nœuds dans l'espace libre	159	122
Configuration totale générée	1 390	1 036
Nb d'appel au détecteur de collision	29 456	30 110
Nb d'appel à la méthode locale	5 013	3 422
Mémoire allouée (en ko)	202 448	118 608
Temps d'initialisation du détecteur de collisions	~ 20 s	~ 30 s
Temps d'optimisation	13.6 s	10.1 s

Tableau 4.2 – Comportement des détecteurs de collision MRP pour l'exemple TPA

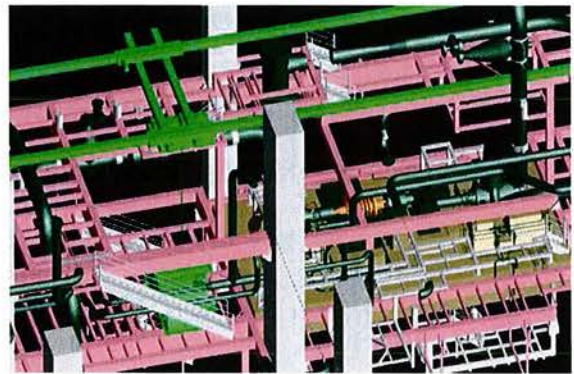
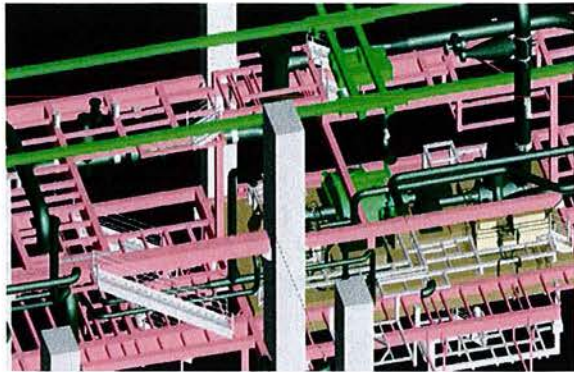
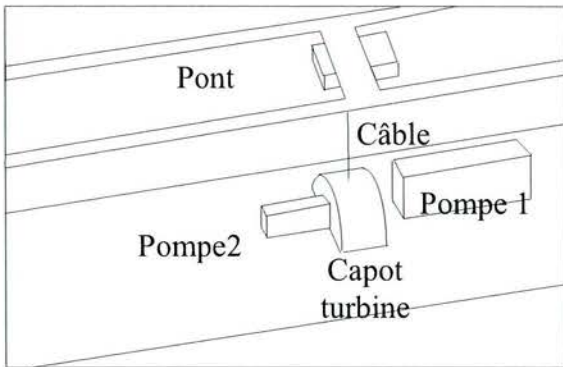
Les temps sont donnés en secondes CPU

() Valeur non disponible.*

Nous avons utilisé pour les tests les paramètres suivants :

```
solve_mpg -x -b -i 3 -tol 33 -s manhattan -p visibility {-c vcollide | -c kcd -sd 0.5 }
```

Pour des performances voisines, le détecteur *KCD* n'entraîne pas de collision dynamique inopportune au voisinage des obstacles dans les environnements industriels testés et plus particulièrement pour les exemples de la turbopompe (TPA) et du générateur de vapeur (GV). Nous présentons ci-dessous quelques images extraites du mouvement de l'exemple TPA après optimisation.



Figures 4.7 – Extrait de la trajectoire continue obtenue après optimisation (méthode locale et distance de type Manhattan)

2. Les évolutions de l'architecture du démonstrateur MoLog

Compte tenu des difficultés rencontrées lors des premiers essais, la conception d'une architecture d'intégration apte à valoriser les méthodes à réseaux probabiliste au sein du logiciel *PDMS* est apparue comme un point important de mon travail. *PDMS* dispose d'une bibliothèque d'environnement et de pièces standards pour le nucléaire et est capable de visualiser de très vastes locaux industriels. Sur ces points, il est irremplaçable à l'heure actuelle. Il est de plus utilisé dans de nombreuses unités d'ingénierie nucléaire d'EDF. Cependant, il ne permet pas de gérer le mouvement et les liaisons d'objets. L'essentiel des difficultés rencontrées sont liées à ces limites.

2.1 Le cadre des tests des méthodes à réseaux probabilistes pour des environnements industriels

L'**objectif** du premier niveau d'intégration était de pouvoir lancer la résolution d'un problème de planification de mouvement depuis *PDMS*. Cette première structure est décrite sur la figure suivante.

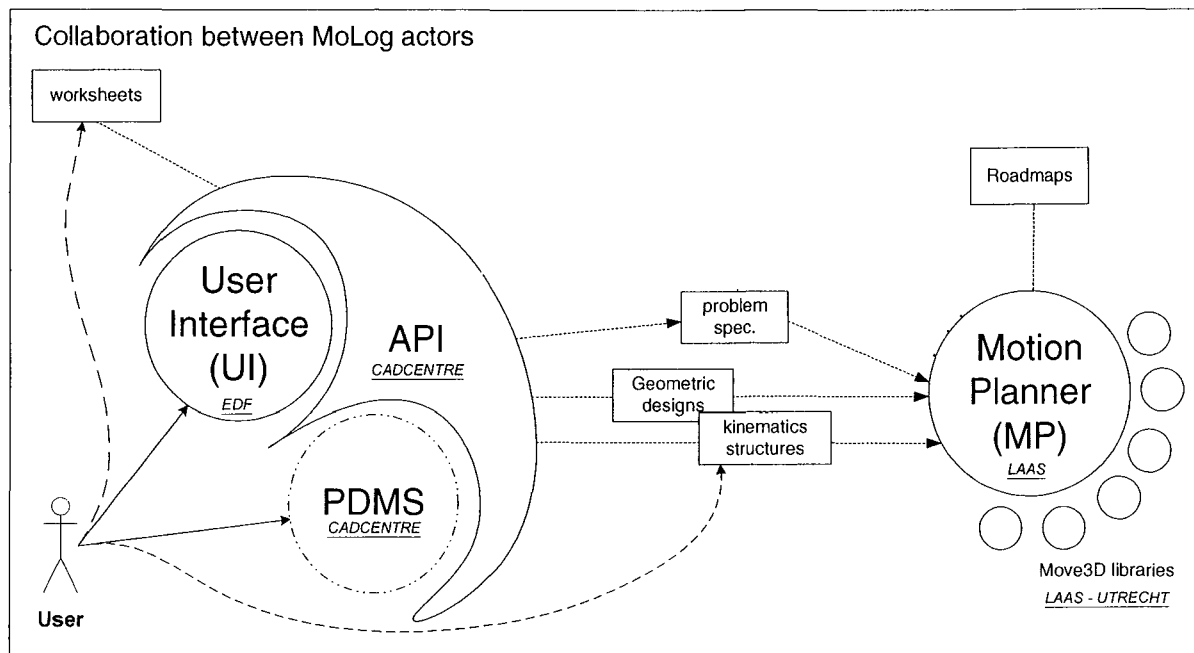


Figure 4.8 – Premier niveau d'intégration: échange "monolithique" de fichiers

Nous énumérons ci-dessous les différents éléments qui interviennent pour l'intégration entre le logiciel CAO *PDMS* et le module de planification de mouvement (MP). Ces éléments comprennent :

Deux modules logiciels fondamentaux :

- **PDMS** : Le logiciel CAO *PDMS* est utilisé par EDF pour la modélisation et la visualisation de maquettes numériques statiques de centrales nucléaires et a été retenu pour constituer le noyau du logiciel MoLog. La société Cadcentre, éditrice du logiciel *PDMS*, a eu pour charge d'introduire une interface logicielle (Application Programmable Interface) autour de *PDMS* pour visualiser des mouvements au sein de son interface graphique et communiquer avec les différents modules du projet.
- **MP** : le module de planification de mouvements (Motion Planner - MP) qui produit un réseau à partir de contraintes géométriques **fixées**. Si ce réseau est déjà construit, il peut en extraire rapidement des mouvements et les optimiser.

Interfaces :

- **UI** : l'interface utilisateur (User Interface - UI) du démonstrateur MoLog permet de **décomposer** les étapes critiques d'un chantier (tâches, opérations) en fichiers de spécification

des contraintes ou "worksheet" et en spécification des problèmes à résoudre au moyen du module de planification du mouvement.

- **API : l'interface programmable d'application** de chaque module, et plus particulièrement celle de *PDMS*, qui gère l'interface graphique. Le logiciel *PDMS* ne permettant pas de définir directement un mouvement rigide ou articulé, la définition du placement des objets mobiles dans l'espace à partir d'un vecteur de réels paramétrant les configurations correspond à l'exécution d'une macro dans le langage PML de *PDMS*.

Données à échanger entre les deux modules logiciel :

- **L'index ("worksheet")** de toutes les contraintes géométriques à prendre en compte (design de la géométrie statique, design de la géométrie mobile, règles) qui doivent être collectées avant de définir une procédure de planification de mouvements. Ces informations correspondent à des fichiers qui spécifient les contraintes géométriques d'un problème de planification du mouvement.
- **La spécification du problème de planification de mouvement ("Problem spec")** : Ces données contiennent au minimum la zone délimitant le réseau ("Roadmap Zone"), la méthode de construction du réseau (et son critère d'arrêt) et les paramètres de recherche et d'optimisation du chemin dans le réseau.
- **Les listes de configurations** ("placement data" en Anglais) qui permettent de positionner tout objet mobile à un instant donné pour le dessiner dans *PDMS*. Ces données de placements correspondent à un vecteur des configurations qui est interprétée par l'interface API de *PDMS*. Les trajectoires solutions produites par le module MP se présentent comme une liste de configurations qui peut être lue et interprétée séquentiellement pour produire un mouvement dans l'interface graphique de *PDMS*.

La première étape a été la conception d'une Interface Utilisateur (IU) et d'une Interface Programmable des Applications (API) qui soient à même de gérer les fonctions CAD. La communication a été assurée par le biais d'échanges de fichiers jusqu'à la deuxième année du projet. La traduction automatique de fichiers a été le premier obstacle pour le couplage avec les fichiers parce que le format des données géométriques et les règles cinématiques étaient l'un comme l'autre

spécifiques à *PDMS* et *Move3D*. C'est pourquoi il a fallu attendre la fin de la première année du projet avant d'obtenir les premiers résultats du couplage effectifs des deux modules logiciels.

2.2 La gestion d'une série de problèmes standards de planification du mouvement

L'**objectif** est de pouvoir modifier de manière interactive (à partir de l'interface utilisateur du logiciel MoLog) les contraintes géométriques (modification des infrastructures ou des liaisons parmi les structures prédéfinies) ainsi que les différents paramètres de génération et de visualisation du mouvement (raffinement de l'optimisation, passage par des points intermédiaires,...). La nouvelle architecture doit permettre en particulier de lancer en série des requêtes de construction de plusieurs réseaux répondant à des contraintes géométriques distinctes. Ceci apporte des potentialités nouvelles à court terme quant à l'intégration des procédures de planification de manutention évoquées dans la partie précédente.

Création d'un nouveau module pour la gestion de plusieurs problèmes de planification

- le serveur de planification de mouvements ("Motion Planner Server" - MPS) décompose toute requête provenant de l'utilisateur en sous-tâches traitables par le module MP. Il réinitialise le module MP lors des **changements sur les contraintes géométriques** et gère le stockage des résultats sous forme de bibliothèques de réseaux à contraintes géométriques **fixées**.

Nous avons introduit une architecture de type client-serveur appelée MPS pour "Motion Planning Server" à la fin de la deuxième année du projet dans le but de :

- réaliser une réinitialisation après des modifications géométriques, de manière à permettre des études d'impact géométrique en supprimant les obstacles critiques,
- pouvoir utiliser *PDMS* pendant que MP est en cours d'utilisation,
- être en mesure d'utiliser *PDMS* et MP sur deux stations Unix distinctes.

Le diagramme ci-dessous présente un aperçu global de l'architecture qui a été présentée à la fin de la seconde année du projet MoLog.

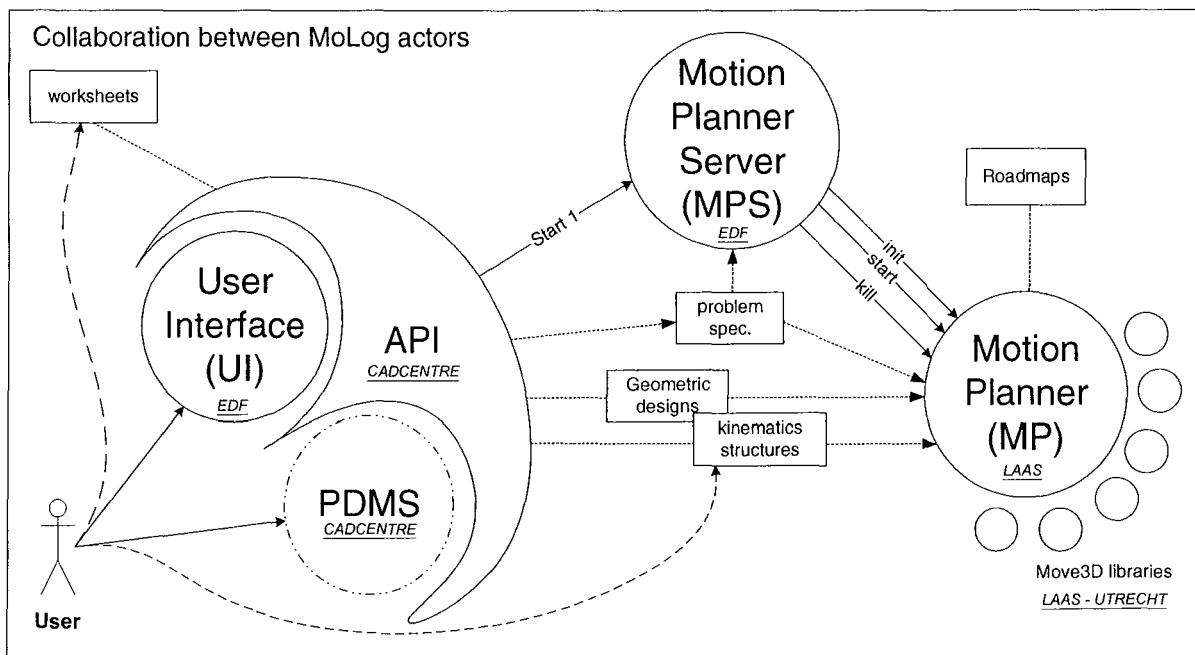


Figure 4.9 – Deuxième niveau d'intégration par échange de fichiers

Le premier prototype d'une telle architecture a été mis en place en mars 2001 et a permis la spécification et la résolution en quelques heures du problème de changement d'une turbo-pompe dans la salle des machine d'une centrale électrique à l'aide d'une grue [SCH01]. La modification interactive des infrastructures a été testée avec succès sur l'exemple de la turbo-pompe alimentaire. Un passage pour l'évacuation de la pompe a été créé, le réseau a été complété et le mouvement de la pompe à travers le passage a été généré rapidement. Ce problème avait été précédemment résolu avec des moyens logiciels classiques en une semaine et l'efficacité de la démarche a ainsi pu être démontrée globalement.

