

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Automatique et robotique

Présentée et soutenue par :

M. SIMON CAILHOL

le lundi 22 juin 2015

Titre :

PLANIFICATION INTERACTIVE DE TRAJECTOIRE EN REALITE VIRTUELLE SUR LA BASE DE DONNEES GEOMETRIQUES, TOPOLOGIQUES ET SEMANTIQUES

Ecole doctorale :

Systèmes (Systèmes)

Unité de recherche :

Laboratoire de Génie de Productions de l'ENIT (E.N.I.T-L.G.P.)

Directeur(s) de Thèse :

M. JEAN-YVES FOURQUET

M. PHILIPPE FILLATREAU

Rapporteurs :

Mme VERONIQUE PERDEREAU, INST. DES SYST. INTELLIGENTS& ROBOTIQUES

M. XAVIER FISCHER, ESTIA BIDART

Membre(s) du jury :

M. MICHEL TAIX, LAAS TOULOUSE, Président

M. FOUAD BENNIS, ECOLE CENTRALE DE NANTES, Membre

M. JEAN-YVES FOURQUET, ECOLE NATIONALE D'INGENIEUR DE TARBES, Membre

M. PHILIPPE FILLATREAU, ECOLE NATIONALE D'INGENIEUR DE TARBES, Membre

Remerciements

Je tiens tout d'abord à remercier mes directeurs de thèse, Jean-Yves Fourquet et Philippe Fillatreau. Je les remercie tous deux d'avoir trouvé dans leurs emplois du temps, durant ces années, des espaces à me consacrer. Leurs qualités scientifiques et humaines ont rendu chacun de ces instants riches et formateurs, aussi bien professionnellement que personnellement. La thèse est une aventure scientifique certes, mais aussi humaine. Je ne pouvais souhaiter meilleure compagnie pour vivre la mienne.

J'adresse aussi mes remerciements aux membres de mon jury de thèse pour l'attention qu'ils ont porté à mes travaux : M. Michel Taïx pour avoir présidé ce jury, Mme Véronique Perdereau et M. Xavier Fischer pour avoir accepté d'évaluer mon mémoire, M. Fouad Bennis pour avoir assisté à ma soutenance en tant qu'examinateur.

Je tiens aussi à remercier Yingshen Zhao pour sa participation à mes travaux de thèse. Sans sa contribution pratique qui m'a permis de me consacrer plus grandement aux aspects théoriques et scientifiques, cette thèse serait toute autre.

Je remercie les directeurs et responsables qui se sont succédés, au cours de ma thèse, à la tête des différents organes de l'École Nationale d'Ingénieurs de Tarbes (ENIT) auxquels j'ai appartenu. Je remercie M. Talal Masri, directeur de l'ENIT ; MM. Daniel Noyes et Jean-Yves Fourquet, directeurs du Laboratoire Génie de Production (LGP) ; M. Jean-Yves Fourquet, feu M. Philippe Charbonneau et M. Bernard Archimède, responsables de l'équipe Décision et Interaction Dynamique pour les Systèmes (DIDS). J'étends aussi ces remerciements à l'ensemble des personnels de l'ENIT qui m'a accueilli.

Je remercie particulièrement les chercheurs de l'équipe DIDS qui ont fait le choix d'associer grandement leurs doctorants à la vie de l'équipe. Ils leurs offrent ainsi la possibilité d'entrevoir la réalité du travail de chercheur après la thèse, j'ai abondamment profité de cette possibilité. Parmi ces chercheur, je remercie tout particulièrement Paul-Etienne Vidal qui m'a poussé dans la voie de la recherche et qui me permet aujourd'hui d'y demeurer davantage.

Je souhaite, dans ces remerciements, mentionner spécialement les collègues du LGP avec qui nous avons partagé la condition de "chercheur précaire" au moins un temps. Les doctorants qui m'ont accueillis à la fin de leur thèse tout d'abord, en particulier Adrien qui m'a formé à certains concepts scientifiques utilisés dans mes travaux. Une mention spéciale aussi à Julie, Malik, Éric, Thibaud, Yannick et Christian qui sont encore dans la place. Ceux avec qui nous avons vécu nos aventures de concert : Muhammad Ali, Damien, Paula,... Une petite mention spéciale pour Juan Diego avec qui nous avons partagé quelques expéditions à Toulouse. Je

remercie aussi particulièrement Vincent et Corentin, ma cellule psychologique des moments difficile. Un divan pour moi dans votre bureau aurait été à sa place. Enfin, je souhaite le plus grand succès à ceux qui arrivent à la fin de leur thèse : Karima, Floran, Victor, Corentin,... ou qui la commencent : Mathias, Yingshen, Nassime, Xu Da,...

Du point de vue personnel, je voudrait tout d'abord remercier ma famille de cœur, les Éclaireuses et Éclaireurs De France (EEDF ou éclés). Cette association porte et défend des valeurs auxquelles je souscrit. Même ma thèse n'y a pas rompu mon engagement. Je m'excuse d'ailleurs auprès de mes "collègues" bénévoles du Comité de Gestion et d'Animation de Bécours (Cégéhabé) pour la légèreté avec laquelle j'ai assumé mes fonctions de trésorier ces dernières années.

Enfin, je remercie ma famille de sang qui ne l'est pas moins de cœur. Merci à mes parents, ma sœur, mon frère et leurs compagnons. Leur soutien sans faille et leurs conseils ont constitué un réconfort permanent. Je remercie également mes grands-parents, tantes, oncles, cousins et cousines qui ont suivi avec un peu plus de distance ma progression. Mes remerciements vont pour les mêmes raisons à ma belle famille. Je remercie en particulier Emeline, Michou et Julien pour les pieds à terre lors de mes escapades toulousaines.

Finalement, mes derniers remerciements, et non les moindres, vont aux plus importantes personnes de ma vie. Mon épouse, Jennifer, qui au delà de partager ma vie et de me soutenir au quotidien, m'a donné un sujet d'émerveillement de chaque jour depuis un peu plus d'un an : Zoé. Je remercie aussi cette dernière de distribuer encore inconsciemment tout ce bonheur autour d'elle.

Pour finir, une première citation qui justifie qu'elle soit la dernière :
Par des citations on affiche son érudition, on sacrifie son originalité.
Arthur Schopenhauer

À mes grand-parents

Résumé

Pour limiter le temps et le coût de développement de nouveaux produits, l'industrie a besoin d'outils pour mener ces développements avec des prototypes virtuels. Ces prototypes virtuels permettent de tester le produit à toutes les étapes du Product Lifecycle Management (PLM).

Beaucoup d'opérations du cycle de vie du produit impliquent la manipulation par un humain des composants du produit (montage, démontage). La Réalité Virtuelle (RV) permet à des opérateurs d'exécuter ces opérations avec des prototypes virtuels.

Ce travail de recherche introduit une nouvelle architecture de planification de trajectoire permettant la collaboration d'un utilisateur de RV et d'un système de planification de trajectoire automatique. Cette architecture s'appuie sur un modèle d'environnement original comprenant des informations sémantiques, topologiques et géométriques.

Le processus de planification de trajectoire est scindé en deux phases. Une planification grossière d'abord exploitant les données sémantique et topologiques. Elle définit un chemin topologique. Une planification fine ensuite exploitant les données sémantiques et géométriques détermine une trajectoire géométrique dans le chemin topologique préalablement défini.

La collaboration entre le système de planification et l'utilisateur de RV s'articule autour de deux modes : en premier lieu, l'utilisateur est guidé sur une trajectoire pré-calculée à travers une interface haptique ; en second lieu, l'utilisateur peut quitter la solution proposée et déclencher ainsi une re-planification.

Ces deux modes d'interaction sont mis en œuvre grâce à des méthodes de partage de contrôle : tout d'abord, l'autorité du système automatique est modulée afin de fournir à l'utilisateur à la fois un guidage prégnant lorsqu'il le suit, et plus de liberté (un guidage atténué) lorsqu'il explore des chemins alternatifs. Lorsque l'utilisateur explore ces chemins alternatifs, ses intentions sont prédites et intégrées dans le processus de re-planification pour guider la planification grossière.

Ce mémoire est organisé en cinq chapitres. Le premier expose le contexte industriel ayant motivé ces travaux. Les trois chapitres suivant abordent les trois thématiques principales des travaux : la modélisation de l'environnement, la planification de trajectoire et l'interaction et le partage de contrôle. Chacun de ces chapitre intègre, pour le thème concerné, un état de l'art, la description de notre proposition et de son implémentation. Enfin le dernier chapitre présente les expérimentations menées avec le planificateur de trajectoire.

Mots clé : Réalité Virtuelle, modèle de l'environnement, planification de trajectoire, interaction, partage de contrôle, partage d'autorité, prédiction d'intention, géométrie, topologie, sémantique.

Abstract

To save time and money while designing new products, industry needs tools to design, test and validate the product using virtual prototypes. These virtual prototypes must enable to test the product at all Product Lifecycle Management (PLM) stages.

Many operations in product's lifecycle involve human manipulation of product components (product assembly, disassembly or maintenance). Cue to the increasing integration of industrial products, these manipulations are performed in cluttered environment. Virtual Reality (VR) enables real operators to perform these operations with virtual prototypes.

This research work introduces a novel path planning architecture allowing collaboration between a VR user and an automatic path planning system. This architecture is based on an original environment model including semantic, topological and geometric information.

The automatic path planning process split in two phases. First, coarse planning uses semantic and topological information. This phase defines a topological path. Then, fine planning uses semantic and geometric information to define a geometrical trajectory within the topological path defined by the coarse planning.

The collaboration between VR user and automatic path planner is made of two modes : on one hand, the user is guided along a pre-computed path through a haptic device, on the other hand, the user can go away from the proposed solution and doing it, he starts a re-planning process.

Efficiency and ergonomics of both interaction modes is improved thanks to control sharing methods. First, the authority of the automatic system is modulated to provide the user with a sensitive guidance while he follows it and to free the user (weakened guidance) when he explores possible better ways. Second, when the user explores possible better ways, his intents are predicted (thanks to geometrical data associated to topological elements) and integrated in the re-planning process to guide the coarse planning.

This thesis is divided in five chapters. The first one exposes the industrial context that motivated this work. The tree following chapter deal with the tree main topic of this study : environment modeling, path planning and interaction and control sharing. Both of these chapters include a state of the art of the concerned field, a description of our proposition and of its implementation. Finally, last chapter introduces the experimentations performed with our path planner and analyses their results.

Keywords : Virtual Reality, environment model, path planning, interaction, control sharing, authority sharing, intent prediction, geometry, topology, semantics.

Table des matières

Remerciements iii
Table Page

Table des matières xi

Table des figures xv

Liste des tableaux xix

Liste des Algorithmes xxi

Acronymes xxiv

Introduction 1

1 Motivation et contexte des travaux 5
1.1 Contexte Industriel . . . 6
1.1.1 Industrie et PLM . . . 7
1.1.2 Le prototype virtuel : une réponse aux contraintes économiques et temporelles . . . 8
1.1.3 La RV comme fil rouge du processus PLM . . . 10
1.1.4 Synthèse . . . 16
1.2 Assemblage et planification interactive de trajectoire . . . 17
1.2.1 Travaux précurseurs de planification interactive de trajectoire . . . 17
1.2.2 Enjeux, verrous . . . 19
1.2.3 Contributions . . . 20
1.3 Communications . . . 22

2 Modèle multi-niveaux de l'environnement 25
2.1 État de l'art . . . 26
2.1.1 Espaces et composantes libres d'obstacles . . . 27
2.1.2 Représentation numérique des corps rigides . . . 28
2.1.3 Modélisation de l'espace libre . . . 29
2.1.4 Synthèse . . . 35

2.2 Contribution . . . 35
2.2.1 Modélisation des Corps rigides . . . 36
2.2.2 Modélisation de l'Espace libre . . . 37
2.2.3 Synthèse du modèle multi-niveaux de l'environnement proposé . . . 40
2.3 Implémentation du modèle de l'environnement . . . 41
2.3.1 Modèle statique : classes et responsabilités . . . 42
2.3.2 Algorithmes, processus et parallélisation . . . 46
2.4 Conclusion sur la modélisation de l'environnement réalisée . . . 52

3 Planification de trajectoire en environnement multi-niveaux 55
3.1 État de l'art de la planification de trajectoire automatique . . . 56
3.1.1 Les approches . . . 57
3.1.2 Les méthodes . . . 57
3.1.3 Synthèse . . . 61
3.2 Contribution . . . 62
3.2.1 Planification grossière . . . 63
3.2.2 Planification fine . . . 67
3.2.3 Synthèse de la planification de trajectoire multi-niveaux proposée . . . 69
3.3 Implémentation de la planification de trajectoire . . . 70
3.3.1 Modèle statique : classes et responsabilités . . . 70
3.3.2 Algorithmes, processus et parallélisation . . . 72
3.4 Conclusion sur la planification de trajectoire réalisée . . . 75

4 Planification interactive et partage de contrôle 79
4.1 État de l'art de la planification interactive et du partage de contrôle . . . 80
4.1.1 Planification interactive de trajectoire . . . 81
4.1.2 Partage de contrôle . . . 85
4.2 Contribution . . . 91
4.2.1 Planification de trajectoire et interaction . . . 91
4.2.2 Partage de contrôle proposé . . . 94
4.2.3 Partage de contrôle et architecture multi-niveaux . . . 98
4.2.4 Synthèse de l'interaction pour la planification de trajectoire proposée . 100
4.3 Implémentation de l'interaction pour la planification de trajectoire . . . 102
4.3.1 Modèle statique : classes et responsabilités . . . 102
4.3.2 Algorithme, processus et parallélisation . . . 103
4.4 Conclusion sur l'interaction pour la planification de trajectoire . . . 109

5 Expérimentations et résultats 113
5.1 Plateforme de RV du LGP . . . 115
5.1.1 Interfaces comportementales . . . 116
5.1.2 Logiciels . . . 119

5.2 Applications de planification 124
5.2.1 Première application : instance 3D de l'exemple illustratif 124
5.2.2 Seconde application : jeu d'encastrement 126
5.3 Validation de la planification de trajectoire non-interactive 128
5.3.1 Stratégies de planification comparées 129
5.3.2 Critères d'évaluation 132
5.3.3 Résultats 132
5.3.4 Synthèse sur la planification de trajectoire multi-niveaux 138
5.4 Validation de la planification de trajectoire interactive 139
5.4.1 Réglages du planificateur interactif de trajectoire 139
5.4.2 Résultats 141
5.4.3 Synthèse sur la planification de trajectoire interactive 144
5.5 Conclusion sur les expérimentations menées 145

Conclusion et perspectives **147**

Bibliographie **153**

Remerciements **172**

Table des figures

Table **Page**

1.1 Cycle de développement d'un produit industriel. 8
1.2 Cycle de développement d'un produit industriel avec prototypes virtuels. 9
1.3 Simulations physiques réalisées sur prototypes virtuels (proposition laboratoire PEARL/PRIMES [Lasserre 12]). 10
1.4 Utilisation de prototypes virtuels pour la simulation de processus. 11
1.5 Les interfaces comportementales pour l'immersion et l'interaction (inspiré de [Fuchs 06a]). 12

2.1 Environnement 2D. 30
2.2 Décompositions en cellules d'environnement 2D. 31
2.3 *Rodmaps* d'environnement 2D. 33
2.4 Exemples de Corps rigides d'un environnement 2D. 37
2.5 Lieux du modèle sémantique. 39
2.6 Représentation des Lieux. 40
2.7 Représentation des Lieux. 41
2.8 Représentation de l'Espace libre. 42
2.9 Modèle de domaine UML de l'environnment. 43
2.10 Calcul des distances de la matrice D (pour le Lieu L_4). 44
2.11 Diagramme de classes UML de l'Information sémantique. 46
2.12 Processus de décomposition octree. 47
2.13 Espace libre considéré. 49
2.14 Cellules géométriques libres des différents niveaux de l'octree (en gris). . . . 50
2.15 Création des premiers Lieux (niveau 3). 50
2.16 Extension des Lieux aux Cellules géométriques libres du niveau 4. 51
2.17 Extension des Lieux aux Cellules géométriques libres du niveau 5. 51
2.18 Création de Lieux au niveau 4. 52
2.19 Création de Lieux au niveau 5. 53

3.1 Processus de planification grossière 65
3.2 Processus de planification fine 69
3.3 Modèle de domaine UML du planificateur de trajectoire multi-niveaux. 71

3.4 Diagramme de classes UML des `Stratégies de planification` 72
3.5 Modèle de domaine UML du planificateur de trajectoire et de l'environnement. 73
3.6 Diagramme de séquence de la planification grossière 74
3.7 Diagramme de séquence de la planification fine 76

4.1 *Planificateur probabiliste avec approche globale* de Ladevèze [Ladevèze 10a] . 83
4.2 RRT interactif proposé par Flavigné [Flavigné 10a] 83
4.3 Charge de calcul vs charge cognitive pour les différents planificateurs interactifs proposé par [Ladevèze 10a]. 84
4.4 Cadre général du partage de contrôle. 86
4.5 Partage d'autorité pour une tâche donnée. 87
4.6 Partage de contrôle en téléopération avec prédiction d'intention. 89
4.7 Prédiction d'intention proposée par Dragan [Dragan 13] 90
4.8 Guidage sur une trajectoire pré-calculée 92
4.9 Partage d'autorité dynamique 95
4.10 Choix du point représentatif de la frontière. 97
4.11 Allocation dynamique de l'autorité 99
4.12 Modèle de domaine UML du planificateur interactif de trajectoire sur environnement multi-niveaux. 104

5.1 Système de visualisation 3D. 116
5.2 Système de visualisation 3D. 117
5.3 Bras haptique Virtuose 6D35-45. 118
5.4 Environnement de développement Virtools. 120
5.5 Script comportemental Virtools. 121
5.6 Exemple de blocs comportementaux fournis avec IPP. 121
5.7 BB pour la planification interactive de trajectoire. 122
5.8 Script de planification interactive de trajectoire en RV. 123
5.9 Environnement d'expérimentation inspiré de l'exemple illustratif 2D. 125
5.10 Projection 2D des six déclinaisons de l'application inspirée de l'exemple illustratif. 126
5.11 Environnement inspiré des jeux d'encastrement. 127
5.12 RRT bidirectionnel en 2D. 130
5.13 `Chemins topologiques` retenus par la stratégie GTS (exemple illustratif). . . . 133
5.14 Performances des différentes stratégies de planification (exemple illustratif). . . 134
5.15 Distribution du nombre de configurations aléatoires utilisées par les stratégies GT et GTS (exemple illustratif). 135
5.16 Performances des différentes stratégies de planification (jeu d'encastrement). . 137
5.17 Trajectoire planifiée sans replanification dans la déclinaison 1a. 141
5.18 Trajectoire planifiée sans replanification pour le rond. 142
5.19 Trajectoire planifiée avec replanification dans la déclinaison 2a. 143

5.20 Trajectoire planifiée avec replanification dans la déclinaison 1a. 143
5.21 Trajectoire planifiée pour l'étoile. 144
5.22 Trajectoire planifiée avec replanification pour le rond. 145

Liste des tableaux

Table		Page
2.1	Information sémantique des différents Lieux.	46
3.1	Principales approches de planification de trajectoire	61
3.2	Exemple de choix de Stratégie de planification.	68
4.1	Moyens d'interaction pour les différents niveaux.	101
5.1	Information sémantique des Lieux de l'instance 3D de l'exemple illustratif.	126
5.2	Information sémantique attachée aux Lieux de l'application inspirée des jeux d'encastrement.	128
5.3	Compatibilité objet/trou.	128
5.4	Qualité des stratégies de planification de trajectoire (exemple illustratif).	133
5.5	Qualité des stratégies de planification de trajectoire (jeu d'encastrement).	136

Liste des Algorithmes

Table		Page
2.1	Construction du modèle multi-niveaux de l'environnement	47
2.2	Décomposition(*cellule*, *niveau*)	48
2.3	Définition des `Lieux`	49
3.1	Planification grossière	64
4.1	Processus de re-planification	106
4.2	Prédiction des intentions de l'opérateur	107
4.3	Contrôle de l'exécution de la tâche	107
4.4	Application de planification interactive de trajectoire	108

Acronymes

2D 2 dimensions.

3D 3 dimensions.

6D 6 dimensions.

ABS *Antiblockiersystem* (allemand).

ACP Analyse en composantes principales.

B-Rep *Boundary Representation*.

B-spline *Basic spline*.

BB *Building Blocks*.

BG *Behaviour Graphs*.

CAO Conception Assistée par Ordinateur.

CSG *Constructive Solid Geometry*.

ddl degrés de liberté.

dll *dynamic link library*.

ESP *Electronic Stability Program*.

FCL *Flexible Collision Library*.

IPP *Interactive Physic Pack*.

IPSI *Interactive Physics Simulation Interface*.

IR Infrarouge.

IRRT *Interactive Rapidly-exploring Random Tree*.

LAAS Laboratoire d'Analyse et d'Architecture des Systèmes.

LGP Laboratoire Génie de Production.

nD n dimensions.

NURBS *Non Uniform Rational B-Spline*.

OpenMP Open Multi Processing.

PEARL *Power Electronics Associated Research Laboratory*.

PLM *Product Lifecycle Management*.

PRIMES *Platform for Research on power electronic Integration and Management of Energy and Storage devices*.

PRM *Probabilistic RoadMap*.

RA Réalité Augmentée.

RDT *Rapidly-exploring Dense Tree*.

RRT *Rapidly-exploring Random Tree*.

RV Réalité Virtuelle.

SDK *Software Development Kit*.

TCP/IP *Transmission Control Protocl/Internet Protocol*.

UML *Unified Modeling Language*.

VRAD *Virtual Reality Aided Design*.

VRPN *Virtual Reality Peripheral Network*.

Introduction

Les délais et les coûts du processus de développement de nouveaux produits par les industriels sont de plus en plus contraints. Ce contexte fait apparaitre un besoin, pour les industriels, d'outils permettant d'accélérer toutes les étapes du *Product Lifecycle Management* (PLM) au cours du développement de nouveaux produits tout en réduisant leurs coûts. L'utilisation généralisée de maquettes numériques à travers les outils de Conception Assistée par Ordinateur (CAO) a permis un grand progrès dans cette direction. Ces maquettes numériques sont de plus utilisées pour subir un ensemble de tests numériques permettant une pré-validation des développements réalisés. On parle alors de prototype virtuel.

Ces prototypes virtuels ont en premier lieu été utilisés pour mener des tests non-interactifs. Dans ce cas, le test vise à vérifier le comportement du produit développé vis à vis d'une sollicitation (p.ex. simulation de comportements multi-physiques) ou lors d'opérations industrielles (p. ex. simulation du processus de fabrication).

Les industriels sont aussi demandeurs d'outils de tests interactifs permettant de vérifier le comportement du produit à l'aide de prototypes virtuels au cours de processus faisant intervenir un humain. La Réalité Virtuelle (RV), en offrant la possibilité à un humain d'interagir avec un prototype virtuel et de s'y immerger, semble être en mesure de répondre à ce besoin.

Nous nous intéressons dans ce mémoire à la simulation des opérations de montage, de démontage ou de maintenance de produits industriels. Ces opérations nécessitent de manipuler des composants de systèmes dans des environnements encombrés (le système lui-même). Ces manipulations s'effectuent donc avec de fortes contraintes géométriques, contraintes d'autant plus importantes que le système à assembler présente une forte intégration.

Pour simuler de telles opérations, il existe comme nous l'avons évoqué précédemment des outils de simulation non-interactifs et interactifs :

outils de simulation non-interactifs : ces outils mettent en œuvre des techniques de planification automatique de trajectoire issues de la robotique pour définir des trajectoires de montage/démontage ;

outils de simulation interactifs : ces outils sont généralement des applications de RV dans lesquelles un opérateur humain peut manipuler les composants d'un système virtuel pour le monter, le démonter.

Les travaux présentés dans ce mémoire sont motivés par l'idée que l'utilisation conjointe des

techniques de planification automatique de trajectoire et de RV permet d'assister un opérateur dans une application immersive de simulation d'assemblage ou de démontage.

Cette idée est toutefois difficile à mettre en œuvre du fait de la différence des objectifs adressés par chacun de ces domaines. L'immersion et l'interaction représentent les objectifs principaux en RV. Elles impliquent une maîtrise des temps de calcul des différents processus exécutés. En planification automatique de trajectoire, l'objectif prépondérant porte davantage sur la définition d'une trajectoire sans collision permettant de garantir l'intégrité de l'environnement et de l'objet pour lequel la trajectoire est recherchée.

Les techniques de planification automatique de trajectoire exploitent essentiellement des données géométriques. La mise en œuvre de ces techniques dans une application immersive et interactive nécessite de les adapter :

- la précision des modèles, ainsi que les techniques (souvent probabilistes) employées impliquent une probabilité d'échec de la planification de trajectoire importante pour des environnements fortement contraints,
- les modèles manipulés (communément des maillages polyédriques des objets) sont lourds et impliquent des traitements lourds rendant la définition d'une trajectoire sans collision longue,
- les modalités d'interaction d'un humain avec les techniques de planification automatique de trajectoire restent à développer puisque ces dernières sont généralement utilisées hors-ligne,
- les données géométriques utilisées restent pauvres pour interpréter les actions d'un humain en interaction qui évolue, lui, dans un modèle plus structuré comportant des informations symboliques de plus haut niveau.

Les travaux présentés dans ce mémoire se sont donc attachés à :

- définir un modèle de l'environnement original permettant de mettre à disposition des informations de différents niveaux d'abstraction (topologiques et sémantiques) en plus des informations géométriques classiquement utilisées ;
- définir une nouvelle technique de planification de trajectoire exploitant l'ensemble des informations du modèle de l'environnement pour accélérer le processus tout en augmentant la pertinence des résultats obtenus. La mise en œuvre de cette technique dans une simulation immersive et interactive nécessite d'en contrôler le temps d'exécution ;
- définir les modalités d'interaction entre le planificateur de trajectoire et un opérateur humain exploitant elles aussi l'ensemble des informations de l'environnement afin d'interpréter au mieux l'action de l'opérateur et ainsi de l'intégrer de manière efficace dans le processus de planification de trajectoire.

Chapitre 1

Motivation et contexte des travaux

Table **Page**

1.1 Contexte Industriel 6

1.1.1 Industrie et PLM 7

1.1.2 Le prototype virtuel : une réponse aux contraintes économiques et temporelles . 8

1.1.3 La RV comme fil rouge du processus PLM 10

1.1.3.1 Définition de la RV 11

1.1.3.2 RV, industrie et PLM 12

1.1.4 Synthèse 16

1.2 Assemblage et planification interactive de trajectoire 17

1.2.1 Travaux précurseurs de planification interactive de trajectoire 17

1.2.2 Enjeux, verrous 19

1.2.3 Contributions 20

1.3 Communications 22

Le contexte économique impose une pression de plus en plus forte sur les industriels pour réduire le coût et le temps de développement de nouveaux produits. Cette pression s'exerce alors que l'exigence de qualité croît et que l'ensemble des phases du cycle de vie des produits doit être considéré dès la conception (fabrication, utilisation, maintenance, démantèlement, retraitement/recyclage,...).

Pour faire face à ces contraintes, les industriels mettent en œuvre des méthodologies telles que le PLM pour optimiser la conception de leurs produits. Ces outils sont de plus couplés à une utilisation toujours plus étendue des maquettes numériques ou "prototypes virtuels", qui, à travers les tests qu'on leur fait subir, permettent de diminuer le recours aux prototypes physiques dans l'ensemble des phases du développement du produit (conception, implémentation, test, évaluation,...). Les outils de RV permettent d'accroitre les capacités de manipulation du prototype virtuel par l'apport de périphériques d'interaction et la possibilité d'immerger un opérateur

humain dans la description numérique du produit. Ils permettent d'envisager une nouvelle réduction du recours aux prototypes physiques.

En plus de la pression économique croissante sur l'industrie, les produits industriels sont de plus en plus complexes et de plus en plus intégrés. Cette complexité et cette intégration croissante posent un problème particulier pour l'assemblage, le démontage et donc la maintenance des systèmes industriels. En effet, ces tâches industrielles consistent à trouver une trajectoire pour les composants au sein du système (pour le démontage et l'assemblage), la complexité et l'intégration importantes des systèmes augmentent les contraintes géométriques lors de ces manipulations des composants.

Cette problématique de définition de trajectoire dans un environnement encombré est depuis longtemps étudiée dans le cadre de la recherche en robotique. Les techniques développées dans ce cadre ont été intégrées dans les logiciels de CAO pour définir des trajectoires d'assemblage des composants des systèmes conçus à l'aide de ces logiciels.

Les travaux présentés dans ce mémoire consistent à définir et mettre en œuvre de nouvelles méthodes de génération de trajectoire en environnement encombré. Ces méthodes tirent parti des possibilités d'interaction qu'offre la RV d'une part, et réutilisent les méthodes de planification automatique d'autre part. Ces nouvelles méthodes constituent une brique nécessaire au développement ultérieur de simulations interactives d'assemblage et de démontage.

Ce chapitre décrit le contexte dans lequel s'inscrivent ces travaux. Tout d'abord le contexte industriel y est précisé (section 1.1). Les apports du PLM au processus de conception de produits industriels sont introduits (section 1.1.1) avant de détailler l'utilisation des prototypes virtuels dans ce processus que ce soit de manière conventionnelle (section 1.1.2) ou interactive en RV (section 1.1.3). Ensuite, la problématique centrale de ce mémoire est présentée (section 1.2), tout d'abord en introduisant quelques travaux précurseurs de planification de trajectoire interactive (section 1.2.1) puis en détaillant les problématiques liées à la mise en œuvre conjointe de stratégies d'interaction en RV et de techniques de planification automatique de trajectoire (section 1.2.2). Enfin, les principales idées structurant les travaux défendus dans cette thèse seront décrites (section 1.2.3).

1.1 Contexte Industriel

Le contexte économique, écologique et normatif actuel implique, pour les entreprises, un processus de développement de nouveaux produits de plus en plus contraint. En effet, le temps et les coûts de ce processus se réduisent, tandis que les contraintes sanitaires, environnementales, et de qualité auxquelles il est soumis augmentent. Dans ce contexte, dès 2003, la conception de nouveaux produits apparait comme un champ particulièrement approprié pour appliquer les idées émergentes des technologies de l'information [Nambisan 03]. En effet, la gestion des processus, la gestion de projets, la gestion de l'information et des connaissances comme la communication et la collaboration des différents acteurs impliqués dans la conception du produit sont autant de processus que les technologies de l'information permettent de consolider.

[Nambisan 03] pose ainsi les bases des applications de PLM qui visent à considérer l'ensemble du cycle de vie du produit dès sa conception. Au delà d'optimiser l'utilisation du produit, cette approche permet d'en faciliter aussi la fabrication, la maintenance et le traitement en fin de vie.

Plus récemment, le bilan de l'implémentation des applications de PLM dans l'industrie aérospatiale a été dressé [Cantamessa 12]. Après avoir noté que le déploiement de tels outils doit être fait de manière efficace (la stratégie de déploiement doit être adaptée à la structure de l'entreprise), Cantamesa constate que le déploiement des outils de PLM présente de nombreux bénéfices. Ceux-ci peuvent être classés en trois catégories :

- effets opérationnels pour les individus impliqués dans le processus de conception : facilitation de la conception d'alternatives (ouverture à la personnalisation des produits), meilleure compréhension de l'architecture des produits, réutilisation des conceptions antérieures, réduction des erreurs de conception, assistance pour l'exécution de tâches complexes, optimisation du design, réduction du temps de collecte de données, réduction des activités à faible valeur ajoutée, assistance au travail en équipe et à la collaboration interdisciplinaire.

- effets organisationnels sur les processus : réduction des données redondantes, meilleure gestion de l'information et de la connaissance, réduction des difficultés liées à l'échange de données, coordination et contrôle de projets plus efficaces, gestion de la conception plus rigoureuse, diversification des produits, augmentation de la qualité de la conception, détection précoce des anomalies, diminution de l'impact de la redéfinition du produit sur les processus.

- effets stratégiques sur les performances industrielles : augmentation du degré d'innovation des produits, réduction des coûts du produit, réduction du temps de conception, diminution des coûts de conception.

1.1.1 Industrie et PLM

L'approche classique du développement de produits dans le cadre du PLM part d'un besoin exprimé par un client. Ce besoin est décliné en spécifications, elles-mêmes affinées pour aboutir aux spécifications de fonctions élémentaires. Ces fonctions sont alors spécifiées, conçues et réalisées indépendamment avant d'être intégrées pour constituer le produit attendu. Cette approche par fonctions élémentaires permet de mettre en œuvre une démarche de développement par étapes comme le cycle PLM en V (figure 1.1). La spécification et la conception du système sont réalisées dans la première phase (flèche descendante), et la validation et l'approbation du produit et des processus dans la seconde (flèche ascendante). Chaque étape de validation et d'approbation est liée à une étape de spécification ou de conception. L'objectif de cette démarche est de détecter les anomalies dans les étapes précoces du développement afin d'en minimiser le nombre dans les dernières étapes de la conception (où le coût et le délai de modification seraient plus importants).

La complexité croissante des systèmes implique un grand nombre de vérifications et de tests pour la validation du produit et des procédures associées, et donc, un grand nombre de prototypes nécessaires pour réaliser les tests permettant ces validations. Or, les prototypes sont par nature générateurs de coûts et de délais. La conception de produits devant se faire à coûts et temps optimisés, la section suivante montre comment le prototype virtuel représente une alternative efficace aux prototypes réels dans le processus de conception.

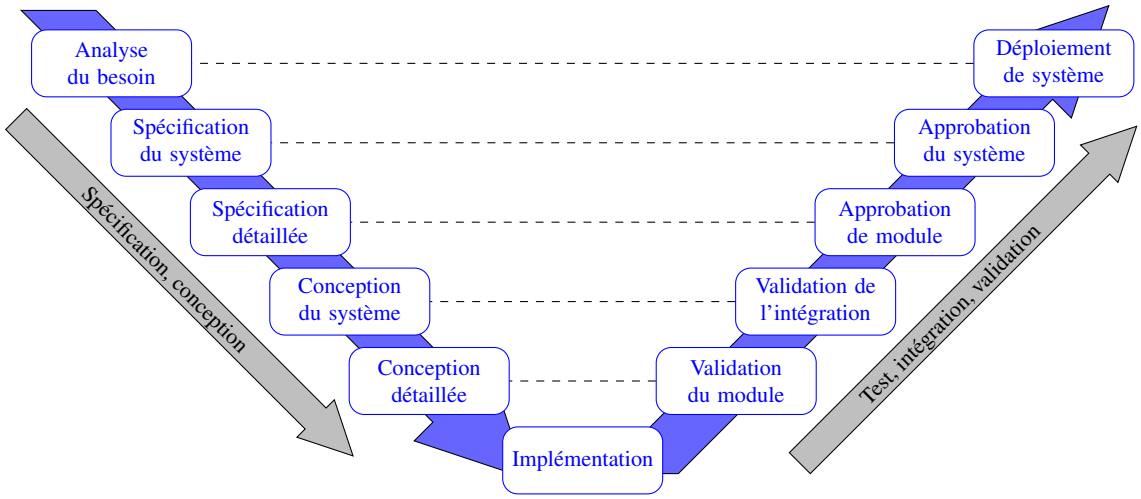


FIGURE 1.1 – Cycle de développement d'un produit industriel.

1.1.2 Le prototype virtuel : une réponse aux contraintes économiques et temporelles

L'utilisation généralisée de la CAO dans l'industrie a généralisé l'utilisation de modèles numériques des produits développés. Ces modèles numériques peuvent faire l'objet de traitements (toujours numériques) visant à tester leurs fonctionnalités [Solomalala 07, Cottet 08, Cottet 12]. On parle alors de prototypes virtuels. Les tests réalisés sur les prototypes virtuels permettent de diagnostiquer des anomalies de manière précoce pour un coût significativement réduit (temps et coût de mise en œuvre du prototype réduits), mais sont menés sur des modèles numériques avec les imprécisions liées aux approximations des modèles utilisés. Ainsi, les prototypes virtuels ne permettent pas nécessairement une validation exhaustive du produit mais réduisent de manière drastique le nombre de prototypes physiques nécessaires. Le cycle de développement à travers le PLM devient alors celui proposé dans la figure 1.2 où les tests sur prototypes physiques succèdent aux tests sur prototypes virtuels après que ceux-ci aient permis une pré-validation du système, du module ou de la fonctionnalité considérés.

L'utilisation de prototypes virtuels s'est donc généralisée pour la simulation de comportements physiques. Différents logiciels sont utilisés pour la simulation des différents comportements physiques souhaités. Par exemple, le laboratoire *Power Electronics Associated Research Laboratory* (PEARL)/*Platform for Research on power electronic Integration and Management of Energy and Storage devices* (PRIMES) propose de simuler l'ensemble du comportement physique de composants d'électronique de puissance en mettant en œuvre différents outils de calcul

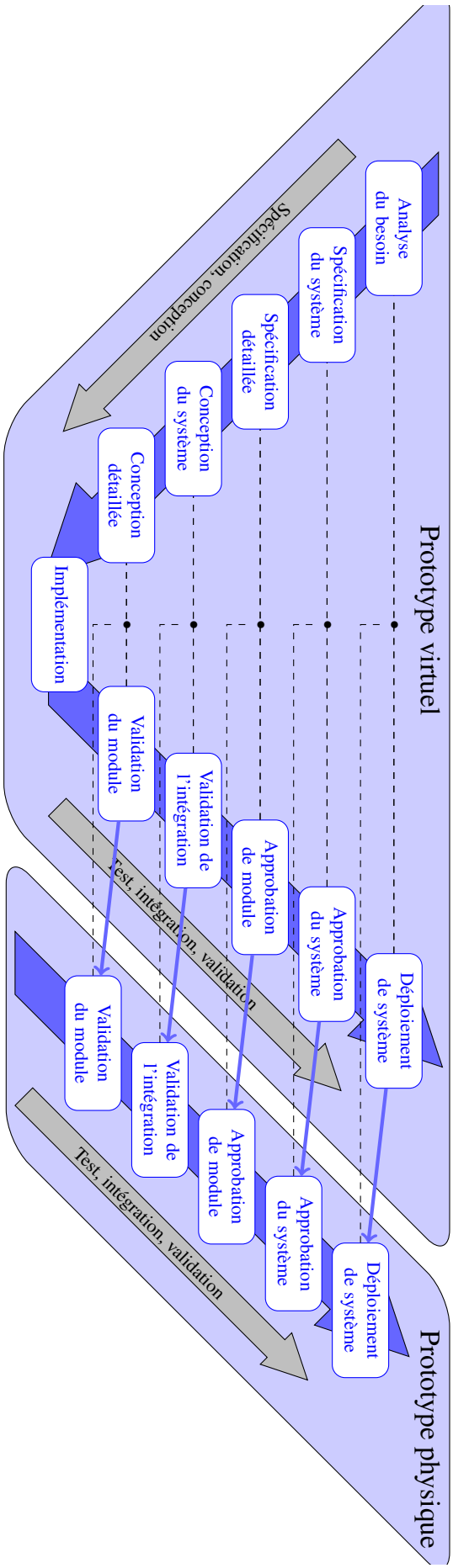


FIGURE 1.2 – Cycle de développement d'un produit industriel avec prototypes virtuels.

numérique [Solomalala 07] (figure 1.3). Ainsi, des simulations de comportement mécanique, thermique, magnétique, électrique et de dynamique des fluides sont réalisées pour pré-valider le produit avant la réalisation de prototypes physiques.

Les simulations réalisées à l'aide de prototypes virtuels peuvent aussi inclure des tests de fonctionnement cinématique. Des modules de gestion des degrés de liberté cinématique des systèmes, intégrés aux logiciels de CAO (Motion module pour Solid Works [Dassault Systemes 14c], DMU Kinematics pour CATIA [Dassault Systemes 14a], Motion Simulation pour UniGraphics [Siemens 14b]) permettent de réaliser une analyse cinématique du système et ainsi de l'animer pour en vérifier le fonctionnement.

D'autres outils définissent des trajectoires d'assemblage ou de démontage de systèmes et permettent donc de vérifier la possibilité d'intégration des différents éléments d'un produit (KineoCam figure 1.4(a) [Siemens 14a]). Ces outils rencontrent cependant de grandes difficultés à définir des trajectoires d'assemblage de systèmes fortement intégrés où les manipulations des composants sont fortement contraintes (plusieurs heures de calcul nécessaires).

Des logiciels comme Delmia permettent, eux, de simuler l'ensemble d'une ligne de production (figure 1.4(b)) [Dassault Systemes 14b]. Les différents postes d'un atelier sont alors modélisés (matériel utilisé, produits manipulés, agencement, opérations effectuées) afin de vérifier la cohérence des différents postes d'une part et de la ligne de production d'autre part.

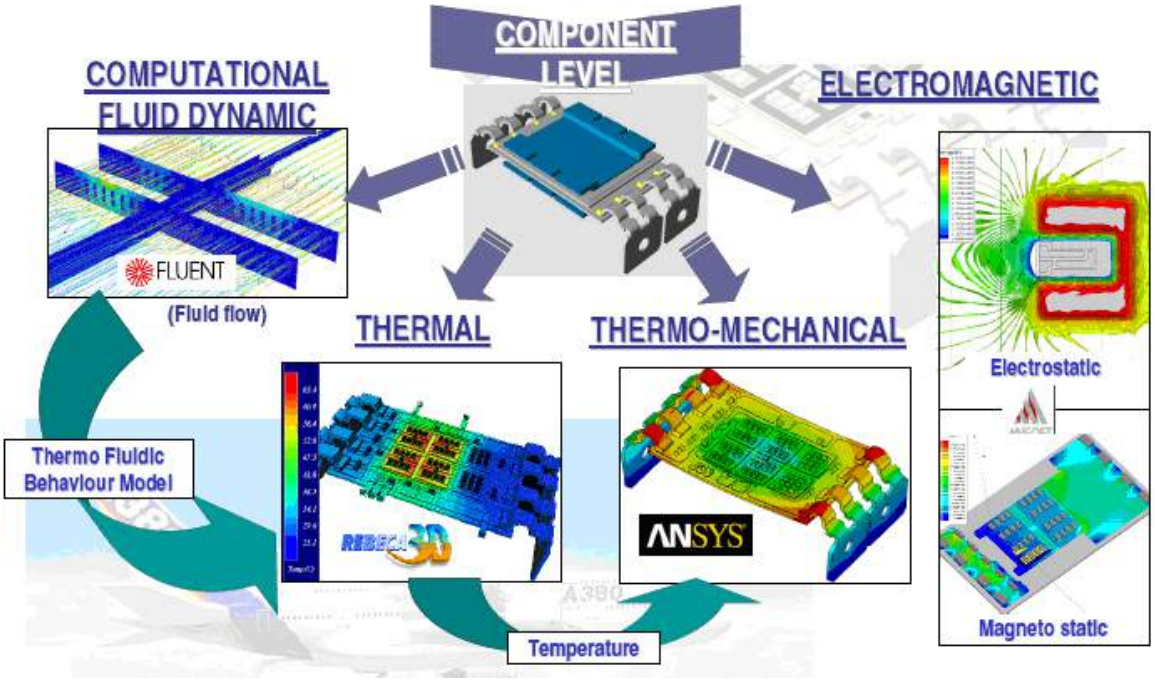
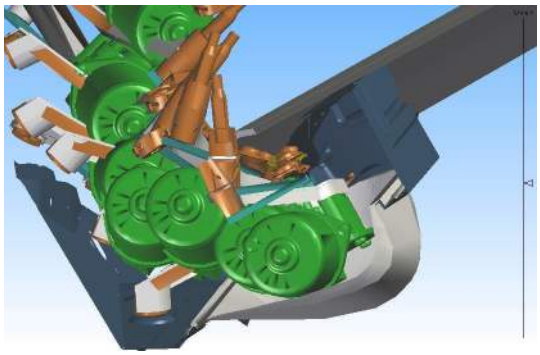


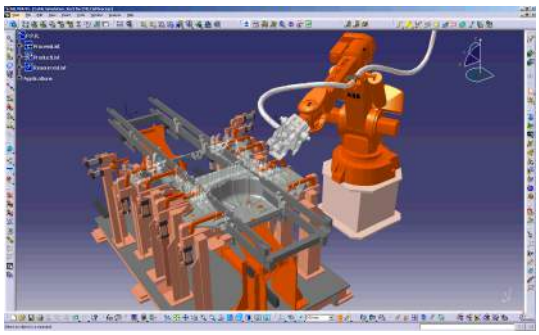
FIGURE 1.3 – Simulations physiques réalisées sur prototypes virtuels (proposition laboratoire PEARL/PRIMES [Lasserre 12]).

1.1.3 La RV comme fil rouge du processus PLM

Les outils précédents permettent de valider les conceptions par des tests de comportements physiques (mécanique, électrique, thermique,...) ou de procédés industriels (assemblage, pro-



(a) Assemblage d'un moteur d'essuie-glace avec KineoCam.



(b) Simulation d'un atelier d'une ligne de production avec Delmia.

FIGURE 1.4 – Utilisation de prototypes virtuels pour la simulation de processus.

cessus de fabrication, ligne de production,...) sur des prototypes virtuels. La RV, en immergeant un humain dans un environnement virtuel et en lui fournissant la capacité d'interagir avec lui, étend les possibilités offertes par les prototypes virtuels. On peut désormais envisager de faire simuler à un humain, les tâches qu'il aura à réaliser sur le produit réel, sur un prototype virtuel (design, visualisation, manipulation). La RV s'intègre ainsi à l'ensemble des étapes du PLM et propose de nouveaux paradigmes permettant la conception de produits dans un environnement virtuel immersif.

1.1.3.1 Définition de la RV

La RV s'inscrit dans la recherche par l'homme de moyens de s'évader de la réalité pour un univers issu de son imagination. Les peintures rupestres en sont les premières traces et cette recherche vers un univers passé ou complètement imaginaire n'a pas cessé depuis. La technique a évolué, les moyens de représenter ces univers se sont diversifiés (dessin, peinture, littérature, photographie, cinéma,...), mais l'objectif reste le même : représenter et partager un univers virtuel.

Dans les dernières décennies, le jeu vidéo a ouvert la voie de l'*interaction* avec ces univers virtuels en permettant au joueur d'influencer l'évolution du monde virtuel. La RV pousse cette idée plus loin en cherchant à *immerger* l'utilisateur dans ces univers virtuels, et à lui procurer des moyens intuitifs (sinon naturels) d'*interaction*. Ainsi, l'oxymore désignant la Réalité Virtuelle (RV) se justifie par le fait que les moyens d'immersion et d'interaction mis en œuvre tendent à rendre tangible l'univers virtuel représenté. Pour le grand public, cet oxymore accentue l'image de science fiction de la RV qui est perçue comme un rêve mêlant réalité et technologie.

Le traité de la réalité virtuelle en cinq volumes, précise les fondements, buts et méthodes de la RV [Fuchs 06a, Fuchs 06c, Fuchs 06d, Fuchs 06b, Fuchs 09]. La RV y est définie comme : *un domaine scientifique et technique exploitant l'informatique et des interfaces comportementales en vue de simuler dans un monde virtuel le comportement d'entités 3D, qui sont en interaction en temps réel entre elles et avec un ou des utilisateurs en immersion pseudo-naturelle par*

l'intermédiaire de canaux sensori-moteurs.

La RV est donc un domaine visant à interfacer la boucle perception/décision/action d'un utilisateur humain avec un monde virtuel à l'aide d'interfaces comportementales (figure 1.5). Les actions de l'utilisateur humain sont perçues à travers des interfaces motrices. En fonction des modes d'interaction choisis, ces actions sont répercutées sur le monde virtuel qui est simulé avant d'être restitué à l'utilisateur à travers des interfaces sensorielles.

Les travaux en réalité virtuelle traitent donc des trois problématiques suivantes :

1. l'analyse et la modélisation de l'humain, en environnement réel pour l'interaction, et en environnement virtuel pour la simulation ;
2. l'analyse et la modélisation de l'interfaçage du sujet avec l'environnement virtuel pour l'interaction et l'immersion ;
3. la modélisation, la construction et la simulation en temps réel de l'environnement virtuel.

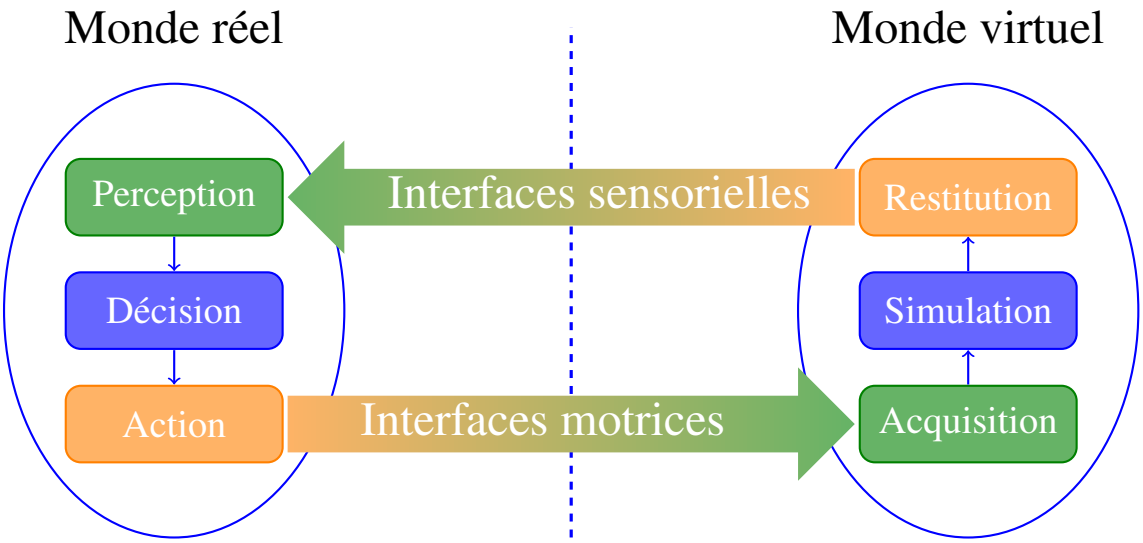


FIGURE 1.5 – Les interfaces comportementales pour l'immersion et l'interaction (inspiré de [Fuchs 06a]).

1.1.3.2 RV, industrie et PLM

Du fait de cette capacité à interagir avec le monde virtuel offerte à l'utilisateur par la RV, celle-ci se présente comme une technique intéressante pour toutes les phases du processus de développement PLM [Bennis 05, Fischer 11]. En effet, d'une part, la virtualité permet de conserver tous les avantages des prototypes virtuels précédemment évoqués (délais de disponibilité et coût), et d'autre part, la capacité d'interaction en temps réel avec ces prototypes permet à la fois d'envisager le test de tâches réalisées par des opérateurs humains, et de modifier le design des produits de manière interactive dans des simulations immersives. Nous proposons ici un aperçu de travaux ayant permis de développer des applications de RV destinées à l'industrie. Ceux-ci sont ici classés en fonction des tâches de développement de produits qu'ils abordent :

la conception mécanique : l'objectif est ici de permettre à un utilisateur de modifier la géométrie d'objets virtuels. Plusieurs applications, présentant divers degrés d'intégration aux applications de CAO ont été proposées. Un démonstrateur pour la CAO en réalité virtuelle appelé *Virtual Reality Aided Design* (VRAD) permet de manipuler les éléments classiques des logiciels de CAO (arbre de construction, *Boundary Representation* (B-Rep) de la topologie des objets, mailleur 3 dimensions (3D) pour la visualisation) tout en l'enrichissant (étiquetage des éléments de la B-Rep) pour en faciliter la manipulation dans un contexte immersif [Bourdot 10]. Cet outil manipulant directement les éléments de CAO, il est destiné à être utilisé par des opérateurs qui en sont familiers (dessinateurs industriels, ingénieurs CAO). Ce démonstrateur a permis de tester différents modes d'interaction, et ainsi de privilégier les interactions haptiques aux interactions par capture de mouvement qui souffrent d'un manque de précision du fait de l'absence de retour d'effort. Les modalités d'interactions haptiques ont par la suite été affinées [Picon 10]. Les rendus haptiques sont alors structurés en fonction des opérations réalisables en CAO (révolutions, congés, opérations booléennes,...). L'enjeu est ici de fournir un rendu haptique cohérent malgré la diversité des opérations disponibles.

Une autre approche est proposée pour les non-experts [Meyrueis 11]. Cette fois, n'importe qui peut déformer localement les objets d'un environnement à l'aide de la capture des mouvements d'une Wiimote [1]. Le *design* des produits peut ainsi être modifié en immersion par des utilisateurs non experts en CAO. Les modifications réalisées demandent ici à être reprises par la suite, par un expert, sur un logiciel de CAO afin de prendre en compte les contraintes liées au métier et/ou aux techniques.

Une approche encore différente, destinée à des *designers* issus du milieu artistique leur permet de re-exploiter, en RV, leurs compétences acquises dans le monde réel [Bordegoni 06]. L'application proposée vise à leur permettre de mettre en forme la matière par *scraping* [2]. Le modelage est ici réalisé grâce à une interface haptique et à des modèles physiques plastiques et viscoélastiques des matériaux pour la sculpture. Les géométries générées ici sont modélisées à l'aide de maillages 3D mais ne sont pas compatibles avec les modèles CAO (absence de B-Rep et d'arbre de construction).

les processus de production : La réalité mixte (mixant RV et Réalité Augmentée (RA)) permet de tester l'intégration de nouveaux systèmes dans une ligne de production en intégrant des éléments virtuels dans une scène réelle. Pour cela, les éléments virtuels sont positionnés dans les prises de vues du système réel grâce à la reconnaissance et à la localisation de marqueurs 2 dimensions (2D) préalablement positionnés sur la ligne de production [Lee 11].

Les logiciels de simulation de processus de production peuvent aussi être interfacés à des systèmes de RV et de RA pour la conception d'un système de production [Dangelmaier 05]. La mise en œuvre de la RV et de la RA est alors utile à différentes étapes de la modélisation du nouveau système de production : analyse de l'existant, modélisation, simulation, évaluation,

1. Contrôleur Bluetooth originellement conçu pour une console de jeu (Nintendo Wii) équipé d'accéléromètres permettant d'en capturer le mouvement.
2. Technique de modelage par raclement.

ajustement et présentation de la solution retenue. Une contribution forte de ces travaux est la capacité de l'outil à faire coopérer différents acteurs (y compris non spécialistes des systèmes de production) dans chacune de ces étapes.

l'analyse ergonomique : La RV peut être mise en œuvre pour l'évaluation de l'ergonomie du système de production. La RV permet de visualiser et de tester le poste de travail et les différentes opérations qui devront y être menées de manière réaliste [Bennes 12]. Ces tests des postes de travail permettent d'en améliorer l'ergonomie avant leur implémentation, dès la phase de conception.

La performance des systèmes de RV pour mener de telles études a été évaluée en comparant la simulation de tâches sur un poste de travail réel et la simulation de tâches en RV. Par exemple l'ergonomie d'un poste de travail informatique a été testée [Gude 07]. Cette étude a permis de mettre en évidence une plus forte implication des utilisateurs finaux dans la conception de leur futur poste de travail grâce à la mise en œuvre de la RV. De même, une comparaison de la RV et du poste travail réel a aussi été réalisée pour l'étude d'un poste de travail sur une presse plieuse [Marc 07]. Dans ce dernier cas, les opérateurs réalisant des simulations en RV ressentent plus une sensation de télé-opération qu'une sensation d'immersion. Les auteurs attribuent cette différence à l'interfaçage avec l'environnement virtuel qui ne permet pas une interaction assez naturelle. Toujours pour simuler les opérations sur une presse plieuse, le développement d'une main virtuelle (et de ses contrôles) a permis d'évaluer la sécurité des postes de travail grâce à une simulation réaliste de la saisie des objets virtuels [Pouliquen 07].

En plus de l'ergonomie du système de production, celle du produit peut aussi être évaluée en RV. Différentes alternatives de conception peuvent être testées et évaluées en RV. Par exemple, différents comportements mécaniques d'un bouton poussoir (raideur, course,...) devant intégrer le tableau de bord d'une voiture peuvent être comparés [Moreau 04]. Cette étude a permis de déterminer que dans ce cas, la conception d'une interface dédiée (interface haptique à 1 degré de liberté) est plus efficace que l'adaptation d'interfaces génériques (adaptation d'un Phantom Desktop [3]). Une autre application d'essai d'un fauteuil roulant motorisé équipé d'un bras robotisé vise à tester différentes alternatives pour le contrôle du fauteuil et du bras, et ainsi à définir celles à implanter sur le produit réel [Di Gironimo 12].

l'aide à la décision : Les outils de RV dédiés aux différentes étapes du PLM aident à la décision en permettant une évaluation subjective (il s'agit de ressentis individuels) des différentes alternatives possibles tout au long du processus de conception des nouveaux produits. Cependant, des outils plus spécifiquement dédiés à la prise de décision existent.

Par exemple, l'analyse conjointe [4] peut être intégrée à une application de visualisation immersive [Dijkstra 03]. Cet outil est utilisé pour comparer différentes alternatives de cloisons de partitionnement de l'espace d'un nouveau bâtiment de travail. L'analyse conjointe a permis de

3. Interface haptique à 6 degrés de liberté (ddl).
4. Méthode d'évaluation de différentes alternatives basée sur l'étude de plusieurs attributs [Timmermans 84].

converger vers une solution optimisée (satisfaisant la majorité) incluant des séparations translucides entre les espaces publics et les espaces de travail, et des cloisons opaques entre les différents postes de travail.

Différentes applications de RV et RA permettent d'évaluer des projets d'urbanisme [Kieferl 03]. Les projets étudiés ont permis de déterminer les avantages et limites de différentes techniques. Par exemple, la RV permet de réaliser facilement des expériences sur des groupes d'individus (utilisation de *caves*[5]) alors que la RA fonctionne mieux pour des expériences individuelles [Kieferl 03]. Ces exemples montrent que RA et RV permettent, à partir de la visualisation de projets, d'apporter une aide pour faire des choix de conception.

La visualisation d'un environnement virtuel réaliste peut ne pas être suffisante pour prendre des décisions. La RV permet aussi de visualiser des données abstraites comme des nuages de points. La nature interactive de la RV permet alors de choisir dynamiquement le nombre d'affichages (fenêtres) utilisés mais aussi les variables associées à chacune des dimensions (visualisation en 3D) de ces affichages [Eddy 02]. Ces différentes représentations peuvent de plus être enrichies d'une part par l'utilisation de couleurs pour différencier les points selon divers critères, et d'autre part par l'affichage d'éléments tels qu'un front de Pareto ou d'autres éléments d'aide à la décision. L'intérêt de cet outil de visualisation de points est démontré en l'appliquant à la visualisation des résultats issus d'algorithmes génétiques pour l'optimisation du design d'une poutre multi-matériaux [Eddy 02].

l'assemblage, la manipulation : Différentes applications de RV fournissent à l'utilisateur la possibilité de manipuler les composants virtuels d'un système afin de les assembler. On note alors la diversité des interfaces disponibles pour la manipulations d'objets virtuels. On peut par exemple profiter d'interfaces peu onéreuses tels qu'une Wiimote pour effectuer un pré-positionnement, et bénéficier du retour sensoriel des interfaces haptiques pour les manipulations fines [Bordegoni 09]. L'entrainement aux tâches de montage et de démontage pour la maintenance de systèmes industriels peut être réalisé à l'aide d'un outil de RV de bureau comprenant des périphériques de visualisation et de manipulation [Li 03]. Cependant, si les modes de manipulation des composants offrent une ergonomie satisfaisante, les produits industriels modernes présentent une intégration élevée qui rend la perception des contraintes géométriques et donc la définition d'une trajectoire d'assemblage difficile malgré l'immersion de l'utilisateur dans le prototype virtuel.

La simulation peut alors être enrichie d'outils permettant d'assister l'utilisateur de RV. Des graphes des contraintes d'assemblage modélisant les degrés de libertés relatifs des différents composants du système permettent par exemple de générer automatiquement les séquences de démontage et montage proposées à l'utilisateur. Pour définir automatiquement les contraintes d'assemblage, il existe différents outils d'analyse et de modélisation de l'assemblage et du démontage de systèmes industriels [Iacob 12]. Ces outils permettent, à partir d'une analyse de

5. Salle dont les mur, le sol et/ou le plafond sont des écrans permettant une forte immersion de l'utilisateur de RV dans l'environnement virtuel

l'assemblage, de déterminer les contraintes d'assemblages et les ddl des différents composants. Ils s'appuient soit sur l'arbre d'assemblage du système créé dans le logiciel de CAO (ils sont alors dépendants du format associé au logiciel de CAO choisi), soit sur l'analyse des contacts entre les différents composants du système assemblé (analyse plus difficile, mais cette fois générique puisque basée uniquement sur les modèles géométriques).

Une fois les contraintes d'assemblages identifiées, des guides virtuels [6] peuvent être utilisés pour faciliter l'assemblage des composants par l'utilisateur (au détriment toutefois de sa liberté de mouvement et donc de sa capacité d'initiative) [Tching 10]. Ces travaux adressent la question de l'activation et de la désactivation de la détection de collision qui représente un point clé pour l'assemblage de composants présentant un ajustement serré (l'absence de jeu pour les ajustements serrés rend l'assemblage de prototypes virtuels impossible sans contact). Les manipulations de l'objet virtuel par l'utilisateur sont alors progressivement contraintes jusqu'à la position finale du composant et donc l'obtention du système assemblé.

Une assistance plus poussée peut aussi être proposée pour l'assemblage en utilisant une trajectoire d'assemblage calculée à l'aide d'algorithmes de planification de trajectoire issus de la robotique [Ladevèze 10b]. La trajectoire calculée est utilisée pour guider l'utilisateur lors des manipulations d'assemblage à l'aide d'une interface haptique. Les manipulations de l'utilisateur sont aussi prises en compte puisqu'en ne suivant pas la trajectoire proposée, l'utilisateur peut déclencher la définition d'une nouvelle trajectoire.

La simulation d'assemblage en RV peut encore être enrichie par la simulation de corps souples (cables, durites,...) [Loock 01]. Il convient alors de développer un modèle de corps souple pouvant être simulé en temps réel dans le cadre d'une application interactive.

1.1.4 Synthèse

Nous avons pu voir dans cette section 1.1 que la mise en œuvre de prototypes virtuels lors du développement de produits industriels a pour objectif de diminuer les coûts et temps de développement en réduisant le recours aux prototypes physiques. L'utilisation de ces prototypes virtuels dans des applications de RV permet de plus à un utilisateur en immersion et en interaction de manipuler le prototype virtuel de manière sinon naturelle, au moins intuitive. Ces modes d'interaction intuitifs permettent d'associer une plus large gamme d'utilisateurs au processus de développement en permettant à des non-experts d'y intervenir facilement.

Nous avons aussi pu voir que les prototypes virtuels sont utilisés pour vérifier la faisabilité de différentes opérations des phases du cycle de vie du produit. En particulier, la possibilité d'assemblage et de démontage du produit industriel est vérifiée à l'aide de simulations mettant en œuvre des prototypes virtuels. Ces simulations sont principalement de deux natures : des applications de RV permettant à un opérateur humain de manipuler les composants virtuels d'un système pour l'assembler ou le démonter, et des simulations hors ligne de calcul de trajectoire permettant de définir des trajectoires d'assemblage de manière automatique à l'aide de

6. *Virtual fixtures*

techniques issues de la recherche en robotique.

Nous allons maintenant voir comment ces deux approches complémentaires peuvent être utilisées conjointement. L'enjeu est dans ce cas de faire collaborer un opérateur humain et des techniques de planification automatique de trajectoire en temps réel pour la définition d'une trajectoire d'assemblage. Nous allons dans un premier temps voir comment des travaux précurseurs ont permis la collaboration de ces deux systèmes très différents (planificateur automatique et opérateur humain), avant de pointer les verrous techniques et scientifiques que cela implique (section 1.2.2) puis d'introduire comment nous nous proposons de les adresser (section 1.2.3).

1.2 Assemblage et planification interactive de trajectoire

On constate donc que la simulation d'assemblage avec des prototypes virtuels se fait au travers de deux outils distincts ayant des finalités différentes :

1. les outils de calcul de trajectoire issus de la robotique qui permettent de vérifier la faisabilité d'un assemblage ;

2. les applications d'assemblage en RV qui permettent à un opérateur humain de manipuler les prototypes virtuels, et ainsi de tester les tâches d'assemblage et de s'y entrainer.

Dans nos travaux, nous défendons l'idée que ces outils sont complémentaires et que leur mise en œuvre dans une application interactive unique est bénéfique sous plusieurs aspects :

- Lors de la manipulation de composants en environnement virtuel, la mise en œuvre de techniques de planification automatique de trajectoire permet de guider l'utilisateur dans un environnement qui peut être difficile à appréhender soit du fait de sa complexité propre, soit du fait de l'interfaçage (visualisation, navigation).

- Les algorithmes de planification automatique de trajectoire peuvent bénéficier des connaissances métier de l'utilisateur humain et de sa capacité à appréhender son environnement de manière synthétique grâce à son expérience acquise.

- La définition de la trajectoire d'assemblage peut être réalisée par le système de planification de trajectoire et/ou par l'utilisateur humain de manière complémentaire en fonction, par exemple, de la capacité de chacune de ces entités à appréhender l'environnement.

L'enjeu est donc de mettre en œuvre des techniques de planification automatique de trajectoire dans des simulations interactives en RV. Pour cela, il convient de fournir à l'utilisateur de RV des moyens pour intervenir dans le processus de planification de trajectoire.

1.2.1 Travaux précurseurs de planification interactive de trajectoire

Ces différents travaux seront détaillés dans le chapitre 4. Nous nous contenterons pour l'heure d'en donner les principes pour faire apparaitre les différences d'approche et les limites.

Les travaux de Ladevèze [Ladevèze 10b], réalisés au Laboratoire Génie de Production (LGP), ont exploré l'intérêt de l'utilisation conjointe de la planification automatique de trajectoire et de la RV. Ils ont permis de montrer l'intérêt de la collaboration entre planificateur de trajectoire et opérateur humain en définissant différentes modalités de planification de trajectoire et d'interaction temps-réel. Cette utilisation de la réalité virtuelle et des techniques de planification automatique de trajectoire permet bien, d'une part, de conserver l'aspect didactique des simulations de RV, et d'autre part de faire bénéficier la planification de trajectoire de l'expertise de l'utilisateur immergé dans la simulation de RV.

Ladevèze a défini deux planificateurs interactifs de trajectoire :

1. *Le planificateur probabiliste avec approche globale* utilise une trajectoire pré-calculée pour guider l'utilisateur. Cette trajectoire est calculée en deux étapes. La première, met en œuvre une approche globale pour définir un chemin évitant les obstacles dans un modèle de l'espace libre 3D de l'environnement. La seconde explore ce chemin aléatoirement pour y définir une trajectoire combinant position et orientation. L'interaction est ici articulée autour du guidage de l'utilisateur sur la trajectoire pré-calculée, mais aussi sur la possibilité pour l'opérateur de déclencher la définition d'une nouvelle trajectoire selon un processus similaire à celui décrit en s'écartant de celle qui lui est proposée.

2. *Le planificateur local* se contente de tirer l'utilisateur vers son objectif en faisant abstraction des obstacles de l'environnement. Il revient alors à l'utilisateur de réaliser le contournement des obstacles sur son chemin. L'interaction est ici encore articulée autour du guidage de l'utilisateur, mais cette fois, la prise en compte des actions de l'utilisateur ne consiste qu'à mettre à jour ce guidage en fonction des mouvements qu'il réalise.

Dans ces deux approches, la trajectoire obtenue est celle des manipulations réalisées par l'utilisateur soumis au guidage du système de planification interactive.

D'un autre coté, les travaux de Flavigné [Flavigné 10a] réalisés au Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) se sont penchés sur l'interaction temps-réel entre un opérateur humain et un planificateur automatique de trajectoire en définissant un mode de contrôle temps-réel de l'algorithme de planification de trajectoire par l'opérateur humain.

Pour cela, l'algorithme de planification de trajectoire *Interactive Rapidly-exploring Random Tree* (IRRT) proposé par Flavigné est dérivé de l'algorithme de planification automatique de trajectoire *Rapidly-exploring Random Tree* (RRT) (chapitre 3). Flavigné a pour cela défini un algorithme RRT explorant une direction privilégiée. L'interaction est alors articulée autour de cette direction d'exploration. D'une part, l'algorithme, en fonction de l'exploration préalable, indique à l'utilisateur une première direction qui lui semble intéressante à explorer. D'autre part, l'utilisateur suggère lui aussi une direction définie en fonction de sa perception de l'environnement. Ces deux directions sont ensuite fusionnées pour définir celle dans laquelle l'algorithme RRT va explorer l'environnement.

Ces trois planificateurs de trajectoire interactifs ont donc permis de définir des modes d'interaction entre un système de planification automatique et un opérateur humain. Ils ont établi les informations échangées par le système automatique et l'opérateur humain pour cette interaction.

L'implication d'un opérateur humain dans le processus de planification fait apparaitre la problématique du temps de calcul comme cruciale. En effet, alors que pour les robots, les trajectoires sont calculées hors ligne, dès lors qu'un opérateur est intégré dans le processus de planification, il faut que la réponse du système automatique à ses actions soit faite dans un délai suffisamment court. Cette problématique a été adressée de façon différente dans les trois planificateurs de trajectoire interactifs :

- *planificateur probabiliste avec approche globale :* nouveau processus de planification en deux phases,
- *planificateur local :* simplification du problème (obstacles ignorés),
- *IRRT :* intégration de l'opérateur dans le processus de planification.

Cependant, bien qu'elles explorent des voies prometteuses, ces solutions présentent toutes des limites de nature différente :

- le processus de planification de trajectoire en deux phases du *planificateur probabiliste avec approche globale* permet certes de mettre rapidement une trajectoire à disposition de l'utilisateur mais augmente par la même occasion les chances d'échec ;
- l'assistance fournie par le *planificateur local* permet seulement de pointer à l'utilisateur la direction dans laquelle se trouve l'objectif mais peut ne pas être pertinente dans des environnements complexes difficiles à appréhender pour l'utilisateur ;
- la fusion des directions d'exploration fournies par le planificateur automatique de trajectoire et par l'utilisateur dans le planificateur IRRT doit faire l'objet d'une attention particulière pour que le planificateur ne se bloque pas lorsque ces directions sont opposées.

Ces limites peuvent être attribuées pour partie au fait qu'à l'instar des planificateurs automatiques de trajectoire issus de la robotique, les trois planificateurs proposés ont une approche purement géométrique de la planification de trajectoire. Il reste donc, en exploitant des données de nature différente (d'un plus haut niveau d'abstraction), un large champ d'investigations pour améliorer à la fois le processus de planification de trajectoire et l'interaction entre ce dernier et un opérateur humain.

1.2.2 Enjeux, verrous

Nous considérons donc qu'un progrès significatif de la planification interactive de trajectoire doit venir de l'exploitation, dans ce processus de données d'un haut niveau d'abstraction. Ces données doivent permettre de réaliser des raisonnements plus riches permettant un traitement optimisé des données géométriques.

À partir de cette idée, différents verrous ont été identifiés :

1. **La mise à disposition de données hétérogènes sur l'environnement.** Les données de haut niveau d'abstraction doivent compléter les données géométriques classiquement utilisées dans une architecture cohérente intégrant cette diversité de données. On peut de plus considérer que l'expertise mise en œuvre dans différentes tâches est différente. Dans ce cas, les données de haut niveau d'abstraction permettant de modéliser cette expertise doivent être adaptées à la tâche traitée par l'application. Cette architecture doit donc être modulaire afin de pouvoir l'adapter aux différentes tâches que nous souhaitons lui faire traiter.

2. **La définition et la mise en œuvre d'une technique de planification de trajectoire exploitant les données de l'environnement disponibles.** L'exploitation de l'ensemble de ces données et en particulier des données d'un haut niveau d'abstraction doit permettre de réaliser des raisonnements permettant de diminuer le temps de calcul nécessaire à la définition d'une trajectoire, tout en augmentant la pertinence des solutions retenues. Nous rappelons ici que le contrôle du temps des processus dans le cadre d'applications interactives est un point clé pour permettre l'interaction temps-réel entre ces derniers et les humains avec lesquels ils sont en interaction. Il convient donc de définir comment l'ensemble des données et leur diversité peuvent être mis en œuvre de manière efficace pour la définition d'une trajectoire.

3. **La définition et la mise en œuvre de modes d'interaction permettant la collaboration d'un opérateur humain avec la technique de planification de trajectoire développée.** Ces modes d'interaction doivent exploiter les différentes données de l'environnement pour améliorer l'interprétation des actions réalisées par l'opérateur humain en interaction. Cette meilleure interprétation des actions de l'opérateur humain doit aboutir à une amélioration de l'efficacité de la collaboration entre planificateur automatique de trajectoire et opérateur humain en permettant au premier de mieux adapter l'assistance qu'il fournit au second.

Les enjeux traités par les travaux présentés dans ce mémoire portent donc aussi bien sur la définition des concepts permettant une planification interactive de trajectoire basée sur des données hétérogènes de l'environnement que sur leur mise en œuvre dans une architecture modulaire. En effet, si, comme nous l'avons dit la modélisation de l'environnement doit s'adapter à l'expertise liée à la tâche à traiter, l'ensemble des processus exploitant ces données (planification de trajectoire et interaction) doivent s'y adapter de même.

1.2.3 Contributions

Pour notre planification de trajectoire interactive, nous retenons dans les travaux précurseurs présentés précédemment :

- l'interaction au travers d'une interface haptique permettant au planificateur automatique de suggérer une trajectoire d'une part, et à l'utilisateur de s'affranchir de cette suggestion d'autre part jusqu'à solliciter la définition d'une nouvelle trajectoire.

- le choix des informations géométriques comme données échangées entre planificateur et utilisateur (direction du mouvement suggérée par le système automatique, positionnement de l'objet manipulé par l'opérateur humain)

- la planification temps-réel permettant au planificateur d'intégrer les mouvements exécutés par l'utilisateur dans la trajectoire planifiée, et à l'utilisateur de manipuler les objets virtuels sans perturber son immersion.

Nous nous proposons de conserver ces acquis et de les améliorer en dotant le système de données de plus haut niveau d'abstraction. Au cours du travail exposé dans ces pages, nous nous sommes attachés à définir une planification de trajectoire interactive basée sur une représentation de l'environnement multi-niveaux permettant d'intégrer des aspects topologiques et sémantiques en plus de ceux géométriques traditionnellement utilisés.

En fournissant au planificateur de trajectoire automatique une représentation de l'environnement incluant des données sémantiques et topologiques, d'une part, nous améliorons la planification automatique de trajectoire réalisée, et, d'autre part, nous améliorons l'interaction entre planificateur automatique et opérateur humain par une meilleure prise en compte de ce dernier dans le processus de planification. La contribution de ces travaux s'articule donc autour de trois points :

Une représentation multi-niveaux de l'environnement complète les informations géométriques traditionnellement utilisées avec des données topologiques et sémantiques (chapitre 2). Cette représentation est modulaire et évolutive. Elle peut être adaptée ou enrichie en fonction, des tâches à exécuter et de la connaissance métier définissant les concepts utiles à la planification de trajectoire. Elle est constituée de :

1. Un niveau sémantique permettant, en fonction des informations de l'environnement utiles pour la planification, de caractériser l'environnement. L'information ajoutée permet de choisir quel chemin emprunter (choix entre plusieurs alternatives), et comment (choix de la modalité de planification géométrique).

2. Un niveau topologique pour modéliser les lieux de l'environnement, leurs frontières et leur connectivité. Cette modélisation de la connectivité des lieux de l'environnement facilite l'exploration de ce dernier pour y déterminer le chemin le plus approprié.

3. Un niveau géométrique pour modéliser les corps rigides et l'espace libre 3D de l'environnement.

Un planificateur automatique de trajectoire multi-niveaux adapté à la représentation de l'environnement multi-niveaux permet une interaction temps-réel plus efficace (chapitre 3). Cette interaction est facilitée par l'accélération de la définition des trajectoires, mais aussi par la pertinence des trajectoires définies. Ces deux améliorations sont obtenues par le partitionnement de la planification de trajectoire en deux étapes :

1. Une première étape de planification grossière sur le niveau topologique, contrôlée par les données du niveau sémantique permet de déterminer les lieux à traverser.

2. Une seconde étape de planification fine permet par la suite de déterminer une trajectoire permettant de traverser chaque lieu. Le mode de planification fine est adapté à chacun des lieux en fonction de l'information sémantique qui y est attachée.

Le partitionnement du processus de planification en deux phases (grossière et fine) semble particulièrement adapté à une utilisation en interaction avec un opérateur humain puisqu'un partitionnement similaire a été identifié en neurobiologie [Ahmadi-Pajouh 07].

Des modes d'interaction adaptés à l'architecture multi-niveaux de modélisation de l'environnement et de planification de trajectoire qui permettent à un opérateur humain et au planificateur automatique de trajectoire de collaborer. Ces modes d'interaction sont enrichis par l'exploitation des différents types d'informations disponibles dans la représentation de l'environnement (sémantique, topologique et géométrique). Ils sont aussi adaptés au partitionnement de la planification de trajectoire en deux phases et leur ergonomie est améliorée par la mise en œuvre de techniques de **partage de contrôle** issues de la télé-opération de robot et du pilotage de véhicules semi-autonomes (chapitre 4). Ces modes d'interaction et leur mise en œuvre permettent au système automatique :

1. d'assister l'opérateur en temps réel y compris lorsque le planificateur automatique de trajectoire n'est pas en mesure de déterminer complètement une trajectoire géométrique réalisable (portions de trajectoire en collision avec les obstacles de l'environnement), l'opérateur doit alors diagnostiquer et définir ces portions de trajectoires manquantes ;

2. d'adapter dynamiquement l'autorité entre le planificateur automatique et l'opérateur pendant la manipulation pour assister l'utilisateur lorsqu'il est en difficulté, mais aussi, le laisser libre d'explorer des alternatives lorsqu'il le souhaite ;

3. d'interpréter le plus tôt possible, en temps réel, et dans un modèle de données de haut niveau (comme celui de l'utilisateur), les intentions de l'opérateur à partir des mouvements qu'il a réalisés. Cette interprétation précoce permet d'adapter le reste du chemin planifié à cette intention perçue.

L'interaction proposée ici permet au planificateur automatique et à l'utilisateur immergé dans la simulation en RV de collaborer pour la définition de la trajectoire.

1.3 Communications

Les travaux présentés dans ce document ont fait l'objet de communications internationales :

1. Simon Cailhol, Philippe Fillatreau & Jean-Yves Fourquet. *Interactive multimodal Path Planning in immersion*. In ECMSM, Electronics, Control, Measurement, Signals and their application to Mechatronics, Toulouse (France), June 2013.

2. Simon Cailhol, Philippe Fillatreau, Jean-Yves Fourquet & Yingshen Zhao. *A multi-layer approach for path planning control in Virtual Reality simulation*. In IDMME, Integrated Design and Manufacturing in Mechanical Engineering, Toulouse (France), June 2014.

3. Simon Cailhol, Philippe Fillatreau, Jean-Yves Fourquet & Yingshen Zhao. *A multi-layer approach for interactive path planning control*. In ICINCO, International Conference on Informatics in Control, automation and robotics, Vienna (Austria), September 2014.

4. Simon Cailhol, Philippe Fillatreau, Jean-Yves Fourquet & Yingshen Zhao. *A multi-layer approach for interactive path planning control in Virtual Reality simulation*. In ICVRV, International Conference on Virtual Reality and Visualization, Shenyang (China), August 2014, Shenyang (China)

5. Simon Cailhol, Philippe Fillatreau, Yingshen Zhao & Jean-Yves Fourquet. *A hierarchic approach for path planning in Virtual Reality*. IJIDeM, International Journal on Interactive Design and Manufacturing, pages 1–12, 2015. Version étendue de la communication faite à IDMME 2014.

6. Simon Cailhol, Philippe Fillatreau, Yingshen Zhao & Jean-Yves Fourquet. *A Multi-Layer Approach of Interactive Path Planning for Assisted Manipulation in Virtual Reality (poster)*. In VR 2015, IEEE Virtual Reality, Arles (France). IEEE, March 2015

   Soumis :

7. Simon Cailhol, Philippe Fillatreau, Yingshen Zhao & Jean-Yves Fourquet. *Hierarchic interactive path planning in Virtual Reality*. LNEE, Lecture Notes in Electrical Engineering, 2015. Version étendue de la communication faite à ICINCO 2014.

Chapitre 2

Modèle multi-niveaux de l'environnement

Table **Page**

2.1 État de l'art 26

2.1.1 Espaces et composantes libres d'obstacles 27

2.1.2 Représentation numérique des corps rigides 28

2.1.2.1 Modèles volumiques 28

2.1.2.2 Modèles surfaciques 28

2.1.3 Modélisation de l'espace libre 29

2.1.3.1 Modélisation géométrique 29

2.1.3.1.1 Décomposition en cellules 29

2.1.3.1.2 Définition de *roadmaps* 31

2.1.3.2 Modélisation topologique 32

2.1.3.2.1 Notion de lieu 33

2.1.3.2.2 Information associée 34

2.1.3.3 Modélisation sémantique 34

2.1.4 Synthèse 35

2.2 Contribution 35

2.2.1 Modélisation des `Corps rigides` 36

2.2.1.1 Niveau géométrique 36

2.2.1.2 Niveau sémantique 37

2.2.2 Modélisation de l'`Espace libre` 37

2.2.2.1 Choix préalables de représentation 37

2.2.2.2 Niveau sémantique 38

2.2.2.3 Niveau topologique 39

2.2.2.4 Niveau géométrique 39
2.2.3 Synthèse du modèle multi-niveaux de l'environnement proposé 40
2.3 Implémentation du modèle de l'environnement 41
2.3.1 Modèle statique : classes et responsabilités 42
2.3.1.1 Vue générale 42
2.3.1.2 `Graphe topologique` 43
2.3.1.3 `Information sémantique` 45
2.3.2 Algorithmes, processus et parallélisation 46
2.3.2.1 Création et mise à jour de l'octree 47
2.3.2.2 Définition des `Lieux` 48
2.3.2.3 Construction du `Graphe topologique` 50
2.3.2.4 Parallélisation des processus 51
2.4 Conclusion sur la modélisation de l'environnement réalisée 52

L'objet de ce chapitre est de décrire le modèle retenu dans nos travaux pour l'environnement et les objets qui s'y trouvent. Ce modèle doit favoriser la planification interactive de trajectoire d'un objet mobile dans un environnement encombré. Afin de permettre la collaboration entre un opérateur humain et le planificateur automatique de trajectoire, nos travaux explorent différents niveaux de modélisation de l'information.

Du point de vue géométrique, l'environnement est une partie fermée bornée de l'espace cartésien dans lequel sont définis les obstacles. Dans notre travail, on considère que les obstacles et les objets mobiles sont des corps rigides.

Dans un premier temps, un état de l'art est proposé en s'appuyant sur la littérature en informatique graphique, en CAO et en robotique (section 2.1). Celui-ci s'intéresse successivement aux espaces considérés, aux corps rigides manipulés et à l'*espace libre*, composante de l'environnement ou de son modèle, atteignable pour l'objet manipulé. Cet état de l'art permet de distinguer différents niveaux de modélisation : géométrique, topologique et sémantique.

Ensuite, le modèle proposé pour l'environnement et l'objet manipulé est détaillé (section 2.2) avant d'en décrire l'implémentation (section 2.3). Afin de favoriser la collaboration entre un opérateur humain et le planificateur automatique de trajectoire, le modèle proposé inclut, en plus des données géométriques classiques, des informations topologiques et sémantiques.

2.1 État de l'art

Pour améliorer les performances de la planification automatique de trajectoires, et l'intégration d'un opérateur humain dans le processus de planification, le planificateur interactif de trajectoire proposé exploite des informations sur l'environnement ayant différents niveaux

d'abstraction. Nous allons donc voir, dans cet état de l'art, les outils existants pour représenter la géométrie d'un environnement, mais aussi sa topologie et sa sémantique. Ces outils sont principalement issus de l'infographie et de la CAO pour les modèles géométriques 3D, et de la robotique pour les aspects topologiques et sémantiques. Nous distinguons parmi ces outils les outils de modélisation permettant de fournir une représentation détaillée des objets de l'environnement, et les outils de cartographie synthétisant l'information utile à la modélisation de l'espace libre.

2.1.1 Espaces et composantes libres d'obstacles

En planification, on s'intéresse au mouvement d'un ou plusieurs solides, éventuellement contraints les uns par rapport aux autres par des liaisons mécaniques dans un environnement encombré par des obstacles. Ce mouvement s'effectue dans l'espace cartésien et chacun des points de l'environnement et des corps en mouvement peut-être décrit par un vecteur de $\mathbb{R}^3$: les coordonnées cartésiennes du point, par exemple. Lorsqu'on veut caractériser la position de tous les points d'un solide dans $\mathbb{R}^3$, on peut se ramener à la définition de la *situation* du solide qui requiert localement 6 paramètres indépendants : les coordonnées cartésiennes de 3 points non alignés du solide (9 paramètres liés par 3 relations de distance) ou encore, par exemple, les 3 coordonnées cartésiennes d'un point du solide et 3 rotations indépendantes qui décrivent son orientation par rapport à une base vectorielle de référence. On parlera alors parfois dans la littérature de *représentation 6D* ou d'*espace 6D*.