Néanmoins, le fait que l'ensemble du problème de spécification doive être prédéfini dans un fichier n'était pas bien adapté pour une utilisation interactive des fonctionnalités de MP. Par exemple, les paramètres définissant l'optimisation et la visualisation d'un mouvement doivent être choisis pendant la spécification du problème. Cela constitue une entrave à un travail interactif. Ce besoin d'interactivité a constitué un réel problème qui nous a conduit à identifier et à décomposer chaque fonction pour l'intégrer dans une architecture globale.

2.3 La gestion interactive des phases d'apprentissage, de recherche et d'optimisation du mouvement

Une évolution de l'architecture a donc été proposée au début de la troisième année du projet pour :

- décomposer et intégrer les différentes phases de la planification de mouvements dans l'Interface Utilisateur (IU) de sorte qu'elles puissent être appelées de manière interactive. Ceci implique l'élimination de la communication par fichiers. Le problème demeure défini dans l'Interface Utilisateur (IU) mais est désormais décomposé en plusieurs requêtes, transmises au serveur MPS par messages. Une requête est reçue dans MPS puis décomposée en tâches élémentaires MP (figure 4.10). Chaque requête peut être réitérée si besoin est (flèches rouges épaisses de la figure 4.11)
- gérer plusieurs problèmes distincts de planification dans le module MP. Chaque problème correspond à une situation où le design géométrique et les règles cinématiques ont été fixées de manière à pouvoir construire et gérer son propre réseau.

Le diagramme ci-dessous présente un aperçu global de l'architecture correspondante avec l'ensemble des tâches MP possibles pouvant être utilisées de manière interactive. Les interfaces ont aussi évolué de manière à être en adéquation avec les nouvelles fonctionnalités offertes.

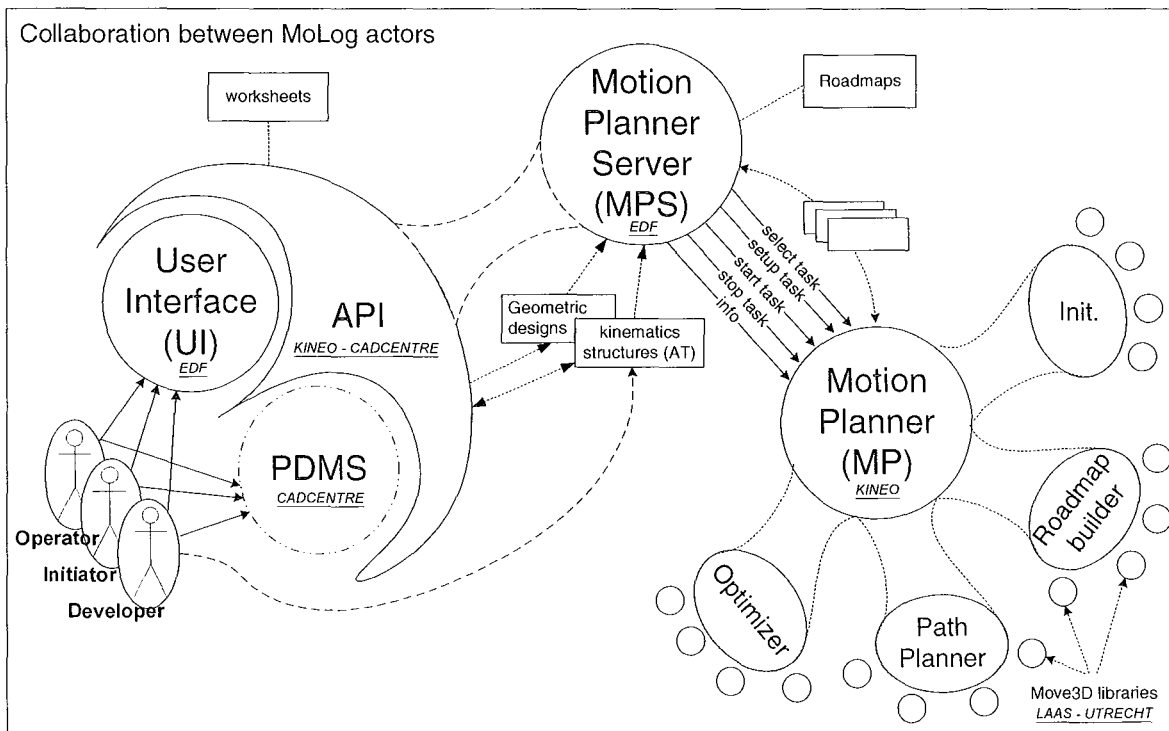
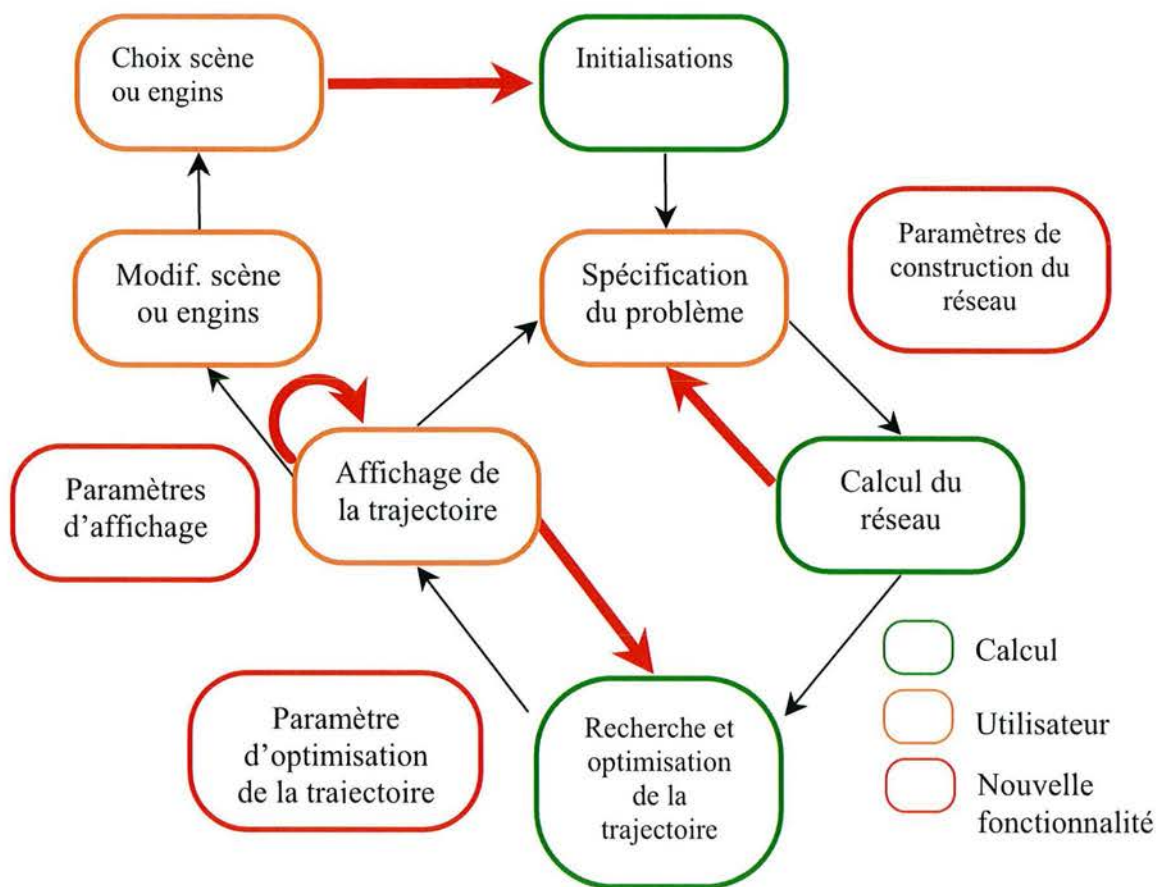


Figure 4.10 – Troisième niveau avec communication du serveur client

Données et requêtes plus détaillées :

Les données de spécification d'un problème de planification de mouvement sont décomposées de manière à être accessibles et modifiables indépendamment les unes des autres. L'utilisateur peut avoir accès si nécessaire, aux paramètres suivants depuis l'interface utilisateur:

- **les paramètres de construction du réseau.** La zone délimitant le réseau ("Roadmap Zone" en Anglais), la méthode de construction du réseau et les critères d'arrêt de la construction du réseau peuvent être modifiés,
- **les paramètres d'optimisation de la trajectoire.** Le gain escompté et le temps maximum consacré à l'optimisation de la longueur du chemin peuvent être modifiés,
- **les paramètres de visualisation.** Ils permettent de « zoomer » sur une portion donnée du mouvement avec une plus grande finesse temporelle.



Figures 4.11 – Nouvelles fonctionnalités offertes par la dernière architecture logicielle

Les limites

- L'utilisation de règles cinématiques de formats différents pendant la spécification et la résolution du problème constituait une source d'erreur et rendait le design cinématique d'un nouvel engin plutôt difficile.
- La lenteur de l'affichage d'une trajectoire (plus de 5 minutes pour un affichage séquentiel de 100 points de la trajectoires).

2.4 La centralisation du calcul cinématique : Vers l'intégration d'un module de design et de calcul cinématique

Des "nouveaux" objets de communications

- **les données de placement** ("placement data" en anglais) permettent de positionner tout objet mobile à un instant donné pour le dessiner dans *PDMS*. Ces données de placements peuvent correspondre à un vecteur des configurations, si les configurations peuvent être interprétées grâce à des règles cinématiques dans chaque module, ou plus généralement à un ensemble de matrices homogènes que nous appellerons PARMO (Position of All Rigid Mobile Objects) (cf. §2.4).

Bien que des outils algorithmiques aient été développés au cours du projet MoLog pour traiter les problèmes de planifications de manipulation d'un objet au niveau du module MP, la définition des règles et attachements cinématiques nécessite des fonctions spécifiques et exige que l'interface utilisateur soit accessible à ce dernier. Ces fonctions sont habituellement fournies par des logiciels CAM et se trouvent en dehors du champ du projet MoLog. Cependant, la redondance dans les règles cinématiques interprétées dans *PDMS* et dans le module MP, peut être évitée en définissant une requête appropriée au module MP (ou à un autre module dédié) qui centralise les procédures de design et de calcul cinématique. Une telle centralisation de la définition et des calculs cinématiques pose les bases de la création d'un module cinématique dédié. Comme indiqué au paragraphe 1.2, nous notons PARMO (Positions of All Rigid Mobile Objects) les "Positions de Tous les Objets Mobiles Rigides" à un instant donné et dans un référentiel lié à la scène statique. Comme nous l'avons déjà dit, PARMO est représenté par un ensemble de matrices homogènes. La tâche PARMO de MP consiste à créer le

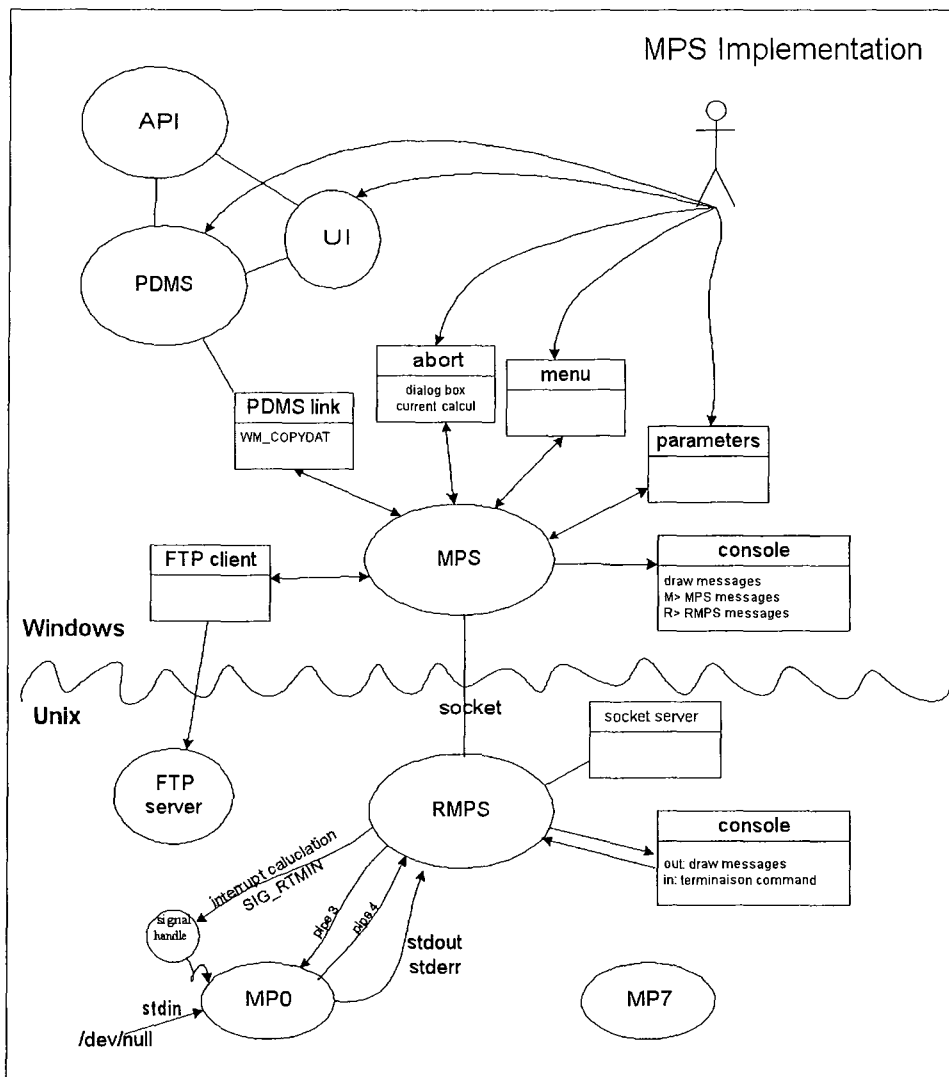


Figure 4.13 – Implémentation physique de l'architecture

Evaluation des performances

Le tableau ci-dessous donne les temps obtenus avec une station Silicon Graphics Workstation (Octane avec 2Gb de mémoire vive, OS : Irix 6.5) et avec un PC (Dell Optiplex x86 Family 6 Model 7 Stepping 3 avec 260 Mb de mémoire vive, OS NT4.00).

<i>Temps exprimés en secondes CPU</i>	Turbo-pompe alimentaire	Générateur de vapeur
Ecriture et transfert des fichiers de données géométriques et de spécification du problème	68	70
Chargement de la scène dans <i>PDMS</i> et initialisations du MP	111	135
calcul du réseau (arrêt dès que la trajectoire est trouvée)	68	62
Optimisation de la trajectoire (avec un critère d'arrêt standard)	<1	<1
Affichage séquentiel de 100 points de la trajectoire	349	471

Tableau 4.3 Evaluation des performances globales.

PDMS n'affiche pas suffisamment rapidement la réactualisation simultanée de plusieurs solides dans l'espace. Le problème n'est pas lié à l'exécution de "macro de cinématique" mais il est inhérent à *PDMS* et n'a pu être résolu par le concepteur de *PDMS* Cadcentre dans le cadre du projet MoLog. L'affichage rapide du mouvement est donc un obstacle important d'un point de vue utilisateur et nécessitera une étude spécifique de la part de Cadcentre. Cette étude dépasse le cadre du projet MoLog.

Conclusion

Le projet MoLog avait pour objectif de contribuer à la résolution de problèmes de planification de mouvements pour des dispositifs industriels correspondant à des installations complexes telles que les centrales nucléaires. Les résultats obtenus sur des cas tests représentatifs que nous avons choisis ont permis de valider les méthodes à réseaux probabilistes pour la planification de mouvement dans ce contexte [SCH00a,SCH00b,SCH01a,SCH01b]. Jusqu'à présent, aucune méthode de planification de trajectoire n'avait donné de résultats significatifs du fait de leur incapacité à gérer un nombre trop important de contraintes de non-collisions. De nouveaux algorithmes de planification de mouvement [BOO99, SIM00] et en particulier un nouveau détecteur de collision KCD [GEE00] ont été testés et ont permis d'améliorer les performances des MRP dans de telles situations. **Le traitement algorithmique de la planification de mouvement pour la préparation de chantier nucléaire est aujourd'hui beaucoup plus performant** avec des temps de résolution qui ne sont plus rédhibitoires pour les unités d'ingénierie d'EDF.

Notre approche d'intégration d'outils de résolution des contraintes géométriques pour la préparation de chantier de maintenance et de démantèlement à l'aide de maquettes numériques a été validée sur les exemples considérés et ce malgré l'hétérogénéité des composants du logiciel devant être intégrés. En effet, les logiciels existant dans le commerce ne sont pas conçus pour pouvoir traiter simultanément une forte complexité géométrique et cinématique. Actuellement, ces logiciels (type *PDMS*) sont soit dédiés à la définition et à la représentation d'objets très gros comme des bâtiments industriels, soit dédiés à la définition et à l'utilisation de mécanismes. L'introduction de nouveaux outils de planification de mouvements capables de gérer à la fois la complexité géométrique et cinématique a rendu le projet d'intégration particulièrement ambitieux. Cependant, si on se place du point de vue de l'utilisateur, ce travail d'intégration doit nécessairement être réalisé afin de permettre la conception de prototypes virtuels d'opérations de maintenance ou de démantèlement des centrales nucléaires avec des engins de manutention. **Ce travail a permis la définition d'interfaces utilisateurs orientées métier, simples et efficaces permettant de décomposer des problèmes de planification de manipulation.**

Les efforts d'intégration ont mis en évidence l'intérêt à court terme des résultats algorithmiques du projet MoLog et leur évolution permet de dégager des perspectives à court terme ou moyen terme.

Perspective à court terme de l'évolution du démonstrateur MoLog

- limiter le champ d'action des planificateurs de mouvements à un ou plusieurs objets en déplacement libre ou planaire.
- intégrer un outil de visualisation rapide du mouvement, compte tenu des difficultés d'affichage du mouvement du fait d'un rafraîchissement assez lent de l'affichage de *PDMS*.
- intégrer la capture des classes d'*homotopie* sur les chemins élémentaires utilisés [SCH02] pour garantir à l'utilisateur une « vision exhaustive » pour le mouvement libre ou planaire d'un ou de plusieurs corps rigides.

Perspective à moyen terme de l'évolution du démonstrateur MoLog

- intégrer un module de génération de modèle géométrique directs et inverses d'engins, de manière par exemple à définir ou modifier interactivement de nouveaux engins ou des liaisons virtuelles pour décrire l'ensemble des saisies ou des poses autorisées lors d'une **opération de manipulation** [SIM02,SAH02] (plutôt que de choisir parmi des contraintes prédéfinies) et en s'appuyant sur les MRPs aptes à gérer les graphes de liaisons cycliques [COR02].
- intégrer un outil de visualisation rapide du mouvement

Conclusion générale

Dans le chapitre §II.3, nous avons identifié deux axes d'adaptations d'outils de planification de trajectoire au contexte de la préparation d'un chantier :

"Définir une démarche permettant de se ramener à une situation standard pour l'outil de planification de trajectoires pour la plus large partie des situations apparaissant lors de la préparation de chantiers nucléaires"

"Réduire le temps de calcul de planification au moyen d'un pré-calcul global de manière à ce qu'ils soient adaptés à la préparation de chantiers ".

Nous détaillons ci-dessous les diverses contributions apportées à leur résolution.

Pour résoudre le **premier point**, un travail concret a été réalisé dans le cadre du projet européen MoLog. Il a été décrit dans le chapitre §IV. Nous avons tout d'abord été amenés à décomposer les problèmes géométriques complexes liés à la préparation d'un chantier nucléaire. Nous avons spécifié en collaboration avec des ergonomes et utilisateurs potentiels d'EDF **une interface utilisateur** qui guide les choix de l'utilisateur pour se ramener à une séquence de cas standards pouvant être traités par un outil de planification de trajectoires. Nous avons montré que cette procédure nécessitait une architecture logicielle spécifique qui, au-delà du module graphique PDMS, de l'interface utilisateur et du module de planification, comprend aussi un **module MPS** (Motion Planner Server) qui orchestre la résolution simultanée de plusieurs cas standards en laissant la « main » à l'utilisateur pour spécifier d'autres problèmes de planifications de trajectoires. Nous avons décrit l'architecture complexe de l'ensemble des éléments du démonstrateur Molog autour du module MPS et spécifié le développement du module MPS. La description et l'implémentation de

l'architecture, qui s'est étalée sur 3 ans, a constitué un travail lourd comprenant différents niveaux d'évolution.

Ainsi, le premier niveau d'évolution a permis de distinguer la phase d'apprentissage de la phase de recherche d'une trajectoire ; un deuxième niveau d'évolution a permis de séparer ces deux premières phases de la phase d'optimisation et de pouvoir mener plusieurs recherches de trajectoires d'affilée, *etc.* (cf. chapitre §IV). Alors qu'au départ, avait été conçue une architecture où tous les éléments étaient transmis ensemble, ce travail de thèse a permis de faire évoluer l'architecture de manière à pouvoir décomposer les problèmes complexes caractérisés par une grande diversité de contraintes et à introduire par ce biais une plus grande interactivité avec l'utilisateur du logiciel. L'efficacité de la démarche a été démontrée et a fait l'objet de démonstrations, de rapports européens [SCH00a,b,01a,b] et d'une publication dans le principal congrès d'ingénierie nucléaire [SCH01a]. Néanmoins, les problèmes d'affichage rapide de mouvements dans PDMS demeurent et sont liés de manière intime au logiciel PDMS, qui, rappelons le, n'est pas conçu pour gérer et afficher des mouvements. Une solution à ce problème consisterait à utiliser le module de visualisation (Viewer) de PDMS pour cette tâche.

Pour résoudre le **second point**, un travail concret a été réalisé dans le cadre du projet européen MoLog. Cette étape a consisté à adapter les méthodes probabilistes existantes (MRP-basique, MRP-gaussien et MRP-Visi décrites dans §III.2) aux situations compliquées rencontrées par EDF et à les tester dans le démonstrateur MoLog. En effet, si l'on sait que, en pratique, les méthodes probabilistes réagissent mieux que les autres face aux problèmes complexes en terme de contraintes, elles n'ont jamais été testées pour des problèmes aussi difficiles que ceux posés par EDF. J'ai travaillé durant ma thèse à intégrer ces adaptations et à tester leur fonctionnement dans les différents modules du démonstrateur MoLog. Les différents tests effectués notamment sur les deux exemples caractéristiques des chantiers de maintenance que nous avons choisis nous ont permis de **valider les méthodes PRM** car les temps de pré-calcul correspondant à la phase d'apprentissage se sont révélés largement inférieurs à une nuit de calcul. Les temps de recherche et d'optimisation permettent quant à eux une réelle interactivité (inférieurs à une minute).

J'ai mené, dans le cadre de la résolution du second point, un travail plus prospectif. J'ai montré que les réseaux probabilistes peuvent pendant un pré-calcul d'apprentissage capturer d'autres propriétés topologiques que la *connexité* d'un *espace des configurations libre* en identifiant notamment des propriétés d'*homotopie* sur les chemins élémentaires utilisés dans le réseau pré-calculé. Cette information permet de ramener tous les traitements ultérieurs d'un chemin (introduction de

nouvelles contraintes géométriques, de nouvelles contraintes différentielles pour définir un mouvement cohérent d'un point de vue physique, optimisation de la longueur du chemin,...) dans une classe d'homotopie donnée et ainsi rendre son traitement beaucoup plus rapide [GRI88, FER96] et donc plus interactif. Dans le chapitre §II.3, nous avons montré qu'il était possible d'un point de vue théorique de capturer les classes d'homotopie et nous avons précisé les hypothèses à définir pour y parvenir. Dans le chapitre §III.3, nous avons proposé un **nouvel algorithme MRP-CH** exploitant les qualités des Méthodes à Réseaux Probabilistes basé sur les domaines de visibilité pour capturer les classes d'homotopie de manière complète (toutes les classes d'homotopie sont capturées) et concise (c'est-à-dire non redondante).

Nous avons montré par des simulations réalisées sous le logiciel Matlab que nous parvenons dans des cas simples mais représentatifs, à capturer les classes d'homotopie. Pour que cet algorithme fonctionne, il a été nécessaire de développer une approche nouvelle dans le contexte des décomposition convexes hiérarchisées: celle de la **détection de collision pour des 2-simplexes** (i.e. des triangles), c'est à dire une méthode de détection de collision en dimension 2 dans l'espace des configurations. Il s'agit d'un résultat important de cette thèse car même la détection de collision en continu sur un chemin (droite ou courbe), c'est à dire une méthode de détection de collision en dimension 1, n'existe pas dans les logiciels disponibles sur le marché. Elle existe uniquement dans les logiciels développés dans certains laboratoires de recherche (§III.1.1.3), comme c'est le cas pour KCD développé par le LAAS dans le cadre du projet MoLog.

Nous pensons que cette approche est essentielle pour une introduction plus interactive des contraintes géométriques lors des étapes successives de la préparation d'un chantier. Elle conditionne en fait l'utilisation des méthodes de planification de trajectoire dans le contexte complexe de la préparation d'un chantier où l'expertise des ingénieurs est essentielle à chaque étape de la préparation et ne peut pas toujours être formalisée *a priori*.

Une **première perspective liée à la résolution du premier point** est de faire évoluer le projet MoLog vers l'intégration d'un module cinématique pour faciliter le design de nouveaux engins et disposer aisément de ses modèles géométriques directes et inverses si cette intégration est faite dans PDMS. Cette évolution peut se faire au moyen de la structure **PARMO**, que nous avons intégré à l'architecture MoLog, qui permet de gérer l'affichage dans PDMS de la position des solides indépendamment d'un modèle géométrique.

Une **seconde perspective liée à la résolution du premier point** consiste au contraire à ne pas développer la partie cinématique, et à exploiter les aspects intéressants de MoLog sans cette composante. MoLog peut en effet permettre de résoudre des problèmes importants pour les ingénieurs tels qu'assembler et désassembler **des objets en déplacement libre** (espace des configuration de type $SE(3)^n$) ou planaire (espace des configuration de type $SE(2)^n$), les extraire d'environnements complexes sans prendre en compte les engins. Cette approche correspond à une modélisation géométrique de tâches et répond aux besoins évoqués à ce propos (§I.3).

Perspectives liées à la résolution du second point

Si l'on sait que l'algorithme MRP-CH capture toutes les classes d'homotopie avec une probabilité qui tend vers un avec le temps de calcul, sa capacité de toujours capturer toutes les classes d'homotopie en un temps raisonnable doit être confirmée. Car même si nous avons prouvé que ce nombre était fini, l'existence d'un trop grand nombre de cycles réductibles à ce stade peut empêcher le logiciel de parvenir à les réduire dans un temps acceptable.

Il conviendrait pour cela **d'évaluer MRP-CH dans le contexte d'un environnement industriel** (notamment dans le cas plus accessible à court terme de la planification d'objets en déplacement libre ou planaire) et d'évaluer de manière précise et en fonction du type d'environnement considéré le nombre de cycles réductibles créés pendant la première phase de l'algorithme.