Lorsque l'on considère un ensemble de solides contraints par des liaisons holonomes, la situation d'un corps par rapport à un autre est caractérisée par la *classe* de la liaison qui les contraint. Ainsi, si on étudie une chaîne cinématique robotique constituée de n liaisons à un degré de liberté (rotation ou translation), la situation relative de deux solides est décrite par un seul paramètre. Le vecteur qui regroupe l'ensemble de ces paramètres constitue le vecteur de *configuration* de la chaîne cinématique. De cette manière, planifier pour cette chaîne cinématique revient à étudier les valeurs possibles d'un point dans l'*espace des configurations* à n dimensions, noté C [Lozano-Perez 80]. En planification, on peut donc être amené à manipuler des triplets de coordonnées cartésiennes dans $\mathbb{R}^3$, des situations décrites localement par 6 paramètres ou encore des vecteurs de configuration à n dimensions.

Dans les espaces considérés, on caractérise la composante libre d'obstacle, ou *espace libre*, noté de manière générique E_{libre}. Dans $\mathbb{R}^3$, cette notion est évidente : elle regroupe tous les points de $\mathbb{R}^3$qui n'appartiennent pas aux obstacles. Dans l'espace des configurations, l'ensemble des configurations des chaînes cinématiques qui correspondent aux obstacles est noté $C_{obstacle}$. La composante libre de collision avec les obstacles C_{libre} est alors le complément de $C_{obstacle}$ dans l'espace des configurations. Cette composante dépend donc à la fois de la partie de $\mathbb{R}^3$occupée par les obstacles et de la chaîne cinématique. De manière similaire, lorsque l'on va considérer un corps rigide, l'ensemble des situations libres de collision, va dépendre de la partie de $\mathbb{R}^3$occupée par les obstacles et de la forme du corps rigide lui-même.

Les techniques de planification consistent alors à trouver une trajectoire Γ joignant le point

initial au point final et toute entière contenue dans la composante libre de l'espace considéré. Elles peuvent faire appel à des techniques qui utilisent un modèle de l'espace libre dans $\mathbb{R}^3$, indépendant des objets manipulés mais ne manipulant que des points de $\mathbb{R}^3$, ou bien raisonner dans l'espace des configurations ou l'espace 6D pour prendre en compte la nature de l'objet manipulé ou de la chaîne cinématique.

On décrit maintenant d'une part la représentation numérique des corps rigides et les méthodes de modélisation de l'espace libre.

2.1.2 Représentation numérique des corps rigides

Différents modèles peuvent être mis en œuvre : volumiques ou surfaciques, continus ou discrets. Après une brève introduction des modèles volumiques, nous nous concentrerons particulièrement sur les modélisations surfaciques qui sont les plus utilisées en informatique graphique, et donc en CAO et en RV.

2.1.2.1 Modèles volumiques

En CAO et dans le domaine de la conception ou du *design* mécanique, les représentations volumiques des objets sont principalement obtenues par la technique appelée *géométrie de construction des solides* [Requicha 80] (ou *Constructive Solid Geometry* (CSG)). Celle-ci permet de décrire un volume à l'aide de solides élémentaires (cylindres, pavés, sphères, cônes, tores, etc) assemblés à l'aide d'opérations ensemblistes (par exemple union, intersection, soustraction). Le volume est ainsi modélisé dans un arbre dont les nœuds représentent les opérations, et les feuilles les solides élémentaires. En informatique graphique, la surface délimitant le volume de l'objet est le principal élément utilisé et on fait alors appel aux modèles surfaciques.

2.1.2.2 Modèles surfaciques

En informatique graphique, ce sont les surfaces qui sont décrites. Les objets 3D sont alors définis par leur enveloppe. On trouve différents modèles parmi lesquels ont peut distinguer les modèles polyédriques et différents modèles paramétriques (carreaux de Coons, surfaces de Bézier [Bézier 66], *Basic spline* (B-spline) [Schoenberg 46, De Boor 76] et *Non Uniform Rational B-Spline* (NURBS) [Piegl 97]) associés dans des B-Rep pour former un solide fermé. Les modèles les plus utilisés et qui concernent notre application sont les modèles polyédriques.

Les modèles polyédriques Ils consistent à représenter les surfaces à l'aide de maillages liés à un repère local. La surface est donc discrétisée pour être décrite par un ensemble de facettes planes. La surface est alors représentée par l'ensemble des facettes, arêtes et sommets qui la décrivent. Pour de tels modèles, la précision est étroitement liée à la taille des facettes et donc à la résolution de l'échantillonnage de la surface modélisée. Plus les facettes sont petites, plus l'échantillonnage est dense, plus la résolution est dense et la précision importante. Différentes

formes de facettes polygonales peuvent être utilisées, mais des facettes quadrangulaires (discrétisation selon les directions paramétriques) ou triangulaires (utilisation de la triangulation de Delaunay [Delaunay 34]) sont généralement mises en œuvre.

2.1.3 Modélisation de l'espace libre

Nous avons vu que l'environnement peut être modélisé selon différents points de vue :

1. un espace cartésien, permettant de construire un modèle d'environnement géométrique similaire au nôtre par la représentation des surfaces ou des volumes des obstacles ;
2. un espace des configurations où chaque dimension est liée à un degré de liberté du robot.

Ces deux modèles coexistent généralement en robotique (les capteurs extéroceptifs perçoivent leur environnement dans l'espace cartésien et le contrôle des actionneurs s'effectue dans l'espace des configurations). Différents types de modèles de l'environnement sont possibles selon les informations que l'on souhaite exploiter. La robotique mobile est le principal pourvoyeur de ces modèles pour la navigation et la planification de trajectoire. On peut classer les niveaux d'information présents dans la littérature en trois grandes familles [Dufourd 05] :

1. le niveau géométrique,
2. le niveau topologique,
3. et le niveau sémantique.

Ces différentes informations ne traitent pas forcément de l'ensemble des dimensions de l'espace cartésien ou de l'espace des configurations, mais seulement d'une partie, suffisante pour définir les trajectoires réalisables (par exemple, projection sur le plan horizontal pour les plateformes robotiques mobiles terrestres).

Les cartographies purement topologiques ou sémantiques sont rares. Elles sont généralement combinées et/ou associées à des informations géométriques [Dufourd 05].

2.1.3.1 Modélisation géométrique

Pour synthétiser la représentation de l'espace libre, deux grandes familles de techniques existent : la décomposition de l'espace libre en cellules géométriques, et la synthèse directe de la connectivité de l'espace libre dans des graphes constituant des *roadmaps*. Les différents modèles introduits dans cette section sont illustrés sur l'environnement 2D carré (encombré de 2 obstacles) introduit dans la figure 2.1.

2.1.3.1.1 Décomposition en cellules

Cette technique consiste à diviser l'espace en cellules géométriques élémentaires. Les portions d'espace ainsi définies sont qualifiées comme libres ou occupées (même partiellement) par des obstacles. L'espace libre est alors constitué de l'ensemble des cellules libres. On ne

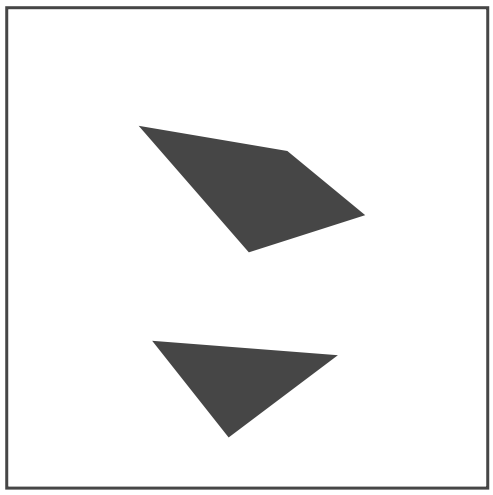


FIGURE 2.1 – Environnement 2D.

considère alors plus un espace continu, mais un ensemble de portions d'espace continu dont on connait la connectivité. On trouve ici une grande variété de décompositions (figure 2.2) :

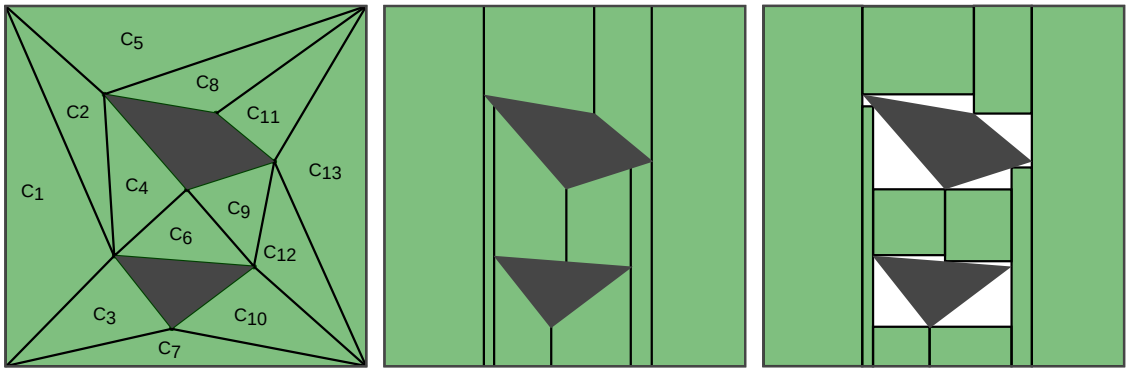
- **cellules exactes :** Ces techniques sont mises en œuvre dans des espaces 2D, et en présence d'obstacles aux géométries simples (polygones). Elles permettent de décrire exactement E_{libre} à travers un ensemble de cellules libres. On trouve alors différentes formes de cellules : cellules triangulaires (figure 2.2(a) générées avec la triangulation de Delaunay à partir des sommets des obstacles [Demyen 06], cellules trapézoïdales (figure 2.2(b)) toujours générées à partir des sommet des obstacles, et dont les côtés parallèles sont alignés avec une des deux dimensions de l'espace considéré [Preparatat 85].

- **cellules rectangulaires :** toujours pour des environnements 2D, la décomposition en cellules rectangulaires (figure 2.2(c)) consiste à diviser l'espace en cellules rectangulaires dont les frontières sont alignées avec les dimensions paramétriques de l'espace et sont positionnées tangentiellement aux obstacles de l'environnement (elles passent par les sommets des obstacles) [Brooks 82b]. La décomposition ainsi obtenue produit une approximation de E_{libre} puisque les obstacles ne sont pas, eux, forcément alignés avec les axes des dimensions de l'espace décomposé.

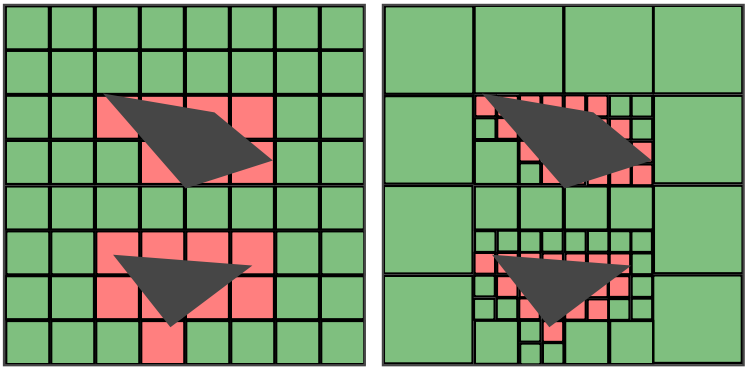
- **cellules régulières :** l'objectif est ici de construire un pavage de l'espace à l'aide de cellules de même forme (figure 2.2(d)). Chacune des cellules est ensuite analysée pour déterminer si elle est libre ou occupée (entièrement ou partiellement). Des exemples de pavages réguliers existent pour des espaces 2D [Samet 84]. Pour les espaces ayant un plus grand nombre de dimensions, les pavages sont moins originaux, et on trouve des cellules cubiques (3D) ou hypercubiques.

- **arbres déséquilibrés :** ces arbres représentent une structure de données facilitant l'exploration. Le pavage est ici réalisé d'abord avec une faible résolution. La résolution de la décomposition est ensuite affinée en décomposant les cellules partiellement occupées à l'aide de cellules de même forme, mais plus petites. En 2D, même si différents pavages

infiniment décomposables sont possibles, un pavage carré (quadtree) (figure 2.2(e)) est généralement utilisé [Samet 84]. [Meagher 82] a proposé l'utilisation de tels arbres en 3D composés de cellules cubiques (octree) qui nécessitent que les bornes de l'espace décomposé forment un cube aligné avec les axes du repère monde. La généralisation de cette idée à un espace n dimensions (nD) y est aussi introduite en précisant que, pour que la décomposition soit possible, l'espace doit disposer d'une métrique permettant de définir des arêtes de l'hypercube de même longueur sur chacune des dimensions.



(a) Décomposition par triangulation. (b) Cellules trapézoïdales. (c) Cellules rectangulaires.



(d) Cellules régulières carrées. (e) Arbre déséquilibré (quadtree).

FIGURE 2.2 – Décompositions en cellules d'environnement 2D.

2.1.3.1.2 Définition de *roadmaps*

La définition de *roadmaps* consiste à échantillonner l'espace à modéliser avec des points d'intérêt, et à interconnecter ces points particuliers afin de synthétiser la connectivité de l'espace considéré. Pour cela, on distingue trois grandes familles de *roadmaps* (figure 2.3) :

- **diagramme de Voronoï :** Le diagramme de Voronoï constitue le diagramme maximisant la distance aux obstacles (figure 2.3(a)). L'utilisation de représentations similaires est très ancienne (utilisation de structures naturelles semblables par Descartes dans *Le monde de monsieur Descartes, ou Le Trait de la Lumière* [Descartes 64]). Cependant, la formalisation mathématique de ces représentations ne date que de la deuxième moitié du XIXe

siècle pour les problèmes à 2 et 3 dimensions, et du début du XXe siècle pour la généralisation à *n* dimensions par Georges Voronoï [Voronoï 08]. Plus tard, [Aurenhammer 91] propose un état de l'art sur cette structure de données et ses utilisations. Il définit à cette occasion les propriétés mathématiques des diagrammes de Voronoï. Des algorithmes de construction des diagrammes de Voronoï y sont aussi proposés. Le temps d'exécution de ces algorithmes est étudié, ce qui permet de conclure qu'il croît exponentiellement avec le nombre de dimensions du problème à traiter. Les diagrammes de Voronoï ont été introduits en robotique à la fin des années 1970 pour la perception, la navigation et la recherche de chemin [Sowat 79].

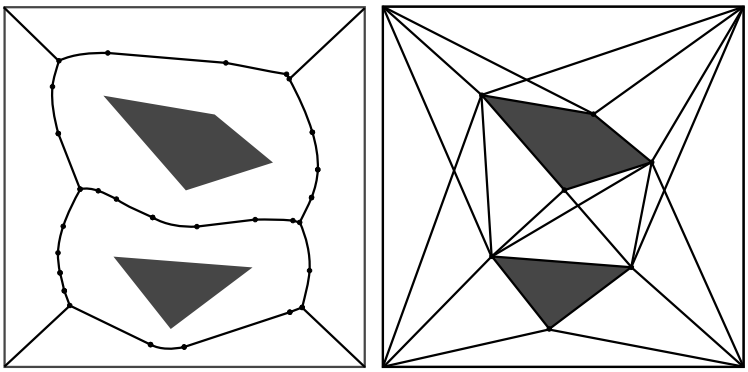
- **graphes de visibilité :** de tels graphes connectent tous les sommets des obstacles (polygonaux en 2D, polyédriques en nD) qui peuvent être joints deux à deux par un segment sans intersection avec les obstacles (figure 2.3(b)). Pour cela, un graphe constitué de nœuds (tous les sommets des obstacles de l'environnement) et d'arcs (connexion entre les sommets visibles deux à deux) est construit [Alt 88]. Un graphe de visibilité réduit peut être obtenu par la triangulation de Delaunay [Delaunay 34] qui permet de diminuer le nombre d'arcs tout en garantissant une régularité optimale du maillage (figure 2.3(c)). On obtient alors un graphe dual du diagramme de Voronoï.

- ***Probabilistic RoadMap* (PRM) (figure 2.3(d)) :** enfin, pour remédier aux problèmes de temps de calcul limitant la capacité des algorithmes classiques à cartographier l'espace des configurations lorsque celui-ci présente un grand nombre de dimensions, une approche probabiliste a été proposée [Overmars 92]. Cette approche consiste à définir aléatoirement un ensemble de points dans l'espace des configurations, et à les interconnecter pour constituer une *roadmap* probabiliste. Le postulat est qu'avec un nombre d'échantillons aléatoires suffisant, l'ensemble de la connectivité de l'environnement sera saisie. Le problème est donc de définir ce nombre d'échantillons pour éviter un suréchantillonnage de l'espace des configurations qui serait trop coûteux en temps de calcul.

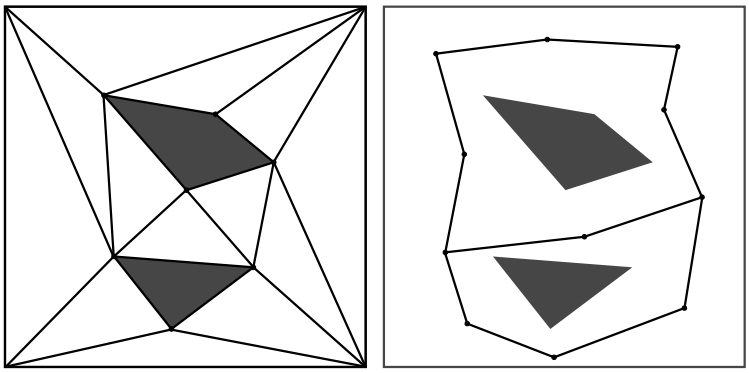
En résumé, une *roadmap* correspond à un graphe dont les nœuds sont les points d'intérêt de l'espace et où les arcs véhiculent une information métrique. Cette notion annonce celle, plus générale, de graphe topologique.

2.1.3.2 Modélisation topologique

Les modèles topologiques utilisés en robotique mobile visent généralement à synthétiser la connectivité de l'environnement. Pour cela, des graphes sont généralement utilisés même si d'autres outils peuvent être mis en œuvre conjointement ou non (modèles de réseaux ou hiérarchiques). Le graphe est donc l'outil commun pour les modèles topologiques, mais les nœuds (représentant les *lieux*) comme les arcs (représentant la connectivité entre les lieux) qui le constituent peuvent être de différentes natures. Dans un premier temps, nous allons voir les différentes méthodes utilisées pour déterminer les lieux et donc les différentes natures de lieux



(a) Diagramme de Voronoï. (b) Graphe de visibilité.



(c) Graphe de visibilité réduit. (d) PRM.

FIGURE 2.3 – *Rodmaps* d'environnement 2D.

identifiés. Ensuite, nous détaillerons les informations qui peuvent être associées aux arcs et aux nœuds du graphe topologique et qui viennent ainsi enrichir le modèle.

2.1.3.2.1 Notion de lieu

La notion de lieu recouvre une large variété de concepts. Les lieux identifiés dans les modélisations topologiques sont liés à la manière de les identifier. On trouve dans la littérature une classification de lieux selon leur mode d'identification [Filliat 03] :

- Les lieux à identifier peuvent être définis par l'homme. Dans ce cas, le robot est doté de procédures lui permettant d'identifier les lieux préalablement définis [Kunz 99, Dedeoglu 99].

- Les lieux peuvent être définis canoniquement. Dans ce cas, le robot n'est plus doté d'une description complète des différents lieux à identifier, mais plutôt d'une description des situations dans lesquelles un lieux doit être identifié. Il revient alors au robot de définir les lieux qu'il identifie [Kuipers 91, Kortenkamp 94].

- Les lieux peuvent encore être définis automatiquement. Dans ce cas, on suppose que la situation sensorielle du robot est relativement constante, et un nouveau lieu est identifié lorsque sa variation dépasse un seuil donné [Kurz 95, Nehmzow 00].

2.1.3.2.2 Information associée

Dans les cartographies purement topologiques de robotique mobile, les lieux identifiés sont liés à des positions dans l'espace. L'information associée aux nœuds du graphe topologique (représentant les lieux) vise à localiser le robot dans l'environnement. On utilise alors, pour chaque nœud, une description de la situation sensorielle du robot (état capteur en général, ou point de vue lorsque la vision pour la robotique est utilisée). Par contre, même si les arcs des graphes topologiques représentent en général la connectivité des lieux identifiés, ils peuvent aussi recouvrir d'autres notions comme la visibilité [Taylor 98]. Un degré de confiance peut aussi être associé aux arcs pour gérer les changements dans l'environnement (déplacement d'obstacles, fermeture de portes) [Yamauchi 96]. Enfin, comme cela a été abordé plus tôt, les cartographies topologiques sont souvent enrichies de données métriques comme la distance entre les nœuds pour lever des ambiguïtés (différencier deux lieux similaires d'un point de vue sensoriel), ou faciliter l'exploitation du graphe (recherche du plus court chemin) [Dufourd 05].

2.1.3.3 Modélisation sémantique

La modélisation sémantique a été utilisée principalement pour la robotique mobile terrestre. On parle d'attributs sémantiques et, dans certains cas, de carte sémantique. Elle est utilisée pour permettre au robot de raisonner sur des concepts symboliques. Elle consiste donc à associer aux lieux des informations permettant au robot :

- d'interagir avec un humain de manière quasi-naturelle : notification des opérations en cours et planifiées ("Je vais à la salle de bain puis, j'irai au bureau"), interprétation de requêtes ("Amène-moi au salon"),
- de classer les lieux en fonction des éléments qui les composent,
- de traiter les ambiguïtés,
- de détecter les erreurs de localisation.

Cette modélisation est systématiquement associée à un modèle topologique et à un ancrage géométrique permettant de localiser les informations sémantiques. On peut aussi voir, dans certains travaux, une démarche incrémentale entre la modélisation topologique et la modélisation sémantique. Le modèle sémantique vient alors comme une sur-couche du modèle topologique. Les lieux définis au niveau sémantique agrègent dans ce cas plusieurs lieux définis au niveau topologique [Pronobis 10, Pronobis 11]. Parmi les lieux définis au niveau topologique, ceux permettant de changer de lieu sémantique sont alors identifiés comme des "lieux de passage".

Plus généralement, les lieux identifiés dans les modèles sémantiques ne correspondent plus à des positions dans l'espace, mais à des portions de l'espace. Par exemple, dans un environnement intérieur, l'information sémantique distingue les différentes pièces d'un bâtiment pour permettre au robot de s'y localiser [Ulrich 00]. Dans cette application, les lieux identifiés sont interconnectés dans un graphe (topologique) en fonction de leur connectivité. Cette cartographie

sans métrique est faite à la main au préalable de la navigation (le modèle de l'environnement est alors connu *a priori*).

L'amélioration des techniques de vision au cours des dernières années a permis de construire ces modèles en ligne en associant aux différents lieux une information sémantique déduite des éléments qui le composent [Viswanathan 11].

Les lieux identifiés se rapprochent généralement de ceux définis par [Ulrich 00] : on utilise une décomposition naturelle (couloir, hall, pièce avec leur fonction,...). Ces lieux sont identifiés en ligne [Torralba 03, Vogel 04] par la reconnaissance des objets qu'ils contiennent (une pièce avec une douche est une salle d'eau, avec un ordinateur, un bureau,...). Leurs liens avec des modèles topologiques et géométriques les enrichissent et facilitent la navigation et la planification de trajectoires [Galindo 05] tout en permettant aux humains d'interagir plus naturellement avec le robot.

2.1.4 Synthèse

Nous avons pu voir, dans cet état de l'art, qu'il existe différentes manières de modéliser un environnement (espace cartésien ou des configurations) et ses obstacles (modèles volumiques ou surfaciques, paramétriques ou discrets). Une fois les corps rigides de l'environnement modélisés, son espace libre peut lui aussi être modélisé afin de synthétiser l'information utile à une application visée. On trouve principalement trois niveaux d'information utilisés en robotique pour la navigation et la planification de trajectoire :

1. le niveau géométrique synthétisant les données nécessaires à la navigation,
2. le niveau topologique permettant la localisation et la planification en synthétisant la connectivité de l'environnement,
3. le niveau sémantique permettant principalement l'identification et la désignation de lieux pour l'interaction avec un humain.

Cet état de l'art nous a permis de constater que bien qu'il existe des cartographies sémantiques des environnements, celles-ci ne sont pas utilisées pour la planification de trajectoire mais pour la résolution des ambiguïtés, la détection des erreurs et l'interaction avec l'humain (langage naturel pour la désignation de portions de l'espace). Nous allons voir, dans la section suivante, le modèle de l'environnement multi-niveaux que nous proposons afin d'exploiter conjointement l'information sémantique, topologique et géométrique dans le processus de planification de trajectoire et l'interaction de ce dernier avec un opérateur humain.

2.2 Contribution

Dans nos travaux, nous souhaitons améliorer les performances de la planification automatique de trajectoires, et de l'intégration d'un opérateur humain dans le processus de planification, pour la simulation interactive de tâches de manipulation réalisées par un opérateur humain.

L'état de l'art de ces domaines montre que la quasi-totalité des travaux manipule des modèles de l'environnement purement géométriques. Pourtant, un opérateur humain réalisant une manipulation dans son environnement perçoit celui-ci à un plus haut niveau d'abstraction que sa seule géométrie. Nous nous sommes inspirés des travaux dans le domaine de la robotique mobile référencés à la section 2.1 pour proposer un modèle de l'environnement combinant des données sémantiques, topologiques et géométriques dans un même modèle. L'originalité de notre démarche repose sur :

- le modèle proposé et son architecture ;
- la mise en œuvre d'un modèle contenant ce type d'informations dans le cadre de la planification interactive et de la RV.

> Dans la suite du document, les concepts notés `de cette manière` représentent des notions dont nous donnons notre définition pour notre contribution. Ces concepts font l'objet du développement d'une classe dédiée dans l'architecture objet décrite dans l'implémentation de ces travaux.

La section section 2.2.1 présente la modélisation des `Corps rigides` de l'environnement. La section section 2.2.2 présente le modèle de l'`Espace libre`.

2.2.1 Modélisation des `Corps rigides`

2.2.1.1 Niveau géométrique

Pour le niveau géométrique du modèle des `Corps rigides` de l'environnement, nous avons décidé d'utiliser la représentation classiquement utilisée en infographie et en RV. Nous utilisons des maillages polyédriques de la surface de ces corps. Chaque maillage est exprimé dans un repère local propre à l'objet décrit. Ce repère local situe (position et orientation) le maillage dans l'espace cartésien, à l'aide par exemple de matrices de transformation homogène.

Bien que les géométries que nous exploitons soient issues de logiciels de CAO, les traitements qu'elles subissent pour être mises en œuvre dans les simulations immersives et interactives en font perdre les modèles analytiques (B-Rep et arbre de construction). Ainsi, exploiter des modèles analytiques des objets 3D demande soit de reconstruire ces modèles à partir des maillages 3D (ce qui implique un temps de développement non négligeable), soit d'accéder à ces données dans les fichiers CAO (ce qui implique de s'adapter aux formats propriétaires utilisés). Nous avons donc choisi de profiter de la disponibilité naturelle des maillages des surfaces des objets 3D dans les simulations de RV et ainsi d'opter pour le modèle polyédrique pour modéliser les environnements 3D.

2.2.1.2 Niveau sémantique

Le niveau sémantique de notre modèle est constitué d'`Attributs` attachés aux `Corps rigides`, tels que leur forme géométrique (information textuelle) ou la qualification de corps "fixe" ou "mobile" (information binaire). Dans les applications que nous visons, les `Corps rigides` de l'environnement peuvent être fixes (obstacles fixes dans l'environnement), ou mobiles (objet manipulé, obstacles mobiles).

En particulier, la prise en compte de la mobilité des obstacles permettra d'évoluer à terme vers la simulation de tâches de manipulation plus complexes, dans lesquelles il est nécessaire de déplacer des obstacles mobiles dans l'environnement pour permettre la réalisation de la tâche.

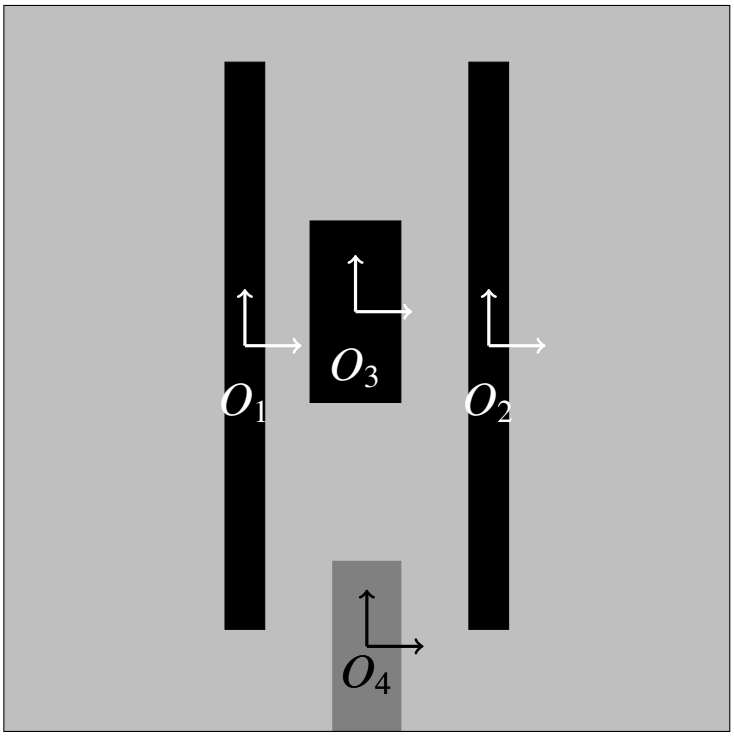


FIGURE 2.4 – Exemples de `Corps rigides` d'un environnement 2D.

La figure 2.4 montre un environnement 2D encombré de quatre obstacles (O_1 à O_4). Trois de ceux-ci sont fixes (O_1 à O_3), et un (O_4) est mobile.

2.2.2 Modélisation de l'`Espace libre`

2.2.2.1 Choix préalables de représentation

Notons que l'application visée (la manipulation d'objets virtuels en RV pour l'assemblage de produits industriels) concerne des corps rigides. On sera donc amené à manipuler l'information géométrique dans $\mathbb{R}^3$ et également dans l'*espace 6D*. Pour l'application visée, les objets manipulés (les différents composants d'un système industriel) ayant des géométries différentes, la composante libre de l'*espace 6D* est à chaque fois différente alors que la composante libre pour un point de $\mathbb{R}^3$ est indépendante de ceux-ci.

Le modèle de l'espace libre retenu concerne $\mathbb{R}^3$: celui-ci est peu couteux en temps de calcul et peut être exploité pour l'ensemble des objets manipulés. La définition des trajectoires dans l'espace 6 dimensions (6D) sera ensuite réalisée avec des techniques d'exploration locales [Cailhol 13].

De plus, pour faciliter la mise à jour de cette modélisation de l'espace libre, et donc sa réutilisation, les Corps rigides fixes et mobiles y sont traités de manière distincte. Notamment, les Lieux identifiés font abstraction des Corps rigides mobiles. On définit donc une partie statique de E_{libre} comme l'espace libre faisant abstraction des Corps rigides mobiles.

L'architecture de modélisation proposée comporte trois niveaux :

- Un niveau sémantique (section 2.2.2.2), constitué d'informations textuelles ou numériques utiles à la planification de trajectoire interactive. Ces informations sont attachées aux lieux du niveau topologique (voir ci-après), définissant par exemple leur difficulté à être traversés, leur occupation ou non par des obstacles mobiles, ou leur forme géométrique.

- Un niveau topologique (section 2.2.2.3), qui représente les Lieux et leurs Frontieres, et les relations topologiques (notamment la connexité) entre eux (une Frontiere connecte deux Lieux). Ce niveau topologique se ramène à un Graphe topologique modélisant la connectivité de l'espace libre. Les nœuds de ce graphe représentent les Frontieres, et les arcs sont associés aux Lieux.

- Un niveau géométrique (section 2.2.2.4) modélise la composante de l'espace cartésien à l'aide d'un arbre déséquilibré (octree) qui permet de distinguer des Cellules géométriques libres et des Cellules géométriques occupées complètement ou partiellement par des Corps rigides fixes ou mobiles.

Cette modélisation originale de l'espace libre, permettra un contrôle accru de la planification de trajectoire interactive en RV. Ce contrôle accru permettra par exemple de compenser grâce à l'Information sémantique l'utilisation d'un modèle de l'espace 3D au lieu de celui de l'espace 6D.

2.2.2.2 Niveau sémantique

Le niveau sémantique (section 2.2.2.2) de notre modèle est constitué d'informations textuelles ou numériques utiles à la planification de trajectoire interactive. Ces informations sont des Attributs attachés aux lieux du niveau topologique. Un lieu donné est caractérisé par des Informations sémantiques telles que la complexité à le traverser (information numérique), son encombrement, lié à la présence ou non de obstacles mobiles (information binaire), ou sa forme (information textuelle).

La figure 2.5 illustre la sémantique des Lieux de l'espace libre de l'environnement donné dans la figure 2.4. Les Lieux sont associés à des couleurs liées à la sémantique qui leur est attachée. Les couleurs vert, orange et rouge correspondent à trois niveaux croissants de difficulté

de traversée (complexité faible, complexité moyenne, complexité forte respectivement), et les Lieux encombrés d'un obstacle mobile sont associés à la couleur jaune.

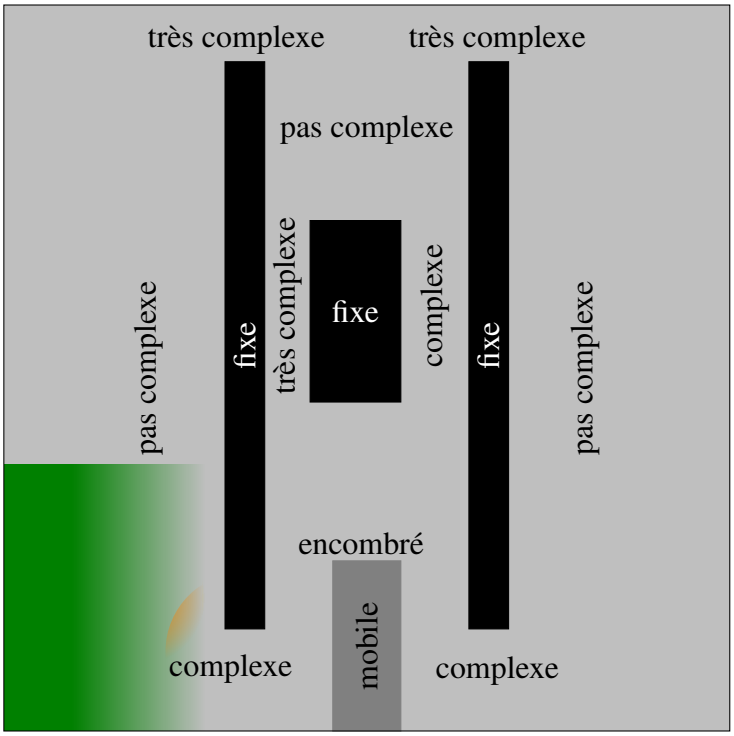


FIGURE 2.5 – Lieux du modèle sémantique.

2.2.2.3 Niveau topologique

Nous avons choisi de modéliser la connectivité des différents Lieux L_i et L_j grâce à leurs Frontieres $F_{i,j}$. Les Frontieres et les Lieux sont des ensembles fermés bornés de $\mathbb{R}^3$, la Frontiere $F_{i,j}$ connecte les Lieux L_i et L_j. L'ensemble des Frontieres et des Lieux modélise donc l'intégralité de la connectivité de l'espace libre. La figure 2.6 représente les Lieux L_i et leur sémantique, et les Frontieres $F_{i,j}$ définis pour notre cas d'étude.

Nous avons choisi d'utiliser un Graphe topologique dans lequel les nœuds représentent les Frontieres, et les arcs les Lieux de l'espace libre. La définition d'un tel Graphe topologique nous permet d'associer à la traversée d'un Lieu (d'une Frontiere à une autre) un arc unique.

Notre Graphe topologique modélise les lieux et frontières correspondant à la partie statique de l'espace libre. Le Graphe topologique obtenu pour l'environnement de la figure 2.4 est illustré dans la figure 2.7.

2.2.2.4 Niveau géométrique

La cartographie retenue pour l'espace 3D est une décomposition en Cellules géométriques à l'aide d'un arbre déséquilibré (octree). La profondeur de l'octree (nombre de niveaux de décomposition) peut être adaptée à l'application visée.

Cette décomposition est illustrée dans la figure 2.8 à l'aide d'un quadtree réalisé sur l'environnement 2D illustré dans la figure 2.4.

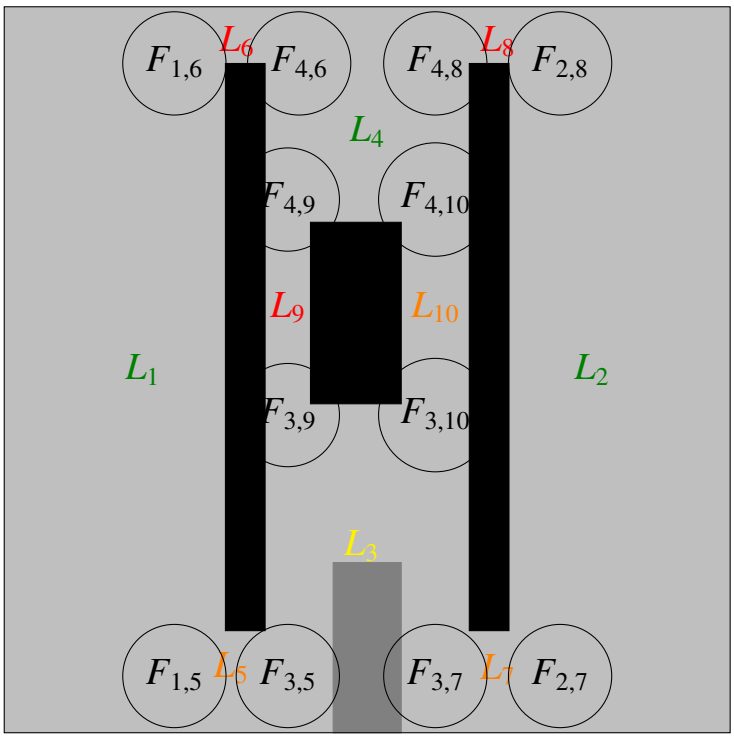


FIGURE 2.6 – Représentation des Lieux.

On peut voir dans la figure 2.8 les Cellules géométriques vertes constituant l'espace libre. Les Cellules géométriques occupées de manière permanente (par des Corps rigides fixes) apparaissent en rouge, et les cellules occupées par des Corps rigides mobiles apparaissent en orange. Cette distinction permet de différencier la partie statique de la décomposition octree (en rouge) de la partie dynamique (en orange et vert). Cette différenciation des Cellules géométriques occupées permet de disposer simultanément d'une décomposition de l'espace libre statique, et d'une décomposition dynamique qui est localement mise à jour avec les déplacements des objets mobiles (section 2.3.2.1).

2.2.3 Synthèse du modèle multi-niveaux de l'environnement proposé

Le modèle de l'environnement proposé se décompose donc en deux parties : d'une part un modèle multi-niveaux des Corps rigides (obstacles fixes ou mobiles et objet manipulé), et d'autre part un modèle multi-niveaux de l'espace libre.

Le modèle des Corps rigides comporte deux niveaux :

un niveau sémantique constitué d'informations textuelles ou numériques caractérisant chaque corps rigide (mobile ou fixe, forme géométrique, ...).

un niveau géométrique représentant les surfaces des Corps rigides sous forme de maillages géométriques.

Le modèle de l'espace libre comporte trois niveaux :

un niveau sémantique caractérisant les Lieux de l'environnement (difficulté de traversée, encombrement par un (plusieurs) Corps rigide(s) mobile(s), forme géométrique...)

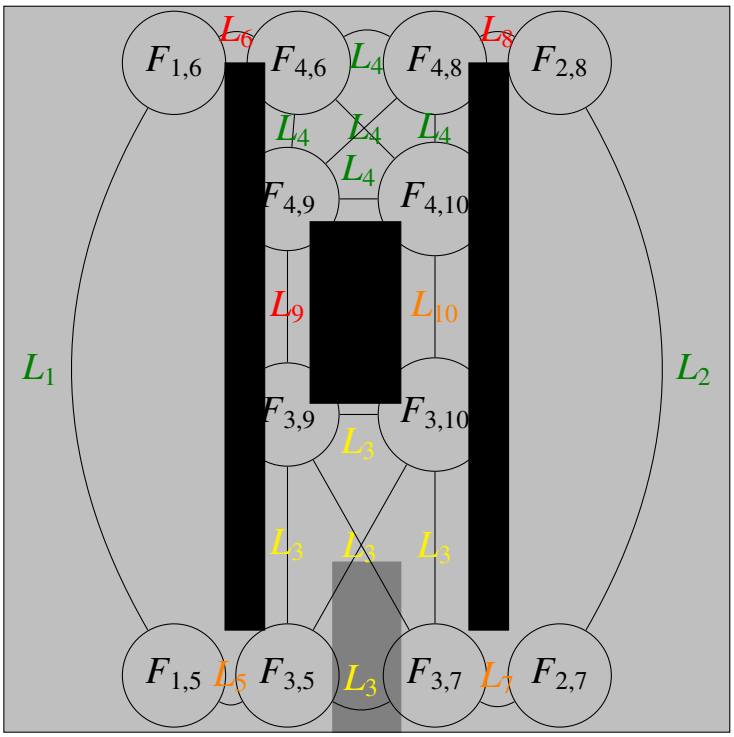


FIGURE 2.7 – Représentation des `Lieux`.

un niveau topologique saisissant la connectivité des `Lieux` par l'intermédiaire de leurs `Frontieres` (synthétisée dans un `Graphe topologique`)

un niveau géométrique constitué d'un octree permettant d'isoler la partie statique de l'espace libre.

Alors que la majeure partie des travaux en planification de trajectoires ou en simulation interactive et immersive en RV de tâches réalisées par un opérateur humain reposent sur l'utilisation de modèles purement géométriques, l'approche originale de modélisation que nous proposons permet l'introduction d'informations (topologiques et sémantiques) d'un plus haut niveau d'abstraction. Ces informations permettront un meilleur contrôle de la planification de trajectoires pour une meilleure efficacité (chapitre 3). Elles permettront aussi d'améliorer l'intégration d'un opérateur humain dans la planification de trajectoire (chapitre 4).