Dans le chapitre §II, nous avons posé le problème de la capture des classes d'homotopie dans un cadre très général, puisque nous avons défini son utilisation dans le cas standard d'une cellule de configurations. Les perspectives d'application de **MRP-CH au-delà d'une cellule paramétrable** sont donc envisageables. Ceci comprend l'étude d'espace des configurations comprenant plusieurs cellules ($SO(d)^n \times [0,1]^m$) et l'étude d'espace des configurations comprenant des cellules non paramétrables. Il s'agit là d'un nouveau défi à relever, qui permettrait d'intégrer les notions développées aux problèmes

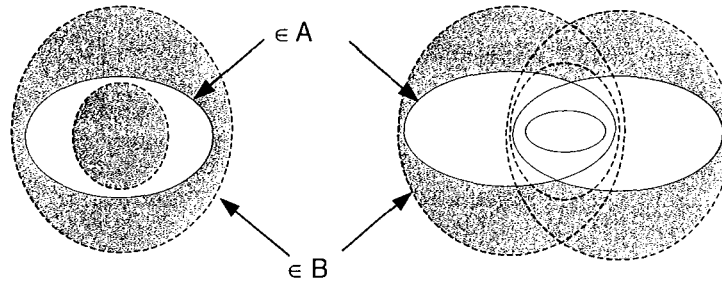
caractérisés par des boucles fermées de liaisons, rencontrés, par exemple, pour les problèmes d'usinage ou dans le pilotage de robots parallèles (les espaces de configurations, dans le cas de la plate-forme de Steward comprennent plusieurs classes d'homotopie avant même de considérer d'éventuels obstacles [CHA98]). Ces perspectives sont d'autant plus envisageables à court terme que les méthodes à réseaux probabilistes peuvent être adaptées efficacement pour générer des mouvements de mécanisme comprenant des boucles fermées de liaisons [COR02]. Ceci ouvre la perspective de décliner le problème de la capture des classes d'homotopie par un réseau (probabiliste) dans de nombreux domaines de la robotique, voire au delà, car les MRP ont déjà montré leur intérêt dans des domaines très divers comme l'étude de mécanismes réactionnels en biochimie.

ANNEXE 1: ESPACE TOPOLOGIQUE ET GROUPE FONDAMENTAL D'HOMOTOPIE

Espaces topologiques

Une *topologie* sur un ensemble X est une collection B de sous-ensembles de X comprenant X tout entier et l'ensemble vide, l'union d'éléments de B est élément de B , l'intersection d'un nombre fini d'éléments de B est un élément de B . Un ensemble X muni d'une topologie B est un *espace topologique* (X, B) . Les ensembles de B sont des ensembles ouverts et les compléments des ensembles *ouverts* sont des ensembles *fermés* de X . Un *voisinage* d'un point x de X est un ensemble ouvert qui contient x . Un *recouvrement* est une collection d'ensembles dont l'union est X . X est *compact* si tout recouvrement de X à l'aide d'ensembles ouverts admet un *sous-recouvrement* fini. X est *connexe* si les seuls sous-ensembles de X qui sont à la fois ouverts et fermés sont X et l'ensemble vide. La topologie *induite* (ou de *sous-espace*) d'un ensemble Y inclus dans X est l'ensemble des traces de la topologie de B sur Y i.e. $Y \subseteq X \quad B_Y = \{Y \cap X, X \in B\}$. Y est alors appelé *sous-espace* de l'espace topologique X . L'espace *euclidien* de dimension n est l'ensemble des n -uplets de réels. La *norme euclidienne* et la *distance euclidienne* sont respectivement définies par : $x \in \mathfrak{R}^n, \|x\| = (\sum x_i^2)^{1/2}$ et $x, y \in \mathfrak{R}^n \quad d(x, y) = \|x - y\|$. Une boule ouverte de centre $z \in \mathfrak{R}^n$ et de rayon $\varepsilon > 0$ est définie par $B(z, \varepsilon) = \{x / d(x, z) < \varepsilon\}$.

Une famille \mathcal{A} d'ouverts de X tels que $X = \bigcup_{A \in \mathcal{A}} A$ et $\forall A_1, A_2 \in \mathcal{A}, \forall x \in A_1 \cap A_2, \exists A_x \in \mathcal{A} / x \in A_x \subseteq A_1 \cap A_2$ est une *base topologique* de X . Deux familles \mathcal{A} et \mathcal{B} sont des *équivalents topologiques* si et seulement si $\forall A \in \mathcal{A}, \exists B \in \mathcal{B} / B \subset A$ et $\forall B \in \mathcal{B}, \exists A \in \mathcal{A} / A \subset B$. Si \mathcal{B} est une base topologique de X équivalente à \mathcal{A} , alors \mathcal{A} est une base topologique de X .



L'ensemble des boules ouvertes est une base topologique de \mathfrak{R}^n et génère la *topologie euclidienne*. Tout sous-ensemble Y de \mathfrak{R}^n définit une topologie de sous-espace topologique.

Groupe fondamental d'homotopie

Soit X un espace topologique. Un *chemin* de X est une carte continue $\sigma : [0,1] \rightarrow X$. Le chemin relie le point initial $\sigma(0)$ au point terminal $\sigma(1)$. X est *connexe par arcs* si chaque paire de points de X peut être reliée par un chemin. Deux chemins sont *équivalents* s'ils sont joints par une homotopie qui laisse les points initial et final fixes. L'*inverse d'un chemin* σ est $\sigma^{-1}(t) = \sigma(1-t)$. Le *produit* de deux chemins σ et ρ est défini si $\sigma(1) = \rho(0)$ et $(\sigma \cdot \rho)(t)$ vaut $\sigma(2t)$ pour $t \in [0, 1/2]$ et $\rho(2t-1)$ pour $t \in [1/2, 1]$. La relation d'équivalence et le produit sont compatibles, c'est-à-dire que si (σ_1, ρ_1) et (σ_2, ρ_2) sont deux paires de chemins équivalents alors $\sigma_1 \cdot \rho_1$ et $\sigma_2 \cdot \rho_2$ sont équivalents. Un chemin σ est un *lacet* si $\sigma(0) = \sigma(1) = x$, on parle alors de lacet de point-base x . Par exemple $\sigma \cdot \sigma^{-1}$ est un lacet équivalent à la carte constante $[0,1] \rightarrow x$. La classe d'équivalence des lacets de point-base x forme avec l'opération produit un groupe nommé *groupe fondamental* de X en x noté $\pi_1(X, x)$. Pour un espace X connexe par arcs les groupes $\pi_1(X, x)$ et $\pi_1(X, y)$ sont isomorphes pour tout $x, y \in X$. Le groupe fondamental est alors noté simplement $\pi(X)$.

ANNEXE 2 : TERMES GENERAUX DE GEOMETRIE DIFFERENTIELLE ET APPLICATION A $SO(3)$ et $SE(3)$

Les définitions ci-dessous sont extraites notamment de [LAF96,REI97].

Variétés différentielles

Une *variété topologique* M de dimension n est un espace topologique séparé (ce qui est le cas d'un espace métrique) dont tout point est contenu dans un ouvert homéomorphe à un ouvert de \mathfrak{R}^n . La donnée d'un ouvert et de l'homéomorphisme correspondant définit une *carte* (U, φ) . Une famille $A = (U_i, \varphi_i)_{i \in I}$ de cartes dont les ouverts recouvrent M définit un *atlas*. Deux cartes (U_i, φ_i) et (U_j, φ_j) sont *compatibles* d'ordre k si $\varphi_j \circ \varphi_i^{-1} : \varphi_i(U_i \cap U_j) \rightarrow \varphi_j(U_i \cap U_j)$ est un difféomorphisme de régularité C^k ou si $U_i \cap U_j = \emptyset$. Un atlas est de classe C^k si tout couple de cartes est compatible d'ordre k . Un atlas de classe C^k tel que toute carte compatible avec les cartes de l'atlas appartient elle-même à l'atlas définit une structure différentielle pour une variété topologique. On parlera alors de variété de classe C^k (par défaut on considérera par la suite que $k = \infty$).

Vecteur et espace tangent

Etant donné une variété M , une fonction scalaire $\lambda : p \in M \mapsto \lambda(p) \in \mathfrak{R}$ de l'ensemble Λ des fonctions scalaires définie sur M et $\sigma : t \in \mathfrak{R} \mapsto \sigma(t) \in M$, une courbe paramétrée de $\Sigma(M)$ alors on peut définir :

$$v_p^\sigma : \lambda \in \Lambda \mapsto v_p^\sigma(\lambda) = \frac{d(\lambda(\sigma(t)))}{dt} \in \mathfrak{R}$$

v_p^σ définit une relation d'équivalence sur les courbes paramétrées qui passent par p que l'on appellera un *vecteur tangent* v_p . (i.e. $\sigma_1 \sim_p \sigma_2 \Leftrightarrow \forall \lambda \in \Lambda, v_p^{\sigma_1}(\lambda) = v_p^{\sigma_2}(\lambda)$). Un vecteur définit une dérivation sur les fonctions scalaires, qui est linéaire $v_p(\alpha.f + \beta.g) = \alpha.v_p(f) + \beta.v_p(g)$ et vérifie la règle de Leibniz $v_p(f.g) = v_p(f).g + f.v_p(g)$.

Le *crochet* de deux vecteurs, défini par $\forall f, [v, w]_p(f) = v_p(w(f)) - w_p(v(f)) \in T_p M$, est un vecteur alors que par exemple ce n'est pas le cas de $(v \circ w)_p \notin T_p M$. Un résultat fondamental de la géométrie différentielle est que l'ensemble des vecteurs tangents en p , noté $T_p M$, a une structure d'espace vectoriel pour l'addition et la multiplication par un scalaire. $(v_p + w_p)(f) = v_p f + w_p f$ et $v_p(\alpha.f) = (\alpha.v_p)(f)$.

De ce fait, étant donnée une carte (U, φ) de coordonnées locales (x_1, \dots, x_n) ,

l'ensemble des dérivations $\frac{\partial \cdot}{\partial x_i}$ forme une *base naturelle* de $T_p M$ où nous pouvons écrire

$v_p = \sum_{i=1}^n v_i \frac{\partial \cdot}{\partial x_i} \in T_p M$. La base duale de cet espace vectoriel notée $(dx_1, \dots, dx_n) \langle dx_i, \frac{\partial \cdot}{\partial x_j} \rangle = \delta_{i,j}$

représente les formes $\omega_p(\cdot) = \sum_{i=1}^n \omega_i dx_i(\cdot) \in (T_p M)^*$ avec $\omega \in \mathfrak{R}$. La différentielle de f s'obtient

alors par $df(x) = \sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i$. La structure locale d'espace vectoriel permet de définir le produit

extérieur (\wedge) et tensoriel (\otimes) des formes.

Définition : Applications entre deux variétés

Soient M et N deux variétés C^k . Une application continue F de M dans N est dite de classe C^k si quel que soit $a \in M$, il existe une carte (U, φ) de M , avec $a \in U$, une carte (V, ψ) de N , avec $F(a) \in V$ tel que l'application $\psi \circ F \circ \varphi^{-1} : \varphi(F^{-1}(V) \cap U) \rightarrow \psi(V)$ soit de classe C^k .

$$\begin{array}{ccc}
 & F & \\
 U \subset M & \rightarrow & V \subset N \\
 \downarrow \varphi & & \downarrow \psi \\
 \varphi(U) & \rightarrow & \psi(V) \\
 & \psi \circ F \circ \varphi^{-1} &
 \end{array}$$

Un difféomorphisme de classe C^k entre M et N est une bijection qui est C^k ainsi que sa bijection réciproque. L'application linéaire tangente est déduite par passage au quotient de l'application $\sigma \mapsto F \circ \sigma$.

$$\begin{array}{ccc}
 & T_p F & \\
 T_p M & \xrightarrow{\quad} & T_{F(p)} N \\
 \downarrow \theta\phi & & \downarrow \theta\phi \\
 R^m & \xrightarrow{\quad} & R^n \\
 & d_{\phi(p)}(\phi \circ F \circ \phi^{-1}) &
 \end{array}$$

Composition et difféomorphismes $T_p(f \circ g) = T_{f(p)}g \circ T_p f$

Algèbre de Lie

Par définition un vecteur tangent vérifie les propriétés d'une dérivation sur les fonctions scalaires Λ (linéarité, dérivation d'un produit $\forall \lambda, \gamma \in \Lambda, v_p(\lambda\gamma) = \lambda(p)(v_p(\gamma)) + v_p(\lambda)\gamma(p)$) et une structure d'algèbre pour l'opération interne du crochet de Lie de deux vecteurs définit par :

$$[v, w]_p(\lambda) = v_p(w(\lambda)) - w_p(v(\lambda)) \in T_p M$$

Dérivée de Lie d'une fonction

$L_v \lambda$ ou $v(\lambda): p \in M \mapsto v_p(\lambda) \in R = \sum_{i=1}^n v^i \partial_i \lambda$ vérifie $v_p(\lambda\gamma) = \lambda(p)(v_p(\gamma)) + v_p(\lambda)\gamma(p)$ et permet définir champ de vecteur lisse.

Courbe intégrale d'un champ de vecteur

Etant donné un champ de vecteur définit par $v: p \in M \mapsto v_p \in TM$, la courbe paramétrée sur un intervalle ouvert I qui vérifie :

$$\forall \lambda \in \Lambda, \forall t \in I, \frac{d(\lambda(\sigma(t)))}{dt} = v_{\sigma(t)}(\lambda)$$

est appelée courbe intégrale du champ de vecteur. (En appliquant le théorème de Frobenius on retrouve les résultats d'existence et d'unicité d'une telle courbe sont analogues à celles d'un système différentiel dans le cas de variétés et de champs lisses).

Application aux groupes de Lie linéaires $SO(3)$ et $SE(3)$

Le placement de deux repères (ou de deux solides) l'un par rapport à l'autre dans un plan ($d=2$) ou dans l'espace ($d=3$) revient à choisir un élément g du groupe Spécial Euclidien $SE(d) = SO(d) \times \mathbf{R}^d$ avec :

$$SO(d) = \left\{ R \in GL(d), R^T R = Id(d), \det R = 1 \right\} \quad (1)$$

$(SO(d), \cdot)$ est un sous-groupe du groupe multiplicatif des matrices inversibles $(GL(d), \cdot)$, $(\mathbf{R}^d, +)$ est un groupe additif. Une représentation matricielle de $g \in SE(d)$ est donnée par la matrice suivante dite matrice homogène:

$$g = \left(\begin{array}{ccc|c} R & & & t \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \text{ avec } t \in \mathbf{R}^d \text{ et } R \in SO(d).$$

$SE(d)$ muni du produit de matrice homogène est aussi un groupe. De plus, pour $d=3$, on notera $p_1 = [px_1, py_1, pz_1, 1]^T$ et $p_2 = [px_2, py_2, pz_2, 1]^T$ les coordonnées homogènes d'un point p quelconque exprimées dans les repères $r=1$ ou $r=2$. Ces coordonnées sont liées par la relation matricielle :

$$p_1 = g_{1,2} \cdot p_2 \quad (2)$$

Tous ces groupes et leur produits cartésiens présentent par ailleurs des caractéristiques de variétés différentielles. En tant que variété différentielles, l'ensemble des définitions présentées dans l'appendice 1 s'appliquent et l'on peut définir un *espace tangent*¹ et une *algèbre de Lie*¹ en tout point de ces variétés. Mais le fait que les relations de groupes soient compatibles avec la structure différentielle donnent lieu à de nombreuses propriétés caractéristiques. On parle dans ce cas de groupe de Lie. Dans le cas du mouvement de solides, les groupes de Lie utilisés peuvent être représentés par des matrices et les opérations de groupe par des produits matriciels ce qui rend leur étude plus aisée. On parle de groupes de Lie linéaires.

Définition : Un groupe de Lie est un groupe G muni d'une structure de variété C^∞ de façon que les applications $(g, h) \in G \times G \mapsto gh \in G$ et $g \in G \mapsto g^{-1} \in G$ soient C^∞ .

Nous noterons qu'il existent de nombreuses autres représentations des ensembles $SO(d)$ ou $SE(d)$ que les représentations matricielles. Elles sont parfois plus efficaces en terme de calculs. Nous pouvons notamment évoquer l'utilisation de quaternions [SHO 85] dont le produit nécessite moins d'opérations élémentaires que le produit de matrices de rotations. L'utilisation de doubles quaternions ou de l'algèbre de Clifford [HEST 00] permet alors de représenter $SE(3)$.

Mais nous garderons cette représentation qui ne nécessite pas de définitions supplémentaires quitte à supposer que le traitement informatique sous-jacent utilise des quaternions. Le principal avantage d'une telle représentation matricielle, (possible uniquement pour les groupes de Lie linéaires), réside dans le fait qu'elle met en évidence une *immersion*¹ dans l'ensemble des coefficients réels des matrices utilisées. La plupart des résultats présentés ci-dessous pourront être retrouvés simplement en considérant par exemple qu'un chemin $\sigma : t \in \mathbf{R} \mapsto SE(3)$ est une courbe paramétrée $\sigma : t \in \mathbf{R} \mapsto \mathbf{R}^{12}$ qui doit respecter (1). Néanmoins, la plupart des propriétés présentée ci dessous peuvent se généraliser aux groupes de Lie généraux et à l'étude de l'action d'un groupe sur variété [LAF 96].

Par ce biais on peut montrer que les espaces suivants :

$$so(d) \equiv T_{id}SO(d) = \left\{ \hat{\omega} \in \mathbf{R}^{d \times d}, \hat{\omega}^T = -\hat{\omega} \right\}$$

$$se(d) \equiv T_{id}SE(d) = \left\{ \left(\begin{array}{c|c} \hat{\omega} & v \\ \hline \mathbf{0} & 0 \end{array} \right) \hat{\omega} \in \mathbf{R}^{d \times d}, \hat{\omega}^T = -\hat{\omega}, v \in \mathbf{R}^d \right\}$$

sont les *espaces tangents*¹ de $SO(d)$ ou de $SE(d)$ en l'identité. L'exploitation des propriétés de groupes permet de ramener la définition d'un vecteur tangent en un élément g de $SO(d)$ ou de $SE(d)$ quelconque à celle de l'espace tangent en l'identité de $SO(d)$ ou de $SE(d)$. Ceci se traduit de par les relations suivantes :

$$\sigma \subset SE(d) \Rightarrow \forall t_0, \exists V_{t_0}^\sigma \in se(d) / \dot{\sigma}(t_0) = \sigma(t_0).V_{t_0}^\sigma$$

ou, de manière similaire

$$\sigma \subset SE(d) \Rightarrow \forall t_0, \exists W_{t_0}^\sigma \in se(d) / \dot{\sigma}(t_0) = W_{t_0}^\sigma \cdot \sigma(t_0)$$

Un élément W ou $V \in se(d)$ (les indices sont omis par la suite) est, au sens de la géométrie différentiel, un *vecteur tangent*¹ qui peut se décomposer sur une base de l'espace tangent en l'identité.

Pour $so(3)$ on peut choisir la base suivante,

$$\hat{w}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad \hat{w}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad \hat{w}_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$V = \sum_{k=1}^{k=3} \alpha^k \cdot \hat{w}_k \in so(3)$$

et pour $se(3)$,

$$v_1 = \begin{pmatrix} \hat{w}_1 & | & \mathbf{0} \\ \mathbf{0} & | & 0 \end{pmatrix} \quad v_2 = \begin{pmatrix} \hat{w}_2 & | & \mathbf{0} \\ \mathbf{0} & | & 0 \end{pmatrix} \quad v_3 = \begin{pmatrix} \hat{w}_3 & | & \mathbf{0} \\ \mathbf{0} & | & 0 \end{pmatrix}$$

$$v_4 = \begin{pmatrix} | & 1 \\ \mathbf{0} & | & 0 \\ \mathbf{0} & | & 0 \\ \mathbf{0} & | & 0 \end{pmatrix} \quad v_5 = \begin{pmatrix} | & 0 \\ \mathbf{0} & | & 1 \\ \mathbf{0} & | & 0 \\ \mathbf{0} & | & 0 \end{pmatrix} \quad v_6 = \begin{pmatrix} | & 0 \\ \mathbf{0} & | & 0 \\ \mathbf{0} & | & 1 \\ \mathbf{0} & | & 0 \end{pmatrix}$$

$$V = \sum_{k=1}^{k=6} \alpha^k \cdot u_k \in se(3)$$

Ces vecteurs correspondent dans l'algèbre vectoriel de Gibbs à un torseur cinématique. Ils permettent de décrire toutes les vitesses instantanées des points fixes d'un repère en mouvement par rapport à un autre. V et W sont l'expression d'un même torseur cinématique exprimé dans le premier ou le second repère avec le changement de repère suivant :

$$V_{t_0}^\sigma = \sigma(t_0)^{-1} \cdot W_{t_0}^\sigma \cdot \sigma(t_0).$$

Si l'on considère que $\sigma(t) = g_{1,2}(t)$ et la relation (2) on obtient en effet $\dot{p}_1 = (g_{1,2} \cdot \mathcal{V}^{1,2}) p_2$. Si le point p est fixe par rapport au repère 2. On retrouve sinon les lois classiques de compositions des vitesses dans l'expression $\dot{p}_1 = (g_{1,2} \cdot \mathcal{V}^{1,2}) p_2 + g_{1,2} \dot{p}_2$. Ces relations s'étendent au produit cartésien de groupe de Lie [TSA 95], ce qui permet de généraliser la composition des vitesses à une chaînes de

repères. La dérivée d'une chaîne de repères $g_{i,j} = g_{i,i+1} \cdot g_{i+1,i+2} \cdots g_{j-1,j}$ donne la loi de composition des torseurs :

$$V_{i,j} = \sum_{k=i+1}^{k=j} g_{i,k} \cdot V_{k-1,k} \cdot g_{k,j} \cdot$$

Si p est fixe dans le repère j on obtient alors :

$$\dot{p}_i = \dot{g}_{i,j} p_j = g_{i,j} V_{i,j} p_j = \sum_{k=i+1}^{k=j} g_{i,k} \cdot V_{k-1,k} \cdot g_{k,j} \cdot p_j \cdot (3)$$

Pour un torseur $V \in se(3)$ fixé, le système d'équations différentielles sur les réels $\dot{\sigma}(t) = \sigma(t) \cdot V$ et la donnée de $\sigma(0)$ définit un mouvement de vissage fournit par :

$$\begin{cases} \sigma(t) = \sigma(0) \cdot \exp(V \cdot t) \\ t \in \mathbf{R} \end{cases}$$

où l'on utilise l'exponentielle définie sur les matrices par

$$\exp(V \cdot t) = \sum_{k=0}^{\infty} \frac{V^k t^k}{k!}$$

Ce faisant, on met en évidence un sous-groupe isomorphe à $(\mathbf{R}, +)$ qui est appelé un sous-groupe à un paramètre de $SE(3)$. Nous utiliserons dans la partie suivante la fonction exponentielle pour le paramétrage des liaisons glissières et pivots. Dans ce contexte le paramètre est appelé une coordonnée articulaire. Les coordonnées articulaires d'une chaîne de liaisons glissières ou pivots seront regroupées en un vecteur q . L'expression (3) de la vitesse d'un point devient :

$$\begin{aligned} \dot{p}_i &= \sum_{k=i+1}^{k=j} g_{i,k}(q) \cdot U_{k-1,k} \cdot \dot{q}_k \cdot g_{k,j}(q) \cdot p_j = \sum_{k=i}^{k=j} A_{i,j}^k(q) \cdot \dot{q}_k \cdot p_j \\ A_{i,j}^k(q) &= g_{i,k}(q) \cdot U_{k-1,k} \cdot g_{k,j}(q) \end{aligned}$$

où les matrices $U_{k-1,k}$ sont des torseurs constants.

En fait, le résultat obtenu pour une courbe donnée peut se généraliser [LAF96, MUR94] à l'aide d'arguments de géométrie différentielle et la fonction exponentielle définit des difféomorphismes locaux de l'espace vectoriel $se(3)$ (ou \mathbf{R}^6 ses coordonnées) dans un voisinage de g_0 de $SE(d)$ appelés respectivement les coordonnées de Lie-Cartan du premier et second type.

$$g(q) = g_0 \prod_{i=1}^n \exp(v_i q_i) \text{ ou } g(q) = g_0 \cdot \exp\left(\sum_{i=1}^n v_i q'_i\right)$$

Par exemple, pour $SO(3)$, les coordonnées de Lie-Cartan de premier type donnent

$$R = \exp(\hat{w}_1 q_1) \exp(\hat{w}_2 q_2) \exp(\hat{w}_3 q_3)$$

$$R = \begin{pmatrix} \cos(q_1) & -\sin(q_1) & 0 \\ \sin(q_1) & \cos(q_1) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(q_2) & 0 & \sin(q_2) \\ 0 & 1 & 0 \\ -\sin(q_1) & 0 & \cos(q_2) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(q_3) & -\sin(q_3) \\ 0 & \sin(q_3) & \cos(q_3) \end{pmatrix}$$

qui sont un exemple de coordonnées d'Euler xyz.

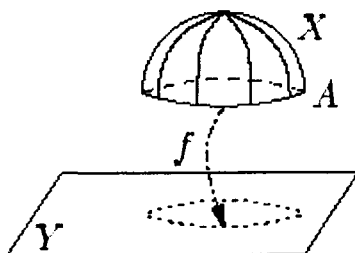
Pour achever cette brève présentation des propriétés locales des groupes de Lie linéaires nous mentionnerons leur emploi notamment pour le pilotage de satellite []. Les relations mises en évidence nous serviront surtout pour la majoration des vitesses des points d'un solide en fonction de ses coordonnées articulaires et de leur dérivées dans le paragraphe suivant. Certains auteurs travaillent directement dans $SE(3)$ notamment pour la génération de trajectoire optimales d'outils [] mais la définition de connexion ou de métriques sur $SE(3)$ n'est pas triviale. Nous chercherons plutôt à construire une paramétrisation de manière explicite.

ANNEXE 3 : DEFINITIONS RELATIVES A LA CONSTRUCTION DES CW-COMPLEXES

Nous présentons ci-dessous quelques définitions extraites de [PAU02] qui permettent de décrire les opérations de recollement de cellules pour la construction et la mise en évidence des propriétés topologiques d'un CW-complexe.

Recollements

Soient X ; Y deux espaces topologiques, A une partie de X et $f : A \rightarrow Y$ une application continue. Le *recollement* de X sur Y par f est l'espace topologique quotient $X \cup_f Y = (X \amalg Y) / \mathfrak{R}$ avec \mathfrak{R} la relation d'équivalence engendrée par $x \sim f(x)$ pour tout x dans A . (\amalg est la somme disjointe de deux ensembles)



On vérifie que si A est fermé (resp. ouvert), et $\pi : X \amalg Y \rightarrow X \cup_f Y$ est la projection canonique, alors $\pi|_Y : Y \rightarrow X \cup_f Y$ est un homéomorphisme sur son image, qui est fermée (resp. ouverte). On vérifie que si A est non vide, et si X ; Y sont connexes (resp. connexes par arcs), alors $X \cup_f Y$ aussi. Si $u : X \rightarrow Z$ et $v : Y \rightarrow Z$ sont deux application continues ,telles que $u(x) = v(f(x))$ pour tout x dans A , il existe une unique application continue $w : X \cup_f Y \rightarrow Z$ telle que $w \circ \pi|_X = u$ et $w \circ \pi|_Y = v$. Si Y est réduit à un point $*$, alors f est l'application constante de A dans Y , et l'inclusion de X dans $X \amalg \{*\}$ induit un homéomorphisme $X / \langle A \rangle \approx X \cup_f \{*\}$. En particulier, si A est fermé, alors

l'application canonique $Y \rightarrow X \cup_f Y$ est un homéomorphisme sur son image, qui est fermée. On identifie alors Y avec son image dans $X \cup_f Y$.

Recollement de cellules. Si e est une cellule (cf. §II.2), on note ∂e son bord en tant que variété topologique et $\overset{\circ}{e} = e - \partial e$. Si $n = 0$, on convient que e est un singleton, que ∂e est vide, et $\overset{\circ}{e} = e$. Si $f: e \rightarrow X$ est une application continue, on note $\partial f = f|_{\partial e}$ la restriction de f au bord de e .

On dit qu'un espace topologique X est obtenu par *recollement de cellules de dimension n* à partir d'un espace topologique Y s'il existe une famille $(e_\alpha)_{\alpha \in A}$ avec e_α une cellule de dimension n munie d'une application continue $g_\alpha: \partial e_\alpha \rightarrow Y$, telle qu'il existe un homéomorphisme

$$\left(\coprod_{\alpha} e_\alpha \right) \cup_{\coprod_{\alpha} g_\alpha} Y \rightarrow X.$$

(Les g_α ne sont pas supposés être injectifs.) La donnée d'une telle famille et d'un tel homéomorphisme s'appelle une *décomposition cellulaire* de X relative à Y . Si $f_\alpha: e_\alpha \rightarrow X$ est l'application induite, alors f_α est continue et appelée l'*application caractéristique* de e_α . (Ce ne sont pas toujours des homéomorphismes sur leur image). Sa restriction $f_\alpha|_{\overset{\circ}{e}_\alpha}$ est un homéomorphisme sur son image, qui est un ouvert de X . Si Y est séparé, l'image de g_α est un compact de Y , donc un fermé de Y . Si Y est séparé, la topologie de X est la topologie faible définie par la famille $\{ Y \} \cup \{ f_\alpha(e_\alpha) \}$ fermés de X . Les e_α (par abus $f(e_\alpha)$) s'appellent les *cellules de X relatives à Y* . On identifie $\overset{\circ}{e}_\alpha$ avec son image dans X par f_α , et on appelle les $\overset{\circ}{e}_\alpha$ les *cellules ouvertes de X relatives à Y* . Ce sont les composantes connexes de $X - Y$. L'application $\partial f_\alpha = g_\alpha: \partial e_\alpha \rightarrow Y \subset X$ est appelée l'*application d'attachement* de la cellule e_α de X sur Y . Si $n = 0$, alors X est (homéomorphe à) l'espace topologique somme disjointe de X et de l'ensemble A muni de la topologie discrète.