2.3 Implémentation du modèle de l'environnement

Cette section décrit l'implémentation du modèle de l'environnement proposé, qui a fait l'objet du développement d'une librairie dédiée. Un modèle objet statique de l'architecture développée est tout d'abord décrit (section 2.3.1). Ensuite, les algorithmes et processus de construction de la modélisation de l'environnement sont donnés (section 2.3.2). À cette occasion, deux processus clés (définition de l'octree et des `Lieux`) sont détaillés.

Le modèle de l'environnement proposé a été développé en C++ et met en œuvre différentes bibliothèques libres (*Flexible Collision Library* (FCL) [FCL 14] pour les modèles polyédriques

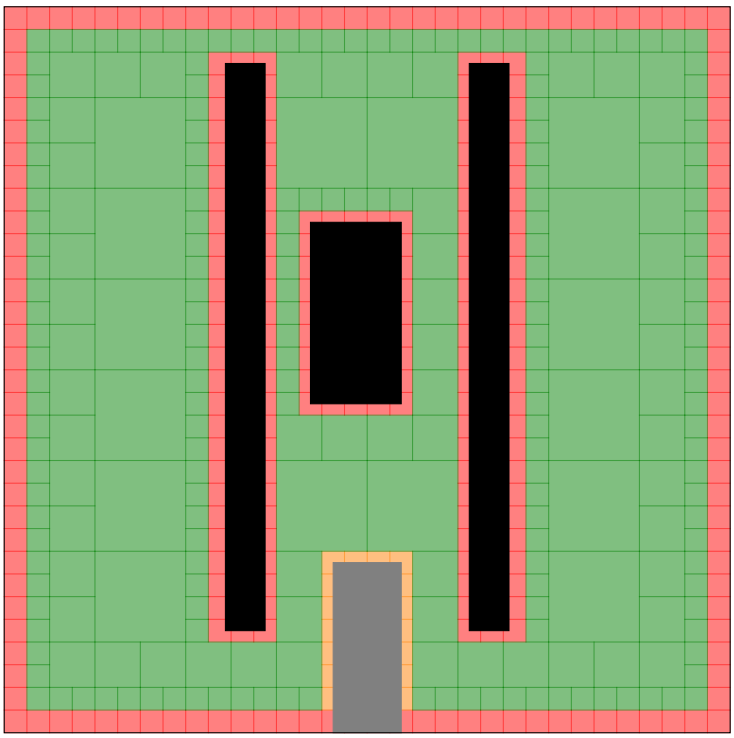


FIGURE 2.8 – Représentation de l'Espace libre.

des différents corps rigides, Open Multi Processing (OpenMP) [omp 14] pour le *multi-threading* et Eigen [Eig 14] pour la représentation matricielle du Graphe topologique).

2.3.1 Modèle statique : classes et responsabilités

Le modèle multi-niveaux de l'environnement proposé est illustré par le modèle de domaine *Unified Modeling Language* (UML) [uml 14] de la figure 2.9. Elle reprend les différents concepts introduits dans toute la section 2.2.

2.3.1.1 Vue générale

Les classes en jaune constituent le niveau géométrique du modèle de l'environnement. Celui-ci est d'abord constitué de Corps rigides modélisant les différents objets 3D de la scène. La définition des modèles polyédriques des Corps rigides a été réalisée à l'aide de la librairie FCL. L'Espace libre est cartographié grâce à une décomposition en Cellules géométriques. Pour cela, la décomposition octree définie par Ladevèze [Ladevèze 10a] a été reprise et modifiée afin de prendre en compte l'aspect dynamique de l'environnement (distinction des Cellules géométriques occupées de manière permanente de celles occupées par des Corps rigides mobiles).

La classe Zone agrège plusieurs Cellules géométriques connexes et définit ainsi un ensemble borné fermé de l'espace auquel peut être associé un ensemble de propriétés. La Zone constitue donc la classe permettant l'ancrage géométrique des Lieux et des Frontieres en permettant d'associer, à chacune des instances de ces classes, une portion d'Espace libre définie comme un ensemble de Cellules géométriques connexes.

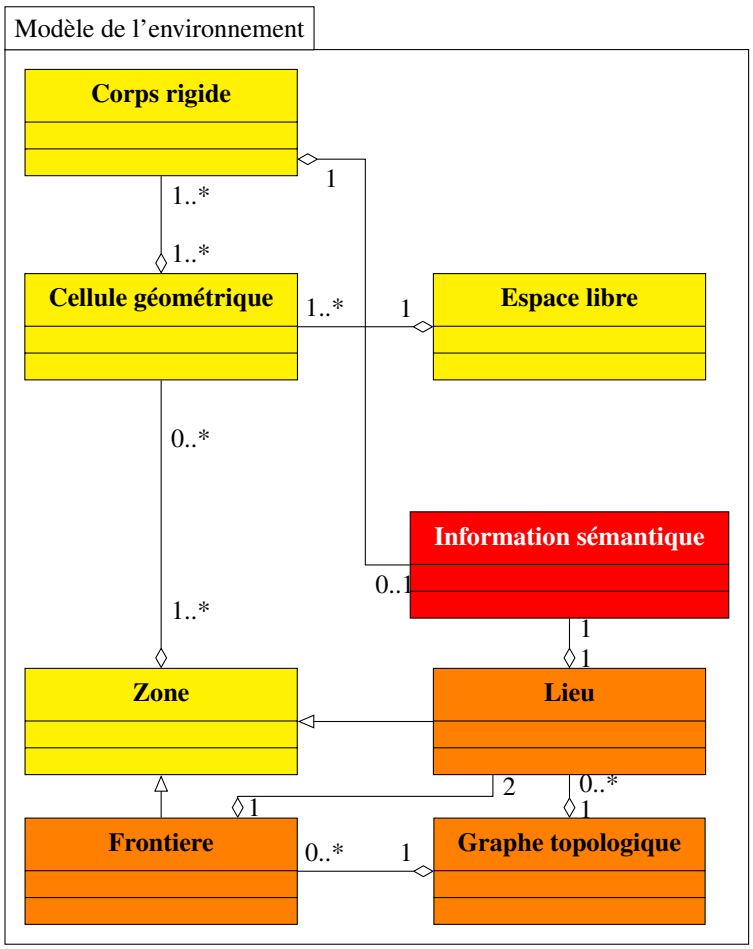


FIGURE 2.9 – Modèle de domaine UML de l'environnment.

Ainsi, la classe Lieu est dérivée de la classe Zone. Les Lieux constituent en effet une portion de l'espace à laquelle est attachée l'Information sémantique (section 2.3.1.3 pour les détails d'implémentation de l'Information sémantique).

La classe Zone est aussi dérivée pour former la classe Frontiere qui représente l'espace de transition entre deux Lieux. Ainsi, les Frontieres sont des Zones enrichies de références aux Lieux qu'elles lient.

L'ensemble des Lieux et Frontieres est agrégé au sein du Graphe topologique. L'implémentation choisie pour les Lieux et le Graphe topologique est détaillée dans la section 2.3.1.2.

2.3.1.2 Graphe topologique

La connectivité entre les Frontieres dans le Graphe topologique est exprimée à l'aide d'une représentation matricielle mettant en œuvre la librairie Eigen. Cette représentation matricielle comprend une matrice de connectivité C (donnée dans l'équation 2.1 pour l'environnement 2D illustré dans la figure 2.7), une matrice de distance D (donnée dans l'équation 2.2 où la distance $d_{i_{j,k}}$ désigne la distance dans L_i entre $F_{i,j}$ et $F_{i,k}$) et enfin une table L liant chaque arc au lieu concerné (donnée dans l'équation 2.3). Le calcul des distances agrégées dans la matrice D est illustré dans la figure 2.10 pour le Lieu L_4. Ces distances correspondent à la distance entre les centres des frontières et sont notées $d_{i_{j,k}}$ pour la distance à parcourir sur le Lieu i entre les

Lieux j et k.

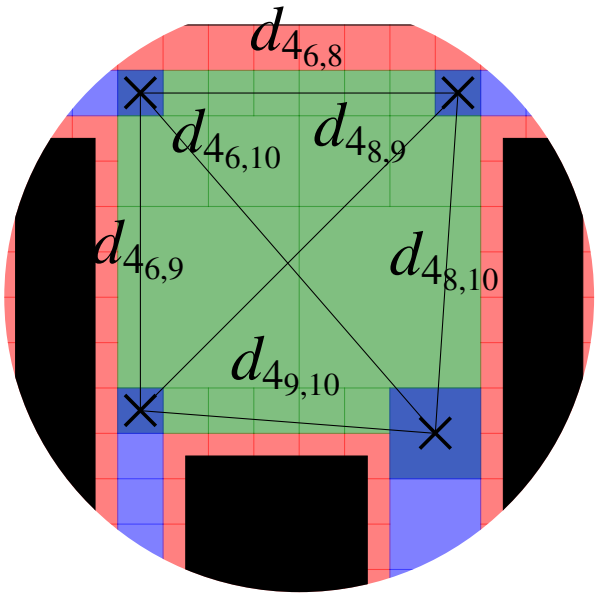


FIGURE 2.10 – Calcul des distances de la matrice D (pour le Lieu L_4).

Ainsi, les trois matrices permettent de saisir la connectivité de l'environnement, sa métrique, et sa topologie au sein d'un graphe unique. On obtient donc pour l'environnement donné en exemple (figure 2.7), les matrices C (équation 2.1), D (équation 2.2) et la table L (équation 2.3) :

$$C = \begin{array}{c|cccccccccccc} & F_{1,5} & F_{1,6} & F_{2,7} & F_{2,8} & F_{3,5} & F_{3,7} & F_{3,9} & F_{3,10} & F_{4,6} & F_{4,8} & F_{4,9} & F_{4,10} \\ \hline F_{1,5} & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ F_{1,6} & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ F_{2,7} & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ F_{2,8} & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ F_{3,5} & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ F_{3,7} & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ F_{3,9} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ F_{3,10} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ F_{4,6} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ F_{4,8} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ F_{4,9} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ F_{4,10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array} \quad (2.1)$$

$$
D = \begin{array}{c|cccccccccccc}
 & F_{1,5} & F_{1,6} & F_{2,7} & F_{2,8} & F_{3,5} & F_{3,7} & F_{3,9} & F_{3,10} & F_{4,6} & F_{4,8} & F_{4,9} & F_{4,10} \\
\hline
F_{1,5} & 0 & d_{1_{5,6}} & 0 & 0 & d_{5_{1,3}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
F_{1,6} & d_{1_{5,6}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{6_{1,4}} & 0 & 0 & 0 \\
F_{2,7} & 0 & 0 & 0 & d_{2_{7,8}} & 0 & d_{7_{2,3}} & 0 & 0 & 0 & 0 & 0 & 0 \\
F_{2,8} & 0 & 0 & d_{2_{7,8}} & 0 & 0 & 0 & 0 & 0 & 0 & d_{8_{2,4}} & 0 & 0 \\
F_{3,5} & d_{5_{1,3}} & 0 & 0 & 0 & 0 & d_{3_{5,7}} & d_{3_{5,9}} & d_{3_{5,10}} & 0 & 0 & 0 & 0 \\
F_{3,7} & 0 & 0 & d_{7_{2,3}} & 0 & d_{3_{5,7}} & 0 & d_{3_{7,9}} & d_{3_{7,10}} & 0 & 0 & 0 & 0 \\
F_{3,9} & 0 & 0 & 0 & 0 & d_{3_{5,9}} & d_{3_{7,9}} & 0 & d_{3_{9,10}} & 0 & 0 & d_{9_{3,4}} & 0 \\
F_{3,10} & 0 & 0 & 0 & 0 & d_{3_{5,10}} & d_{3_{7,10}} & d_{3_{9,10}} & 0 & 0 & 0 & 0 & d_{10_{3,4}} \\
F_{4,6} & 0 & d_{6_{1,4}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{4_{6,8}} & d_{4_{6,9}} & d_{4_{6,10}} \\
F_{4,8} & 0 & 0 & 0 & d_{8_{2,4}} & 0 & 0 & 0 & 0 & d_{4_{6,8}} & 0 & d_{4_{8,9}} & d_{4_{8,10}} \\
F_{4,9} & 0 & 0 & 0 & 0 & 0 & 0 & d_{9_{3,4}} & 0 & d_{4_{6,9}} & d_{4_{8,9}} & 0 & d_{4_{9,10}} \\
F_{4,10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_{10_{3,4}} & d_{4_{6,10}} & d_{4_{8,10}} & d_{4_{9,10}} & 0
\end{array} \tag{2.2}
$$

$$
L = \begin{array}{c|cccccccccccc}
 & F_{1,5} & F_{1,6} & F_{2,7} & F_{2,8} & F_{3,5} & F_{3,7} & F_{3,9} & F_{3,10} & F_{4,6} & F_{4,8} & F_{4,9} & F_{4,10} \\
\hline
F_{1,5} & 0 & L_1 & 0 & 0 & L_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
F_{1,6} & L_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & L_6 & 0 & 0 & 0 \\
F_{2,7} & 0 & 0 & 0 & L_2 & 0 & L_7 & 0 & 0 & 0 & 0 & 0 & 0 \\
F_{2,8} & 0 & 0 & L_2 & 0 & 0 & 0 & 0 & 0 & 0 & L_8 & 0 & 0 \\
F_{3,5} & L_5 & 0 & 0 & 0 & 0 & L_3 & L_3 & L_3 & 0 & 0 & 0 & 0 \\
F_{3,7} & 0 & 0 & L_7 & 0 & L_3 & 0 & L_3 & L_3 & 0 & 0 & 0 & 0 \\
F_{3,9} & 0 & 0 & 0 & 0 & L_3 & L_3 & 0 & L_3 & 0 & 0 & L_9 & 0 \\
F_{3,10} & 0 & 0 & 0 & 0 & L_3 & L_3 & L_3 & 0 & 0 & 0 & 0 & L_{10} \\
F_{4,6} & 0 & L_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & L_4 & L_4 & L_4 \\
F_{4,8} & 0 & 0 & 0 & L_8 & 0 & 0 & 0 & 0 & L_4 & 0 & L_4 & L_4 \\
F_{4,9} & 0 & 0 & 0 & 0 & 0 & 0 & L_9 & 0 & L_4 & L_4 & 0 & L_4 \\
F_{4,10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & L_{10} & L_4 & L_4 & L_4 & 0
\end{array} \tag{2.3}
$$

2.3.1.3 Information sémantique

L'Information sémantique associée au Lieux et aux Corps rigides est définie comme un ensemble d'Attributs. Un Attribut est défini par un type et une valeur. On dispose par exemple d'Attributs dont la valeur est booléenne pour spécifier si un Corps rigide est mobile ou non, ou pour spécifier si un Lieu est encombré ou non par un Corps rigide mobile ; mais aussi d'Attributs de valeur entière pour spécifier le niveau de complexité des Lieux (1 pour une complexité faible, 2 pour une complexité moyenne, et 3 pour une complexité forte). Ainsi, la table 2.1 résume l'Information sémantique associée aux différents Lieux de l'environnement donné en exemple dans la figure 2.4 qui ne fait appel qu'aux concepts de complexité et d'encombrement.

Lieu	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
Complexité	1	1	1	1	2	3	2	3	2	3
Encombrement			✓							

Tableau 2.1 – Information sémantique des différents Lieux.

Une classe abstraite Attribut a été créée afin de pouvoir agréger les Attributs malgré leur diversité. Cette classe est ensuite dérivée pour modéliser tous les différents types d'Attributs nécessaires à la modélisation de la sémantique de l'environnement. Tous les Attributs de nature différente étant issus de la même classe abstraite, ils peuvent être agrégés dans une liste attribut de l'Information sémantique (figure 2.11). Le niveau sémantique du modèle de l'environnement est obtenu en associant une Information sémantique aux différents Lieux et Corps rigides qui le constituent.

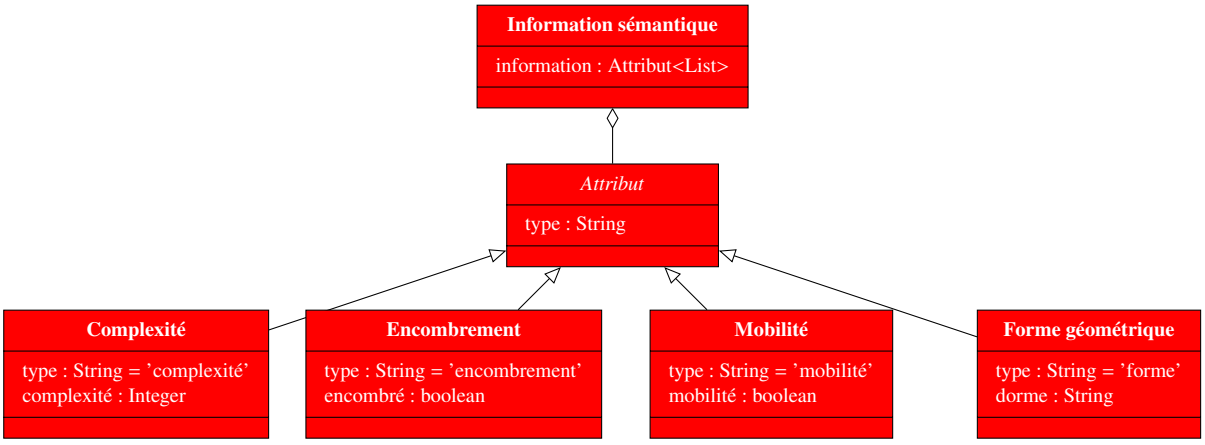


FIGURE 2.11 – Diagramme de classes UML de l'Information sémantique.

Cette définition des différentes classes d'Attributs comme la déclinaison d'une classe abstraite permet d'introduire une certaine flexibilité au niveau sémantique de notre modèle de l'environnement. Elle permet en effet de facilement ajouter de nouveaux types d'Attributs lorsque le besoin de modéliser de nouveaux concepts est identifié.

Cependant, l'identification automatique de l'Information sémantique à associer aux différents Lieux et Corps rigides de l'environnement constitue la difficulté principale. Il convient donc, lors de la définition de nouveaux types d'Attributs, soit de définir comment les éléments qui les portent sont identifiés, soit de fournir à l'opérateur la possibilité de l'associer manuellement aux éléments de l'environnement.

2.3.2 Algorithmes, processus et parallélisation

Différents outils ont été développés pour automatiser une partie de la construction du modèle de l'environnement. Celle-ci est réalisée selon la séquence décrite par l'algorithme 2.1. Les couleurs de chaque chaque ligne correspondent au niveau de l'environnement mis en jeu (jaune pour le niveau géométrique, orange pour le niveau topologique et rouge pour le niveau sémantique).

Algorithme 2.1 : Construction du modèle multi-niveaux de l'environnement

```
1 début
2     Chargement des maillages des Corps rigides ;
3     Construction du modèle de l'Espace libre ;
4     Définition des Lieux ;
5     Identification des Frontieres ;
6     Construction du Graphe topologique ;
7     Définition de l'Information sémantique des Lieux ;
```

En premier lieu, les maillages représentant les différents Corps rigides 3D de l'environnement sont importés depuis la simulation de RV (où ils sont définis comme fixes ou mobiles) (ligne 2). La décomposition de l'Espace libre est ensuite construite (ligne 3) en distinguant pour chaque Cellule géométrique occupée les Corps rigides mobiles des Corps rigides fixes qui l'occupent (chaque Cellule géométrique est doté d'une liste de Corps rigides fixes et une autre de Corps rigides mobiles en collision). L'octree est par la suite parcouru pour définir les Lieux (ligne 4). Les Frontieres entre les Lieux sont ensuite identifiées (ligne 5) puis connectées au sein du Graphe topologique (ligne 6). Pour finir, l'opportunité est offerte à l'utilisateur de compléter et/ou corriger la sémantique associée à chaque Lieu (ligne 7).

2.3.2.1 Création et mise à jour de l'octree

L'octree est une décomposition de l'Espace libre multi-résolution. Il est obtenu en décomposant les cellules cubiques occupées par des Corps rigides en 8 cellules filles (figure 2.12). Ce processus fait donc apparaitre des cellules de différents niveaux. La construction de l'octree est stoppée à une profondeur (niveau) déterminée fixant la résolution de la décomposition.

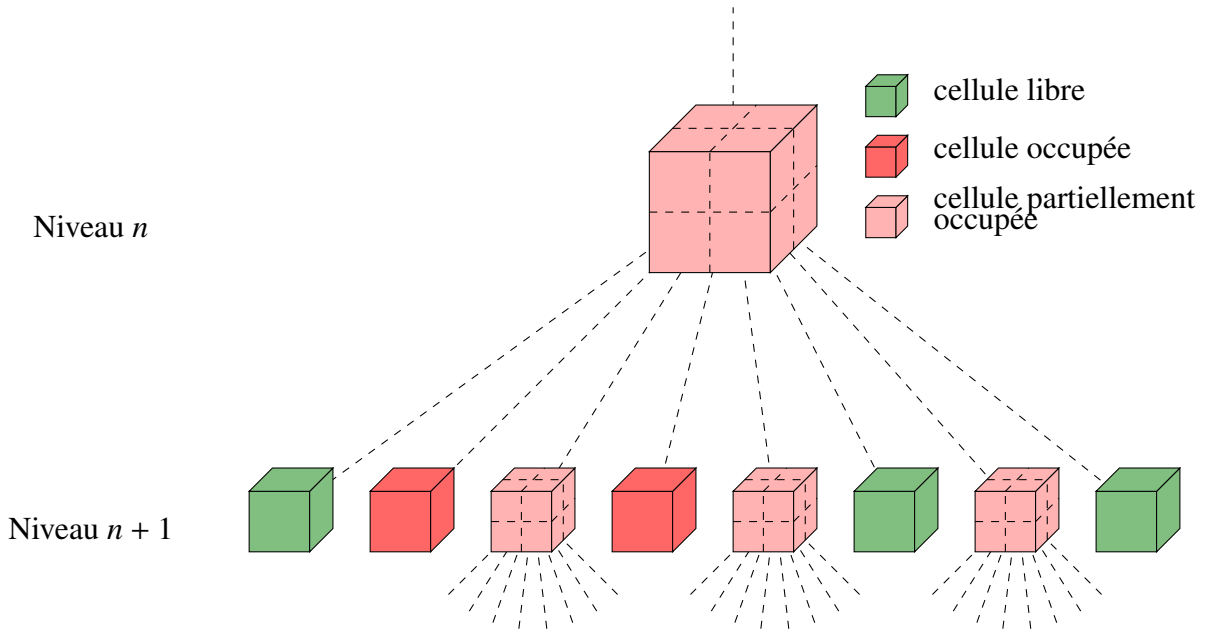


FIGURE 2.12 – Processus de décomposition octree.

Ce processus est basé sur une fonction récursive (algorithme 2.2) appelée pour chaque Cellule géométrique.

Algorithme 2.2 : Décomposition(*cellule*, *niveau*)

```
1 début
2     Vérifier occupation(cellule) ;
3     si cellule = partiellement occupée & niveau < profondeur max alors
4         filles = Créer filles de cellule ;
5         pour chaque f de filles faire
6             Décomposition(f, niveau+1) ;
```

Dans cette fonction, l'occupation de la cellule considérée est vérifiée pour déterminer la présence ou non de Corps rigides de l'environnement dans la cellule. La fonction de vérification de l'occupation de la cellule permet de lui associer, lorsqu'il y a lieu, les Corps rigides qui l'occupent. Cette vérification peut présenter cinq issues différentes qui permettent de définir l'attribut de type de la Cellule géométrique :

1. la cellule est libre,
2. la cellule est totalement occupée par des Corps rigides fixes,
3. la cellule est totalement occupée uniquement par des Corps rigides mobiles,
4. la cellule est partiellement occupée par des Corps rigides fixes,
5. la cellule est partiellement occupée uniquement par des Corps rigides mobiles.

Dans les deux derniers cas, et si la profondeur maximale de l'octree n'est pas atteinte, la Cellule géométrique est divisée en 8 Cellules géométriques filles qui subissent elles-mêmes un traitement identique. La liste des Corps rigides occupant leur Cellule géométrique parent est alors utilisée pour limiter le nombre de Corps rigides considérés lors de la vérification de l'occupation.

La liste des Corps rigides occupant chaque Cellule géométrique est aussi utilisée pour faciliter la reconstruction locale de l'octree lorsque des Corps rigides mobiles (obstacles ou objets manipulés) sont déplacés. Cette reconstruction est réalisée en changeant le type des Cellules géométriques libérées et nouvellement occupées par le Corps rigide. Les Cellules géométriques libérées sont fusionnées lorsque c'est possible, et celles nouvellement occupées sont décomposées.

2.3.2.2 Définition des Lieux

L'identification des Lieux est réalisée par un traitement de l'octree illustré par l'algorithme 2.3. Dans cet algorithme, les Cellules géométriques considérées comme libres sont les Cellules géométriques de type 1, 3 et 5 (cellules vertes dans la figure 2.13).

Cet algorithme est articulé autour de deux procédures détaillées en suivant :

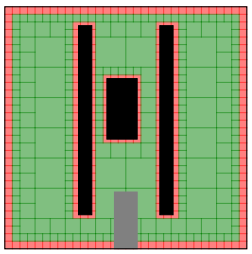


FIGURE 2.13 – Espace libre considéré.

1. l'extension des Lieux existants avec les Cellules géométriques libres d'un niveau donné de l'octree (lignes 8 à 11),

2. la création de nouveaux Lieux à un niveau donné de l'octree (lignes 12 à 17).

Algorithme 2.3 : Définition des Lieux

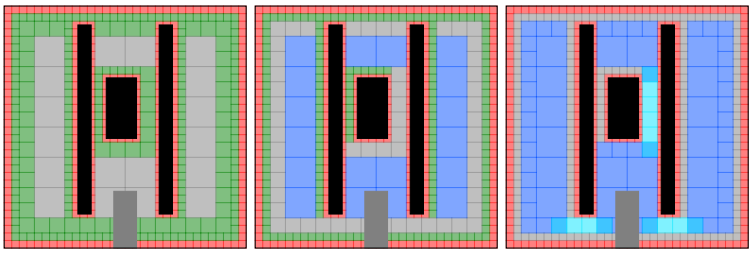
```
1  début
2      ListeLieux = ∅ ;
3      pour i de 0 à ProfondeurOctree faire
4          CellLibres = Cellules géométriques vides de niveau i ;
5          si CellLibres ≠ ∅ alors
6              Voisines = ∅ ;
7              si ListeLieux ≠ ∅ alors
8                  pour chaque L de ListeLieux faire
9                      L ← Cellule géométrique de CellLibres voisines de L ;
10                     Voisines ← Cellule géométrique de CellLibres voisines de L ;
11                 Retirer Voisines à CellLibres ;
12             tant que CellLibres ≠ ∅ faire
13                 Choisir c une Cellule géométrique dans CellLibres ;
14                 R=Région connexe de c ;
15                 Nouveau Lieu L (R) ;
16                 Retirer R à CellLibres ;
17                 ListeLieux ← L ;
```

Lors de l'exécution de cet algorithme (ligne 4) les Cellules géométriques libres de chaque niveau de l'octree sont identifiées (cellules grises dans la figure 2.14).

Les premiers Lieux sont créés au premier niveau de l'octree auquel figurent des Cellules géométriques libres (figure 2.15). Les Lieux créés constituent les régions connexes de Cellules géométriques de même taille.

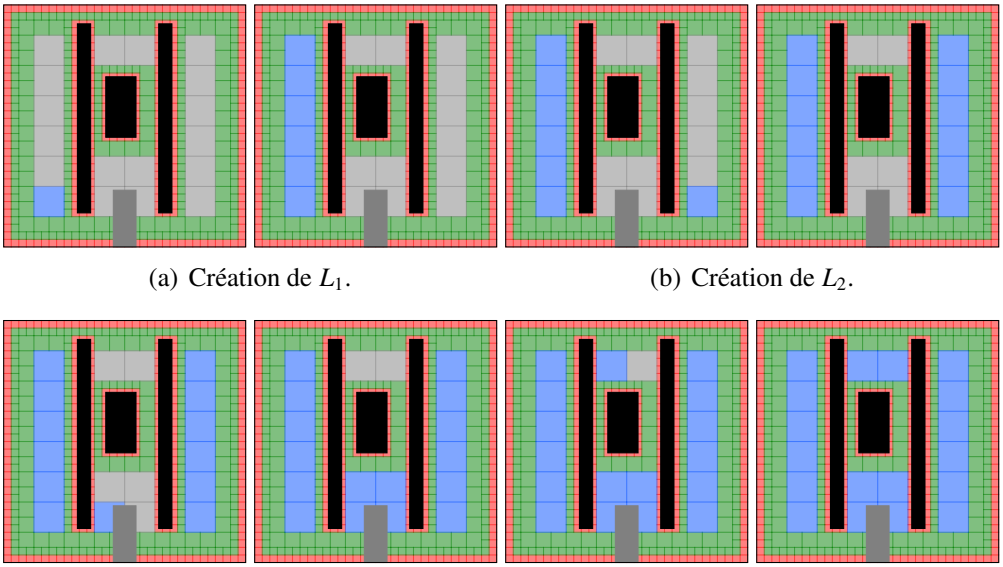
Aux niveaux suivants, l'extension des Lieux existants est tout d'abord effectuée (figures 2.16 et 2.17). Cette extension consiste à agréger aux différents Lieux des Cellules géométriques voisines plus petites.

Les Cellules géométriques libres du niveau courant qui restent non affectées après l'extension des Lieux existants sont traitées par la procédure de création de Lieux (figures2.18 et



(a) Niveau 3. (b) Niveau 4. (c) Niveau 5.

FIGURE 2.14 – Cellules géométriques libres des différents niveaux de l'octree (en gris).



(a) Création de L_1. (b) Création de L_2.

(c) Création de L_3. (d) Création de L_4.

FIGURE 2.15 – Création des premiers Lieux (niveau 3).

2.19). Cette procédure de création de nouveaux Lieux est définie de telle manière que les Lieux créés couvrent au moins une Cellule géométrique des Lieux existants qu'ils atteignent.

2.3.2.3 Construction du Graphe topologique

Pour identifier les frontières, les ensembles de Cellules géométriques formant les Lieux sont comparés deux à deux pour constituer des Zones regroupant les Cellules géométriques connexes communes à deux Lieux. Ces Zones constituent les Frontieres.

Comme présenté dans la section 2.3.1.2, le Graphe topologique construit est constitué de nœuds représentant les Frontieres, et d'arcs représentant les connections entre celles-ci. Ces connections sont représentées à l'aide de matrices :

- une matrice de booléens représente la connectivité entre les Frontieres,
- une matrice de réels positifs, les distances entre les centres des Frontieres concernées,
- une dernière matrice référençant les Lieux concernés par chaque arc du Graphe topologique.

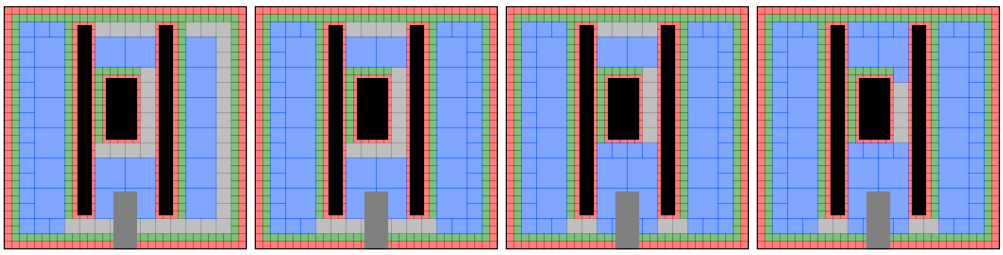
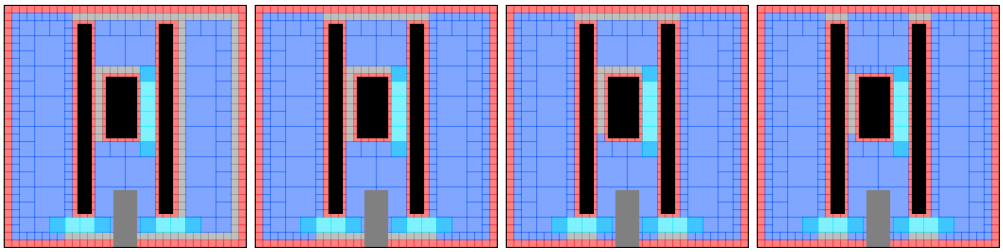
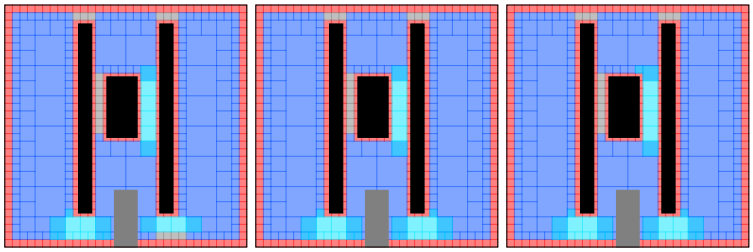


FIGURE 2.16 – Extension des Lieux aux Cellules géométriques libres du niveau 4.



(a) Extension des Lieux créés au niveau 3.



(b) Extension des Lieux créés au niveau 4.

FIGURE 2.17 – Extension des Lieux aux Cellules géométriques libres du niveau 5.

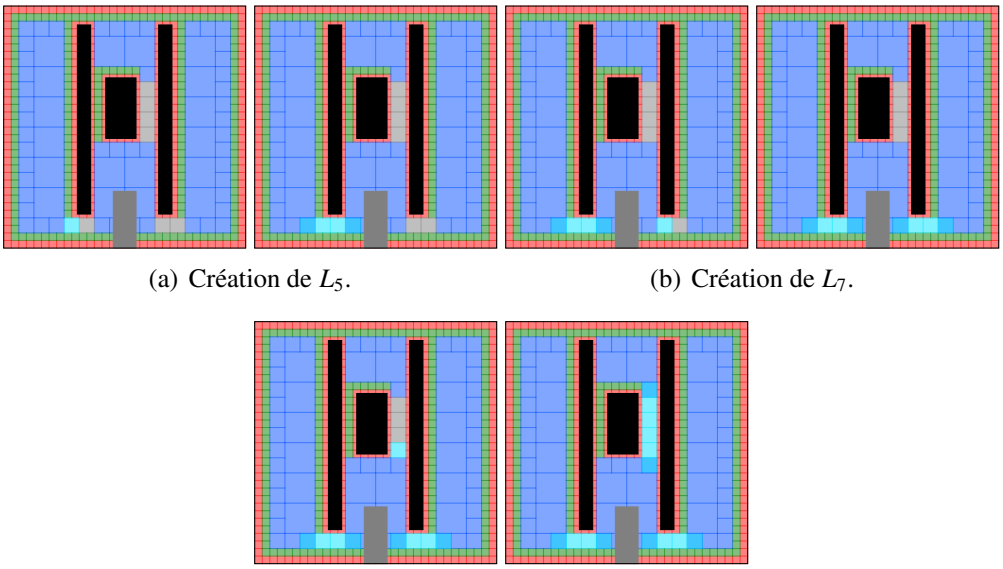
2.3.2.4 Parallélisation des processus

La librairie OpenMP permet une parallélisation aisée des boucles (for, while) lorsque les processus réalisés dans la boucle à paralléliser n'écrivent pas dans les mêmes zones de mémoire.

Cette librairie est donc utilisée lors de la construction du modèle multi-niveaux de l'environnement pour paralléliser différentes opérations :

- la construction de l'octree (chaque itération de la boucle récursive est réalisée dans un *thread* dédié),
- l'extension des Lieux (lignes 8 à 11 de l'algorithme 2.3) lors du passage au niveau inférieur (les opérations illustrées dans les figures 2.16 et 2.17 sont réalisées en parallèle),
- la définition des Frontieres par l'identification de l'intersection des Lieux deux à deux (chaque paire de Lieux est alors traitée dans un *thread* dédié afin d'extraire, s'il y en a, leur(s) Frontiere(s)).

Lors de la création de l'octree illustré dans la figure 2.12, un *thread* est créé chaque fois qu'une nouvelle Cellule géométrique est créée. Ainsi, dans cette figure, chaque arc pointillé correspond à la création d'un nouveau *thread*. Ce *thread* est dédié à une Cellule géométrique. Le type (libre, occupé ou partiellement occupé, par des Corps rigides fixes ou non) de cette



(a) Création de L_5. (b) Création de L_7.

(c) Création de L_{10}.

FIGURE 2.18 – Création de Lieux au niveau 4.

dernière y est défini, et, si nécessaire, sa décomposition en cellules filles (elles-mêmes traitées dans des *threads* dédiés) est réalisée.

Lors de l'identification des Lieux, au moment du changement de niveau, chaque Lieu fait l'objet de la création d'un *thread* dans lequel l'extension du Lieu aux Cellules géométriques voisines est effectuée. Une fois tous les *threads* d'extension des Lieux existants clos après avoir effectué ces opérations, la création de nouveaux Lieux reprend en un *thread* unique.

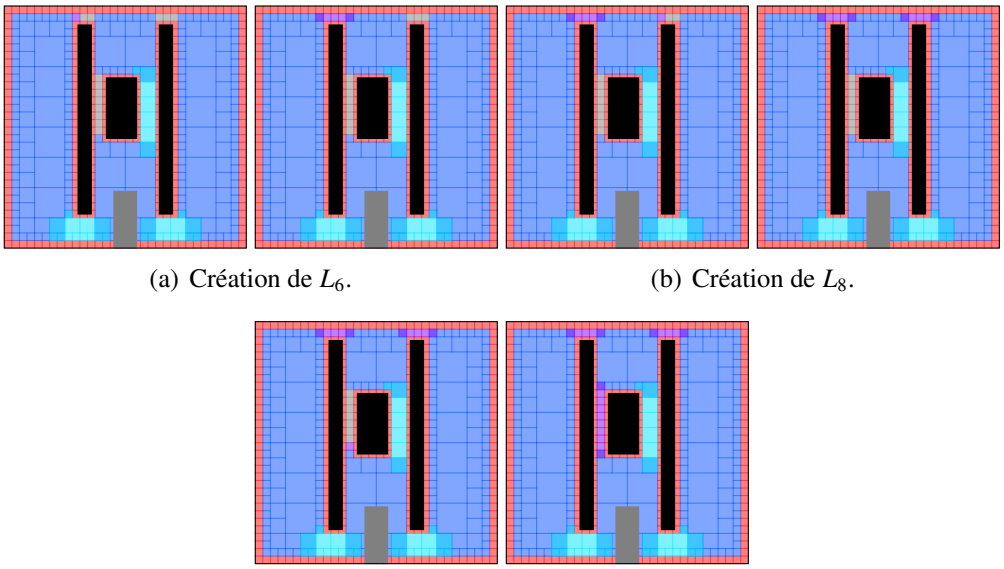
2.4 Conclusion sur la modélisation de l'environnement réalisée

La représentation de l'environnement proposée intègre donc un modèle de l'environnement virtuel 3D comprenant une description des Corps rigides et une description de l'Espace libre. Le modèle de chacun de ces éléments est constitué de différents niveaux permettant d'en représenter différents niveaux d'abstraction. Les Corps rigides disposent d'un niveau géométrique et d'un niveau sémantique. L'Espace libre est lui modélisé par des niveaux sémantique, topologique et géométrique.

Ces trois niveaux sont liés les uns aux autres :

- les éléments du niveau topologique sont ancrés géométriquement à travers les Zones qui regroupent des Cellules géométriques du niveau géométrique qui permettent de les localiser,

- les Attributs du niveau sémantique viennent enrichir la connaissance aussi bien des Lieux du niveau topologique que des Corps rigides modélisés par des maillages polyédriques.



(a) Création de L_6. (b) Création de L_8.

(c) Création de L_9.

FIGURE 2.19 – Création de Lieux au niveau 5.

L'architecture d'implémentation de ce modèle de l'environnement offre une grande modularité et peut être enrichie.

On peut par exemple envisager de compléter la modélisation polyédrique des Corps rigides par l'utilisation de B-Rep, fournissant ainsi un niveau topologique à la représentation des Corps rigides. L'utilisation de modèles analytiques peut aussi être envisagée pour augmenter la précision des traitements réalisés avec les surfaces des Corps rigides (détection de collision). De tels éléments permettraient d'ouvrir la possibilité de réaliser une planification de trajectoire au contact des surfaces des Corps rigides.

Le modèle de l'environnement peut aussi être facilement enrichi de nouveaux Attributs sémantiques permettant de mieux modéliser l'expertise liée à l'application et ainsi de mieux adresser la problématique posée.

La planification de trajectoire, ainsi que l'interaction avec un opérateur humain dans un modèle de l'environnement présentant une telle hétérogénéité d'informations peut être complexe. Les chapitres suivants décrivent la planification de trajectoire et son architecture multi-niveaux proposée pour tirer parti de l'architecture de notre modèle de l'environnement (chapitre 3). De même, l'intégration d'un opérateur humain dans le processus de planification nous a amené à développer des modes d'interaction qui exploitent l'ensemble des niveaux du modèle de l'environnement (chapitre 4).

Chapitre 3

Planification de trajectoire en environnement multi-niveaux

Table **Page**

3.1 État de l'art de la planification de trajectoire automatique 56

3.1.1 Les approches 57

3.1.2 Les méthodes 57

3.1.3 Synthèse 61

3.2 Contribution 62

3.2.1 Planification grossière 63

3.2.1.1 Objectif de la planification grossière 63

3.2.1.2 Processus de planification grossière 63

3.2.1.2.1 Définition des coûts du `Graphe topologique` 64

3.2.1.2.2 Recherche du `Chemin topologique` 66

3.2.1.2.3 Définition des `Étapes topologiques` 66

3.2.2 Planification fine 67

3.2.2.1 Les différentes modalités de planification géométrique 67

3.2.2.2 Choix des modalités de planification fine 67

3.2.3 Synthèse de la planification de trajectoire multi-niveaux proposée 69

3.3 Implémentation de la planification de trajectoire 70

3.3.1 Modèle statique : classes et responsabilités 70

3.3.1.1 Planificateur de trajectoire 70

3.3.1.2 Relation avec le modèle de l'environnement 71

3.3.2 Algorithmes, processus et parallélisation 72

3.3.2.1 Planification grossière 74

3.3.2.2 Planification fine 75

3.4 Conclusion sur la planification de trajectoire réalisée 75

La planification de trajectoire vise à déterminer, dans un espace donné comportant des zones interdites, une trajectoire évitant les zones interdites entre un point de départ S et un point d'arrivée G. L'espace considéré et les points S et G constituent une requête de planification. En robotique, la planification de trajectoire s'effectue dans l'espace des configurations C ($C = C_{libre} \cup C_{obstacle}$) où les zones interdites $C_{obstacles}$ représentent la projection des obstacles de l'espace cartésien. Ce problème est depuis longtemps connu comme difficile [Reif 79]. C'est justement pour y répondre que la notion d'espace des configurations a été développée (section 2.1.1). Grâce à cette notion ; la recherche de trajectoire, complexe lorsque le système dispose d'un grand nombre de degrés de liberté, est ramenée à la recherche d'une trajectoire pour un point dans un espace ayant un grand nombre de dimensions.

Le modèle de l'environnement introduit dans le chapitre précédent permet de simplifier davantage la planification de trajectoire en déterminant, dans un premier temps, un chemin dans le niveau topologique du modèle de l'`Espace libre`. L'`Information sémantique` est pour cela utilisée pour contrôler le processus de détermination de ce chemin de manière à ce que celui-ci soit pertinent et adapté à la requête de planification. Le chemin déterminé sur le modèle de l'`Espace libre` est ensuite utilisé pour diviser la requête de planification en sous-requêtes plus simples à traiter, et pour limiter l'exploration des techniques de planification de trajectoire 6D sur des portions spécifiques de l'`Espace libre`. L'`Information sémantique` est ici encore une fois utilisée pour mettre en œuvre, sur chaque portion de l'espace, la technique la plus adaptée.

Ce chapitre présente un état de l'art des techniques développées en robotique pour la planification de trajectoire (section 3.1). Ensuite, la planification de trajectoire proposée y est présentée (section 3.2) avant la description de son implémentation (section 3.3).

3.1 État de l'art de la planification de trajectoire automatique

Bien que le problème de la définition d'une trajectoire soit rigoureusement formulé en mathématiques, en robotique, les solutions de planification de trajectoire apparaissent comme une bibliothèque de techniques et leurs déclinaisons [Choset 05]. Nous proposons ici de classer ces mêmes techniques selon leur façon d'appréhender l'environnement (approches globales ou locales raisonnant sur l'ensemble de l'environnement pour la première et sur le voisinage de la position courante pour la seconde), mais aussi en fonction de la méthode mise en œuvre pour déterminer la trajectoire (déterministes ou probabilistes). Nous allons d'abord voir comment les approches globales et locales permettent de traiter des problématiques différentes. Nous verrons ensuite comment les méthodes déterministes et probabilistes adressent le problème de la pla-

nification de trajectoire en introduisant les principales techniques de chacune de ces méthodes pour les deux approches possibles (globales ou locales).

3.1.1 Les approches

Les techniques avec **approche globale** s'appuient sur un modèle de l'environnement (section 2.1.3.1). Ce modèle est en général synthétisé dans un graphe qui est exploré (de manière exhaustive ou non) pour y déterminer un chemin. Les techniques mettant en œuvre cette approche sont particulièrement utilisées pour des applications où les modèles, une fois construits, permettent de répondre à plusieurs requêtes de planification. Le modèle permettant de répondre plus rapidement à chacune des requêtes, le coût de sa construction préalable (en temps de calcul) devient abordable.

Les techniques avec **approche locale** s'appuient sur une stratégie de progression au voisinage du point courant. La trajectoire obtenue est alors le produit de la séquence des progressions déterminées localement. Ces techniques sont utilisées lorsque la modélisation de l'environnement n'est pas souhaitée pour diverses raisons :

- environnement dynamique rendant le maintien du modèle difficile,
- environnement inconnu (la technique avec approche locale peut alors être utilisée pour réaliser l'exploration nécessaire pour la modélisation),
- requête de planification unique rendant le coût de modélisation (en temps de calcul) prohibitif.

3.1.2 Les méthodes

Méthodes déterministes Les techniques mettant en œuvre des méthodes déterministes sont basées sur des calculs déterministes et fourniront toujours le même résultat pour une même requête.

Les techniques déterministes avec approche globale s'appuient sur un modèle de l'espace des configurations C construit de manière déterministe. Le modèle est ensuite parcouru (toujours de manière déterministe) afin d'y définir une trajectoire. Ces techniques comprennent :

- les techniques basées sur une décomposition en cellules de l'environnement (cf section 2.1.3.1.1). La connectivité des cellules de la décomposition de l'environnement est saisie dans un graphe. La requête de planification est alors traitée en déterminant les cellules auxquelles appartiennent S et G puis en parcourant le graphe des cellules. Le chemin obtenu consiste en une liste de cellules à traverser. Une technique locale est ensuite utilisée pour déterminer la manière de traverser ces cellules (position aux limites des cellules - p. ex. centre des segments - et trajectoire pour les lier - p. ex. interpolation linéaire) [Latombe 91].

- les techniques basées sur des *roadmaps* déterministes (cf Diagrammes de Voronoï et graphe de visibilité section 2.1.3.1.2). Dans ce cas les configurations S et G sont connectées à la *roadmap* qui est ensuite explorée à l'aide d'un algorithme de parcours de graphe (par exemple Disjkstra [Dijkstra 59] ou A* [Hart 68]). L'ensemble des nœuds et arcs composant le chemin trouvé dans le graphe par cet algorithme constitue la trajectoire demandée.

Lorsque les modèles sur lesquels sont utilisées ces techniques sont exacts, ces techniques sont complètes. Lorsque ces techniques sont utilisées sur des modèles approchés de l'environnement (ce qui est généralement le cas), on parle de complétude en résolution (la résolution du modèle utilisé). Ces techniques étant complètes (ou complètes en résolution), si elles ne fournissent pas de trajectoire, c'est donc qu'il n'en existe pas qui permette de répondre à la requête de planification. De plus, si les algorithmes de parcours de graphe utilisés ne sont pas stoppés prématurément (dès qu'un chemin est trouvé dans le graphe pour réduire le temps d'exploration), la solution retenue est alors la solution optimale au regard du critère utilisé pour contrôler l'exploration du graphe. Ces techniques présentent donc des garanties très intéressantes, mais pêchent par la lourdeur des calculs pour la construction du modèle de l'environnement. En effet, le temps de création de ces modèles de l'espace des configurations du robot croît de manière exponentielle avec l'augmentation du nombre de dimensions de cet espace (c.-à-d. du nombre de degrés de liberté du robot) [Canny 88]. Ainsi, ces techniques ne sont en pratique pas mises en œuvre pour des espaces à plus de 3 dimensions.

Les techniques déterministes avec approche locale sont basées sur les champs de potentiels. Elles consistent en la superposition d'un champ attractif pour l'objectif G (potentiel bas) et des champs répulsifs pour les obstacles (potentiels hauts) [Khatib 85]. Le robot suit alors le gradient maximal des potentiels pour atteindre G.

D'autres champs de potentiels peuvent être ajoutés pour prendre en compte d'autres considérations. Par exemple, pour prendre en compte, dans les trajectoires générées, la présence d'humains dans l'environnement, des champs peuvent être ajoutés aux champs précédents pour empêcher le robot de surgir dans le champ de vision de l'humain, ou de se mettre en face de lui lorsqu'il est assis (comportements perçus comme inconfortables par les humains) [Sisbot 08].

Cette technique des champs de potentiel souffre tout de même de limites importantes comme l'incapacité à percevoir un passage entre deux obstacles proches ou le fait de pouvoir tomber dans des minima locaux [Koren 91]. Ces deux limites représentent un inconvénient fort de la technique des champs de potentiels puisqu'elles peuvent aboutir à l'échec de la définition d'une trajectoire alors qu'une trajectoire réalisable existe. Le potentiel répulsif des obstacles peut être diminué pour répondre à la première limite (le fait de ne pas percevoir un passage entre deux obstacles), mais cela augmente les risques d'apparitions de minima locaux. Ainsi, différentes solutions ont été développées pour répondre à la problématique des minima locaux.

Tout d'abord, des solutions basées sur la suppression des minima locaux des champs de potentiels s'appuient sur des calculs complexes (fonctions de navigation [Koditschek 87] ou équations de Laplace [Connolly 90]). Ces calculs permettent de générer des champs de poten-

tiels ne faisant apparaitre qu'un minimum en G. Ils garantissent ainsi de répondre à la requête de planification si celle-ci admet au moins une solution. Ces techniques permettent en effet de remédier au problème des minima locaux, mais cela, au prix de lourds calculs qui ne les rendent exploitables que sur des espaces de faible dimension. Ces calculs pénalisent de plus la facilité de mise en œuvre des champs de potentiel en atténuant l'aspect local de l'approche utilisée dans cette technique.

Avec la même idée de supprimer les minima locaux des champs de potentiel, les potentiels répulsifs des obstacles ont été remplacés par des "potentiels de circulation" permettant de les contourner selon un sens arbitrairement choisi [Singh 96]. Les trajectoires ainsi générées peuvent donc contourner les obstacles de manière non optimale.

Pour répondre au problème des minima locaux sans alourdir la définition des champs de potentiels, d'autres techniques conservent les minima locaux, mais pallient la chute dans un minimum (autre que G) en définissant des méthodes pour s'en "échapper". Par exemple, lorsque un minimum local est rencontré, des objectifs intermédiaires peuvent être ajoutés pour contourner l'obstacle produisant le minimum local et ainsi se placer sur une nouvelle pente de potentiels [Zou 03]. Une autre méthode pionnière consiste, une fois tombée dans un minimum local, à réaliser une "marche" d'une distance aléatoire dans une direction aléatoire pour sortir du puits [Barraquand 91]. Les chemins obtenus ainsi peuvent ne pas être optimaux puisqu'avant de converger vers une solution, le robot aura pu passer par un ou plusieurs minima locaux avant de s'en échapper.

Méthodes probabilistes Les techniques mettant en œuvre une méthode probabiliste sont basées sur une description partielle et aléatoire de l'espace des configurations (échantillonnage aléatoire de l'espace des configurations et inter-connections des échantillons tirés). Ces méthodes garantissent de trouver un chemin, si il existe, dans un temps infini (on parle de complétude probabiliste). Cependant, la durée de l'exploration aléatoire doit, pour des raisons pratiques être limitée. Il convient alors de définir le nombre n de configurations aléatoirement définies nécessaires pour synthétiser la connectivité de l'environnement. Ce nombre n dépend de l'encombrement de l'espace des configurations par des zones interdites (dues aux obstacles). Cet encombrement est souvent hétérogène, trouver un passage entre deux zones interdites avec des jets aléatoires est alors d'autant plus difficile que le passage est étroit. Ainsi, le fractionnement de l'espace des configurations (nombre et largeur des passages étroits) influe sur le nombre n de configurations aléatoires nécessaires et donc le temps de calcul de ces techniques. Ainsi, les enjeux des techniques mettant en œuvre une méthode probabiliste sont, de définir d'une part le nombre n de configurations à utiliser pour disposer d'une description suffisante de l'environnement, et, d'autre part la stratégie à adopter pour le jet aléatoire de ces configurations.

Parmi les techniques basées sur une méthode probabiliste, on distingue encore des techniques mettant en œuvre une approche globale et d'autres mettant en œuvre une approche locale.

Les techniques probabilistes avec approche globale s'appuient, à l'instar des techniques déterministes avec approche globale, sur une *roadmap* de l'environnement pour y exécuter un

recherche de chemin similaire. Cependant, cette fois, la *roadmap* est réalisée à partir d'un échantillonnage aléatoire de C_{libre}. On parle alors de PRM (cf. section 2.1.3.1.2). Ces techniques introduites il y a une vingtaine d'années [Overmars 92, Kavraki 96] permettent d'adresser des problèmes ayant un grand nombre de dimensions. Elles sont par exemple utilisées pour la planification de manipulation dextre [Saut 06]. Dans ces travaux, l'espace des configurations est divisé en sous-espaces. Des PRM saisissent la connectivité de chacun de ces sous-espaces, et la connectivité entre ces sous-espaces est modélisée afin de planifier des trajectoires dans C_{libre}.

Les techniques basées sur les PRM ont fait l'objet de nombreux développements visant à optimiser le mode de jet aléatoire des configurations pour synthétiser plus rapidement la connectivité de C_{libre} avec un nombre de configurations réduit et défini rapidement.

Pour atteindre cet objectif, il a été proposé de construire la *roadmap* à partir d'un échantillonnage des configurations au contact des obstacles. Pour cela, des configurations sont définies au contact des obstacles, puis la *roadmap* est construite en utilisant comme nœuds des configurations prises sur les arcs connectant les configurations au contact de deux obstacles différents (des configurations entre les deux obstacles) [Amato 96]. Dans cette technique, la difficulté d'échantillonner de manière efficace la surface des obstacles (densité surfacique d'échantillons uniforme) est rapidement apparue. Ce processus d'échantillonnage des surfaces a donc fait l'objet d'améliorations afin d'obtenir un échantillonnage uniformément dense des surfaces des obstacles [Amato 98].

Pour trouver les passages étroits, l'utilisation de la loi normale pour définir des configurations à proximité des obstacles (et donc favoriser les passages étroits) a aussi été proposée [Boor 99].

Une autre technique consiste à "dilater" C_{libre} (concrètement en tolérant les interpénétrations entre le robot et les obstacles de l'environnement) pour construire la PRM [Hsu 98]. La tolérance est ensuite progressivement réduite et la *roadmap* localement reconstruite (suppression des portions en collision et définition de nouvelles configurations sans collision) jusqu'à obtenir une *roadmap* opérationnelle.

Le diagramme de Voronoï de l'espace 2D (ou 3D) peut être calculé pour ensuite y projeter les configurations à n dimensions tirées aléatoirement [Holleman 00, Lien 03].

La notion de visibilité similaire à celle utilisée pour la construction des graphes de visibilité (section 2.1.3.1.2) a aussi été introduite pour la construction de PRM [Siméon 00]. On distingue alors, lors de la construction, de la PRM, les "nœuds gardes" qui assurent une visibilité de l'ensemble de l'espace des configurations libres (la zone de l'environnement visible par un "nœud garde" est une "composante" de l'environnement), et les "nœuds de connection" qui connectent les différentes "composantes" de l'espace libre. Enfin, des techniques hybrides mettant en œuvre plusieurs des idées évoquées précédemment ont été proposées [Hsu 05].

Les techniques probabilistes avec approche locale ne nécessitent, elles, aucune modélisation de l'espace des configurations mais définissent une stratégie de progression pour explorer aléatoirement cet espace depuis une configuration spécifique (typiquement la configuration de départ S). On parle alors d'arbres de diffusion et d'algorithmes *Rapidly-exploring Random Tree*

(RRT) ou *Rapidly-exploring Dense Tree* (RDT) [LaValle 98]. À l'instar des PRM, ces techniques rencontrent de grandes difficultés aux passages étroits de l'espace des configurations. Diverses techniques ont donc été développées pour adresser ces difficultés.

Par exemple, lorsqu'un couloir est trouvé, une Analyse en composantes principales (ACP) peut permettre d'accélérer la progression le long de ce couloir en extrayant la direction de progression des configurations précédemment ajoutées à l'arbre [Dalibard 09]). L'exploration de l'environnement peut aussi être accélérée par l'utilisation de plusieurs arbres en parallèle [Strandberg 04]. Ces différents arbres explorent différentes "composantes" de C_{libre} et augmentent ainsi les chances de trouver les passages étroits permettant de les interconnecter. On trouve aussi des techniques similaires à celles développées pour les PRM. Par exemple des constructions d'arbres de diffusion mettant en œuvre la dilatation de C_{libre} [Ferre 04], ou la notion de visibilité [Flavigné 10b].

3.1.3 Synthèse

Les différentes techniques de planification de trajectoire sont donc des adaptations de cinq grandes familles de planification résumées dans la table 3.1 avec leurs travaux fondateurs ainsi que leurs avantages (en vert) et leurs inconvénients (en rouge).

	Approches globales (multiples requêtes)	Approches locales (requête unique)
Méthodes déterministes	• Décomposition en cellules [Brooks 82b] • *Roadmap* [Brooks 82a] Complètes ou complètes en résolution Temps de construction du modèle prohibitif pour plus de 3 dimensions	• Champs de potentiels [Khatib 85] Mise en œuvre aisée Non complètes, problème des minima locaux
Méthodes probabilistes	• PRM [Overmars 92] Facile à mettre en œuvre, complétude probabiliste Difficulté avec les environnements "fractionnés" due aux passages étroits	• RRT and RDT [LaValle 98] Facile à mettre en œuvre, complétude probabiliste Difficulté avec les environnements "fractionnés" due aux passages étroits

Tableau 3.1 – Principales approches de planification de trajectoire

Une grande variété de techniques de planification de trajectoire est donc proposée dans la littérature. Ces techniques souffrent toutes de faiblesses et permettent d'adresser des problématiques différentes. De plus, les techniques de planification existantes ne raisonnent que sur des

modèles géométriques là où l'utilisation de modèles plus symboliques permettant, par exemple, d'identifier les passages étroits, pourrait être d'une aide précieuse.

Cet état de l'art, nous permet également de voir que si différentes techniques de planification de trajectoire (adaptées à différentes problématiques) existent, chaque requête de planification n'utilise qu'une seule technique. Or, pour une même requête, différentes modalités de planification peuvent être utiles sur différentes portions de la trajectoire à définir (pour l'assemblage : approche, pré-positionnement, assemblage). Nous allons voir dans la section suivante comment l'utilisation de données de différentes natures (sémantique, topologique et géométrique) disponibles dans le modèle de l'environnement proposé dans le chapitre 2) permet d'identifier les différentes portions de la trajectoire et de leur associer des techniques de planification de trajectoire. adaptées aux contextes locaux.

3.2 Contribution

Pour répondre aux limites des travaux précédents (traitements lourds sur des modèles purement géométriques volumineux, non-adaptabilité aux différents contextes locaux), nous proposons une technique de planification de trajectoire divisée en deux phases [Cailhol 14b]. Chacune de ces phases met en œuvre des données spécifiques du modèle de l'environnement présenté dans le chapitre 2.

La première phase, que nous appelons planification grossière, met en œuvre les modèles topologique et sémantique de l'environnement pour définir un Chemin topologique. Ce chemin est partitionné en un ensemble d'Étapes topologiques permettant chacune de traverser un Lieu et d'atteindre une Frontiere qui constitue donc un jalon topologique.

Les différentes Étapes topologiques sont ensuite traitées dans une seconde phase que nous appelons planification fine qui met en œuvre le modèle géométrique de l'Espace libre et des Corps rigides à travers une Stratégie de planification. Cette Stratégie de planification définit des jalons géométriques échantillonnant la trajectoire 6D de l'Étape topologique qu'elle traite.

La planification de trajectoire proposée fait intervenir la sémantique pour le contrôle de chacune de ces deux phases : pour le choix des Lieux et Frontieres à emprunter lors de la planification grossière, et pour le choix des techniques de planification mises en œuvre dans la Stratégie de planification de chaque Étape topologique lors de la planification fine.

Ce mode de planification en deux phases peut être rapproché du mode de planification humain décrit par les neurobiologistes [Ahmadi-Pajouh 07]. Cette correspondance facilite l'interprétation mutuelle des deux entités en interaction (système automatique et opérateur humain), et ainsi, fluidifie l'interaction comme nous le verrons dans le chapitre 4.

La planification de trajectoire développée s'appuie sur des planificateurs dédiés à chaque niveau de modèle de l'environnement :

Le Planificateur sémantique exploite le niveau sémantique du modèle de l'environnement. Pour cela, il met en œuvre un Interpréteur sémantique pour traiter les différentes In-

formations sémantiques de l'environnement et utilise ces traitements pour séquencer la mise œuvre des autres planificateurs. Au delà de ces traitements liés à la planification de trajectoire, c'est aussi le Planificateur sémantique qui, à terme, planifiera les opérations à un plus haut niveau (planification de tâche).

Le Planificateur topologique traite avec le Graphe topologique de l'environnement pour définir un Chemin topologique.

Le Planificateur géométrique permet de définir, lorsque cela est utile pour limiter davantage l'Espace libre étudié, un chemin au sein du modèle géométrique de l'Espace libre.

Le Planificateur local traite lui avec les Corps rigides de l'environnement pour définir la trajectoire 6D.

3.2.1 Planification grossière

La planification grossière parcourt le Graphe topologique de l'environnement à l'aide d'un algorithme de parcours de graphe (typiquement Dijkstra ou A*) pour y trouver un chemin. Les coûts des arcs et nœuds du Graphe topologique utilisés permettent d'en contrôler l'exploration. Ils sont déterminés à partir de l'Information sémantique des Lieux auxquels ils se réfèrent ainsi que celle du Corps rigide O_d pour lequel on planifie une trajectoire. Il permettent de traduire la difficulté pour O_d à traverser un Lieu (coûts des arcs) et la difficulté à passer d'un Lieu à un autre (coûts des nœuds). Ainsi, malgré la différence entre espace 3D (dont la connectivité est synthétisée dans le Graphe topologique) et l'espace 6D, le Chemin topologique obtenu est adapté à la requête de planification puisque les Lieux retenus sont ceux privilégiés par l'interprétation de l'Information sémantique.

3.2.1.1 Objectif de la planification grossière

La planification grossière permet, en premier lieu, de déterminer si au moins un chemin existe dans le Graphe topologique pour répondre à la requête de planification. S'il en existe plusieurs, l'implication de la sémantique dans la planification grossière permet de choisir le Chemin topologique le plus adapté à la requête de planification. Une fois déterminé, ce chemin est utilisé pour scinder la trajectoire en Étapes topologiques (chaque Étape topologique associant chaque Lieu à traverser avec une Frontiere à atteindre). Ce partitionnement permet de jalonner la trajectoire (les différentes Frontieres à atteindre constituent des jalons topologiques). De plus, chacune des Étapes topologiques obtenues constitue une requête de planification pour le niveau géométrique pour laquelle la modalité de planification de trajectoire peut être adaptée à l'Information sémantique du Lieu associé à l'Étape topologique considérée.

3.2.1.2 Processus de planification grossière

La figure 3.1 illustre les principales étapes de la planification grossière données dans l'algorithme 3.1. Pour adresser la requête de planification, il est indispensable, en premier lieu,

d'enrichir le Graphe topologique avec les données de cette requête. Pour cela, deux nœuds S et G viennent enrichir le Graphe topologique (ligne 2 de l'algorithme 3.1) pour représenter respectivement les configurations de début et de fin de la requête de planification. Cette opération est illustrée dans la figure 3.1(a) où les carrés rouges et verts représentent les nœuds du graphe topologique associés respactivement aux configurations de début (S) et de fin (G) de la requête de planification.

Algorithme 3.1 : Planification grossière

```
1 début
2     Mise à jour du Graphe topologique avec les nœuds de début et de fin ;
3     Calcul des coûts du Graphe topologique ;
4     Exploration du Graphe topologique ;
5     Construction du Chemin topologique et décomposition en Étapes topologiques ;
6     pour chaque Étape topologique ∈ Chemin topologique faire
7         définir un jalon géométrique dans le jalon topologique de l'Étape topologique ;
```

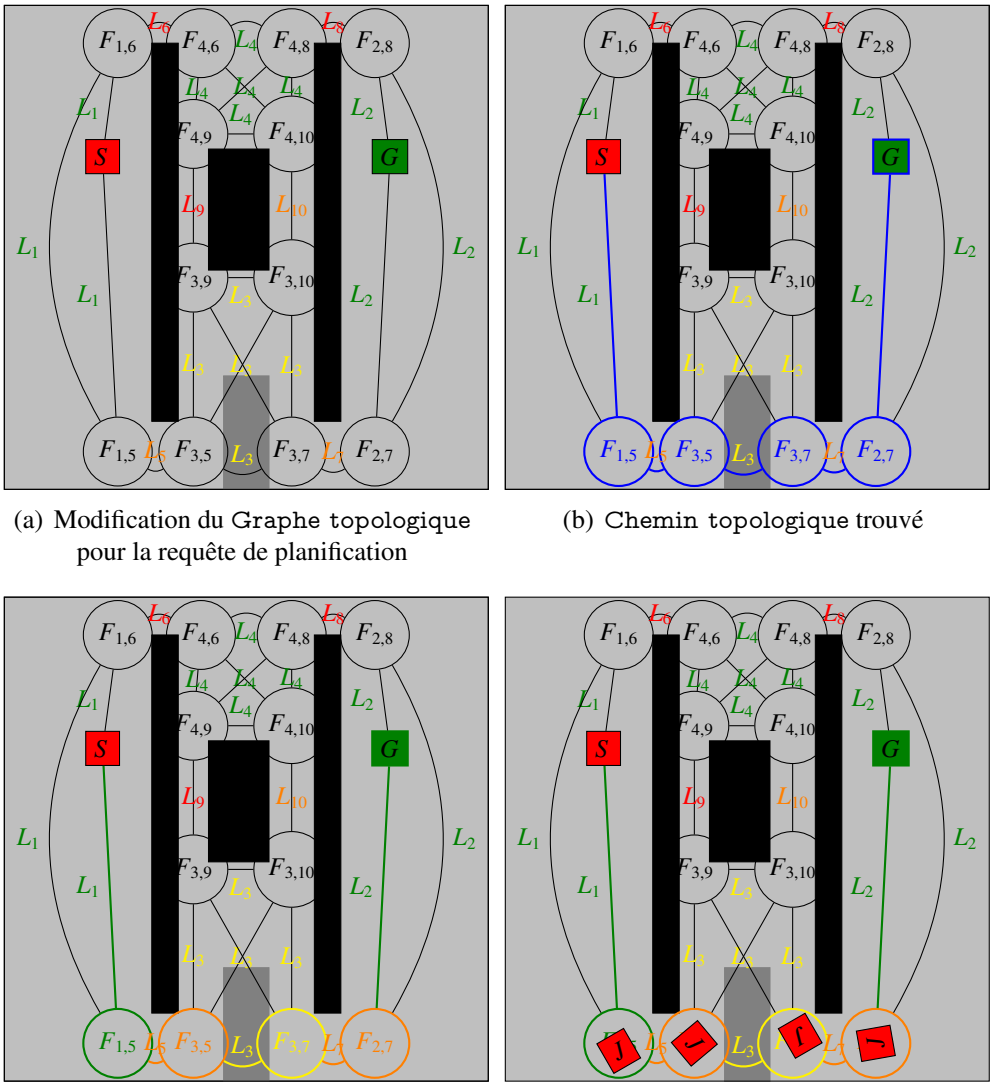
Ensuite, le Planificateur sémantique utilise l'information sémantique des différents Lieux et de O_d pour définir l'ensemble des coûts des arcs et des nœuds du Graphe topologique (ligne 3 de l'algorithme 3.1) (section 3.2.1.2.1). Ces différents coûts permettent de contrôler l'exploration du Graphe topologique par le Planificateur topologique (ligne 4 de l'algorithme 3.1) pour que le Chemin topologique retenu soit adapté à la requête de planification et à la perception humaine de l'environnement (section 3.2.1.2.2). Le Chemin topologique ainsi obtenu est utilisé pour définir chaque Étape topologique (ligne 5 de l'algorithme 3.1) en agrégeant un Lieu à traverser et une Frontiere à atteindre (section 3.2.1.2.3). Les Frontieres ainsi associées aux Étapes topologiques du Chemin topologique constituent donc des jalons topologiques.

Enfin, pour conclure la planification grossière, chaque étape du Chemin topologique est utilisée pour définir aléatoirement un jalon géométrique (position et orientation) au sein de chaque jalon topologique (lignes 6 à 7 de l'algorithme 3.1, figure 3.1(d)).

La construction du Graphe topologique, comme son parcours par un algorithme de parcours de graphe sont des processus déterministes. Ainsi, cette première partie de la planification grossière, jusqu'à la définition de jalons géométriques aux différentes Frontieres du Chemin topologique est un processus déterministe.

3.2.1.2.1 Définition des coûts du Graphe topologique

Les coûts, évoqués précédemment, que le Planificateur sémantique assigne aux arcs et aux nœuds du Graphe topologique, traduisent la difficulté, pour O_d à traverser un Lieu (coût des arcs), ou sa difficulté à passer d'un Lieu au suivant (coût des nœuds). Pour cela, l'équation 3.1 montre que le coût $C_{n_{i,j}}$ associé au nœud correspondant à la frontière $F_{i,j}$ liant les Lieux



(a) Modification du Graphe topologique pour la requête de planification

(b) Chemin topologique trouvé

(c) Éléments topologiques d'une Étape topologique

(d) Définition des jalons géométriques

FIGURE 3.1 – Processus de planification grossière

L_i et L_j est défini en fonction des Informations sémantiques associées à ces Lieux et celle associée à O_d. Ce coût permet donc de traduire la difficulté de passer du contexte de planification de trajectoire interprété à partir de l'Information sémantique associée au Lieu L_i à celui du Lieu L_j (interprété de même à partir de son Information sémantique). Les Frontieres représentant les transitions entre deux contextes et non le parcours d'une distance dans ces contextes, aucune distance n'est utilisée pour définir leurs coûts. Le coût $C_{a_{i_{j,k}}}$ associé à l'arc entre les nœuds correspondant aux frontières $F_{i,j}$ et $F_{i,k}$ est lui défini en fonction de la distance $d_{i_{j,k}}$ qui y est associée, et de l'Information sémantique associée à O_d et au Lieu auquel cet arc est lié (L_i) de manière à traduire la difficulté pour O_d à parcourir la distance $d_{i_{j,k}}$ dans le Lieu L_i.

$$\begin{aligned} C_{n_{i,j}} &= f(sem(L_i), sem(L_j), sem(O_d)) \\ C_{a_{i_{j,k}}} &= f(d_{i_{j,k}}, sem(L_i), sem(O_d)) \end{aligned} \tag{3.1}$$

3.2.1.2.2 Recherche du Chemin topologique

Les coûts du Graphe topologique permettent de définir le coût d'un Chemin topologique $C_{CheminTopo}$ comme la somme des coûts des arcs et nœuds qu'il emprunte. Ce calcul est donné dans l'équation 3.2 où $n_{i,j}$ et $a_{i_{j,k}}$ sont respectivement les nœuds i, j et les arcs $i_{j,k}$ de $CheminTopo$.

$$C_{CheminTopo} = \sum_{n_{i,j} \in CheminTopo} C_{n_{i,j}} + \sum_{a_{i_{j,k}} \in CheminTopo} C_{a_{i_{j,k}}} \tag{3.2}$$

L'algorithme A* [Hart 68] nécessite la définition d'une heuristique permettant d'évaluer le coût entre chaque nœud étudié et un nœud objectif pour estimer le coût du chemin restant à parcourir. Dans notre cas, ce coût intègre à la fois des éléments géométriques et des éléments sémantiques rendant la définition d'une telle heuristique complexe. Nous préférons donc utiliser l'algorithme de Dijkstra [Dijkstra 59] qui n'utilise aucune estimation du coût du chemin restant.

Ainsi, le Planificateur topologique, en parcourant le Graphe topologique depuis le nœud de début S à l'aide d'un algorithme de Dijkstra, définit le chemin présentant le coût le plus faible entre le nœud S et chaque nœud du Graphe topologique.

Le chemin satisfaisant ce critère de coût minimum entre les nœuds de début S et de fin G est retenu lors de la planification grossière. Celui de notre exemple est affiché en bleu dans le Graphe topologique de la figure 3.1(b). Ce chemin est utilisé pour construire chaque Étape topologique composant la trajectoire.

3.2.1.2.3 Définition des Étapes topologiques

Une Étape topologique est composée d'un Lieu à traverser inscrit dans le Chemin topologique et de la Frontiere à atteindre pour passer au Lieu suivant.

La figure 3.1(c) illustre les différentes Étapes topologiques du Chemin topologique illustré dans la figure 3.1(b). Les arcs et nœuds du Graphe topologique associés aux Lieux et Frontieres associés à chaque Étape topologique y sont colorés en fonction du contexte décrit dans l'Information sémantique du Lieu à traverser dans l'Étape topologique (vert : complexité=1, orange : complexité=2, jaune : Lieu encombré).

Cette notion d'Étape topologique permet donc d'une part de partitionner la trajectoire en fonction des Lieux qu'elle traverse et d'autre part d'y disposer de jalons topologiques grâce aux Frontieres.

La phase de planification grossière se conclut en définissant aléatoirement un premier jalon géométrique pour chaque Étape topologique au sein de la Frontiere qui la conclut (sauf pour la dernière Étape topologique où G est retenu comme jalon géométrique) (figure 3.1(d)). Les jalons géométriques ainsi définis permettent de formuler les requêtes de planification géométrique pour la planification fine.

3.2.2 Planification fine

La Planification fine permet de définir la trajectoire permettant de parcourir chaque Étape topologique. Cette trajectoire est décrite à l'aide de jalons géométriques.

Les Étapes topologiques permettent de définir pour chaque étape, une Stratégie de planification de trajectoire géométrique adaptée. Cette stratégie est définie à partir des Informations sémantiques attachées au Lieu que l'Étape topologique traverse.

La planification fine vise à définir la trajectoire, dans l'espace 6D, permettant de couvrir une Étape topologique. Chaque Étape topologique constitue, pour la planification fine, une requête de planification consistant à atteindre le jalon géométrique de l'étape considérée, à partir de celui de l'étape précédente (ou de S pour la première Étape topologique).

3.2.2.1 Les différentes modalités de planification géométrique

Le Planificateur géométrique et le Planificateur local permettent de planifier une trajectoire dans le niveau géométrique du modèle de l'Espace libre pour le premier, et une trajectoire 6D pour le second. Ils permettent de mettre en œuvre des modalités de planification différentes pour les différentes Étapes topologiques. Ce sont donc des notions génériques qui permettent chacun de définir différents modes de planification de trajectoire.

Ces planificateurs peuvent être associés dans une Stratégie de planification. Ils sont alors exécutés en séquence (Planificateur géométrique puis Planificateur local). Dans ce cas, le résultat de la planification du Planificateur géométrique est utilisé pour limiter le domaine 3D exploré par le Planificateur local.

Une Stratégie de planification est associée à chaque Étape topologique. Elle permet d'adapter la technique de planification de trajectoire au contexte du Lieu à traverser décrit dans son Information sémantique. De plus, l'association de Stratégies de planification à chaque Étape topologique permet de limiter le champ d'action des planificateurs qui les composent aux Zones des Lieux de l'Étape topologique pour laquelle ils doivent définir une trajectoire.

3.2.2.2 Choix des modalités de planification fine

Afin de pouvoir adapter au mieux la Stratégie de planification, il est nécessaire de disposer d'une bibliothèque de Planificateurs locaux et de Planificateurs géométriques suffisamment riche.

De plus, il faut, pour adapter convenablement la Stratégie de planification, disposer des informations suffisantes dans le niveau sémantique du modèle de l'environnement pour faire le choix le plus approprié.

La table 3.2 illustre des choix de Stratégie de planification qui peuvent être fait avec un Planificateur géométrique A* et un Planificateur local de type RRT. La colonne "Contexte d'utilisation" décrit en langage naturel les informations dont on doit disposer au travers du niveau sémantique du modèle de l'Espace libre pour adapter la Stratégie de planification à chaque Lieu.

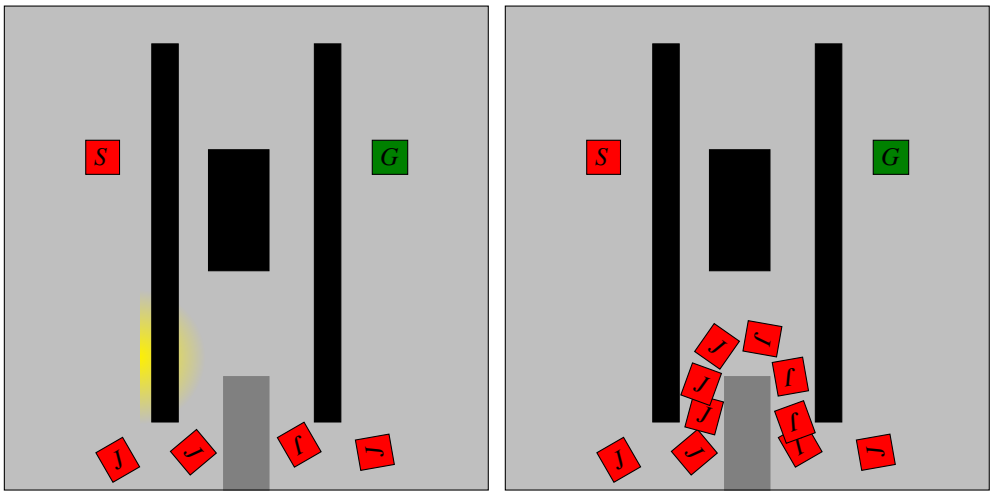
Le symbole "∅" signifie qu'aucun planificateur de ce type n'est mis en œuvre. Pour un niveau décroissant de la complexité d'un Lieu, on peut utiliser les deux planificateurs successivement, le Planificateur local seul, le Planificateur géométrique seul ou aucun des deux. Lorsque le Planificateur géométrique est utilisé seul, des jalons géométriques sont définis au centre de chacune des Cellules géométriques retenues. Quand aucun planificateur n'est utilisé, cela signifie que les jalons définis lors de la planification grossière suffisent à déterminer une trajectoire pour traverser ce Lieu. Dans tous les cas, une interpolation linéaire est utilisée pour connecter deux à deux les jalons consécutifs de la trajectoire 6D.

Stratégie de planification		Contexte d'utilisation
Planificateur géométrique	Planificateur local	(Information sémantique associée au Lieu)
∅	∅	Le Lieu ne présente aucune difficulté. Les jalons déterminés lors de la planification grossière suffisent.
A*	∅	Le Lieu est fractionné mais composé de cellules suffisamment grosses pour que l'orientation soit négligeable
∅	RRT	Le Lieu n'est pas fractionné mais encombré, ce qui nécessite la prise en compte de l'orientation
A*	RRT	Le Lieu est fractionné et nécessite la prise en compte de l'orientation

Tableau 3.2 – Exemple de choix de Stratégie de planification.

La figure 3.2 illustre les choix de Stratégie de planification faits pour la planification grossière illustrée dans la figure 3.1. Ici, l'attribut décrivant l'encombrement du Lieu L_3 (figure 3.2(a)) permet de choisir d'y utiliser un algorithme RRT pour définir une trajectoire permettant de parcourir la troisième Étape topologique et ainsi définir des jalons géométriques supplémentaires dans le Lieu L_3 (figure 3.2(b)). Les autres Lieux retenus dans le Chemin topologique n'étant pas encombrés, aucun jalon supplémentaire n'y est défini lors de la planification fine (la Stratégie de planification ne met en œuvre ni Planificateur géométrique ni Planificateur local).

Il convient d'enrichir les bibliothèques de Planificateurs géométriques et Planificateurs locaux (au moins en rendant A* et RRT paramétrables) pour adresser au mieux des problématiques particulières liées au contexte spécifique des différents Lieux identifiés. L'algorithme A* du Planificateur géométrique peut par exemple explorer de manière privilégiée les grandes Cellules géométriques ou non. Pour le Planificateur local, différentes stratégies de jet aléatoires peuvent être utilisées pour l'algorithme RRT.



(a) Lieu encombré du Chemin topologique (b) Jalonnement du Lieu L_3

FIGURE 3.2 – Processus de planification fine

3.2.3 Synthèse de la planification de trajectoire multi-niveaux proposée

La planification de trajectoire proposée répond à deux limites évoquées des techniques de planification de trajectoire classiques (section 3.1) : elle exploite des informations à haut niveau d'abstraction (sémantiques et topologiques) pour définir une trajectoire pertinente et pour adapter la technique de planification géométrique au contexte (sémantique) local. La pertinence des trajectoires obtenues, ainsi que l'adaptation des techniques de planification le long de la trajectoire permet de réduire les calculs nécessaires à la définition de la trajectoire (et donc de réduire le temps de calcul). Cette contribution est obtenue par l'utilisation des Informations sémantiques lors des deux phases constituantes de notre processus de planification :

1. Tout d'abord, les données sémantiques de l'environnement sont utilisées pour déterminer les coûts des arcs et nœuds du Graphe topologique. Ces coûts permettent de contrôler l'exploration du Graphe topologique et ainsi d'obtenir un Chemin topologique pertinent pour répondre à la requête de planification.

2. Ensuite, une fois le Chemin topologique partitionné en Étapes topologiques, la sémantique de l'environnement est utilisée pour adapter la stratégie de planification à la complexité locale de chaque Étape topologique de la trajectoire.

Cette structure de planification est de plus ouverte à de nombreuses évolutions. En effet, l'intégration de nouvelles techniques de planification se fait à travers la définition de nouveaux Planificateurs géométriques ou Planificateurs locaux qui s'intègreront naturellement à l'architecture avec la définition des contextes (décrits à travers l'Information sémantique) dans lesquels ils doivent être mis en œuvre. Par exemple, des techniques de planification de trajectoire au contact des Corps rigides pourraient se montrer utiles pour les dernières manipulations lors de l'assemblage de composant (utilisation des surfaces des composants à assembler pour guider la planification de trajectoire). Le fait de pouvoir planifier une trajectoire dans différents espace des configurations pour les différentes Étapes topologiques pourrait aussi être utile

pour, par exemple, négliger l'orientation dans les portions de trajectoire dans des Lieux larges par rapport à O_d. Modifier la métrique de l'espace des configurations entre différentes Étapes topologiques en fonction par exemple de la courbure des surfaces des Corps rigides semble aussi intéressant dans le cadre de la planification de trajectoire au contact.

Cependant, le partitionnement du processus de planification en deux phases pose la question du bouclage entre celles-ci. En effet, si la planification fine échoue à définir une trajectoire géométrique, il conviendrait de revenir à la planification grossière pour tenter de définir un autre Chemin topologique sur lequel essayer à nouveau de réaliser la planification fine. L'architecture proposée tend à réduire la survenue de tels problèmes en augmentant la pertinence du Chemin topologique retenu par le contrôle de la planification grossière grâce à l'Information sémantique, mais elle ne l'exclut tout de même pas. Dans le chapitre suivant, nous verrons comment l'interaction avec un opérateur humain permet de pallier cette difficulté.

3.3 Implémentation de la planification de trajectoire

L'implémentation du planificateur de trajectoire multi-niveaux proposé a fait l'objet du développement d'une librairie dédiée.

Cette section présente les principales classes développées pour la planification (section 3.3.1), et comment elles sont mises en œuvre dans le processus de planification de trajectoire (section 3.3.2). Nous verrons aussi que certaines opérations de ce processus peuvent être parallélisées pour accélérer la définition de la trajectoire.

Ce planificateur a été développé en C++ et met en œuvre les bibliothèques libres (FCL [FCL 14] pour la détection de collision et Boost [Boost 14] pour paralléliser certains processus).

3.3.1 Modèle statique : classes et responsabilités

Cette section présente tout d'abord les classes crées pour l'implémentation du planificateur de trajectoire (section 3.3.1.1). L'interfaçage du planificateur avec le modèle de l'environnement est ensuite détaillé (section 3.3.1.2).