Topologie faible. Soit X un ensemble et $(X_i)_{i \in I}$ une famille de parties de topologie finale sur X définie par $(f_i)_{i \in I}$ est appelée la *topologie faible* définie par $(X_i)_{i \in I}$. Une partie F de X est fermée si et seulement si $F \cap X_i$ est fermée dans X_i pour tout i dans I ; une partie O de X est ouverte si et seulement si $O \cap X_i$ est ouverte dans X_i pour tout i dans I ; pour tout espace topologique Y , une application $f: X \rightarrow Y$ est continue si et seulement si sa restriction $f|_{X_i}: X_i \rightarrow Y$ à X_i est continue pour tout i dans I .

Quelques propriétés des CW-complexes finis. Un CW-complexe fini (cf II.3) est compact. La topologie de X est la *topologie faible* définie par la famille de ses cellules. Il est alors de dimension finie. Le sous-espace $X^{(n)}$ s'appelle le *n -squelette* de X . Il est fermé dans X . Le 0-squelette $X^{(0)}$

est un espace discret. On appelle décomposition cellulaire de X la donnée pour tout $n \in \mathbb{N}$ d'une décomposition cellulaire de $X^{(n)}$ relative à $X^{(n-1)}$. Les cellules, cellules ouvertes, applications caractéristiques, applications d'attachement de (la décomposition cellulaire fixée de) X sont celles des $X^{(n)}$ relatives aux $X^{(n-1)}$. L'ensemble X est réunion disjointe de ses cellules ouvertes (qui sont les composantes connexes des $X^{(n)} - X^{(n-1)}$). Une cellule de X de dimension 0 est appelée un *sommet* de X . Une cellule (resp. cellule ouverte) de X de dimension 1 est appelée une *arête* (resp. *arête ouverte*) de X . L'adhérence d'une arête ouverte est formée d'un ou deux sommets de X , appelés les *extrémités* de l'arête. La *dimension* de X est la borne supérieure des dimensions de ses cellules ouvertes (ce qui est bien défini par le théorème d'invariance du domaine 1.1). Un *graphe* (topologique) est un *CW-complexe* de dimension ≤ 1 .

Un sous-*CW-complexe* de X est un sous-espace topologique Y de X , tel que si $Y^{(n)} = X^{(n)} \cap Y$, alors Y muni de la famille des $Y^{(n)}$ est un *CW-complexe*. Il faut et il suffit pour cela que Y soit union de cellules ouvertes de X dont l'adhérence est contenue dans Y . Par exemple, $X^{(k)}$ est un *sous-CW-complexe* de X de dimension $\leq k$. Si $X ; Y$ sont des *CW-complexes*, une application continue de X dans Y est dite *cellulaire* si elle envoie le n -squelette de X dans le n -squelette de Y pour tout n .

ANNEXE 4 : DEFINITIONS RELATIVES A LA CONSTRUCTION DU GROUPE FONDAMENTAL D'UN POLYEDRE CONNEXE

Nous présentons ci-dessous quelques définitions extraites de [DEY99,REI97] qui permettent de décrire la construction et les propriétés topologiques d'un complexe simplicial (indépendamment des CW-complexes dont il constitue une sous-classe).

Nous utilisons la structure d'espace vectoriel de \mathbb{R}^n avec la somme de vecteurs et la multiplication d'un vecteur par un scalaire. Nous utiliserons indistinctement le terme de point ou de vecteur. Pour un ensemble de points de \mathbb{R}^n $T = \{p_0, p_1, \dots, p_t\}$, l'enveloppe affine est :

$$\text{aff}(T) = \left\{ \sum_i \lambda_i p_i, \sum_i \lambda_i = 1 \right\}.$$

T est *indépendant affine* si $\text{aff}(T)$ est différent des enveloppes affines de tout sous-ensemble de T . Un k -*plat* est l'enveloppe affine de $k+1$ points indépendants affines. L'*enveloppe convexe* de T est :

$$\text{conv}(T) = \{x \in \text{aff}(T) \mid \lambda_{i=0..t} \geq 0\}.$$

Un k -*simplexe* est l'enveloppe convexe de $k+1$ points indépendants affines. Un d -simplexe est un sommet pour $d=0$, une arête pour $d=1$, un triangle pour $d=2$, un tétraèdre pour $d=3$. Si $T = \{p_0, p_1, \dots, p_k\}$ est indépendant affine alors $\sigma = \text{conv}(T)$ est un k -simplexe de dimension $\dim(\sigma) = \text{Card}(T) - 1 = k$. Une *face* de σ est un simplexe $\tau = \text{conv}(U)$ avec $U \subseteq T$. Les *coordonnées barycentriques* d'un point x de σ sont les réels λ_i avec :

$$\sum_{i=0..k} \lambda_i \cdot p_i = x \text{ et } \sum_{i=0..k} \lambda_i = 1.$$

Le barycentre de σ est le point $b(\sigma)$ de coordonnées barycentriques $\lambda_i = \frac{1}{k+1}$ pour tout i .

Un *complexe simplicial* est une collection finie K de simplexes avec les propriétés suivantes (1) si $\sigma \in K$ et τ est une face de σ alors $\tau \in K$, et (2) si $\sigma, \sigma' \in K$ alors $\sigma \cap \sigma'$ est vide ou il a une face commune des deux simplexes.

L'ensemble des sommets est $\text{Vert}(K) = \{ \sigma \in K \mid \dim \sigma = 0 \}$. L'espace sous-jacent est la partie de l'espace recouvert par les simplexes de K : $|K| = \cup K$ (un *polyèdre*). La *dimension* est $\dim(K) = \max \{ \dim(\sigma) \mid \sigma \in K \}$. Un *sous-complexe simplicial* est un complexe simplicial $L \subseteq K$.

Une *triangulation* de d'un espace topologique X est un complexe simplicial K muni d'un homéomorphisme $h: |K| \rightarrow X$. X est *triangulable* s'il admet une triangulation. Une *rétraction* d'un espace topologique X à un sous-espace Y est une carte $r: X \rightarrow Y$ avec $r(y) = y$ pour tout $y \in Y$.

Un *recouvrement* d'un espace topologique X est une collection C de sous-ensembles dont l'union est $X = \cup C$. Par exemple l'ensemble des d -faces d'un polyèdre convexe de dimension $d+1$ est un recouvrement de son bord. La *nervure d'un recouvrement* C est un *système de sous-collections* d'intersection non nulle. $\text{Nrv } C = \{ X \subseteq C \mid \cap X \neq \emptyset \}$. La nervure est un complexe simplicial abstrait *i.e.* vérifiant $X \in \text{Nrv } C$ et $Y \subseteq X$ implique $Y \in \text{Nrv } C$. Une réalisation géométrique du complexe abstrait peut être donnée en *cartographiant* chaque $C \in C$ en un point $p(C)$ de \mathbb{R}^k . Chaque *collection* $X \in \text{Nrv } C$ est représentée par l'enveloppe convexe des points $p(C)$ pour $C \in X$. Si C est fini alors $\text{Nrv } C$ est fini et si k est suffisamment grand alors les points peuvent être choisis de telle manière que les simplexes résultant forment un véritable complexe simplicial. Appelons ce complexe K et considérons son espace sous-jacent $|K| = \cup_{\sigma \in K} \sigma$, qui est encore un espace topologique. Le théorème sur la nervure qui est un résultat de Leray annonce que sous des conditions simples sur les ensembles du recouvrement et leurs intersections, l'ensemble sous-jacent de K a le même type d'homotopie que X : **Si C est fini et si $\cap X$ est soit vide soit contractile pour tout $X \subseteq C$ alors $|K|$ et $X = \cup C$ sont équivalents par homotopie.** On trouve dans la littérature différentes conditions sur les ensembles du recouvrement et par voie de conséquence sur les relations topologiques entre X et $|K|$.

On montre dans les parties suivantes comment on peut étudier le groupe d'homotopie $\pi(K, P_0)$ en un point P_0 d'un polyèdre $|K|$ connexe (donc connexe par arcs) en passant à l'une de ses triangulations. On rappelle (fin de l'annexe 1) que le choix de P_0 importe peu, tous les groupes obtenus en faisant varier P_0 dans K étant isomorphes. On peut donc choisir pour P_0 un sommet de K . On

rappelle également qu'il s'agit d'étudier les groupes des classes d'équivalence des lacets tracés sur $|K|$, de point-base P_0 , la relation d'équivalence étant celle de la définie dans l'annexe 1.

Chaînes polygonales

Une suite finie $\alpha = (P_0, \dots, P_r)$ de sommets d'un complexe simplicial K s'appelle *chaîne polygonale* d'origine P_0 et d'extrémité P_r si pour tout couple de points consécutifs distincts P_i, P_{i+1} le segment $[P_i, P_{i+1}]$ est une arête de K . α est une chaîne polygonale *fermée* si $P_0 = P_r$ (Figure A1). Elle définit alors un lacet sur $|K|$ de point-base P_0 .

Remarque : Les sommets d'une chaîne polygonale peuvent se retrouver plusieurs fois dans α . La suite $\alpha_0 = P_0$ est dite *constante*. Elle définit le chemin constant d'image $\{P_0\}$ sur $|K|$.

Homotopie de chaînes polygonales, groupe polygonal

Deux chaînes polygonales fermées sont dites homotopes au sens combinatoire si on peut ramener l'une à l'autre en un nombre fini d'opérations (réalisables dans les deux sens) appartenant à l'un des deux types :

$$(1) (\dots, P, P, \dots) \leftrightarrow (\dots, P, \dots)$$

$$(2) (\dots, P, Q, R, \dots) \leftrightarrow (\dots, P, R, \dots) \text{ si } P, Q, R \text{ définissent un simplexe (cf. figure A2)}$$

Cette propriété est une relation d'équivalence entre chaînes polygonales fermées de même point base P_0 .

L'espace quotient est défini par $\pi(K, P_0)$. On peut définir sur celui-ci une opération interne par :

$$[[\alpha]], [[\hat{\alpha}]] = [[\alpha\hat{\alpha}]] \text{ (ind. des rep.) où } \alpha\hat{\alpha} \text{ est la suite obtenue en prolongeant la suite } \alpha = (P_0, \dots, P_i, \dots, P_0) \text{ par la suite } \hat{\alpha} = (P_0, \dots, Q_j, \dots, P_0).$$

$$\alpha\hat{\alpha} = (P_0, \dots, P_i, \dots, P_0, P_0, \dots, Q_j, \dots, P_0).$$

$(\pi(K, P_0); \cdot)$ est un groupe, le *groupe polygonal* de K relativement au point P_0 . $[[\alpha_0]] = [[P_0]]$ est l'élément neutre et $[[\alpha]]^{-1} = [[\alpha^{-1}]]$ est l'inverse de $[[\alpha]]$: si $\alpha = (P_0, P_1, \dots, P_r, P_0)$. $\alpha^{-1} = (P_0, P_r, \dots, P_1, P_0)$.

Exemple : cf. figure B

En faisant intervenir la notion d'approximation simpliciale (p242/2) on démontre le théorème fondamental suivant :

Théorème 1 : *Le groupe d'homotopie $\pi(|K|, P_0)$ d'un polyèdre $|K|$ est isomorphe au groupe polygonal $\pi(K, P_0)$.*

Comme le groupe $\pi(|K|, P_0)$ est isomorphe au groupe fondamental $\pi(|K|)$, on peut en déduire que $\pi(K, P_0)$ est, à un isomorphisme près, indépendant de P_0 . D'où la notation $\pi(K)$ et : $\pi(|K|) \cong \pi(K)$

(deux triangulations distinctes d'un même polyèdre ont des groupes isomorphes.

Remarque : Dans la démonstration du th. 1, on n'a besoin que du squelette $K^{(2)}$ (ensemble des simplexes éléments de K , de dimension 0, 1 ou 2 : $\pi(K, P_0) = \pi(K^{(2)}, P_0)$) (Si $|K|$ est connexe, il en est de même des polyèdres $|K^{(d)}|$).

Nous allons voir maintenant comment l'introduction de groupes libres permet d'énoncer des résultats remarquables.

Groupe libre de type fini

Soit $(a_1, \dots, a_n, \dots, \bar{a}_1, \dots, \bar{a}_n)$ un ensemble fini de *symboles* ($n \in \mathbb{Z} \setminus \{0\}$). Toute suite ω finie de ces symboles (répétition autorisée) s'appelle un *mot* : par exemple $\omega_1 = a_1 a_4 \bar{a}_5 \bar{a}_1 a_2$. Le mot vide ω_0 est celui qui ne contient aucun symbole. Si l'on juxtapose le mot $\hat{\omega}$ à la droite du mot ω , on dit que l'on construit le mot $\omega \hat{\omega}$ appelé produit de ω par $\hat{\omega}$. Enfin, pour tout i , a_i et \bar{a}_i sont dits conjugués. Deux mots sont dits *équivalents* si l'un d'entre eux peut se ramener à l'autre par suppression ou insertion de tout mot de la forme $\bar{a}_i a_i$ ou $a_i \bar{a}_i$ dans son écriture. Par exemple,

$a_5 a_1 \bar{a}_2 a_2 \bar{a}_1$ est équivalent à a_5 . Il s'agit d'une relation d'équivalence qui permet de partager l'ensemble des mots en classes $[[\omega]]$. Dans l'espace quotient F_n , on peut définir une opération interne par $[[\omega]] \cdot [[\hat{\omega}]] = [[\omega \hat{\omega}]]$, car le résultat ne dépend pas du représentant des deux classes facteurs du produit. $(F_n; \cdot)$ est un groupe d'élément neutre $[[\omega_0]]$, l'inverse $[[\omega]]^{-1}$ de la classe $[[\omega]]$ étant la classe $[[\omega^{-1}]]$ où ω^{-1} est défini de la façon suivante : a) si $\omega = a_i$, resp. \bar{a}_i , $\omega^{-1} = \bar{a}_i$, resp. a_i ; b) si ω est formé de p symboles, ω^{-1} est formé de p symboles conjugués écrits en sens inverse.