3.3.1.1 Planificateur de trajectoire

Le modèle de domaine UML de la figure 3.3 montre les différents concepts évoqués dans ce chapitre qui ont fait l'objet de la création de classes.

Le Planificateur sémantique gère l'ensemble du processus de planification. Il utilise un Interpréteur sémantique pour réaliser les raisonnements sur l'Information sémantique.

L'Interpréteur sémantique permet une adaptation aisée de la structure du planificateur de trajectoire à une large gamme d'applications. Le traitement de l'information sémantique étant concentré au sein de cette classe, l'adaptation du planificateur à de nouvelles applications (à travers une sémantique spécifique traduisant l'expertise associée à cette application) ne nécessitera de re-développer que cette classe. À terme, l'Interpréteur sémantique pourra encapsuler

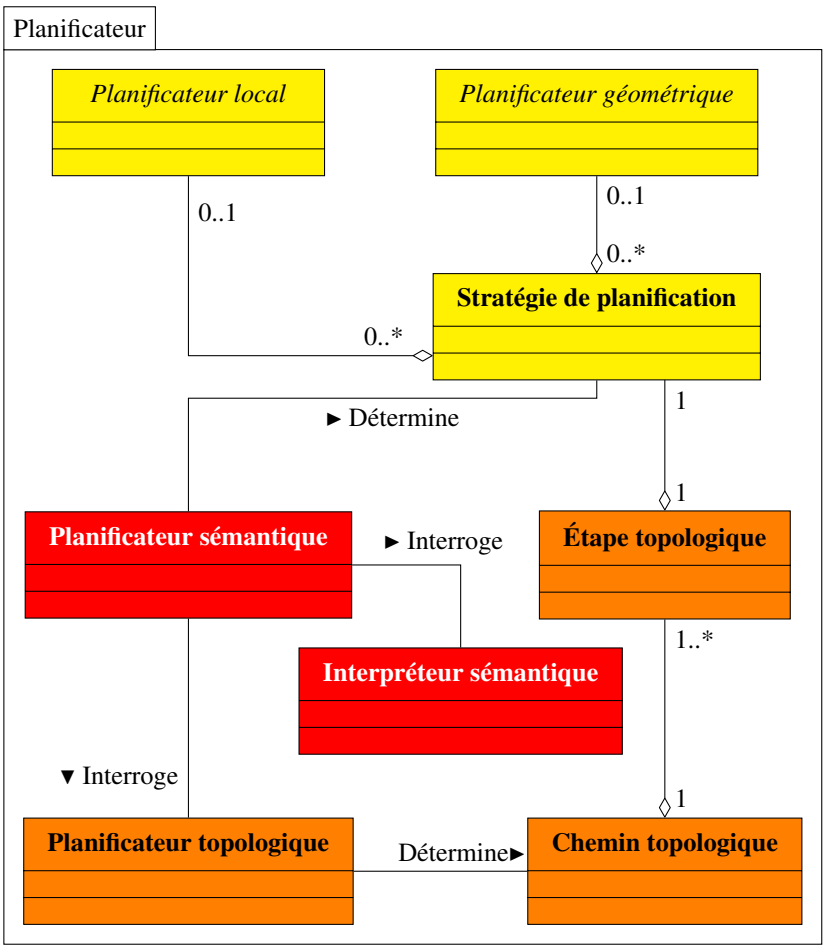


FIGURE 3.3 – Modèle de domaine UML du planificateur de trajectoire multi-niveaux.

des méthodes mettant en œuvre des outils de représentation et d'exploitation de la connaissance (tels que les ontologies ou les graphes conceptuels).

L'Interpréteur sémantique permet donc au Planificateur sémantique de contrôler la définition du Chemin topologique par le Planificateur topologique en déterminant les coûts du Graphe topologique. Le Chemin topologique défini par le Planificateur topologique est l'agrégation des Étapes topologiques qui le composent.

L'Interpréteur sémantique est aussi mis en œuvre par le Planificateur sémantique pour déterminer les Stratégies de planification associées à chaque Étape topologique. Pour cela, il associe, à chaque Étape topologique une technique de planification dans le niveau géométrique du modèle de l'Espace libre et/ou une technique de planification de trajectoire 6D. Ces techniques sont définies dans les classes dérivées des classes Planificateur géométrique et Planificateur local (figure 3.4).

3.3.1.2 Relation avec le modèle de l'environnement

La figure 3.5 montre comment les classes du planificateur de trajectoire s'interfacent avec celles du modèle de l'environnement présenté dans le chapitre 2. Chacun des niveaux du modèle de l'environnement est utilisé par un planificateur dédié :

Le Planificateur sémantique accède à l'Information sémantique des Lieux et Corps ri-

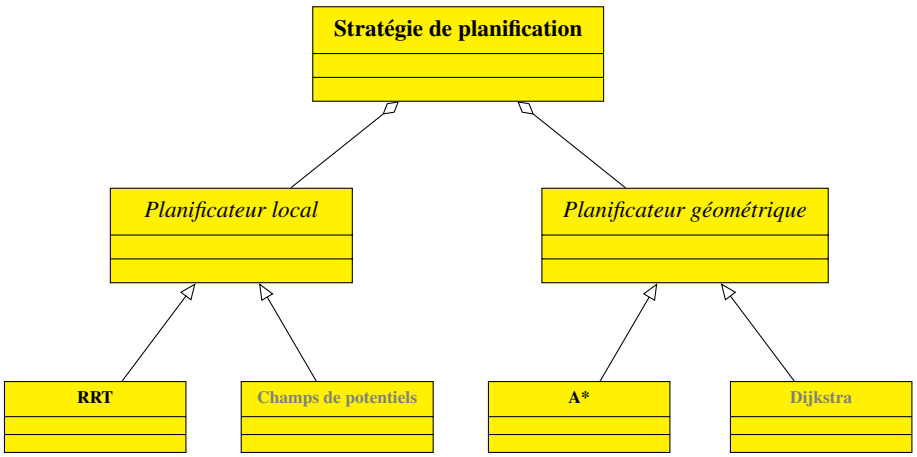


FIGURE 3.4 – Diagramme de classes UML des Stratégies de planification

gides et les transmet à l'Interpréteur sémantique pour réaliser les raisonnements permettant de définir les coûts du Graphe topologique et les Stratégies de planification. Il coordonne la mise en œuvre des différents planificateurs au cours du processus de planification

Le Planificateur topologique exploite la représentation matricielle du Graphe topologique pour définir un Chemin topologique. Une fois le Chemin topologique créé, les différentes Étapes topologiques qui le composent sont instanciées et agrégées dans le Chemin topologique.

Les Planificateurs géométriques utilisent le niveau géométrique du modèle de l'Espace libre (la décomposition octree) pour définir un "tunnel" constitué d'un ensemble connexe de Cellules géométriques libres permettant de parcourir une Étape topologique. La portion du niveau géométrique de l'Espace libre exploré par le Planificateur géométrique est limitée à la Zone du Lieu concerné par l'Étape topologique. Le "tunnel" défini par le Planificateur géométrique est sauvegardé dans l'Étape topologique.

Les Planificateurs locaux définissent une trajectoire 6D sans collision avec les différents Corps rigides de l'environnement. La portion du niveau géométrique du modèle de l'Espace libre considéré par le planificateur local est limité au "tunnel" défini par le Planificateur géométrique s'il existe, à la Zone du Lieu concerné par l'Étape topologique sinon. L'ensemble des jalons géométriques définis pas le Planificateur local est sauvegardé dans l'Étape topologique.

3.3.2 Algorithmes, processus et parallélisation

Cette section décrit comment les processus de planification fine et grossière ont été implémentés en C++ à partir du modèle de domaine introduit dans la figure 3.3.

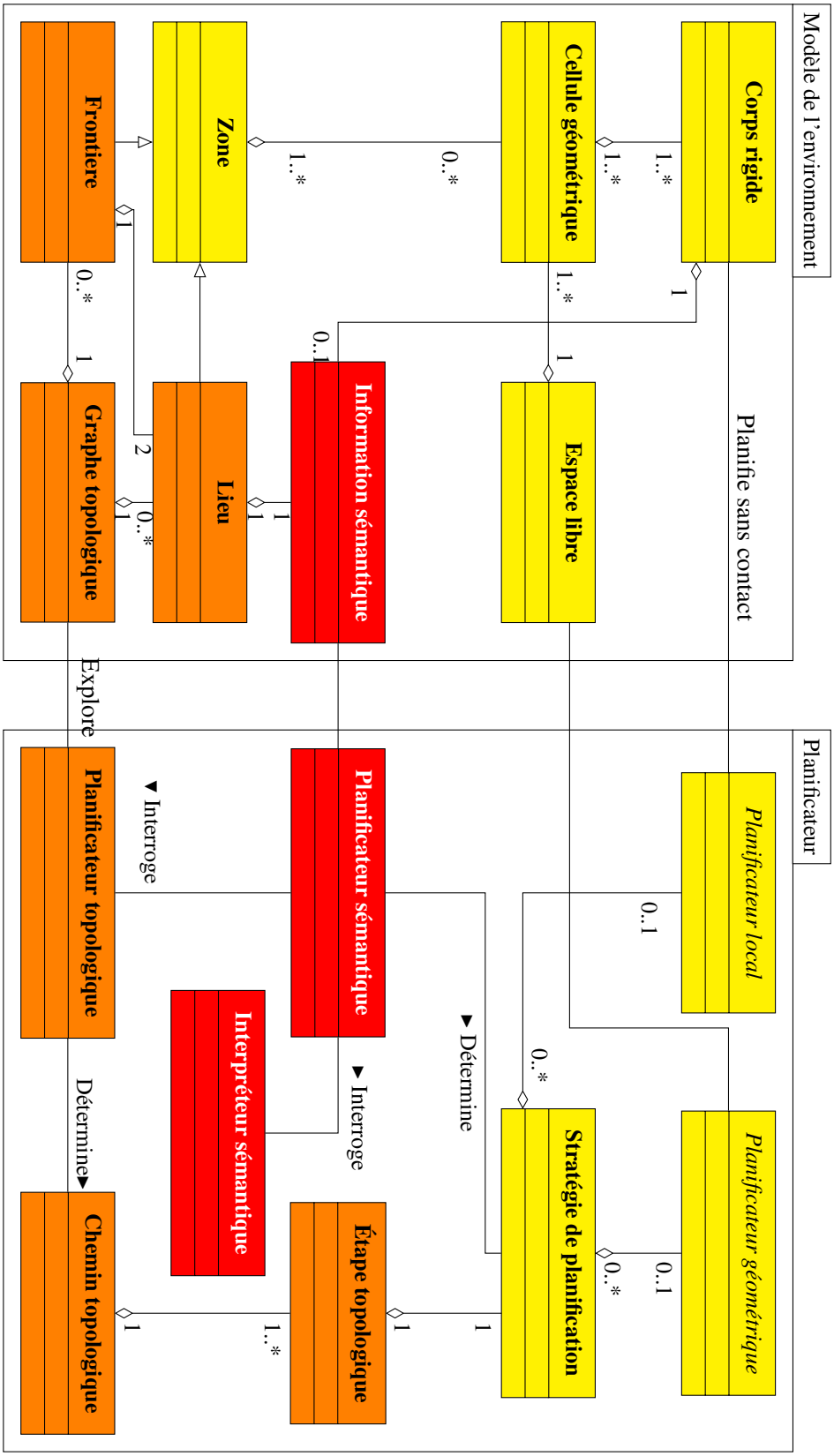


FIGURE 3.5 – Modèle de domaine UML du planificateur de trajectoire et de l'environnement.

3.3.2.1 Planification grossière

La mise en œuvre des différents objets pour la planification grossière est illustrée dans le diagramme de séquence UML de la figure 3.6. On peut y voir une initialisation du processus par la transmission de la requête de planification au `Planificateur topologique` (positions initiale et objectif extraites de la requête de planification par le `Planificateur sémantique`) qui permet de mettre à jour le `Graphe topologique` avec les nœuds correspondants aux configurations initiale et finale de la requête de planification.

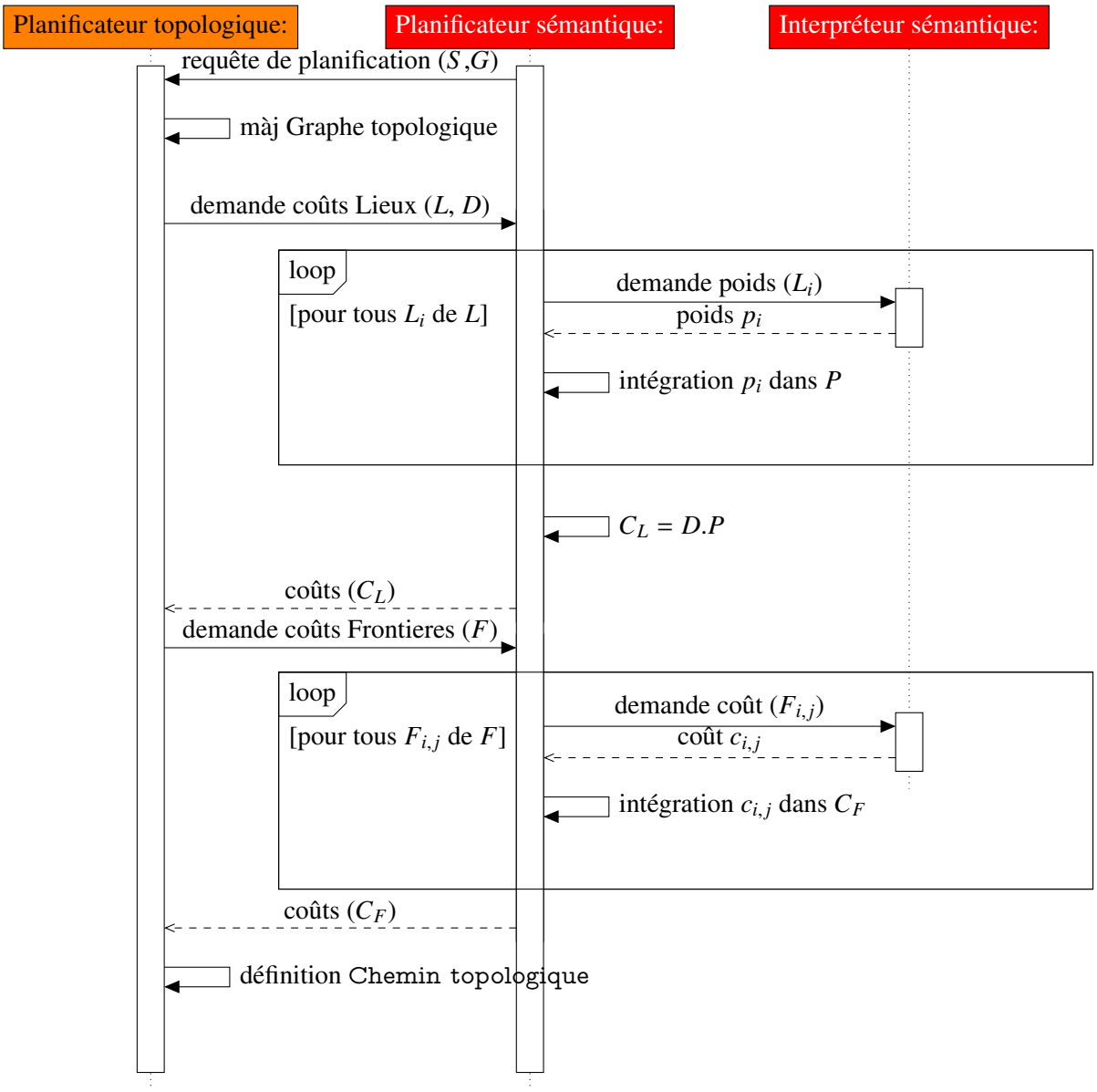


FIGURE 3.6 – Diagramme de séquence de la planification grossière

Une fois cette mise à jour faite, le `Planificateur topologique` interroge le `Planificateur sémantique` pour déterminer les coûts des arcs du `Graphe topologique`. Pour cela, la table de `Lieux` qui est transmise au `Planificateur sémantique` est utilisée pour générer une matrice de poids P grâce à l'`Interpréteur sémantique`. Lorsque la matrice P est constituée, le `Planificateur sémantique` applique ces poids à la matrice des distances D grâce à un produit de

Hadamard [1] pour générer la matrice des coûts C_L. Pour finir, la matrice C_L ainsi constituée est transmise au Planificateur topologique par le Planificateur sémantique.

De même, le vecteur de coûts des Frontieres C_F est construit par le Planificateur sémantique avec l'aide de l'Interpréteur sémantique à partir de la table des Frontieres transmise par le Planificateur topologique.

Enfin, le Planificateur topologique construit le Chemin topologique. Pour cela, il détermine le chemin du Graphe topologique présentant le coût le plus faible en explorant le Graphe à l'aide d'un algorithme de Dijkstra. Le résultat obtenu par cet algorithme est utilisé pour construire les Étapes topologiques qui agrègent un Lieu à traverser et une Frontiere à atteindre. Un jalon géométrique (configuration) est défini aléatoirement au sein de la Frontiere (loi uniforme en position et en orientation) à atteindre de chaque Étape topologique (G pour la dernière). les Étapes topologiques ainsi créées sont agrégées de manière ordonnée dans le Chemin topologique.

3.3.2.2 Planification fine

La planification fine met en œuvre différents objets comme l'illustre le diagramme de séquence UML de la figure 3.7. Pour chaque Étape topologique préalablement définie lors de la planification grossière, le Planificateur sémantique, par l'intermédiaire de l'Interpréteur sémantique définit une Stratégie de planification. Cette Stratégie de planification est ensuite exécutée. Pour cela, le Planificateur géométrique et le Planificateur local sont mis en œuvre s'ils sont instanciés (si aucun autre jalon géométrique n'est nécessaire pour parcourir l'Étape topologique, aucun Planificateur géométrique et Planificateur local n'est instancié dans la Stratégie de planification permettant de la parcourir). La trajectoire ainsi trouvée par la Stratégie de planification est ensuite retournée et mémorisée dans l'Étape topologique.

Les différentes Étapes topologiques étant indépendantes les unes des autres et les jalons géométriques les bornant étant définis lors de la planification grossière, la planification fine des différentes étapes peut être parallélisée pour accélérer la définition de la trajectoire. Ainsi, chaque itération de la boucle sur les Étapes topologiques pour la définition et l'exécution des Stratégies de planification peut correspondre à la création d'un *thread* dédié à l'ensemble de ces opérations pour une Étape topologique.

3.4 Conclusion sur la planification de trajectoire réalisée

La planification de trajectoire proposée exploite l'ensemble des éléments du modèle de l'environnement introduit dans le chapitre 2. La mise en œuvre de cette variété d'informations permet d'accélérer le processus de planification de trajectoire tout en améliorant la pertinence de la trajectoire planifiée. L'amélioration de la pertinence de la trajectoire retenue est obtenue en utilisant l'Information sémantique de l'environnement pour en explorer la topologie et y

1. Produit matriciel dont les éléments de la matrice résultat est le produit terme à terme des matrices multipliées

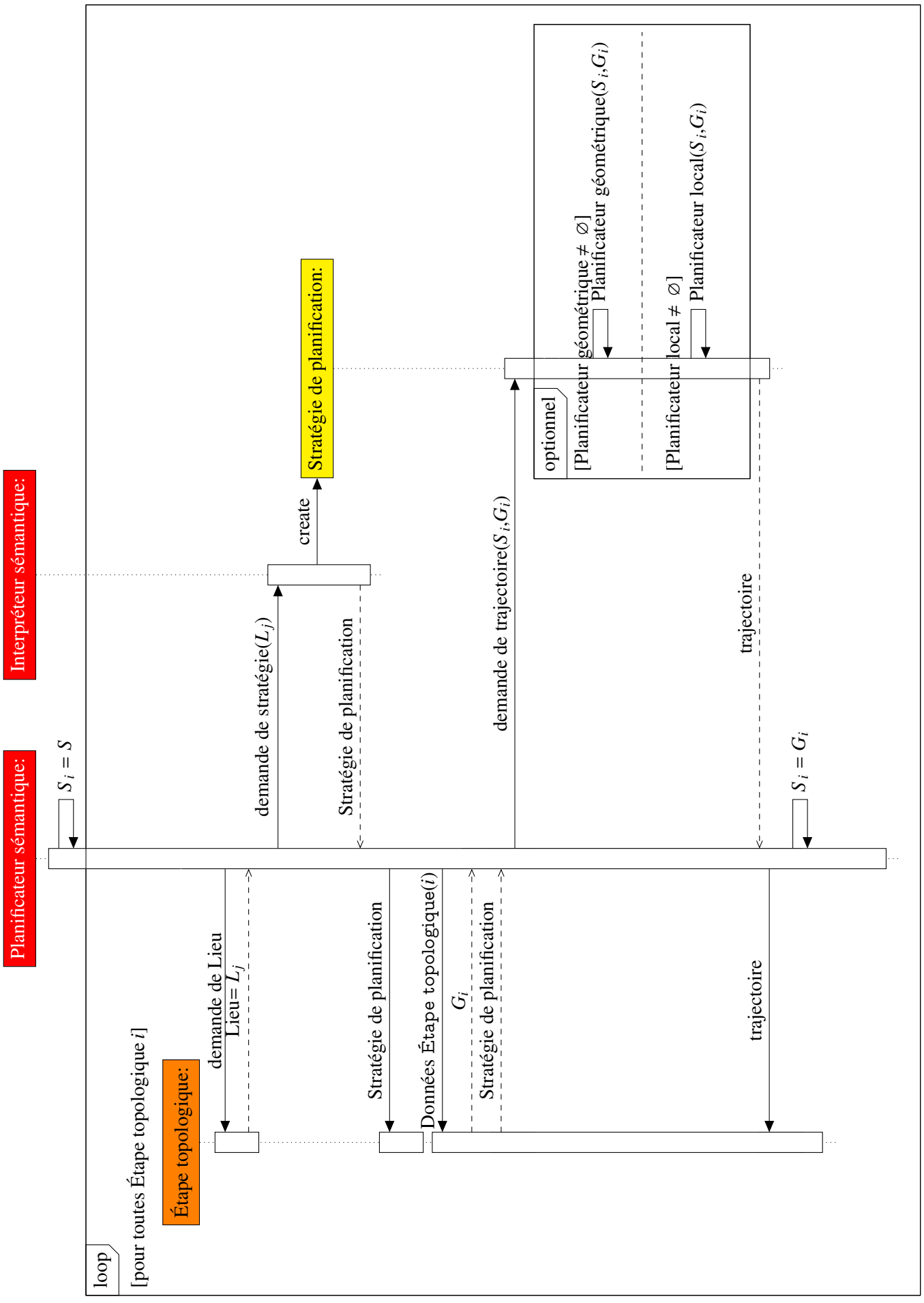


FIGURE 3.7 – Diagramme de séquence de la planification fine

définir un Chemin topologique qui privilégie les Lieux les plus adaptés à la requête de planification. L'accélération du processus est, elle, due à différentes contributions de la planification proposée :

- le Chemin topologique obtenu pour la planification fine est partitionné en Étapes topologiques qui constituent des requêtes de planification fine indépendantes pouvant être traitées en parallèle,
- la planification grossière, sur le niveau topologique du modèle de l'Espace libre, permet de réduire la part de l'environnement à explorer lors la planification fine (les Lieux non retenus dans le Chemin topologique ne sont pas explorés),
- la Stratégie de planification fine est adaptée au contexte local de chaque Étape topologique (décrit dans l'Information sémantique du Lieu à traverser).

L'architecture d'implémentation retenue pour cette stratégie de planification est modulaire et permet d'intégrer de nouvelles techniques de planification géométriques. Ainsi, l'enrichissement des bibliothèques de Planificateurs géométriques et de Planificateurs locaux, avec la définition des contextes dans lesquels ils doivent être mis en œuvre, permettra d'optimiser encore davantage l'ensemble du processus de planification. Par exemple, la définition de Planificateurs locaux permettant de définir des trajectoires au contact des surfaces des obstacles permettra de mieux gérer les Lieux fortement contraints géométriquement et/ou les dernières manipulations pour l'assemblage de composants de systèmes industriels.

Dans le chapitre 4, nous allons voir comment cette planification de trajectoire est mise en œuvre dans des applications de RV (immersives et interactives). Dans ces applications, le planificateur de trajectoire est utilisé pour assister un opérateur lors de la manipulation d'objets virtuels. Il est donc nécessaire de définir les modalités d'interaction et de partage de contrôle entre le planificateur automatique et l'humain immergé dans la simulation de RV.

Ces modalités doivent exploiter l'ensemble des données disponibles du modèle de l'environnement pour que, d'une part, l'opérateur soit en mesure d'intervenir dans la définition de la trajectoire et comprenne la proposition faite par le planificateur de trajectoire, et d'autre part, pour que le planificateur de trajectoire interprète au mieux les actions de l'opérateur et intègre son intention dans la trajectoire planifiée.

Nous allons aussi voir comment l'interaction peut être utilisée pour pallier les limites du processus de planification proposé (absence de feedback entre les planifications fine et grossière) et même d'exploiter au mieux le résultat de la planification grossière, même lorsque la planification fine échoue.

Chapitre 4

Planification interactive et partage de contrôle

Table **Page**

4.1 État de l'art de la planification interactive et du partage de contrôle 80

4.1.1 Planification interactive de trajectoire 81

4.1.1.1 Planification interactive de trajectoire en deux phases 81

4.1.1.2 Planification de trajectoire et interaction continue 81

4.1.1.3 Synthèse de l'état de l'art de la planification interactive 84

4.1.2 Partage de contrôle 85

4.1.2.1 Partage d'autorité 86

4.1.2.2 Prédiction d'intention 88

4.1.2.3 Synthèse de l'état de l'art du partage de contrôle 90

4.2 Contribution 91

4.2.1 Planification de trajectoire et interaction 91

4.2.1.1 Modes d'interaction 91

4.2.1.2 Adaptation du planificateur automatique de trajectoire 93

4.2.1.3 Synthèse de l'interaction pour la planification de trajectoire 93

4.2.2 Partage de contrôle proposé 94

4.2.2.1 Partage d'autorité pour la manipulation de `Corps rigides` virtuels . . . 95

4.2.2.2 Prédiction d'intention et re-planification topologique 96

4.2.3 Partage de contrôle et architecture multi-niveaux 98

4.2.4 Synthèse de l'interaction pour la planification de trajectoire proposée 100

4.3 Implémentation de l'interaction pour la planification de trajectoire 102

4.3.1 Modèle statique : classes et responsabilités 102

4.3.2 Algorithme, processus et parallélisation 103

4.3.2.1 Parallélisation de l'application de RV et de la re-planification 104

4.3.2.2 Génération d'assistance 105

4.3.2.3 Prédiction des intentions de l'opérateur 106

4.3.2.4 Contrôle de l'exécution de la tâche 106

4.3.2.5 Application de planification interactive de trajectoire en RV 108

4.4 Conclusion sur l'interaction pour la planification de trajectoire 109

Les chapitres 2 et 3 ont présenté respectivement le modèle multi-niveaux de l'environnement et le planificateur de trajectoire associé développés au cours de cette thèse.

La mise en œuvre de cette planification de trajectoire dans une simulation interactive nécessite le développement de modes d'interaction et de partage de contrôle entre l'opérateur humain et la planification de trajectoire.

Ce chapitre présente les modes d'interaction et de partage de contrôle que nous proposons. Nous présentons tout d'abord un état de l'art de la planification interactive de trajectoire et du partage de contrôle (section 4.1). Nous décrirons ensuite les modes d'interaction et de partage de contrôle multi-niveaux que nous avons développés (section 4.2). Enfin, nous présenterons leur implémentation (section 4.3).

4.1 État de l'art de la planification interactive et du partage de contrôle

Dans cet état de l'art, nous allons dans un premier temps nous intéresser à la planification interactive de trajectoire. Nous allons voir comment la collaboration entre planificateur automatique de trajectoire et opérateur humain peut s'organiser en deux phases successives, chacune allouée à une des entités en interaction (section 4.1.1.1). Nous verrons aussi que d'autres planificateurs interactifs de trajectoire proposent une collaboration basée sur une interaction continue (section 4.1.1.2).

Nous allons par la suite évoquer des techniques de partage de contrôle comme le partage d'autorité (section 4.1.2.1) et la prédiction d'intention (section 4.1.2.2). Ces techniques de partage de contrôle entre système automatique et humain sont issues de domaines tels que le pilotage de véhicules semi-autonomes et la télé-opération de robots. Nous avons mis en œuvre de telles techniques de partage de contrôle pour améliorer l'ergonomie de l'interaction entre le planificateur automatique de trajectoire et l'opérateur humain.

4.1.1 Planification interactive de trajectoire

Il existe dans la littérature différents planificateurs de trajectoire dit interactifs permettant à un planificateur automatique et un opérateur humain de collaborer pour définir une trajectoire. Parmi ceux-ci, certains proposent de faire intervenir l'opérateur humain et le planificateur automatique de trajectoire dans deux étapes successives distinctes. D'autres proposent une interaction continue entre l'opérateur humain et le planificateur automatique.

Dans cet état de l'art, après avoir introduit le fonctionnement des planificateurs interactifs de trajectoire en deux phases dans la section 4.1.1.1, nous allons nous intéresser aux planificateurs de trajectoire mettant en œuvre une interaction continue tout au long de l'exécution d'une tâche. Nous allons particulièrement nous intéresser à trois planificateurs interactifs de trajectoire permettant de traiter une tâche complexe (6 ddl ou plus). Leur fonctionnement et la manière dont ils permettent à un planificateur automatique de trajectoire et un opérateur humain de collaborer sont détaillés dans la section 4.1.1.2.

4.1.1.1 Planification interactive de trajectoire en deux phases

Lorsque la planification interactive de trajectoire est réalisée en deux phases dédiées chacune à une des entités en interaction, deux solutions se présentent :

L'opérateur intervient en premier : par exemple, dans un premier temps, l'opérateur définit des points de passage avant qu'un algorithme probabiliste les interconnecte [He 09]. L'opérateur peut aussi définir toute une trajectoire comprenant des interpénétrations avec les obstacles avant que des techniques de reconstruction locale soit mises en œuvre pour redéfinir les portions de trajectoire en collision [Bayazit 01, Yan 13].

La planificateur automatique de trajectoire intervient en premier : le planificateur de trajectoire définit alors une trajectoire utilisée pour assister, guider l'opérateur qui peut infléchir la trajectoire calculée en s'en écartant [Vázquez 07].

4.1.1.2 Planification de trajectoire et interaction continue

Lorsque la requête de planification est simple (petit nombre de ddl), le planificateur automatique de trajectoire peut re-planifier une trajectoire en temps réel de manière à intégrer les manipulations faites par l'opérateur dans la trajectoire planifiée [Patil 10]. Cependant, dans les travaux présentés dans ce mémoire, l'objectif est de définir la trajectoire d'un objet virtuel dans un environnement encombré dans une application de RV. La planification d'une telle trajectoire 6D est une tâche trop complexe pour une telle interaction continue.

Dans la littérature, très peu de travaux proposent une interaction continue entre planificateur automatique de trajectoire et opérateur humain pour une telle tâche. Nous allons ici décrire en détail ceux de Ladevèze [Ladevèze 10a] et de Flavigné [Flavigné 10a] évoqués dans la section 1.2.1. Ces travaux ont abouti au développement de trois planificateurs interactifs de trajectoire pour lesquels les modes d'interaction diffèrent. Pour rappel, on trouve, développés par

Ladevèze, un *planificateur local* et un *planificateur probabiliste avec approche globale* utilisés pour des simulations d'assemblage de systèmes industriels en RV ; et, développé par Flavigné, un algorithme *IRRT* permettant à un opérateur d'indiquer la direction d'exploration d'une technique probabiliste avec approche locale.

Le *planificateur local* Le *planificateur local* de Ladevèze utilise la configuration objectif pour déterminer une force attractive. L'interpolation linéaire entre la configuration courante et la configuration objectif permet de définir la direction de cette force sans tenir compte des obstacles présents dans l'environnement. Cette force attractive est ensuite communiquée à l'opérateur en immersion dans une simulation de RV pour l'aider à réaliser la tâche d'assemblage en le guidant par l'intermédiaire d'une interface haptique.

Ainsi, c'est l'opérateur qui doit prendre en compte les obstacles présents dans l'environnement pour les éviter grâce au retour visuel et haptique (simulation du contact). Il lui revient donc de s'écarter de l'assistance proposée pour contourner les obstacles, la direction de l'assistance étant, elle, re-générée en temps réel. La trajectoire d'assemblage est donc obtenue de manière collaborative.

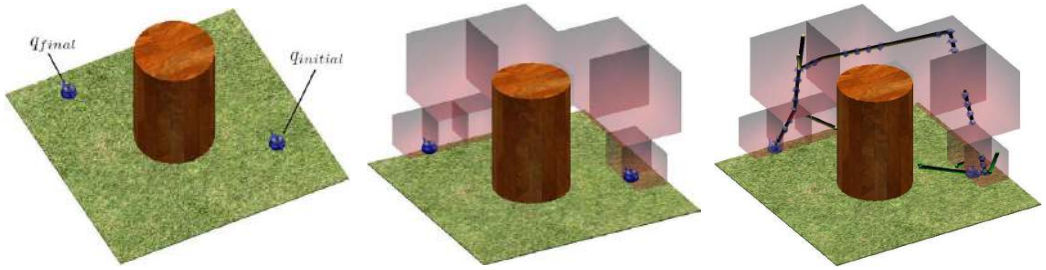
Ce planificateur interactif a le grand avantage de pouvoir en toutes circonstances fournir une assistance à l'utilisateur. En contrepartie, pour des environnements complexes, l'assistance proposée peut être contre-productive. En effet, dans des cas comme l'exploration d'un labyrinthe où certaines portions du chemin demandent de s'éloigner de l'objectif, l'assistance proposée gène la manipulation en s'opposant au mouvement.

Le *planificateur probabiliste avec approche globale* Le planificateur de trajectoire probabiliste avec approche globale proposé par Ladevèze (figure 4.1) s'appuie sur une décomposition octree de l'espace libre 3D en plus de la modélisation des corps rigides de l'environnement pour répondre à une requête de planification (figure 4.1(a)). Le processus de planification est composé de deux phases. Dans la première phase, un algorithme A* est utilisé pour parcourir l'octree et déterminer un "tunnel" 3D (suite de cellules de l'octree connexes) entre les positions des configurations de départ et objectif (figure 4.1(b)). Ce tunnel 3D est utilisé dans la seconde phase pour contraindre l'exploration 6D de l'environnement 3D par un algorithme probabiliste RDT (figure 4.1(c)).

La trajectoire obtenue est alors utilisée pour guider l'utilisateur à l'aide d'une interface haptique. Ce guidage se fait le long de la trajectoire dans une "zone de suivi", et vers la trajectoire hors de cette zone. Lorsque l'utilisateur s'éloigne davantage de la trajectoire calculée, l'ensemble du processus A*/RDT est relancé depuis la configuration courante pour définir une nouvelle trajectoire qui sera utilisée pour guider l'utilisateur. La trajectoire finale est ainsi planifiée interactivement.

Ce planificateur permet de fournir une assistance particulièrement pertinente pour l'assemblage puisque le guidage proposé est défini à partir d'une trajectoire effectivement réalisable. Par contre, il devient inopérant lorsqu'aucune trajectoire ne peut être définie par le planificateur de trajectoire (résolution de l'octree trop faible) ou lorsqu'il y a impossibilité de définir une

trajectoire 6D dans le "tunnel" défini par l'algorithme A*.



(a) Requête de planification (b) tunnel 3D obtenu avec l'algorithme A* sur l'octree (c) trajectoire obtenue avec l'algorithme RDT

FIGURE 4.1 – *Planificateur probabiliste avec approche globale* de Ladevèze [Ladevèze 10a]

L'*IRRT* L'*IRRT* proposé par Flavigné [Flavigné 10a] permet à l'utilisateur d'influencer la propagation d'un algorithme RRT. Pour cela, le planificateur de trajectoire et l'utilisateur échangent des pseudo forces qui sont utilisées pour contraindre le jet aléatoire de l'algorithme RRT (figure 4.2).

Une première pseudo force F_a est calculée par le planificateur automatique de trajectoire à partir des nœuds de l'arbre RRT courant et de leur proximité aux obstacles. Cette pseudo force vise à orienter le jet des prochaines configurations pour permettre à l'arbre RRT de se déployer loin des obstacles. Une seconde force F_u est transmise par l'utilisateur. Une force résultante F_r est, elle, calculée comme la somme pondérée des deux forces précédentes (équation 4.1).

$$F_r = \alpha F_u + (1 - \alpha) F_a \tag{4.1}$$

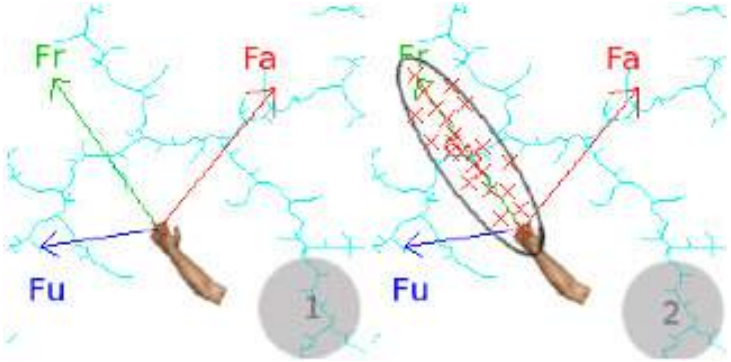


FIGURE 4.2 – RRT interactif proposé par Flavigné [Flavigné 10a]

La force F_r ainsi générée est utilisée pour définir une ellipsoïde dans laquelle est contraint le jet aléatoire de l'algorithme RRT comme illustré par la figure 4.2

Le choix du paramètre α est donc critique pour cet algorithme. En effet, le choix de ce paramètre peut amener l'algorithme à se comporter comme un algorithme RRT classique dont l'exploration est contrôlée par le développement préalable ($\alpha = 0$) ou comme un suivi de cible ($\alpha = 1$).

4.1.1.3 Synthèse de l'état de l'art de la planification interactive

On trouve donc dans la littérature deux formes de planificateurs interactifs de trajectoire.

La première fait intervenir successivement, dans un ordre ou dans l'autre, l'opérateur humain et le planificateur automatique. Dans ce cas, la première phase réalisée par la première des deux entités constitue un pré-traitement facilitant les opérations de la seconde entité lors de la seconde phase.

La seconde forme de planificateurs interactifs de trajectoire fait collaborer l'opérateur humain et le planificateur automatique de trajectoire de manière continue. De tels planificateurs interactifs sont rares pour des problèmes complexes (ayant un grand nombre de ddl).

Les trois planificateurs interactifs dont nous avons détaillé le fonctionnement ne traitent qu'avec des modèles géométriques de l'environnement. Ils mettent différemment en œuvre l'utilisateur immergé. Ladevèze propose de considérer ces planificateurs interactifs vis-à-vis de la charge cognitive qu'ils impliquent pour l'utilisateur, et de la charge de calcul pour le système automatique (figure 4.3) [Ladevèze 10a].

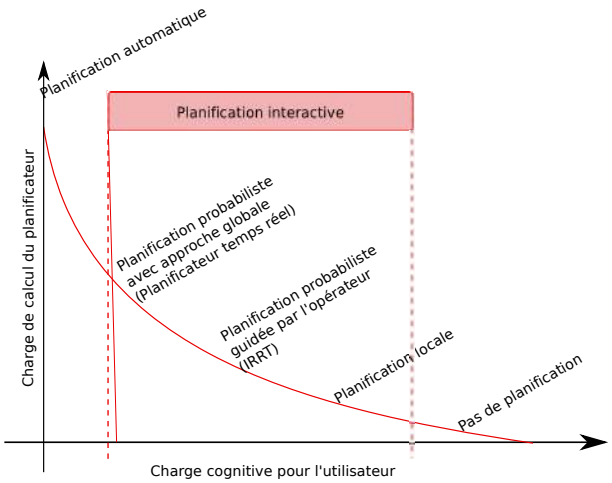


FIGURE 4.3 – Charge de calcul vs charge cognitive pour les différents planificateurs interactifs proposé par [Ladevèze 10a].

Ces trois planificateurs interactifs de trajectoire répondent à des besoins différents. L'un ou l'autre de ces planificateurs devrait donc être mis en œuvre en fonction de la tâche à réaliser. Par exemple, pour des environnements virtuels complexes, difficiles à appréhender, il convient de diminuer la charge cognitive de l'opérateur et donc d'utiliser le *Planificateur probabiliste avec approche globale*. À l'inverse, le *Planificateur local* est tout à fait efficace dans des environnements dégagés. L'*IRRT* constitue lui une solution intermédiaire.

Cependant, si ces planificateurs répondent à des besoins différents, aucune méthode n'est proposée pour déterminer lequel est le mieux adapté à une tâche donnée. De plus, on peut aussi considérer que pour une même tâche, différents planificateurs interactifs devraient être mis en œuvre pour différentes portions de la trajectoire. Par exemple, pour l'assemblage de systèmes industriels, les premières portions de trajectoire à l'extérieur du système se font dans un espace dégagé, alors que les dernières se font dans un espace beaucoup plus encombré.

4.1.2 Partage de contrôle

On parle de partage de contrôle lorsqu'au moins deux entités contrôlent ensemble un même système. Les entités peuvent être des humains ou des systèmes automatiques. Dans notre cas, nous nous intéressons au partage de contrôle entre un humain et un système automatique.

Une telle collaboration est généralement mise en œuvre pour assister un humain dans le pilotage d'un système afin de réaliser une tâche. Ces systèmes de partage de contrôle ont été classés en systèmes d'assistance et systèmes d'extension [Sheridan 78]. Les premiers assistent l'humain avec qui le contrôle est partagé pour réaliser sa tâche en allégeant sa charge cognitive (par exemple, pour le maintien de distance de sécurité entre véhicules). Les seconds, eux, amènent des capacités de contrôle sur le système que l'humain ne peut pas avoir (l'assistance au freinage *Antiblockiersystem* (allemand) (ABS) [Leiber 80] et le système de contrôle de stabilité *Electronic Stability Program* (ESP) [Müller 94] contrôlent individuellement les roues d'un véhicule).

Dès les années 80, différents problèmes ont été identifiés en particulier pour des systèmes d'aide au pilotage d'avions [Wiener 80]. Ces problèmes consistent en :

- la mauvaise interprétation des signaux du système automatisé par l'opérateur,
- la mauvaise utilisation du système par l'opérateur,
- le manquement de dysfonctionnement du système par l'opérateur,
- la perte de compétence des opérateurs.

Notons que les problèmes associés à la collaboration d'humains avec des systèmes automatiques ont été identifiés initialement pour le contrôle de véhicules semi-autonomes (plus précisément les avions [Wiener 80]). Cependant, c'est sur les véhicules terrestres (en particulier, les voitures) et les robots télé-opérés que les recherches portent aujourd'hui le plus, du fait de la facilité d'accès à de tels systèmes, et des possibilités d'implémentation et de test des propositions.

Nous distinguons deux éléments clés pour le partage de contrôle :

1. le système de contrôle automatique génère des commandes pour contrôler le système.
2. le contrôle d'autorité permet de mixer les commandes issues du système automatique et celles de l'humain pour produire celles soumises au système [Mercier 10].

En synthèse, nous pouvons représenter le cadre général du partage de contrôle comme illustré par la figure 4.4. Les éléments en jaune représentent les composants du système automatique qui assistent l'humain.

Dans cette figure, la "Commande système" traduit le fait que les commandes générées par l'utilisateur et le système d'assistance peuvent être définies dans un espace différent que celui des actionneurs du système. Par exemple, dans le cas d'un robot télé-opéré, l'opérateur contrôle

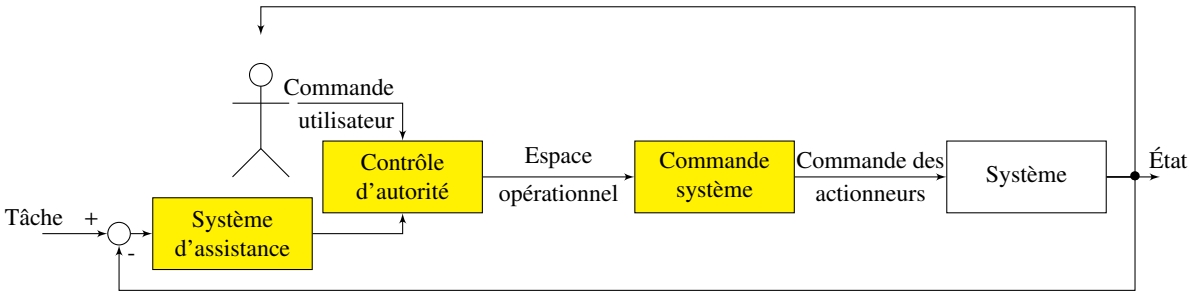


FIGURE 4.4 – Cadre général du partage de contrôle.

souvent le robot dans l'espace opérationnel et la "Commande système" pilote les actionneurs [Sheridan 78, Debus 01, Tarault 05, Kofman 05, Smith 08, You 12].

Lorsque le système peut réaliser une diversité de tâches, le partage de contrôle peut être complété par une système de prédiction d'intention qui a pour rôle de définir la tâche en cours de réalisation sur laquelle le système d'assistance doit se baser pour générer ses commandes.

Il apparait que la conception de systèmes nécessitant la collaboration d'un humain et d'un système automatique passe nécessairement par la définition des rôles de chacun.

La définition des rôles consiste à définir quelles seront les opérations à la charge du système automatique, et donc, de manière complémentaire, celles laissées à l'humain. Elle permet donc, d'une part, de définir pour l'humain ce à quoi il peut s'attendre de la part du système automatique, mais aussi, d'autre part, d'intégrer lors la conception du système automatique, le fait que l'humain prend part au contrôle du système.

En cas d'accident de véhicule semi-autonome, la responsabilité civile et pénale du constructeur comme du conducteur pouvant être engagée, le partage d'autorité entre le système automatique et le conducteur est un point critique. C'est donc à travers ces applications que nous considèrerons les préoccupations liées au partage d'autorité (section 4.1.2.1).

Contrairement aux systèmes d'assistance des véhicules semi-autonomes qui sont fortement spécialisés, les robots télé-opérés sont par nature plus polyvalents. Ces derniers sont donc particulièrement concernés par la prédiction d'intention qui permet de définir la tâche en cours d'exécution et ainsi de fournir une assistance plus pertinente à l'utilisateur (section 4.1.2.2).

4.1.2.1 Partage d'autorité

Le partage d'autorité vise à définir la manière de fusionner les commandes de l'utilisateur et du système d'assistance pour contrôler le système. L'objectif est ici d'arbitrer entre ces deux entités (utilisateur et système d'assistance) et de leur distribuer l'autorité sur le système. L'arbitrage entre les commandes de l'utilisateur et les commandes du système d'assistance peut être dynamique, dans ce cas, l'autorité de ces deux entités sur le système évolue au cours du temps.

Cette question de l'allocation dynamique de l'autorité a été particulièrement étudiée pour les voitures semi-autonomes. En effet, les systèmes d'assistance de ces voitures gèrent des questions critiques. Ils adressent par exemple le problème du maintien de la distance de sécurité entre véhicules [Flemisch 12, Mulder 07, de Winter 04, de Winter 08] et le maintien dans la voie (*lane keeping* en anglais) [Brandt 07, Flemisch 08b, Forsyth 06, Griffiths 05, Mohellebi 09,

Mulder 08, Steele 01, Tsoi 10]. Dans ces applications, le système automatique vise à :

- déterminer une vitesse objectif pour le maintien de la distance de sécurité,
- déterminer l'orientation des roues directionnelles pour le maintien dans la voie.

Pour de tels systèmes, l'assistance est fournie de manière continue et le conducteur reste prioritaire. Cette priorité est toutefois suspendue lorsque l'intégrité de l'équipage (véhicule et passager) et de l'environnement (autres usagers de la route, matériel) est engagée. Ces systèmes doivent donc prendre le contrôle du véhicule lorsque celui ci est menacé, ou lorsque le conducteur le leur laisse. Dans ce dernier cas, il est intéressant de noter que le conducteur peut abandonner explicitement le contrôle du véhicule au système automatique (activation des fonctions). Ou bien, lorsque le système est actif en permanence, celui-ci peut détecter un désengagement du conducteur et, par conséquent confier le pilotage au système automatique.

Cette définition des rôles sert de base pour définir dans quelles circonstances l'autorité doit être attribuée au système automatique ou à l'humain [Bainbridge 83].

Les missions de tels systèmes d'assistance sont établies et leur tâche reste immuable tout au long de leur utilisation. On obtient donc le fonctionnement décrit dans la figure 4.5.

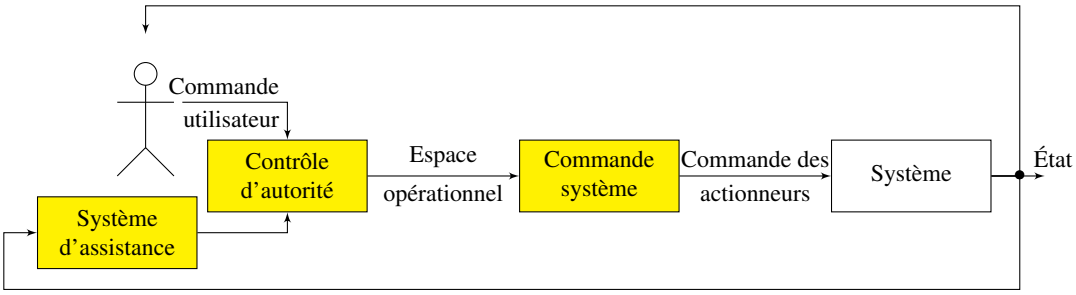


FIGURE 4.5 – Partage d'autorité pour une tâche donnée.

Dès 2003, Flemisch propose de prendre le cheval comme stéréotype de véhicule semi-autonome [Flemisch 03]. Il y perçoit en effet toutes les qualités nécessaires pour deux aspects : la répartition de l'autorité dynamique selon la façon dont le cavalier contrôle sa monture à travers les rênes et les étriers, mais aussi, la capacité à se déplacer en troupeaux (navigation de flottes quand on parle de véhicules semi-autonomes). Flemisch appelle ce paradigme la *H-Metaphor*. Dans notre cas, nous sommes particulièrement intéressés par la répartition dynamique de l'autorité qu'il appelle *H-Mode* [Flemisch 08a]. Le *H-Mode* s'appuie sur :

- une capacité du cheval à assurer la sécurité de l'équipage,
- une grande capacité d'autonomie du cheval,
- une capacité de contrôle fin du cavalier,
- une capacité pour le cheval de percevoir l'implication du cavalier à travers les rênes et les étriers,
- une capacité du cheval à reprendre l'autorité lorsque la sécurité de l'équipage est en jeu.

Cette proposition a été mise en œuvre pour le contrôle de voitures. Par exemple différents modes de retour haptique (utilisation d'un joystick à retour d'effort) ont été testés sur un simulateur de conduite [Kienle 12]. Différents niveaux d'automatisation (et donc différentes répartitions de l'autorité) ont aussi été implémentés, et les modes de basculement entre ces différents niveaux ont été étudiés [Flemisch 12].

De manière générale, pour la mise en œuvre du partage d'autorité entre un humain et un système automatique, trois points doivent faire l'objet d'une attention particulière [Abbink 12] :

1. L'humain doit en permanence rester en situation de contrôle même si le niveau d'automatisation peut varier,

2. L'humain doit en permanence être informé des limites et fonctionnalités du système automatique,

3. L'humain doit être en interaction permanente avec le système automatique, capable d'agir sur les actionneurs et être informé de l'état courant.

Il en résulte que les performances de l'humain doivent être améliorées par le système d'assistance qui, par la même occasion, réduit la charge cognitive pour l'utilisateur.

Ces trois préoccupations peuvent être adressées au travers d'interactions physiques par un partage d'autorité haptique [Abbink 12]. Le partage d'autorité haptique se définit comme une méthode d'interaction système automatique/humain qui *autorise l'humain comme le système automatique à exercer une force sur l'interface de contrôle, dont la sortie (sa position) constitue directement l'entrée du système contrôlé* [Abbink 10]. Dans ce cadre, l'interaction humain/système automatique est une interaction bidirectionnelle riche où les directions et normes relatives des forces exercées par l'humain et le système automatique sur l'interface haptique (et la raideur de ce contrôle) définissent l'autorité que chacun a sur le système.

On trouve, pour le partage d'autorité appliqué à la conduite de voiture, la mise en œuvre d'un retour d'effort sur la pédale d'accélérateur pour aider le conducteur à maintenir une distance de sécurité avec le véhicule qui le précède [de Winter 04, Mulder 07, de Winter 08], ou encore, un retour d'effort appliqué sur le volant pour aider le conducteur à suivre une trajectoire (maintien dans la voie) [Steele 01, Griffiths 05, Forsyth 06, Brandt 07, Flemisch 08b, Mohellebi 09, Mulder 08, Tsoi 10, Flemisch 12].

4.1.2.2 Prédiction d'intention

La prédiction d'intention permet de fournir une assistance lors du pilotage de système polyvalents. Parmi ces systèmes, on trouve par exemple les robots télé-opérés. Pour de tels systèmes, il reste intéressant de fournir une assistance à l'opérateur en dépit de la diversité des tâches qu'ils peuvent mener.

Un premier niveau d'assistance consiste à aider l'opérateur localement. Par exemple, en téléopération de robot, on peut fournir une aide à l'opérateur pour éviter les obstacles (par exemple à l'aide d'un champ de potentiel [Crandall 02]). À partir d'un modèle du robot, une assistance

peut aussi être fournie à l'opérateur en déterminant l'évolution du robot sur un horizon court à l'aide de techniques issues de la commande prédictive [Anderson 10] ou du *minimum jerk* [Weber 09]. Cette assistance est alors associée aux commandes de l'utilisateur à travers une fonction de contrôle d'autorité pour générer les commandes du système.

L'assistance ainsi générée est calculée localement à partir du modèle dynamique du robot et des données de l'environnement. On obtient alors un fonctionnement similaire à celui décrit dans la figure 4.5.

Une assistance plus pertinente peut encore être fournie à l'utilisateur si le système d'assistance ne raisonne plus localement, mais sur un horizon plus large. Autrement dit, l'assistance fournie ne doit plus être générée localement à partir du modèle du système, mais sur une trajectoire globale. Pour cela, il faut que le système automatique connaisse la tâche exécutée par le système télé-opéré.

La tâche peut être connue *a priori* [Aigner 97], dans ce cas, le système d'assistance peut planifier une trajectoire permettant de réaliser la tâche. La trajectoire planifiée est alors utilisée pour guider l'utilisateur à l'aide, par exemple, de champs de potentiels [Aigner 97, Gerdes 01, You 12, Yu 05] ou de guides virtuels (*virtual fixtures* en anglais) [Marayong 02].

Cependant, la tâche en cours de réalisation n'est pas nécessairement explicitement définie et connue par le système d'assistance. Lorsqu'elle n'est pas précisée au système automatique, pour permettre à ce dernier de fournir une assistance plus pertinente à l'utilisateur, il est nécessaire d'identifier automatiquement la tâche en cours. On parle alors de prédiction d'intention de l'utilisateur [Dragan 13]. Il s'agit ici de déterminer, à partir des mouvements initiés par l'opérateur en début d'opération réalisée en téléopération, quelle est la tâche que l'utilisateur est en train de réaliser. On obtient alors le schéma donné dans la figure 4.6.

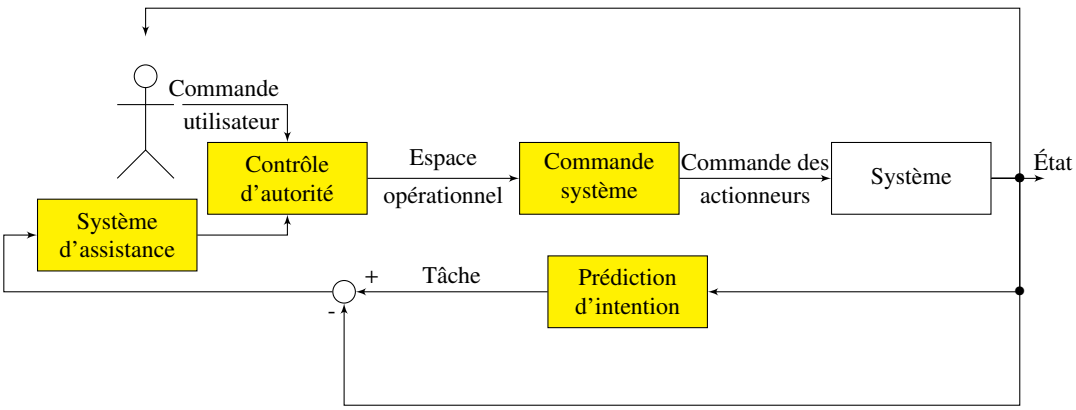


FIGURE 4.6 – Partage de contrôle en téléopération avec prédiction d'intention.

Pour réaliser cette prédiction d'intention, la méthode communément utilisée identifie, dans les premiers mouvements, un comportement parmi ceux modélisés et stockés dans une base de données. Ces comportements peuvent être connus *a priori* [Fagg 04, Aarno 05, Yu 05], ou appris au fur et à mesure que le système est utilisé [Demiris 02, Li 03]. Lorsque la tâche exécutée par l'opérateur est identifiée, une assistance construite grâce à des champs de potentiels ou des guides virtuels peut être proposée à l'opérateur [Fagg 04, Aarno 05, Li 03].

Une méthode de prédiction d'intention qui se dispense de toute bibliothèque de comportements a récemment été proposée [Dragan 13]. Cette méthode exploite des données purement géométriques : d'une part, les positions des objectifs potentiels des mouvements (objets saisissables), et d'autre part, la position et la trajectoire déjà réalisée par le robot téléopéré. Ces éléments permettent d'associer à chaque objectif potentiel une probabilité que cet objectif soit celui ciblé par l'opérateur. La méthode développée compare la trajectoire réalisée à celle qui permet de relier les différents objectifs potentiels depuis la configuration initiale.

La figure 4.7 illustre cette méthode. S et U correspondent aux configurations initiales et courante respectivement. Les G_i sont les objectifs potentiels i. ξ et la trajectoire réalisée et les ξ^* les trajectoires estimées. Dragan propose d'utiliser le produit scalaire entre $\xi_{S\to U}$ et $\xi^*_{U\to G_i}$ pour évaluer la probabilité que G_i soit l'objectif visé par l'utilisateur. Cette technique comporte deux avantages : d'une part, aucune modélisation des comportements associés aux différentes tâches n'est nécessaire, d'autre part, le modèle du robot nécessaire est simple puisque seule la trajectoire de l'organe terminal est utilisée.

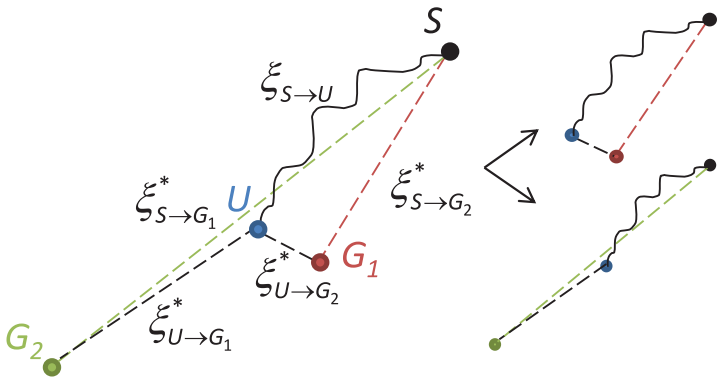


FIGURE 4.7 – Prédiction d'intention proposée par Dragan [Dragan 13]

4.1.2.3 Synthèse de l'état de l'art du partage de contrôle

Cet état de l'art des techniques de partage de contrôle entre systèmes automatiques et opérateur humain pour la conduite de véhicules semi-autonomes et la télé-opération de robots nous permet d'introduire des pistes pour améliorer l'efficacité et l'ergonomie de l'interaction entre le planificateur automatique de trajectoire et un opérateur humain.

En effet, grâce à ces travaux, nous considérons que l'efficacité et l'ergonomie de l'interaction entre le planificateur automatique de trajectoire et un opérateur humain peuvent être améliorées sous deux aspects :

1. le partage d'autorité entre ces deux entités doit permettre, une fois une trajectoire définie par le planificateur de trajectoire, que l'opérateur n'ait pas de difficulté à tester et proposer des alternatives ;

2. la prédiction d'intention doit permettre au planificateur automatique de mieux intégrer l'opérateur dans la planification en interprétant son action de manière précoce et ainsi de tenir compte de ses suggestions au plus tôt lors de l'exécution de la tâche.

4.2 Contribution

Le planificateur de trajectoire décrit dans le chapitre 3 est donc mis en œuvre de manière interactive dans des simulations de manipulation en RV. Cette section décrit tout d'abord les modes d'interaction prévus mais aussi les adaptations amenées au planificateur de trajectoire (section 4.2.1). Ces adaptations visent à limiter le temps de calcul de trajectoires afin de pouvoir intégrer en temps réel les manipulations réalisées par l'opérateur. Elles permettent ainsi de profiter de la complémentarité entre le planificateur automatique de trajectoire et l'opérateur humain pour pallier les limites du planificateur de trajectoire évoquées au chapitre 3. Les modalités de partage de contrôle venant améliorer l'ergonomie de l'interaction entre planificateur automatique et opérateur humain sont ensuite détaillées (section 4.2.2). Le découpage du processus de planification en deux phases, et la mise en œuvre de données sémantiques, topologiques et géométriques dans le processus de planification permettent d'envisager des modes de partage de contrôle très riches. Une brève introduction de ces possibilités offertes par l'architecture est faite (section 4.2.3).

4.2.1 Planification de trajectoire et interaction

Cette section décrit les modes d'interaction définis entre le planificateur de trajectoire et l'opérateur humain en immersion dans l'application de RV (section 4.2.1.1). Elle décrit aussi, les adaptations apportées au planificateur automatique de trajectoire afin de limiter son temps de calcul (nécessaire pour l'utilisation dans un cadre interactif) tout en assurant la définition d'une trajectoire permettant de guider un opérateur humain (section 4.2.1.2).

4.2.1.1 Modes d'interaction

Notre planificateur automatique de trajectoire permettant de jalonner une trajectoire, nous avons choisi d'utiliser un paradigme d'interaction en deux modes adaptés des planificateurs développées par Ladevèze :

1. nous avons décidé d'assister l'opérateur le long de la trajectoire pré-calculée en le guidant successivement vers les différents jalons. Pour réaliser ce guidage, nous avons choisi d'utiliser une interpolation linéaire entre la configuration courante et le prochain jalon à atteindre (guidage inspiré du *planificateur local*) ;

2. par ailleurs, à l'instar de l'utilisation interactive du *planificateur probabiliste avec approche globale*, l'opérateur peut relancer l'ensemble du processus de planification en ne suivant pas le guidage qui lui est proposé.

Nous avons choisi, dans notre cas, d'organiser l'interaction autour d'une force utilisée pour guider l'opérateur dans l'espace 6D. À l'instar du *planificateur local* de Ladevèze [Ladevèze 10b], cette force d'assistance f_a est calculée à partir de l'interpolation linéaire entre la configuration

courante q_c du Corps rigide manipulé O_d et celle du prochain jalon à atteindre q_j. Pour générer f_a, l'interpolation est normalisée et multipliée par un coefficient F permettant d'en définir l'amplitude (équation 4.2).

$$f_a = F\frac{q_j - q_c}{\|q_j - q_c\|} \tag{4.2}$$

Afin que le jalon utilisé pour calculer l'assistance proposée à l'opérateur soit mis à jour au cours de l'exécution de la tâche, il faut détecter lorsque l'opérateur les atteint. La trajectoire exécutée par l'opérateur ne passe pas nécessairement par les jalons mais en passe suffisamment près pour que la distance entre la configuration courante et celle du jalon $(q_j - q_c)$ soit inférieure à un seuil ($\theta_{distance}$). Lorsque c'est le cas, le système automatique détecte que l'opérateur a atteint le jalon et utilise le jalon suivant pour le guider. La figure 4.8 illustre la trajectoire exécutée pour le chemin planifié donnée dans le chapitre 3 (jalons en rouge). Les différentes configurations prises par O_d lors de l'exécution de la tâche y sont affichées en jaune avec une flèche bleue illustrant l'assistance proposée à l'opérateur.

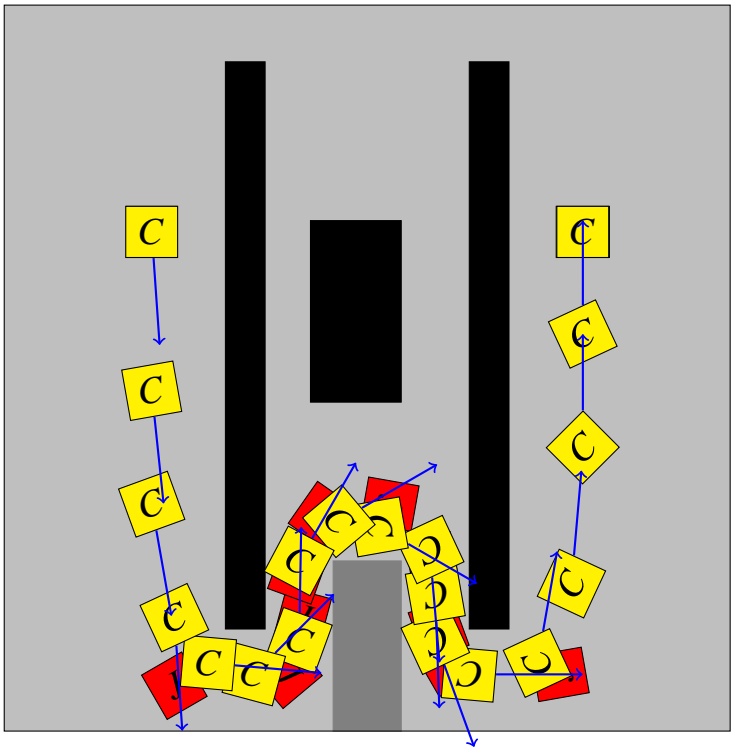


FIGURE 4.8 – Guidage sur une trajectoire pré-calculée

Si l'opérateur ne suit pas l'assistance et déforme fortement la trajectoire définie par le planificateur automatique, il peut produire une trajectoire sortant du Chemin topologique proposé. Lorsque c'est le cas, l'ensemble du processus de planification de trajectoire est relancé afin de déterminer un nouveau Chemin topologique correspondant mieux à la trajectoire réalisée. Ce mode d'interaction est cette fois inspiré du *planificateur probabiliste avec approche globale* de Ladevèze [Ladevèze 10b]. Il a fait l'objet d'adaptations pour mieux exploiter le partitionnement du processus de planification en deux phases et la variété d'informations disponibles dans le modèle de l'environnement développé (sémantique, topologique et géométrique).

4.2.1.2 Adaptation du planificateur automatique de trajectoire

L'interaction entre un opérateur humain et le planificateur automatique de trajectoires permet d'aménager ce dernier pour pallier ses faiblesses introduites par le découpage en deux phases de la planification de trajectoire. L'interaction nous permet en effet de remédier à l'incapacité du planificateur automatique à diagnostiquer la "non-réalisabilité" d'un Chemin topologique en confiant ce diagnostic à l'opérateur. Pour cela, afin de disposer d'une trajectoire permettant d'assister l'opérateur lors du parcours du Chemin topologique défini lors de la planification grossière, la contrainte de non-collision de la trajectoire 6D est localement relaxée.

La relaxation de la contrainte de non collision intervient à deux points du processus de planification automatique :

1. lors de la définition de jalons géométriques au cours de la planification grossière (jalons définis dans les frontières pour construire les requêtes de planification fine),
2. lors de la planification fine, si la stratégie de planification fine n'aboutit pas pour une Étape topologique.

Dans le premier cas, si le planificateur automatique de trajectoire n'est pas en mesure de déterminer un jalon sans collision dans une Frontiere après un nombre n de jets aléatoires, un jalon est défini au centre de la Frontiere sans considérer le problème de collision. Dans le deuxième cas de même, la planification fine de l'Étape topologique est abandonnée et l'utilisateur est guidé vers la prochaine étape grâce au jalon défini dans la Frontiere suivante (interpolation linéaire directe entre les deux jalons définis lors de la planification grossière). Il revient alors à l'opérateur de gérer la présence d'obstacles dans le Lieu à traverser.

Cette relaxation de la contrainte de non-collision permet de plus de limiter le temps de recherche d'une trajectoire. Cet aménagement du processus de planification de trajectoire permet donc de faciliter sa mise en œuvre dans une application interactive temps-réel.

Au cours de l'exécution de la tâche en interaction, lorsque l'opérateur atteint une portion de trajectoire définie par le planificateur automatique comprenant des collisions, il lui revient de diagnostiquer cette difficulté. Il peut alors soit redéfinir localement la trajectoire si le Chemin topologique est réalisable (s'il existe une trajectoire sans collision empruntant ce Chemin topologique), ou déclencher un nouveau processus de planification dans le cas contraire (section 4.2.1.2).

4.2.1.3 Synthèse de l'interaction pour la planification de trajectoire

Grâce aux modes d'interaction choisis et aux dispositions aménagées sur le processus de planification, la complémentarité entre le planificateur automatique de trajectoire et l'opérateur humain permet une collaboration efficace. Le planificateur automatique trouve un chemin dans la topologie de l'environnement et y esquisse une trajectoire qui peut inclure des portions en collision avec les Corps rigides de l'environnement.

Cette trajectoire permet de guider l'opérateur qui peut la corriger localement ou plus radicalement en déclenchant un nouveau processus de planification lors de l'exécution de la tâche.

L'opérateur est ainsi guidé le long du Chemin topologique qui semble le plus pertinent au vu de l'Information sémantique associée aux éléments de l'environnement, même si aucune trajectoire géométrique sans collision n'a pu y être définie.

Lorsque l'opérateur, en exécutant la tâche, atteint une portion de trajectoire que le planificateur automatique n'est pas arrivé à définir sans collision, ses capacités cognitives et son expertise lui permettent, soit de passer la difficulté et ainsi définir une trajectoire géométrique réalisable, soit de constater l'impossibilité d'emprunter cette voie. Dans ce cas, en explorant un chemin alternatif, il déclenche une re-planification réalisant ainsi le bouclage nécessaire entre les deux phases du processus de planification.

4.2.2 Partage de contrôle proposé

L'ergonomie des deux modes d'interaction est améliorée grâce à des techniques de partage de contrôle. Un partage d'autorité pour la manipulation du Corps rigide permet de fournir à la fois une assistance haptique lorsque l'opérateur suit la trajectoire planifiée, et une assistance atténuée (mais toujours perceptible) lorsqu'il explore d'autres alternatives (section 4.2.2.1).

Une prédiction d'intention permet de percevoir de manière précoce quel Chemin topologique l'opérateur souhaite emprunter quand il ne suit pas la trajectoire proposée par le planificateur automatique. Cette prédiction est utilisée pour partager l'autorité sur le niveau topologique de l'environnement en guidant le planificateur de trajectoire sur le Chemin topologique préféré par l'opérateur lors de la re-planification en ligne d'une trajectoire (section 4.2.2.2).

Comme nous ne disposons pour notre système d'aucun modèle analytique pour la dynamique du Corps rigide manipulé ou de l'humain en interaction, le partage de contrôle proposé s'appuie sur la direction instantanée du mouvement m généré par l'opérateur assisté par le système de planification de trajectoire.

La direction du mouvement m est calculée à partir des configurations q_{c_i} prises par le Corps rigide manipulé à chaque échantillonnage i (équation 4.3).

$$m_i = q_{c_i} - q_{c_{i-1}} \tag{4.3}$$

Pour diminuer la sensibilité de nos techniques de partage de contrôle au bruit sur la configuration, on utilise une direction de mouvement filtrée m_{f_i} (équation 4.4).

$$m_{f_i} = \alpha_m m_{f_{i-1}} + (1 - \alpha_m) m_i \tag{4.4}$$

où α_m est choisi entre 0 et 1 en fonction de la fréquence d'échantillonnage et de la bande passante désirée.

Cette direction du mouvement m_{f_i} est utilisée pour :

- moduler la norme de l'assistance proposée afin d'augmenter la sensation de liberté de l'opérateur lorsque celui-ci explore des voies alternatives (l'assistance est moins forte lorsque le mouvement constaté ne suit pas la même direction),

- prédire au plus tôt la volonté, de la part de l'opérateur, d'essayer un autre chemin topologique (il vise une `Frontiere` différente) et ajuster le chemin planifié à cette prédiction d'intention.

4.2.2.1 Partage d'autorité pour la manipulation de `Corps rigides` virtuels

Le partage d'autorité mis en œuvre est inspiré du *H-mode* proposé par Flemisch pour les véhicules semi-autonomes [Flemisch 08a]. L'objectif est ici de diminuer l'autorité du système automatique et ainsi d'augmenter relativement celle de l'humain lorsque ce dernier participe activement à la réalisation de la tâche (il ne se contente pas de suivre l'assistance qui lui est proposée). Dans notre cas, du fait de l'application en réalité virtuelle, nous n'avons pas de considération de sécurité qui, comme dans le cas des véhicules semi-autonomes, doivent donner l'autorité au système automatique pour préserver l'intégrité de l'équipage. L'objectif de la mise en œuvre du *H-mode* est donc de donner plus de liberté à l'opérateur pour explorer l'environnement et rechercher des chemins alternatifs. Ce *H-mode* est illustré dans la figure 4.9 où l'opérateur ne suit pas l'assistance qui lui est proposée (flèche bleue). L'opérateur ne suivant pas l'assistance, la force de cette dernière est diminuée pour laisser l'opérateur libre d'explorer d'autres voies [Cailhol 14a].

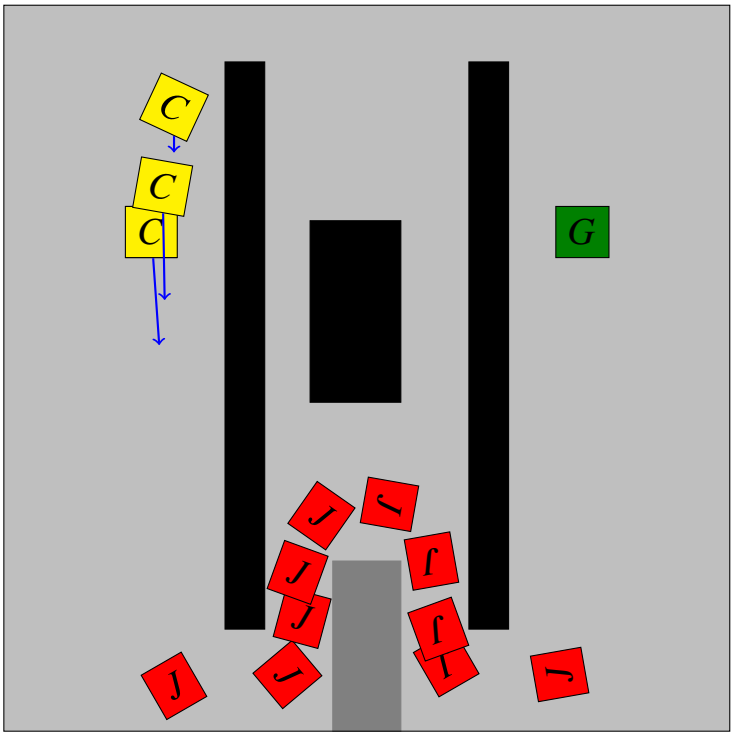


FIGURE 4.9 – Partage d'autorité dynamique

Pour mettre en œuvre ce *H-mode*, nous pondérons la force f_a par un coefficient mod_f traduisant l'implication de l'opérateur dans la manipulation. Ce coefficient est calculé à l'aide du produit scalaire entre le mouvement instantané m et l'assistance proposée f_a (équation 4.5).

$$mod_f = \frac{1 - f_{min}}{2} \left(\frac{m.f_a}{||m.f_a||} + 1 \right) + f_{min} \tag{4.5}$$

où f_{min} permet de maintenir une force résiduelle pour l'assistance et ainsi de maintenir en toutes circonstances l'opérateur informé de l'action du planificateur automatique [Abbink 10].

Le coefficient mod_f est ensuite appliqué à la force f_a pour constituer la force f_g qui guide l'opérateur le long de la trajectoire planifiée (équation 4.6).

$$f_g = mod_f f_a \tag{4.6}$$

La force ainsi obtenue assiste l'opérateur lors de l'exécution de la tâche. Lorsque l'opérateur suit le guidage proposé, cette force est importante et donc facile à suivre. Lorsque l'opérateur cherche des chemins alternatifs, celle-ci est réduite et permet ainsi à l'opérateur d'explorer plus librement l'environnement pour trouver et proposer des trajectoires alternatives.

4.2.2.2 Prédiction d'intention et re-planification topologique

La prédiction d'intention mise en œuvre ici vise à contrôler si la trajectoire exécutée par l'opérateur vise ou pas la Frontiere de l'Étape topologique en cours. Si ce n'est pas le cas, un nouveau Chemin topologique doit être défini pour correspondre à la trajectoire réalisée par l'opérateur et ainsi, permettre au système automatique d'assister l'opérateur dans l'exécution de la tâche en intégrant ses manipulations.

Cette redéfinition interactive de la trajectoire définie automatiquement correspond à un des modes d'interaction introduits précédemment (section 4.2.1). Le premier mode de déclenchement de la re-planification consiste en la détection de sortie du Chemin topologique (en manipulant, l'opérateur quitte le Lieu de l'Étape topologique courante pour un Lieu ne figurant pas au Chemin topologique). Cependant, la détection de la volonté de l'opérateur d'emprunter un autre Chemin topologique peut être anticipée. En effet, lorsque l'opérateur ne suit plus l'assistance comme exprimé dans l'équation 4.7 qui permet de comparer l'angle formé par l'assistance et le mouvement exécuté à un angle de référence θ_{suivi}, il est possible qu'il explore un Chemin topologique alternatif.

$$\frac{f_a.m_f}{||f_a||.||m_f||} \leq \theta_{suivi} \tag{4.7}$$

Il convient alors de prédire quel Chemin topologique alternatif est envisagé par l'opérateur. Pour prédire cette intention de Chemin topologique, nous avons développé une méthode de prédiction d'intention adaptée des travaux de Dragan [Dragan 13]. La prédiction d'intention de Dragan s'appuie sur un ensemble d'objectifs potentiels pour y déterminer lequel est le plus probablement visé par l'opérateur. Cette probabilité est définie grâce à la trajectoire réalisée, et la position des différents objectifs potentiels.

Pour adapter cette méthode, nous avons choisi de considérer l'ensemble des frontières du Lieu de l'Étape topologique courante comme des objectifs potentiels. La probabilité que chacune de ces Frontieres soit visée est définie en fonction de l'angle entre la direction du mouvement m_f et la direction de chacune de ces Frontieres [Cailhol 14a, Cailhol a].

Nous considérons la direction de chacune des frontières comme le vecteur défini par la

configuration courante q_c, et le point de la frontière le plus près de la direction du mouvement. Ce choix est illustré par la figure 4.10 où $N_{i,j}$ représente le point défini pour le calcul de la probabilité associée à la Frontiere $F_{i,j}$.

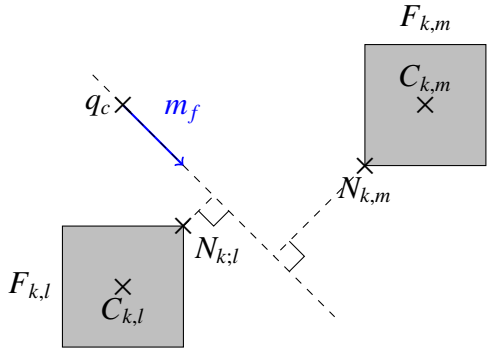


FIGURE 4.10 – Choix du point représentatif de la frontière.

Ce choix de point nous permet d'associer à chaque Frontiere une probabilité calculée à partir du produit scalaire entre m_f et $N_{i,j} - q_c$ mis à l'échelle entre 0 et 1 (équation 4.8). Ce choix de points et de mode de calcul de la probabilité permet de traiter toutes les Frontieres de manière identique malgré la diversité de leurs tailles et de la distance à laquelle elles se trouvent de q_c.

$$P(F_{i,j}) = \frac{1}{2}\left(\frac{m_f.(N_{i,j} - q_c)}{\|m_f\|.\|N_{i,j} - q_c\|} + 1\right) \tag{4.8}$$

Lorsque la direction du mouvement m_f et la direction des frontières $N_{i,j} - q_c$ forment un angle obtus, la Frontiere $F_{i,j}$ ne peut pas être considérée comme visée par l'opérateur. Ainsi, une re-planification de la trajectoire n'est exécutée que lorsque le maximum des probabilités calculées est supérieur à 0,5 et est associé à une Frontiere différente de la Frontiere de l'Étape topologique courante $F_{étape}$ (équation 4.9).

$$\begin{cases} max_{i,j}(P_{F_{i,j}}) > 0.5 \\ max_{i,j}(P_{F_{i,j}}) > P(F_{étape}) \end{cases} \tag{4.9}$$

Lorsque ces deux conditions sont satisfaites, les probabilités calculées sont utilisées pour définir de nouveaux coûts qui viennent s'ajouter aux coûts précédemment établis à l'aide de l'Information sémantique (équation 3.1). Ces nouveaux coûts permettent d'orienter le Planificateur topologique lors de la relance du processus de planification pour que celui-ci privilégie, dans le nouveau Chemin topologique, les Frontieres visées par l'opérateur (équation 4.10). Ces coûts sont donc définis de manière à être faibles pour les Frontieres dont la probabilité est élevée et forts pour celles dont la probabilité est faible. De plus, le coût de la frontière figurant précédemment dans le Chemin topologique F_{tape} est lui défini de manière à pénaliser le chemin précédent (coût prohibitif C_p) pour que le Chemin topologique nouvellement planifié ne soit pas identique au précédent.

$$\begin{cases} C_{n_{a,b}} = k\dfrac{max_{i,j}(P(F_{i,j}))-P(F_{a,b})}{max_{i,j}(P(F_{i,j}))} & \text{if } F_{a,b} \neq F_{étape} \\ C_{n_{a,b}} = C_p & \text{if } F_{a,b} = F_{étape} \end{cases} \tag{4.10}$$

où k est un paramètre de réglage.

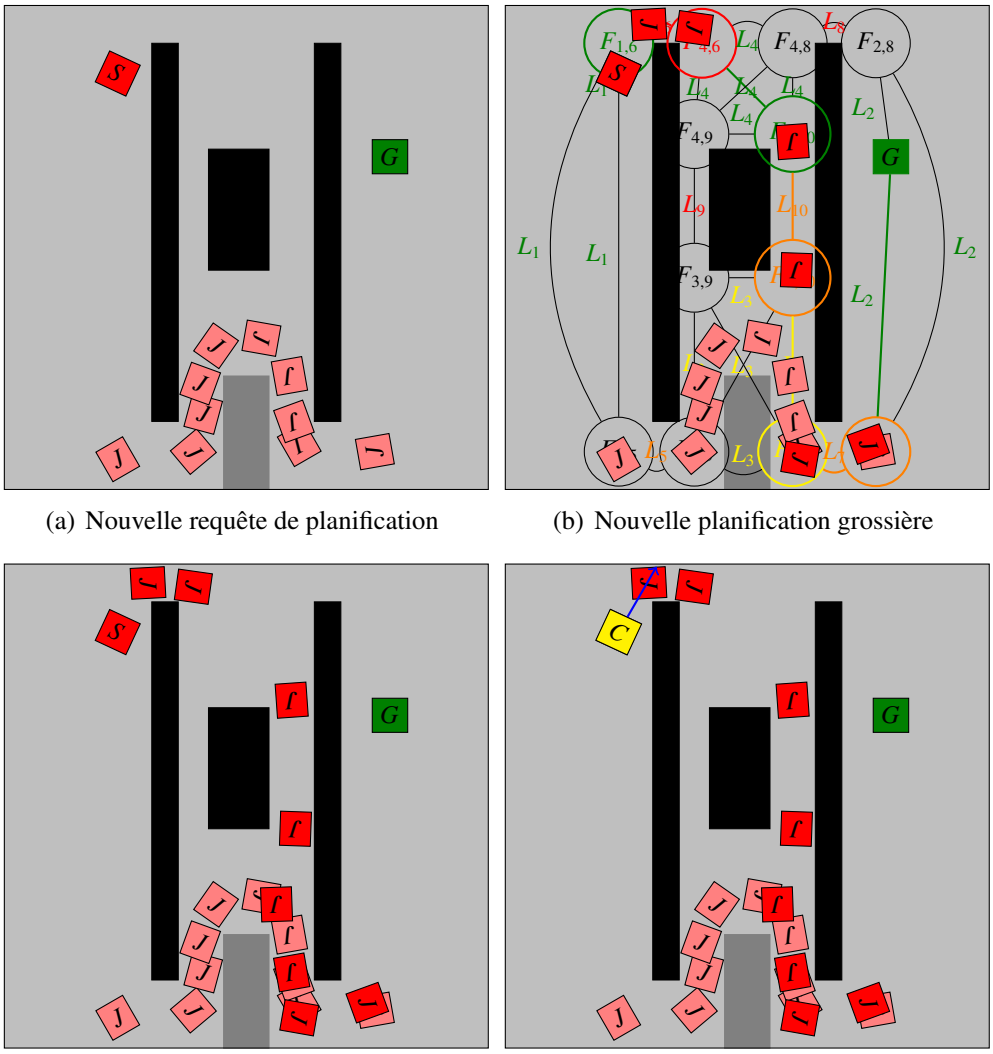
Le processus de re-planification lié à la prédiction d'intention dans le cas des manipulations illustrées dans la figure 4.9 est donné dans la figure 4.11. La figure 4.11(a) fait apparaitre la trajectoire abandonnée en rose tout en illustrant la requête de planification transmise pour une re-planification. La configuration courante y est utilisée comme configuration initiale (S). La figure 4.11(b) illustre la planification grossière réalisée pour répondre à cette requête de planification. Le Chemin topologique retenu traverse les Lieux L_1, L_6, L_4, L_{10}, L_3, L_7 et L_2, et des jalons géométriques sont définis dans les Frontieres $F_{1,6}$, $F_{4,6}$, $F_{4,10}$, $F_{3,10}$, $F_{3,7}$ et $F_{2,7}$. Le Lieu L_6 est traversé malgré sa complexité très élevée car la prédiction d'intention réalisée privilégie la Frontiere $F_{1,6}$ tout en pénalisant la Frontiere $F_{1,5}$. La figure 4.11(c) illustre l'échantillonnage supplémentaire de la trajectoire fait sur le Lieu L_3 du fait de son encombrement par un Corps rigide mobile. Enfin, la figure 4.11(d) illustre l'assistance lorsqu'elle a été mise à jour avec la trajectoire nouvellement planifiée.