Il est d'usage de poser $\bar{a}_i = a_i^{-1}$ et d'une manière générale $a_i^n = a_i a_i \dots a_i$, (n fois pour $n \in \mathbb{Z} \setminus \{0\}$), $= \omega_0$ pour $n = 0$, $= (\bar{a}_i)^{-n}$ pour $-n \in \mathbb{Z} \setminus \{0\}$. Dans ces conditions, $\forall n \in \mathbb{Z}$, $[[a_i]]^n = [[a_i^{-n}]]$.

La partie $E_n = \{[[a_1]], \dots, [[a_n]]\}$ de F_n est un système générateur. On dit que F_n est un groupe libre de type fini ($n \in \mathbb{Z} \setminus \{0\}$) engendré par E_n . Le groupe libre F_1 engendré par $\{[[a]]\}$ est isomorphe à \mathbb{Z} .

Construction du groupe fondamental à l'aide d'un groupe libre de type fini

a) Si le polyèdre est un polyèdre connexe par arcs de dimension 1, il s'identifie à un graphe connexe : celui-ci possède n_s sommets P_i ($i = 1, 2, \dots, n_s$) et n_a arêtes. On peut extraire de ce graphe un *arbre maximal* ou *ossature*, i.e. un arbre connectant tous les sommets. Cet arbre aura $n_s - 1$ arêtes. Il reste donc $(n_a + 1 - n_s)$ arêtes ne faisant pas partie de l'arbre. On oriente et on note ces arêtes $P_i P_j$ dans le sens dit des indices croissants (si $i < j$, sommet origine = P_i , sommet extrémité = P_j : notation a_{ij} ; si $i > j$, sommet origine = P_j , sommet extrémité = P_i : notation a_{ji}). Alors :

Théorème 2 : Le groupe fondamental d'un graphe connexe est isomorphe au groupe libre de type fini F_n ($n = n_a + 1 - n_s$) d'ensemble générateur $E_n \{[a_{pq}]\}$.

b) Pour un polyèdre connexe par arcs quelconques $|K|$, on peut en tenant compte de la remarque faite après l'énoncé du théorème 1 remplacer K par $K^{(2)}$.

Comme $K^{(1)} \subset K^{(2)}$, on peut considérer le graphe associé à $K^{(1)}$ et construire un groupe F_n selon a). L'existence de simplexes 2-dimensionnels dans $K^{(2)}$ conduit alors simplement à opérer des identifications entre éléments de F_n , comme l'indique la figure C ; On obtient par ce procédé un groupe quotient. Celui-ci est isomorphe au groupe fondamental de $|K|$.

Théorème 3 : Le groupe fondamental d'un polyèdre connexe par arcs est isomorphe à un groupe quotient d'un groupe libre de type fini.

Exemple : cf. figures D1, D2.

Figure A1 : Chaînes polygonales et homotopie combinatoire

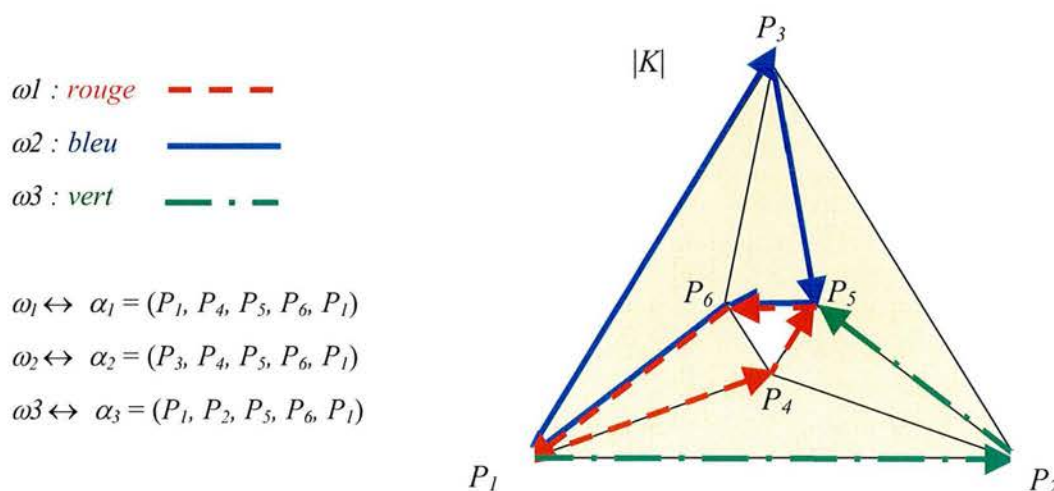


Figure A2 : Chaînes polygonales et homotopie combinatoire

α et β sont homotopes au sens combinatoire car P, Q, R définissent un simplexe de dimension 2, dans le schéma a et de dimension 1 dans le schéma b.

Dans la figure A1, α_1 et (P_1, P_2, P_3, P_1) sont homotopes au sens combinatoire, de même α_2 et (P_1) . Mais α_1 et α_2 ne sont pas homotopes au sens combinatoire.

α : chaîne rouge - - - -
 β : chaîne bleue ————

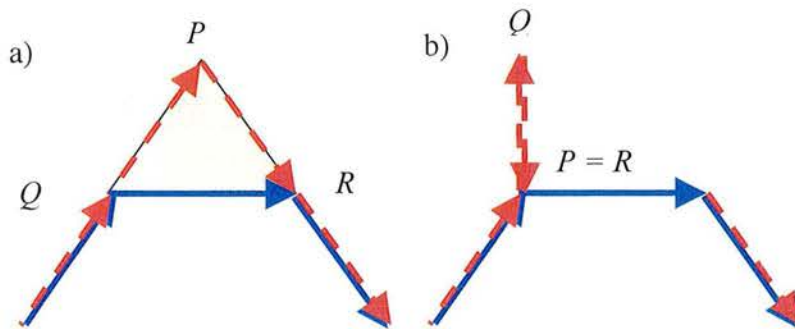


Figure B : Groupe polygonal

\mathbb{Z}^1 est homéomorphe à $|K|$ (notation : $\mathbb{Z}^1 \leftrightarrow |K|$)

$$\begin{array}{ccccccc} \pi(K, P_1) & = & \{ & [[\alpha_0]], & [[\alpha]], & [[\alpha^2]], & \dots, & [[\alpha^{-1}], & [[\alpha^{-2}], & \dots \} \\ \updownarrow & & & \updownarrow & \updownarrow & \updownarrow & & \updownarrow & \updownarrow & \\ \mathbb{Z} & = & \{ & 0, & 1, & 2, & \dots, & -1, & -2, & \dots \} \end{array}$$

Soit $\pi(\mathbb{Z}^1) \cong \mathbb{Z}$ (cf. p. 238)

α : chaîne rouge - - - -
 α^{-1} : chaîne bleue ————

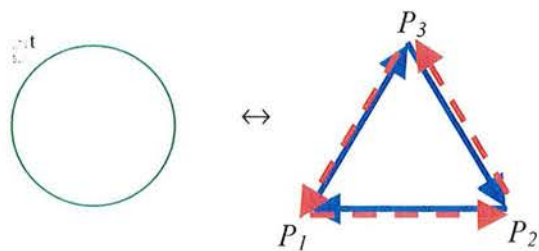


Figure C : Identification

$r < s < t$ (les arêtes figurées en rouge (pointillés) appartiennent à l'arbre maximal choisi)

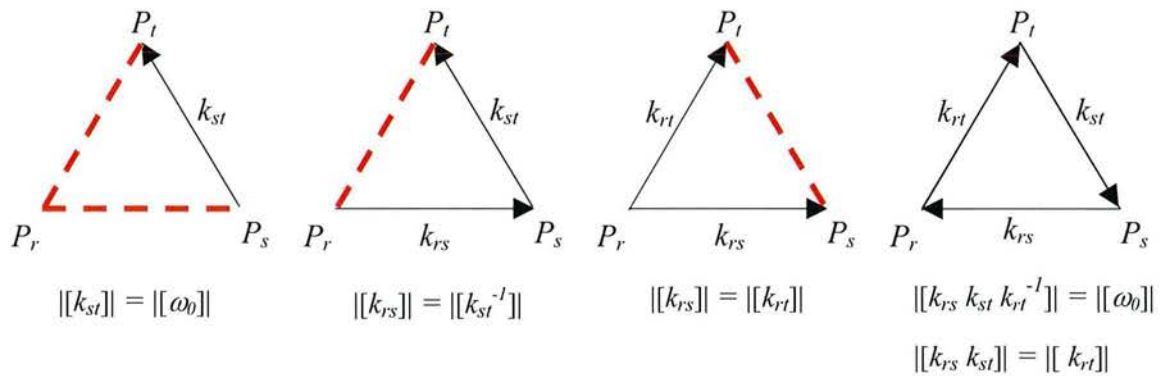


Figure D1 : détermination du groupe fondamental à l'aide d'un groupe libre

Ossature : rouge

Grphe $|K^{(l)}|$: bleu

$\pi(|K^{(l)}|)$ est isomorphe au groupe libre F_7 engendré par $[[k_{23}]]$, $[[k_{24}]]$, $[[k_{25}]]$, $[[k_{35}]]$, $[[k_{36}]]$, $[[k_{46}]]$, $[[k_{56}]]$.

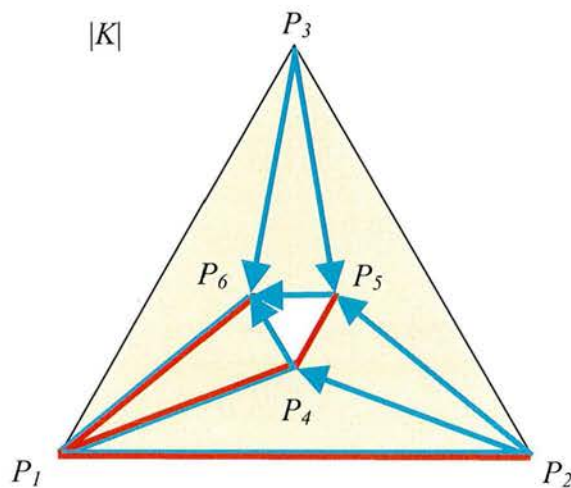
Pour la détermination de $\pi(|K|)$, on s'intéresse aux simplexes de dimension 2. Selon les schémas de C on procède aux identifications :

$$[[k_{24}]] = [[k_{46}]] = [[k_{36}]] = [[\omega_0]]$$

$$[[k_{25}]] = [[k_{24}]] = [[\omega_0]]$$

$$[[k_{23}, k_{35}]] = [[k_{25}]] = [[\omega_0]] \text{ d. h. } [[k_{23}]] = [[k_{35}^{-1}]]$$

$$[[k_{35}, k_{56}]] = [[k_{36}]] = [[\omega_0]] \text{ d. h. } [[k_{56}]] = [[k_{35}^{-1}]]$$



$\pi(|K|)$ est donc isomorphe au groupe libre F_7 engendré par $[[k_{35}]]$, c'est à dire isomorphe à \mathbb{Z} .

Références bibliographiques

- [AHU 94] J.M. Ahuactzin *Le Fil d'Ariane. Une méthode de planification générale. Application à la planification de trajectoires.* Thèse de Doctorat de l'Institut National Polytechnique de Grenoble , 1994
- [AHU 99] Ahuactzin J.M. & Gupta K. *A Motion Planning Based Global Approach for Inverse Kinematics of Redundant Robots : The Kinematic Roadmap.* IEEE Tr. on Robotics and Automation, 15(4). pp 653-669 1999
- [ALA 95] Alami R., Laumond J.P. & Siméon T. *Two manipulation planning algorithms.* In Algorithmic Foundations of Robotics (WAFR94), K. Goldberg et al (Eds), AK Peters, 1995
- [AMA 98] Amato N.M., Burchan Bayazit O., Dale L. K., Jones C., Vallejo D. *OBPRM: An Obstacle-Based PRM for 3D Workspaces.* In Robotics: The Algorithmic Perspective, P. Agarwal and all (Eds), AK Peters. pp 630 -637 1998
- [ANS 00] Anshelevich E., Owens S., Lamiroux F. & Kavraki L. *Deformables Volumes in Path Planning Applications.* In IEEE International Conference on Robotics and Automation, San Francisco. 2000
- [BAR 90] Baraff D. *Curved Surfaces and Coherence of Non-Penetrating Rigid Body Simulation.* 1990 ACM Computer Graphics, 24(4):19-28, 1990
- [BAR91] Barraquand J. & Latombe J. C. *Robot Motion Planning : A Distributed Representation Approach.* The International Journal of Robotics Research 10(6) pp. 628-649 1991
- [BAS 94] Basu S., Pollack R. & Roy M.-F.. *On the combinatorial and algebraic complexity of quantifier elimination.* In Proceedings of the foundations of Computer Science. pp. 632-641 1994
- [BER 99] van den Bergen G. *A Fast and Robust {GJK} Implementation for Collision Detection of Convex Objects* Journal of Graphics Tools: JGT 4(2) pp 7-25 1999
- [BOB85] Bobrow J., Dubowsky S. & al. *Time-Optimal Control of Robot Manipulators* International Journal of Robotics Research 4(3) 1985

- [BOH 00] Bohlin R. & Kavraki L.E. *Path Planning Using Lazy PRM*. IEEE Int. Conf. On Robotics and Automation., 2000
- [BOO 99] Boor V., Overmars M.H. & van der Stappen A.F. *The Gaussian Sampling Strategy for Probabilistic Roadmap Planner*. In IEEE International Conference on Robotics and Automation. pp 1018-1023 1999
- [BRO99] Brock O. & Khatib O. *Elastic Strips: A Framework for Integrated Planning and Execution* In International. Symposium on Experimental Robotics 1999
- [CAM 90] Cameron S. *Collision detection by four-dimensional intersection testing*. IEEE Transactions on Robotics & Automation 1990;6(3):291 pp. 291-302 1990
- [CAM 97] Cameron S. *Enhancing GJK: Computing minimum and penetration distances between convex polyhedra*. In IEEE International Conference on Robotics and Automation., pp 3112-3117 1997
- [CHA 98] Chablat, D. *Domaines d'unicité et parcourabilité pour les manipulateurs pleinement parallèles*. Thèse de Doctorat de l'Université et de l'Ecole Centrale de Nantes. 1998
- [COH 95] Cohen J., Lin M., Manocha D. & Ponamgi M. K., *I-COLLIDE: an interactive and exact collision detection system for large-scale environments*. Proc. ACM Interactive 3D Graphics Conf., pp. 189-196. 1995
- [COL 75] Collins G. *Quantifier elimination for real closed Fields by cylindrical algebraic decomposition*. 2nd Conf. On Automata Theory and Formal Languages, Lectures Notes in Computer Science V33, Springer Verlag. pp134-183 1975
- [COR 02] Cortés J., Siméon T. & Laumond J.P. A Random Loop Generator for planning the motions of closed kinematic chains with PRM methods. In IEEE Int. Conf. On Robotics and Automation. 2002
- [DEY 99] Dey T. K., Edelsbrunner H., & Guha S. *Computational Topology*. In Advances in Discrete and Computational Geometry (Contemporary mathematics 223), ed. B. Chazelle, J. E. Goodman, and R. Pollack, American Mathematical Society. pp 109-143 1999
- [FAU95] Faugère J. & Lazard D. *The combinatorial classes of parallel manipulators*. Mechanism and Machine Theory 30 pp. 765--776. 1995
- [FAV88] Faverjon B. & Tournassoud P. *A local based Approach for path planning manipulators with a high number of degrees of freedom*. In IEEE International Conference on Robotics and Automation., 1987
- [FER 96] Ferbach P. *Contribution à la planification de trajectoire*. Thèse de l'Ecole Polytechnique 1996
- [FER 97] Ferbach P. & Barraquand. J. *A method of Progressive Constraints for Manipulation Planning*. In IEEE Tr. Robotics. and Automation 13(4). pp 473-485 1997

- [FOI 93] Foisy A & Hayward V. *A safe swept volume method for collision detection*. The Sixth International Symposium of Robotics Research, Pittsburgh pp. 61 1993
- [GEE 00] Van Geem, C. *Collision detection : new improved techniques motion for logistics*. Deliverable ESPRIT LTR 28226. 2000
- [GEE 99] Van Geem C. *Literature Survey on Collision Detection and Evaluation of Available Software for Path Planning Purposes*. Deliverable ESPRIT LTR 28226. 1999
- [GIL88] Gilbert E. G., Johnson D. W. & Keerthi.S. S. *A fast procedure for computing the distance between objects in three-dimensional space*. IEEE J. Robotics and Automation, vol RA-4 . pp 193-203 1988
- [GOT 96] Gottschalk S., Lin M. & Manocha D. *OBB-Tree: A Hierarchical Structure for Rapid Interference. Detection*. Proc. ACM SIGGRAPH'96. pp. 171-180 1996
- [GRI 88] Grigor'ev D. *The complexity of deciding Tarski Algebra*. Journal of symbolic Computation, 5. pp 65-108 1988
- [GRI 97] Grigoriev D. & Slissenko A.. *Computing minimum-link path in a homotopy class amidst semi-algebraic obstacles in the plane*. In T. Mora and H. Mattson editors, In Proc. of the Int. Symp. AAEECC'12,. Springer Verlag, Lect. Notes in Comput. Sci, vol. 1255. pp 114-129 1997
- [HES 00] Hestenes D., Li H. & Rockwood A. *New Algebraic tools for classical geometry*. In Sommer G. editor, Geometric Computing with Clifford Algebra. Springer Verlag. 2000
- [HSU 97] Hsu D., Latombe J.C. & Motwani R. *Path Planning in Expansive Configuration Spaces*. Proc. IEEE Int. Conf. Robotics and Automation. pp. 2719-2726 1997
- [HSU98] Hsu D., Kavraki L., Latombe J.C., Motwani R. & Sorkin S. *On Finding Narrow Passages with Probabilistic Roadmap Planners*. In P.K. Agarwal et al. (eds.), Robotics: The Algorithmic Perspective, A K Peters, Natick, MA, pp. 151-153. 1998
- [HUB 95] Hubbard P.M. *Collision Detection for Interactive Graphics Applications*. PhD Thesis, Computer Science Dept., Brown University, Providence, RI, 1995
- [HUD 97] Hudson T., Lin M., Cohen J., Gottschalk S. & Manocha D. *V-COLLIDE: Accelerated Collision Detection for VRML*. in Proc. Of VRML'97. 1997
- [JAO97] Jaouni H. *Etude d'une métrique non-holonyme et applications aux robots mobiles* Thèse de Doctorat de l'Université Paul Sabatier, Toulouse 1997
- [JIM 01] Jimenez P., Thomas F. & Torras C. *3D collision detection: a survey*. Computer & Graphics 25 pp. 269-285 2001
- [JIX 01] Ji X. & Xiao J. *Planning Motion Compliant to Complex Contact States*. In IEEE International Conference on Robotics and Automation 2001

- [KAV 96] Kavraki L. E., Svestka P. & Latombe J-C. *Probabilistic roadmaps for path planning in high dimensional configuration spaces*. In IEEE Trans. Robotics and Automation 12(4). pp 566-580 1996
- [KLO 98] Klosowski J.T., Held M., Mitchell J.S.B., Sowizral H. & Zikan K. *Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs*. In IEEE Transactions on Visualization and Computer Graphics, March 1998 Volume 4, Number 1 1998
- [KUF 00] Kuffner J. & Lavelle S. *RRT-Connect: an efficient approach to single-query path planning*. In IEEE Int. Conf. On Robotics and Automation. 2000
- [LAF 91] Lafferriere G. & Sussmann H. *Motion Planning for Controllable Systems Without Drift*. IEEE Int. Conf. On Robotics and Automation pp. 1148-1153 1991
- [LAF 96] Lafontaine J. *Introduction aux variétés différentielles*. Presses Universitaires de Grenoble. 1996
- [LAR 00] Larsen E., Gottschalk S., Lin M. C. & Manocha D. *Fast Distance Queries with Rectangular Swept Sphere Volumes*. In IEEE International Conference on Robotics and Automation. 2000
- [LAT 91] Latombe J-C. *Robot Motion Planning*. Ed. Kluwer Academic Publisher. 1991
- [LAU 98] Laumond J-P., Sekhavat S. & Lamiroux F. *Guidelines to Non-holonomic Motion Planning*. pp 1-53 1998
- [LAU94] Laumond J-P., Jacobs P., Taix M. & Murray R. *A Motion Planner for Non Nonholonomic Mobile Robots* IEEE Transactions on Robotics and Automation 10(5) 1994
- [LAV01] LaValle S.M. & Kuffner J.J. *Randomized Kinodynamic Planning*. Int. J. of Robotics Research, 20(5) 278-300 2001
- [LIN 91] M. Lin & J. Canny. *A fast algorithm for incremental distance calculation*. In IEEE Conference on Robotics and Automation, pages 1008-1014, 1991
- [LIN 98] Lin M. & Gottschalk S. *Collision Detection between Geometric Models: A Survey*. Appeared in the Proceedings of IMA Conference on Mathematics of Surfaces 1998. 1998
- [MAR 94] .Martinetz T. & Schulten K. *Topology representing networks*. Neural Networks 7. pp 507-522 1994
- [MIS 97] Mishra B. *Computational real algebraic geometry*. Handbook of Discrete and Computational Geometry. 1997
- [MUR 94] Murray R.M., Li Z.X. & Sastry S.S. *A Mathematical Introduction to Robotic Manipulation*. CRC Press. 1994
- [NOT 99] Notte D. & Drouin A. *Projet MoLog Etude des besoins fonctionnels en maintenance - Scénarios et Contraintes*. Rapport EDF/DRD/IMA/Groupe CAO 1999

- [ODU 85] O'Dunlaing & Yap C. *A retraction method for planning the motion of a disk*. J. of Algorithms 6 pp. 104-111 1985
- [OVE 95] Overmars M. & Švestka P. *A Probabilistic learning approach to motion planning*. In Algorithmic Foundations of Robotics, K. Goldberg et al (Eds), AK Peters. 1995
- [PAU 01] Paulin F. *Topologie Algébrique élémentaire*. Cours de Magistère Ecole Normale Supérieure 2001 2001
- [RED 02] Redon S., Kheddar A. & Coquillart S. *Fast Continuous Collision Detection between Rigid Bodies*. Eurographics 2002 / G. Drettakis and H.-P. Seidel (Guest Editors) Vol.21 Number3 2002
- [REIN 97] Reinhardt F. & Soeder H. *Atlas des mathématiques*. Librairie générale Française 1997
- [SAH 02] Sahbani A., Cortés J. & Siméon T. *A probabilistic algorithm for manipulation planning under continuous grasps and placements*. In IEEE Int. Conf. On Intelligent Robots and Systems. 2002
- [SCH 00a] Schmitzberger E. & Bouchet J-L. *Testbed Definition for the MoLog Software Evaluation*. Deliverable ESPRIT LTR 28226. 2000
- [SCH 00b] Schmitzberger E., Ouzeau F. & Bouchet J-L. *MoLog Evaluation of the developed Software*. Deliverable ESPRIT LTR 28226. 2000
- [SCH 01a] Schmitzberger E., Franju A. & Bouchet J-L. *Second Evaluation of the developed Software*. Deliverable ESPRIT LTR 28226. 2001
- [SCH 01b] Schmitzberger E. & Bouchet J-L. *A Test case of Computer Aided Motion Planning for nuclear maintenance operation*. In ASME International Conference On Nuclear Engineering 2001 2001
- [SCH 83a] Schwartz J. T. & Sharir M. *On the Piano Movers' Problem : I. The Case of a Two Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers*. Communications on Pure and Applied Mathematics 36. Pp 345-398 1983
- [SCH 83b] Schwartz J. T. & Sharir M. *On the Piano Movers' Problem : II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds*. Advances in Applied Mathematics. Pp 298-351 1983
- [SHO 85] Shoemake K. *Animating rotation with quaternion curves*. In Computer Graphics , 19(3), 1985
- [SIM 00] Siméon T., Laumond J-P. & Nissoux C. *Visibility-based probabilistic roadmap for motion planning*. In Advanced Robotics Journal 14(6). 2000
- [SIM 01] Siméon T., Laumond J-P., Van Geem C. & Cortes J. *Computer Aided Motion: Move3D within MOLOG*. In IEEE International Conference on Robotics and Automation. 2001

- [SIM 02] Siméon T., Cortès J., Sahbani A. & Laumond J-P. *A Manipulation Planner for Pick and Place Operations under Continuous Grasps and Placements*. In IEEE International Conference on Robotics and Automation. 2002
- [SVE 98] Švestka P. & Overmars M. *Probabilistic path planning*. In Laumond Ed. Lecture Notes in Control and Information Sciences 229, Springer, www.laas.fr/~jpl. pp 255-304 1998
- [TAR 51] Tarski A. *A decision method for elementary algebra and geometry*. University of California Press. 1951
- [TSA 95] Tsakiris D.P. *Motion Control and Planning for Nonholonomic Kinematic Chains*. Thesis Report, Univ. of Maryland. 1995
- [WEN 99] Wenger P. *CAO de site robotisés*. Journées Nationales de la recherche en robotique. 1999
- [WIL 99] Wilmarth S., Amato N. & Stiller P. *MAPRM: A Probabilistic Roadmap Planner with Sampling on the Medial Axis of the Free Space*. In IEEE International Conference on Robotics and Automation. pp. 1024-1031 1999
- [ZEF 97] Zefran M. & Kumar V. *A variational calculus framework for motion planning*. In International Conference on Advanced Robotics, Monterey, CA. 1997

**AUTORISATION DE SOUTENANCE DE THESE
DU DOCTORAT DE L'INSTITUT NATIONAL
POLYTECHNIQUE DE LORRAINE**

o0o

VU LES RAPPORTS ETABLIS PAR :

Monsieur Thierry SIMEON, Chargé de Recherche, LAAS-CNRS, Toulouse

Monsieur Christian LAUGIER, Directeur de Recherche, INRIA, ZIRST, Montbonnot Saint-Martin

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur SCHMITZBERGER Erwin

à soutenir devant un jury de l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE,
une thèse intitulée :

**« Planification globale de trajectoires en milieux industriels : application à la
maintenance et au démantèlement de centrales nucléaires »**

NANCY BRABOIS
2, AVENUE DE LA
FORET-DE-HAYE
BOITE POSTALE 3
F - 5 4 5 0 1
VANCEUVRE CEDEX

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : **« Automatique et traitement du signal »**

Fait à Vandoeuvre, le 29 octobre 2003

Le Président de l'I.N.P.L.,

L. SCHUFFENECKER



le Président
Vice-Président,

J.-L. COJJARD

Planification Globale de Trajectoires en Milieux Industriels : Application à la Maintenance et au Démantèlement de centrales nucléaires

Le problème de planification de trajectoires sans collisions a été posé depuis plusieurs décennies en robotique. Le pré-calcul de réseaux probabilistes dans l'espace des configurations est une méthode efficace pour générer des mouvements sans collision reliant une paire quelconque de configurations d'un robot. Nous avons contribué, au sein d'EDF R&D et du projet européen MoLog, à adapter et à intégrer ces méthodes dans le contexte de la maintenance et du démantèlement de centrales nucléaires. Ce type d'application requiert des déplacements d'objets, d'engins ou de robots dans des environnements très encombrés. Nous proposons de plus une méthode visant à enrichir les réseaux pour qu'il puisse générer tous les chemins non homotopes reliant une paire quelconque de configurations. De cette manière, les contraintes géométriques, cinématiques voire dynamiques spécifiées après le pré-calcul, peuvent s'appliquer à une seule classe d'homotopie choisie par l'utilisateur. Ceci conduit à un gain de temps et d'interactivité entre les ingénieurs et les outils de planification.

Mots clefs : Planification de trajectoire, Méthode à réseaux probabilistes, Calcul d'homotopie, Prototypage virtuel, Maintenance nucléaire, Démantèlement nucléaire.

Global Motion Planning in Industrial Environments: Application for the Maintenance and the Dismantling of Nuclear Power Plants

The problem of motion planning has been addressed for several decades in robotics. Pre-computed probabilistic roadmaps have been recognized as an efficient method to generate rapid collision free motions between any pair of initial and final configurations of the robot. Inside EDF R&D and the European MoLog project, we contribute to adapt and integrate these methods in the context of the maintenance and the dismantling of nuclear power plants. Such applications require the motion of objects, devices or even robots within cluttered environments. Furthermore, we propose a way to enrich the pre-computed roadmaps so that they can generate all the non homotopic paths for any pair of configurations. Hence any further geometrical, kinematical or dynamical constraints specified after the Roadmap pre-computation, can be applied on a single homotopy class which is selected by the user. This leads to a gain of time and interactivity between engineers and motion planning tools.

Keywords : Motion Planning, Probabilistic Roadmaps, Homotopy Computation, Virtual prototyping, Nuclear maintenance, Nuclear dismantling.