4.2.3 Partage de contrôle et architecture multi-niveaux

Le partitionnement de la planification automatique de trajectoire en deux phases (grossière et fine), les modes d'interaction retenus (assistance haptique et re-planification en ligne), et la mise en œuvre de données de différentes natures (sémantiques, topologiques et géométriques) permettent, en impliquant un opérateur dans le processus, d'envisager une grande variété de modes de partage de contrôle. Cette section présente comment le partage d'autorité et la prédiction d'intention peuvent se mettre en œuvre pour les différentes natures de données du modèle de l'environnement.

Utilisation des données géométriques Les données géométriques sont les données de plus bas niveau. C'est à travers ces données que l'interaction s'exprime en premier. En premier lieu, par l'assistance que le planificateur automatique fournit à l'opérateur, et en second lieu, par les manipulations réellement effectuées par l'opérateur. Au niveau géométrique, le partage de contrôle permet de :

Partage d'autorité : comme nous l'avons fait, pour faciliter l'implication de l'opérateur dans le processus de planification, la force de l'assistance qui lui est proposée peut être modulée. L'opérateur peut ainsi être plus libre d'explorer l'environnement, et donc suggérer des alternatives au planificateur automatique de trajectoires. Les modes d'interaction peuvent être complétés pour permettre à l'opérateur, pendant l'exécution de la tâche, de solliciter localement (au niveau d'une Étape topologique) une assistance différente de la part du planificateur de trajectoire automatique. Dans ce cas, la Stratégie de planification de l'Étape topologique peut être redéfinie de manière interactive.



(a) Nouvelle requête de planification

(b) Nouvelle planification grossière

(c) Nouvelle planification fine

(d) Mise à jour de l'assistance

FIGURE 4.11 – Allocation dynamique de l'autorité

Prédiction d'intention : plutôt que d'attendre que l'opérateur entre dans un `Lieu` ne figurant pas dans le `Chemin topologique` pour relancer le processus de planification, nous détectons la volonté de l'opérateur d'essayer des `Chemins topologiques` alternatifs dès qu'un écart entre le mouvement exécuté et l'assistance apparaît. Nous prédisons alors, au plus tôt, par où l'opérateur envisage de quitter le `Lieu` courant pour orienter une nouvelle planification grossière sur un `Chemin topologique` empruntant cette voie.

Utilisation des données topologiques Ces données constitutives du `Chemin topologique` peuvent être utilisées pour partager l'autorité entre le `Planificateur topologique` et l'utilisateur pour la définition du `Chemin topologique`. Au niveau topologique, le partage de contrôle permet de :

Partage d'autorité : comme nous l'avons fait, la prédiction d'intention faite sur les données géométriques peut être utilisée pour relancer le processus de planification pour lequel la planification grossière est orientée vers le `Chemin topologique` que l'opérateur privilégie.

Prédiction d'intention : si de multiples re-planifications sont déclenchées par les actions de l'opérateur, des règles générales sur ses préférences peuvent être déduites et utilisées pour redéfinir les fonction de coûts des arcs et nœuds du Graphe topologique en fonction de leurs Informations sémantiques.

Utilisation des données sémantiques Ces données ayant un haut niveau d'abstraction, elles facilitent l'interprétation de l'humain en interaction. Au niveau sémantique, le partage de contrôle permet de :

Partage d'autorité : L'extraction des préférences de l'opérateur à partir de la redéfinition interactive des Stratégies de planification, et des re-planifications interactives constituent un partage d'autorité sur le niveau sémantique de la planification de trajectoire. Elles permettent de redéfinir dynamiquement l'utilisation de l'information disponible au niveau sémantique du modèle de l'environnement (définition des Stratégies de planification et des coûts du Graphe topologique).

Prédiction d'intention : une forme de prédiction d'intention à ce niveau de l'architecture peut être introduite avec l'ouverture vers une planification de tâche. Par exemple, dans le cas de l'assemblage de systèmes industriels, l'opérateur pourrait manifester (par exemple en langage naturel) des préférences pour le processus d'assemblage et ainsi en modifier la séquence.

L'architecture multi-niveaux proposée offre donc différents moyens d'interaction et de partage de contrôle impliquant chacun des éléments du modèle de l'environnement et de la planification de trajectoire différents. La description de l'utilisation des différentes données décrite précédemment est synthétisée dans la table 4.1. Dans cette table, les éléments qui apparaissent en vert illustrent les techniques que nous avons développées. Les autre représentent des perspectives de développement. L'ensemble des propositions faites ici n'est pas exhaustif mais introduit l'intérêt de l'implication de ces différents types d'informations pour le partage de contrôle. On peut par ailleurs lier le partage d'autorité fait à un niveau donné et la prédiction d'intention au niveau inférieur (le premier exploitant les résultats du second).

On note de plus, que ces différents niveaux d'interaction n'impliquent pas la même vitesse de réaction des processus de partage de contrôle. En effet, alors que l'interaction géométrique doit être réalisée en temps réel afin de ne pas perturber la sensation d'immersion de l'utilisateur, la redéfinition d'un Chemin topologique peut être faite dans un temps plus long.

4.2.4 Synthèse de l'interaction pour la planification de trajectoire proposée

L'interaction d'un opérateur humain avec le planificateur automatique de trajectoire présenté au chapitre 3 est, à l'instar de l'interaction proposée par Ladevèze pour le *planificateur probabiliste avec approche globale*, articulée autour de deux composantes :

	Partage d'autorité	Prédiction d'intentions
Niveau sémantique	Utiliser les actions de l'opérateur pour mettre à jour les fonctions de choix des paramètres des planificateurs topologiques et géométriques en fonction des informations sémantiques.	Utiliser le langage naturel pour formuler des requêtes de planification (*assembler ces deux composants*, *Poser cet objet sur celui-là*,...)
Niveau topologique	Vérifier si l'opérateur accepte le chemin proposé, et, sinon, déterminer son intention (quel lieu veut-il traverser ?) pour l'intégrer dans le `Chemin topologique`.	Apprendre, à partir des mouvements de l'opérateur, ses préférences et les intégrer dans les re-planifications à venir.
Niveau géométrique	Partager dynamiquement l'autorité en modulant la norme du guidage fourni par le planificateur automatique.	Déterminer quel lieu l'opérateur vise pour planifier un nouveau chemin topologique, et ainsi, redéfinir l'objectif géométrique courant.

Tableau 4.1 – Moyens d'interaction pour les différents niveaux.

1. l'opérateur est assisté par le planificateur automatique à travers une force de guidage lui indiquant une trajectoire pré-calculée. Lors de l'exécution de la tâche, l'opérateur peut localement redéfinir la trajectoire en s'écartant de celle qui lui est proposée.

2. en s'écartant plus fortement de la trajectoire proposée, l'opérateur peut déclencher la définition d'une nouvelle trajectoire.

Alors que Ladevèze et Flavigné appuyaient l'ensemble de leurs modes d'interaction sur des éléments géométriques, nous profitons du découpage de notre processus de planification en deux phases (planification grossière et planification fine), et de la diversité des informations de l'environnement (géométriques, mais aussi topologiques et sémantiques) pour mieux intégrer l'opérateur en interaction. En effet, alors que l'assistance proposée par le planificateur automatique à l'opérateur et la redéfinition locale de la trajectoire correspondent à des éléments de la planification fine (échanges de données géométriques), la re-planification s'appuie sur les éléments de la topologie de l'environnement en définissant, lors de la planification grossière, un `Chemin topologique` en fonction des intentions de l'opérateur interprétées sur la base du niveau topologique du modèle de l'`Espace libre`.

De plus, l'efficacité et l'ergonomie de ces deux modes d'interaction ont été améliorées par la mise en œuvre de techniques de partage de contrôle inspirées du pilotage de véhicules semi-autonomes et de la téléopération de robots :

Le partage d'autorité : la force de guidage utilisée pour assister l'opérateur a fait l'objet de la définition et de la mise en œuvre d'une technique de partage d'autorité afin que l'opérateur puisse se sentir plus libre de redéfinir localement une trajectoire ou d'explorer des chemins alternatifs à celui qui lui est proposé.

La prédiction d'intention : la re-planification d'une nouvelle trajectoire a fait l'objet de la définition et de la mise en œuvre d'une technique de prédiction d'intention permettant de détecter au plus tôt la volonté de l'opérateur d'emprunter un Chemin topologique différent de celui qui lui est proposé. Cette intention prédite est intégrée lors de la définition du nouveau Chemin topologique réalisant ainsi un partage d'autorité lors de la phase de planification grossière portant sur le niveau topologique du modèle de l'Espace libre (utilisation conjointe de coûts définis d'une part à partir de l'Information sémantique et d'autre part à partir de l'intention de l'opérateur).

Comme nous avons pu le voir, la mise en œuvre de techniques de partage de contrôle exploitant davantage les données topologiques et sémantiques disponibles dans notre modèle multi-niveaux de l'environnement peut encore améliorer l'efficacité et l'ergonomie de l'interaction. Ces améliorations permettront en effet d'intégrer toujours mieux les manipulations faites par ce dernier dans les trajectoires planifiées interactivement.

4.3 Implémentation de l'interaction pour la planification de trajectoire

L'implémentation de ces modes d'interaction et de partage de contrôle enrichit l'implémentation C++ du planificateur de trajectoire multi-niveaux. Elle fait aussi appel aux diverses librairies évoquées dans les chapitres précédents. La librairie Boost [Boost 14] en particulier a permis de paralléliser les calculs nécessaires à l'interaction et au partage de contrôle et ceux de l'application de RV. Cette parallélisation permet de ne pas perturber l'immersion et l'interaction de l'opérateur lors de l'exécution de processus longs (comme la prédiction d'intentions et la planification automatique de trajectoire).

Cette section décrit l'implémentation réalisée des modes d'interaction et de partage de contrôle, tout d'abord à travers un modèle statique décrivant les classes ajoutées et leurs fonctionnalités (section 4.3.1), puis en décrivant comment ces modalités sont réalisées (section 4.3.2). Nous verrons à cette occasion comment certains processus sont parallélisés et synchronisés.

4.3.1 Modèle statique : classes et responsabilités

Afin d'intégrer un opérateur humain dans le processus de planification de trajectoire, 5 nouvelles classes ont été créées. Cette section présente ces classes et leurs fonctionnalités pour la mise en œuvre du planificateur de trajectoire de manière interactive :

1. La Stratégie interactive est dérivée de la Stratégie de planification. Cette Stratégie interactive permet toujours d'associer un Planificateur local et un Planificateur géométrique pour définir la trajectoire d'une Étape topologique, mais associe en plus un Générateur d'assistance pour y guider l'opérateur.

2. La Générateur d'assistance fournit l'assistance utilisée pour guider l'opérateur lorsqu'il parcourt cette Étape topologique. Le Générateur d'assistance calcule donc l'assistance en temps-réel en fonction des jalons géométriques qui sont définis dans l'Étape topologique dans laquelle il est instancié (premier mode d'interaction, section 4.2.1).

3. Le Contrôleur d'autorité associé au Générateur d'assistance d'une Étape topologique adapte l'autorité du planificateur automatique à l'action de l'opérateur tout au long de l'exécution de la tâche comme décrit dans la section 4.2.2.1.

4. Le Contrôleur de progression surveille la progression de l'opérateur sur la trajectoire et met à jour l'Étape topologique en cours au fur et à mesure que l'opérateur progresse sur le Chemin topologique. Il active par conséquent le Générateur d'assistance approprié. Le Contrôleur de progression détecte aussi les sorties du Chemin topologique (lorsque l'opérateur quitte le Lieu de l'Étape topologique courante pour un Lieu différent de celui de l'Étape topologique suivante). Cette détection constitue une première façon de déclencher la re-planification interactive.

5. Le Prédicteur d'intention fournit une autre façon de déclencher une re-planification. Associé au Contrôleur de progression, lorsque l'opérateur ne suit pas l'assistance qui lui est proposée, le Prédicteur d'intention calcule la probabilité pour chaque Frontiere voisine d'être visée par l'opérateur (section 4.2.2.2).

Ces classes viennent compléter le modèle de domaine du planificateur de trajectoire comme illustré dans la figure 4.12.

4.3.2 Algorithme, processus et parallélisation

Les algorithmes développés pour introduire l'humain dans la planification de trajectoire sont :

- la génération de l'assistance qui lui est proposée (avec le partage d'autorité),
- la prédiction des intentions de l'opérateur et la définition de la trajectoire correspondante,
- la surveillance de la progression sur le Chemin topologique et la détection de la volonté de l'opérateur d'emprunter un Chemin topologique alternatif.

Certains de ces algorithmes comme la génération d'assistance sont simples et rapides à exécuter. D'autres sont plus complexes, donc plus longs à exécuter. Pour ne pas dégrader l'immersion et l'interaction de l'opérateur, ces algorithmes sont réalisés en parallèle de l'application de RV. Cette parallélisation implique une synchronisation des divers processus. Après avoir détaillé les changements apportés à l'algorithme de planification de trajectoire pour la re-planification interactive (exécution en parallèle de l'application de RV), cette section détaille les trois nouveaux algorithmes dédiés à l'interaction. Nous verrons pour finir comment tous ces développements sont mis en œuvre dans une application de RV.

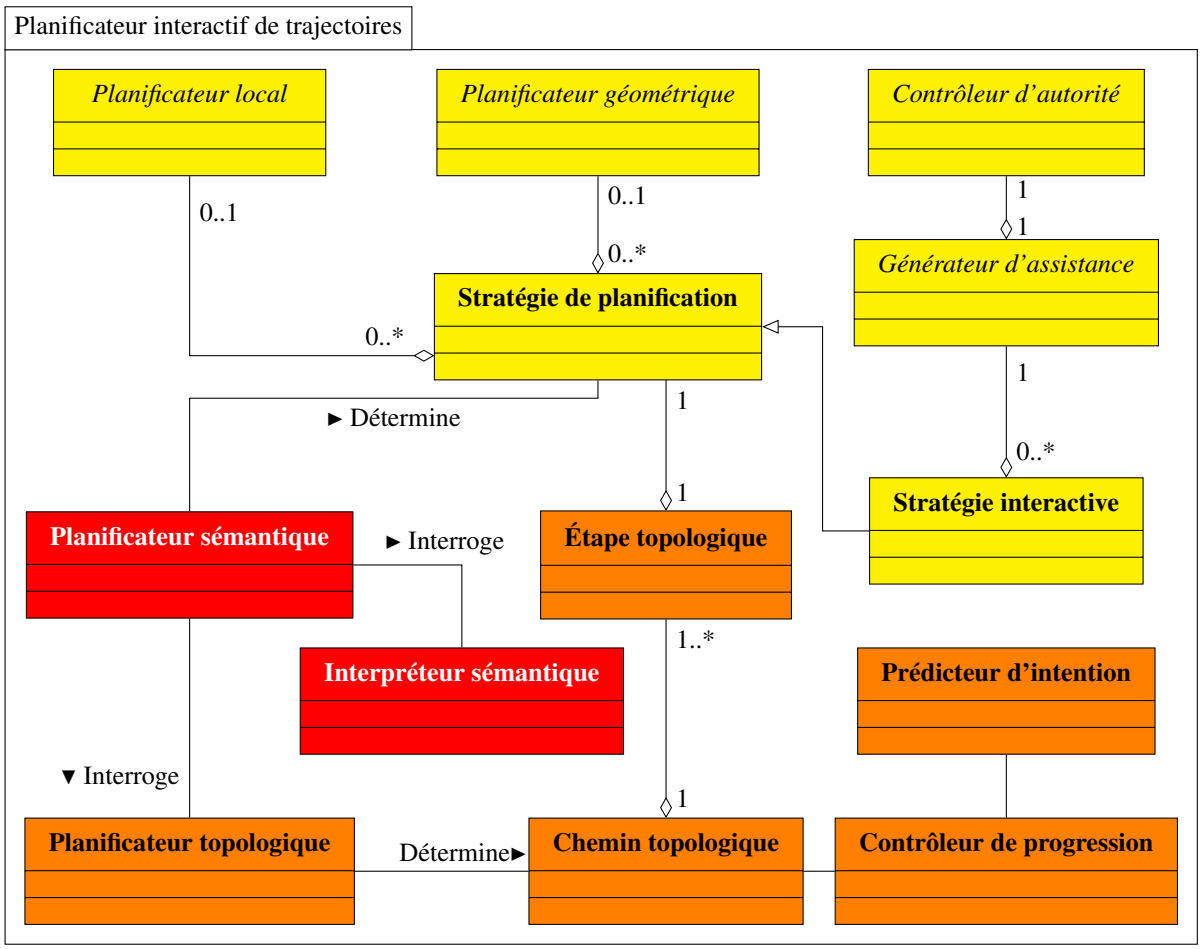


FIGURE 4.12 – Modèle de domaine UML du planificateur interactif de trajectoire sur environnement multi-niveaux.

4.3.2.1 Parallélisation de l'application de RV et de la re-planification

Du fait de l'interaction avec l'opérateur humain, il peut être nécessaire de re-planifier en ligne une nouvelle trajectoire (second mode d'interaction). En effet, nous avons vu comment nous prédisons au plus tôt l'intention de l'opérateur d'emprunter un `Chemin topologique` différent de celui qui lui est proposé (section 4.2.2.2). Cependant, le processus de planification (et donc de re-planification) est consommateur de temps et ne peut pas être exécuté à une vitesse suffisante pour ne pas perturber la sensation d'immersion et la fluidité de l'interaction dans l'application de RV. Pour remédier à cette difficulté, lors de la re-planification de la trajectoire, l'ensemble du processus est réalisé dans un *thread* dédié (algorithme 4.1). Ainsi, la fluidité de l'interaction entre le planificateur automatique de trajectoire et l'opérateur immergé dans l'application de RV est maintenue puisque le processus de re-planification de trajectoire exécuté en parallèle ne bloque pas celui de l'application de RV.

L'exécution des ces deux processus (l'application de RV et la re-planification) en parallèle, dans des *threads* dédiés, à des vitesses différentes nous conduit à introduire les variables *Changement de planification*, *Re-planification en cours* et *Re-planification nécessaire* qui sont des drapeaux permettant de synchroniser les deux *threads*.

- Le drapeau *Changement de planification* est levé dans le *thread* de re-planification lorsque

le processus de re-planification met de nouveaux éléments de la trajectoire planifiée (Chemin topologique, Étape topologique et jalons géométriques) à disposition de l'application de RV. Ce drapeau est levé dès que la planification grossière est achevée pour remplacer, dans l'application de RV le Chemin topologique (refusé par l'opérateur) par celui nouvellement planifié. Il est ensuite levé à la fin de la planification fine de chaque Étape topologique pour mettre à jour la trajectoire géométrique correspondante.

- Le drapeau *Re-planification en cours* est levé dans le *thread* de l'application de RV lors de l'ouverture du *thread* de re-planification, il est baissé dans le *thread* de re-planification juste avant que celui-ci soit clos. Ce drapeau permet de ne pas lancer plusieurs processus de re-planifiaction en parallèle.

- Le drapeau *Re-planification nécessaire* est levé et baissé dans le *thread* de l'application de RV lors du contrôle de l'exécution de la tâche en fonction du comportement de l'opérateur (algorithme 4.3). Il est lu dans le *thread* de re-planification pour déterminer si la re-planification doit être poursuivie ou interrompue.

Durant la définition d'une nouvelle trajectoire, celle que l'opérateur a refusée est maintenue dans le *thread* de l'application de RV pour l'interaction. L'assistance produite à partir de cette trajectoire reste modulée par le partage d'autorité (section 4.2.2.1). Ce maintien de la trajectoire pendant la définition d'un nouveau chemin permet d'exploiter le délai de re-planification pour permettre à l'opérateur de suivre à nouveau l'assistance du planificateur automatique. Lorsque l'opérateur suit à nouveau la trajectoire qu'il a temporairement abandonnée alors qu'une nouvelle est en cours de planification, la re-planification est abandonnée et la trajectoire préalablement perçue comme refusée est maintenue. Pour réaliser la synchronisation entre le *thread* de l'application de RV et celui de l'algorithme de planification présenté dans le chapitre 3, ce dernier se voit enrichi de points de contrôle (après la mise à jour du graphe topologique (ligne 5 de l'algorithme 4.1) et après la planification grossière (ligne 7 du même algorithme).

Lorsque le processus de planification grossière arrive à terme, le Chemin topologique nouvellement défini (et donc ses Étapes topologiques et leurs jalons géométriques) est utilisé en lieu et place de celui refusé par l'opérateur (lignes 8 et 9). Cette actualisation du Chemin topologique avant la planification fine permet de notifier à l'opérateur la prise en compte de sa volonté, mais aussi de commencer à l'assister sur ce nouveau Chemin topologique bien que la trajectoire géométrique n'y ait pas été définie. La planification fine de chaque Étape topologique est réalisée par la suite, alors que l'opérateur est déjà assisté sur ce Chemin topologique.

4.3.2.2 Génération d'assistance

Parmi les algorithmes dédiés à l'interaction, la génération de l'assistance est le plus simple et ne se fait que par une séquence de trois opérations :

1. définition de l'assistance f_a,

Algorithme 4.1 : Processus de re-planification

```
1  début
2      Coûts = Vérification des intentions de l'opérateur;        // algorithme 4.2
3      si Coûts ≠ NULL & Re-planification vrai alors
4          Mise à jour Graphe topologique ;
5          si Re-planification nécessaire vrai alors
6              Planification grossière ;                          // figure 3.6
7              si Re-planification nécessaire vrai alors
8                  Charger nouveau Chemin topologique pour assistance;
9                  Détruire ancien Chemin topologique;
10                 Changement de planification=vrai;
11             sinon
12                 Détruire nouveau Chemin topologique;
13         pour chaque Étape topologique faire
14             Planification fine ;                               // figure 3.7
15             Changement de planification=vrai;
16     Re-planification en cours = faux;
17     Clore Thread;
```

2. définition du coefficient de modulation mod_f (nous avons pour cela choisi $f_{min} = 0,1$ dans l'équation 4.5 pour maintenir une assistance résiduelle de 10% de son amplitude nominale lorsque l'opérateur s'y oppose),

3. calcul de f_g utilisée pour guider l'utilisateur.

4.3.2.3 Prédiction des intentions de l'opérateur

La prédiction des intentions de l'opérateur permet de préparer la re-planification d'une trajectoire par le planificateur automatique quand l'opérateur ne suit pas celle qui lui est proposée (algorithme 4.2). Pour cela, les `Frontieres` bordant le `Lieu` dans lequel se trouve $O-d$ se voient associer des probabilités traduisant le fait qu'elles sont visées par l'opérateur (ligne 2). Si la plus forte de ces probabilités satisfait les conditions de l'équation 4.9 (ligne 3), elles sont utilisées pour calculer des coûts pour chacune des frontières selon l'équation 4.10 (ligne 4). Ces coûts sont ensuite transmis pour être utilisés dans la planification grossière avec ceux définis à l'aide de l'`Information sémantique` lors du processus de re-planification (ligne 6).

4.3.2.4 Contrôle de l'exécution de la tâche

Le contrôle de l'exécution de la tâche (algorithme 4.3) est réalisé dans le *thread* de l'application de RV. Il permet d'une part de contrôler la progression de l'opérateur le long de la trajectoire pré-définie par le planificateur automatique (afin de maintenir une assistance pertinente), et d'autre part, de synchroniser le processus de re-planification avec l'application de RV. Il permet aussi de stopper l'ensemble du processus lorsque la tâche est achevée (ligne 5).

Algorithme 4.2 : Prédiction des intentions de l'opérateur

```
1  début
2      Probabilité = Calculer les probabilités des Frontière s voisines ;
3      si (max(Probabilité) ≥ 0.5 & max(Probabilité) ≥ P(F_tape)) alors
4          Coûts = Calculer les coûts des Frontieres voisines (Probabilité);
5          Re-planification nécessaire = vrai;
6          retourner Coûts;
7      sinon
8          retourner NULL;
```

Algorithme 4.3 : Contrôle de l'exécution de la tâche

```
1  début
2      si Frontierede l'Étape topologiquecourante atteinte alors
3          si Étape topologique courante = dernière Étape topologique alors
4              si Dernier jalon géométrique atteint alors
5                  fini = vrai;
6          sinon
7              Étape topologique courante = prochaine Étape topologique ;
8      sinon
9          si Jalon géométrique atteint alors
10             Transmettre le prochain jalon géométrique au Générateur d'assistance ;
11     si direction du mouvement instantané ≠ direction de l'assistance alors
12         si Re-planification en cours = faux alors
13             Processus de re-planification dans nouveau thread;     // algorithme 4.1
14             Re-planification nécessaire = vrai;
15             Re-planification en cours = vrai;
16     sinon
17         Re-planification nécessaire = faux;
18     si Changement de planification=vrai alors
19         Mise à jour du Chemin topologique;
20         Changement de planification=faux
```

Le contrôle de la progression de l'opérateur sur la trajectoire pré-définie est réalisé aux niveaux topologique et géométrique. Au niveau topologique, l'`Étape topologique` suivante est activée dès que la `Frontiere` concluant l'`Étape topologique` courante est atteinte (ligne 7). Au niveau géométrique, de même, le jalon géométrique utilisé pour générer l'assistance est mis à jour au fur et à mesure que l'opérateur progresse sur la trajectoire (lignes 9 et 10).

La synchronisation de ce contrôle de la progression de l'opérateur avec le *thread* de re-planification est réalisée en lançant le processus de prédiction d'intention et de re-planification dans un nouveau *thread* lorsque l'opérateur ne suit plus l'assistance (ligne 13), ou en stoppant ce même processus lorsqu'il la suit à nouveau (ligne 17). L'intégration des changements sur la trajectoire réalisée dans le *thread* de re-planification est aussi réalisée lors du contrôle de l'exécution de la tâche (ligne 19).

4.3.2.5 Application de planification interactive de trajectoire en RV

L'application de RV mettant en œuvre la planification interactive (algorithme 4.4) regroupe l'ensemble des fonctionnalités décrites précédemment (génération de l'assistance, prédiction des intentions de l'opérateur, contrôle de l'exécution de la tâche). Elle comprend en premier lieu une phase d'initialisation (lignes 2-6) comprenant la construction du modèle de l'environnement (ligne 2) décrite dans le chapitre 2, la définition d'une trajectoire (lignes 3 et 4) selon le processus introduit dans le chapitre 3, et le chargement de la première `Étape topologique` pour l'interaction (ligne 6).

La simulation entre ensuite dans une boucle d'interaction (lignes 7-10) comprenant le calcul de la direction du mouvement (ligne 8), le contrôle de l'exécution de la tâche qui fait appel à la prédiction d'intention lorsque c'est nécessaire (ligne 9) et l'actualisation de l'assistance (ligne 10).

Algorithme 4.4 : Application de planification interactive de trajectoire

```
1  début
2      Construction du modèle de l'environnement ;          // algorithme 2.1
3      Planification grossière ;                                // figure 3.6
4      Planification fine ;                                     // figure 3.7
5      fini = faux;
6      Re-planification nécessaire = faux
       Étape topologique courante = première Étape topologique ;
7      tant que fini = faux faire
8          Calculer la direction du mouvement ;
9          Contrôle de l'exécution de la tâche  ;           // algorithme 4.3
10         Actualisation de l'assistance ;
```

4.4 Conclusion sur l'interaction pour la planification de trajectoire

Dans ce chapitre, nous avons proposé deux modes d'interaction avec le planificateur de trajectoire présenté dans le chapitre 3 (section 4.2.1). Le premier de ces modes consiste en une assistance à l'opérateur fournie par le planificateur de trajectoire pour la manipulation d'objets virtuels. Le second mode permet à l'opérateur de déclencher la définition d'une nouvelle trajectoire par le planificateur automatique en s'écartant de la trajectoire proposée (section 4.2.1.1). Ces deux modes d'interaction permettent d'une part à un opérateur humain de contribuer à la définition de la trajectoire, et d'autre part de relâcher les contraintes sur la planification de trajectoire afin d'en contrôler le délai de convergence (section 4.2.1.2). Ces modes d'interaction exploitent à la fois les différents niveaux du modèle de l'environnement introduit au chapitre 2, et le partitionnement du processus de planification de trajectoire en deux phases (grossière et fine) présenté dans le chapitre 3.

Ces modes d'interaction remédient aux limites du planificateur de trajectoire en confiant à l'opérateur la responsabilité de diagnostiquer et de réaliser les actions correctrices lorsque la planification d'une trajectoire 6D sans collision n'a pas pu être réalisée. Pour cela, tout d'abord, la contrainte de non-collision de la trajectoire 6D a été relaxée, aussi bien lors de la définition des jalons géométriques qui conclut la planification grossière que lors de la planification fine.

Cette relaxation de la contrainte de non-collision permet de définir des jalons géométriques nécessaires au système automatique pour assister l'opérateur. Il revient donc ensuite à l'opérateur, pendant l'exécution de la tâche, lorsqu'il rencontre une portion de trajectoire 6D en collision, de diagnostiquer cette difficulté rencontrée par le planificateur automatique de trajectoire. Une fois que l'opérateur a posé le diagnostic grâce à son expertise, en manipulant l'objet, il peut soit redéfinir localement la trajectoire 6D, soit déclencher un nouveau processus de planification pour trouver une trajectoire différente sur un `Chemin topologique` plus adapté.

De plus, l'interaction entre le planificateur de trajectoire et l'opérateur humain a été enrichie par des techniques de partage de contrôle (section 4.2.2). Ces techniques de partage de contrôle permettent de donner plus de liberté à l'opérateur lors de l'exploration de `Chemins topologiques` alternatifs (section 4.2.2.1). Elles permettent aussi au système de définir une nouvelle trajectoire empruntant un nouveau `Chemin topologique` correspondant mieux aux souhaits de l'opérateur en prédisant au plus tôt ses intentions (section 4.2.2.2).

D'autres modes d'interaction peuvent encore enrichir le système interactif. On peut par exemple permettre à l'opérateur d'intervenir lors de la définition de la `Stratégie de planification` de chaque `Étape topologique` afin d'améliorer la capacité du planificateur à définir une trajectoire géométrique, et donc à assister l'opérateur. Une autre façon d'enrichir l'interaction est de propager les actions de l'opérateur (pour la re-planification et le choix des `Stratégies de planification`) sur les informations sémantiques afin d'apprendre les préférences de l'opérateur lorsque plusieurs requêtes de planification doivent être traitées dans la même application (comme c'est le cas pour l'assemblage de plusieurs composants d'un système industriel). Les

fonctions permettant de définir les coûts des éléments du Graphe topologique, comme le mode de choix des Stratégies de planification, peuvent ainsi être appris au cours des exécutions des tâches.

L'interaction peut encore être enrichie en intégrant à l'architecture des Planificateurs locaux et des Planificateurs géométriques dédiés à une planification fine de manière interactive. L'*IRRT* proposé par Flavigné semble particulièrement adapté à cela. Sa mise en œuvre nécessitera tout de même de reconsidérer la question du partage d'autorité et de la prédiction d'intention puisque la trajectoire géométrique qui intervient dans les deux techniques proposées sera cette fois définie pendant l'exécution de la tâche.

Nous allons voir, dans le chapitre suivant, comment notre planificateur de trajectoire a été mis en œuvre dans des simulations sur la plateforme de RV du laboratoire. Nous montrerons l'intérêt de l'architecture multi-niveaux proposée pour la planification de trajectoire hors-ligne comme pour la planification interactive de trajectoire.

Chapitre 5

Expérimentations et résultats

Table **Page**

5.1 Plateforme de RV du LGP 115

5.1.1 Interfaces comportementales 116

5.1.1.1 Visualisation 3D 116

5.1.1.2 Capture de mouvement 116

5.1.1.3 Interface haptique 118

5.1.2 Logiciels 119

5.1.2.1 Virtools Dev 119

5.1.2.2 Intégration des interfaces comportementales 120

5.1.2.3 Intégration logicielle de la planification de trajectoire proposée 121

5.1.2.4 Script de planification interactive de trajectoire 123

5.2 Applications de planification 124

5.2.1 Première application : instance 3D de l'exemple illustratif 124

5.2.1.1 Description générale 124

5.2.1.2 Déclinaisons 125

5.2.2 Seconde application : jeu d'encastrement 126

5.2.2.1 Description générale 127

5.2.2.2 Déclinaisons 128

5.3 Validation de la planification de trajectoire non-interactive 128

5.3.1 Stratégies de planification comparées 129

5.3.1.1 Stratégie G : géométrie seule 129

5.3.1.2 Stratégie GT : géométrie et topologie 130

5.3.1.3 Stratégie GTS : géométrie, topologie et sémantique 131

5.3.2 Critères d'évaluation 132
5.3.3 Résultats 132
5.3.3.1 Première application : instance 3D de l'exemple illustratif 132
5.3.3.2 Seconde application : jeu d'encastrement 135
5.3.4 Synthèse sur la planification de trajectoire multi-niveaux 138
5.4 Validation de la planification de trajectoire interactive 139
5.4.1 Réglages du planificateur interactif de trajectoire 139
5.4.1.1 Planification de trajectoire 139
5.4.1.2 Modes d'interaction 140
5.4.2 Résultats 141
5.4.2.1 Assistance et partage d'autorité 141
5.4.2.2 Re-planification et détection d'intention 142
5.4.3 Synthèse sur la planification de trajectoire interactive 144
5.5 Conclusion sur les expérimentations menées 145

Le planificateur de trajectoire interactif présenté dans les chapitres précédents a été implémenté et évalué sur la plateforme de RV du LGP. Différentes expérimentations ont été menées pour valider la faisabilité et l'intérêt de l'approche proposée. Cette validation a porté sur deux aspects :

1. L'approche multi-niveaux mettant en œuvre des données géométriques, topologiques et sémantiques pour la planification de trajectoire ainsi que le partitionnement du processus de planification de trajectoire en deux phases (grossière et fine).

   Cette validation a été obtenue par la comparaison des performances de différentes méthodes de planification de trajectoire. Pour valider le processus de planification proposé, ces performances ont été relevées dans un cadre non interactif.

2. L'interaction et le partage de contrôle entre le processus de planification de trajectoire automatique et un opérateur humain pour réaliser des tâches de manipulation en environnement virtuel.

   Pour cela, nous avons vérifié, d'une part l'intérêt de l'assistance fournie par le planificateur de trajectoire interactif, et d'autre part la capacité du système interactif à intégrer les manipulations faites par l'opérateur dans la trajectoire planifiée et dans l'assistance fournie.

La validation de ces deux éléments a été obtenue par des expérimentations de planification de trajectoire non-interactive et interactive menées sur deux applications virtuelles permettant

de décliner différentes requêtes de planification. La première application est basée sur une instance 3D de l'environnement 2D utilisé dans les chapitres 2, 3 et 4 pour illustrer les concepts proposés. La seconde application est inspirée des jeux d'encastrement destinés au développement psycho-moteur des jeunes enfants.

Ce chapitre est organisé comme suit :

- La section 5.1 décrit la plateforme de RV du LGP ainsi que la mise en œuvre du planificateur de trajectoire proposé qui y est faite.

- Ensuite, la section 5.2 présente les différentes applications utilisées pour valider notre proposition. Cette présentation détaille les informations sémantiques et topologiques de chacun de leurs environnements.

- Pour finir, deux sections décrivent les expérimentations menées et leurs résultats, et valident ainsi notre proposition d'une part pour la mise en œuvre de la planification de trajectoire proposée de manière non-interactive dans la section 5.3 et d'autre part pour la planification interactive de trajectoire dans la section 5.4.

5.1 Plateforme de RV du LGP

La plateforme de RV du LGP intègre différentes interfaces comportementales :

- un système de visualisation 3D par stéréoscopie passive,

- un système de capture de mouvement infrarouge,

- un gant de données,

- un bras haptique.

La mise en œuvre de ces interfaces fait appel à des logiciels spécifiques, qui leur sont dédiés.

Pour la planification de trajectoire interactive, nous mettons en œuvre le système de visualisation 3D, le système de capture de mouvements et le bras haptique. Ces interfaces, leurs logiciels associés et leur utilisation pour la planification interactive de trajectoire sont décrits dans la section 5.1.1. La section 5.1.2 présente le logiciel Virtools utilisé pour développer les applications de RV ainsi que la manière dont les interfaces comportementales y sont exploitées. La section 5.1.2 introduit l'outil logiciel Virtools de la plateforme utilisé pour le développement d'applications de RV. L'intégration de la planification de trajectoire proposée dans l'architecture logicielle de Virtools ainsi que la façon de la mettre en œuvre dans une application de RV y est aussi décrite.

5.1.1 Interfaces comportementales

5.1.1.1 Visualisation 3D

Système de visualisation 3D C'est un système de stéréoscopie passive (figure 5.1). Il est constitué de deux vidéo-projecteurs ProjectionDesign F2 munis chacun d'un filtre à polarisation circulaire (polarisation différente pour chaque vidéo-projécteur) (figure 5.1(a)). Ces vidéo-projecteurs rétro-projettent leurs images sur un écran de 3m de large par 2,25m de haut conservant la polarisation de la lumière (figure 5.1(b)). Pour reconstituer l'image stéréoscopique, l'utilisateur doit s'équiper de lunettes (figure 5.1(c)) munies de filtres du même type que ceux des video-projecteurs afin de ne conserver, pour chaque œil, que l'image qui lui est destinée.



(a) Vidéo-projecteurs et filtres. (b) Écran. (c) Lunettes.

FIGURE 5.1 – Système de visualisation 3D.

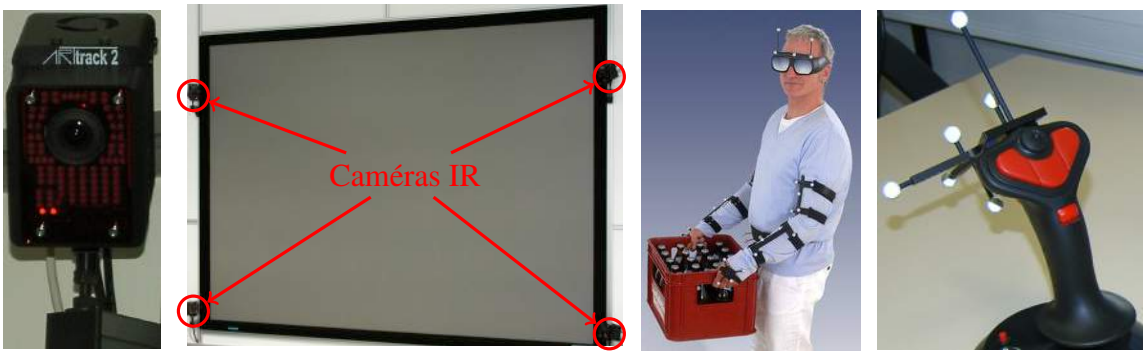
La rétro-projection, bien que difficile à mettre en œuvre, est ici préférée à la projection directe afin d'éliminer les occultations qui pourraient apparaitre du fait de la présence d'un utilisateur en interaction entre les vidéo-projecteurs et l'écran. De même, bien que moins performante qu'une polarisation horizontale/verticale du point de vue du filtrage de la lumière, la polarisation circulaire est mise en œuvre pour autoriser tous les mouvements aux utilisateurs de RV (inclinaison de la tête).

Mise en œuvre de la visualisation 3D Elle permet de fournir une visualisation de la scène et des manipulations exécutées à l'opérateur humain. Elle réalise ainsi un premier bouclage entre l'opérateur et le système qu'il manipule en fournissant un retour visuel temps-réel de l'environnement virtuel. La visualisation 3D permet aussi de fournir une première assistance à l'opérateur en affichant la trajectoire 3D planifiée, mais aussi, localement, la direction (3D) de progression pour atteindre le prochain jalon.

5.1.1.2 Capture de mouvement

Système de capture de mouvement optique Fourni par la société ART (Advanced Realtime Tracking GmbH) (figure 5.2), ce système est constitué de plusieurs caméras Infrarouge (IR) (figure 5.2(a)). Ces caméras émettent des flashs IR, ceux-ci sont réfléchis par des sphères dotées d'un revêtement spécial (réfléchissant la lumière IR). La position des sphères dans les

images de chacune des caméras permet d'en trianguler la position dans l'espace 3D. La position et l'orientation de marqueurs équipés de plusieurs sphères réfléchissantes (chaque marqueur ayant une configuration géométrique de sphères particulière) peuvent alors être déterminées.



(a) Caméra IR. (b) Position des caméras. (c) Utilisateur équipé de marqueurs. (d) Flystick.

FIGURE 5.2 – Système de visualisation 3D.

Le système du LGP est constitué de quatre caméras positionnées autour de l'écran (figure 5.2(b)). Deux caméras suffisent à trianguler la position des sphères, mais en augmentant leur nombre, on augmente la précision du système tout en prévenant les phénomènes d'occultation. Le LGP dispose de 8 marqueurs permettant d'équiper le haut du corps d'un utilisateur (figure 5.2(c)) : tête, abdomen, bras, avant-bras et mains), et d'un *flystick* (figure 5.2(d)) permettant de pointer des éléments de l'environnement 3D tout en disposant de quelques contrôles (4 boutons et un joystick).

Logiciel DTrack Le logiciel DTrack, hébergé sur une station dédiée, traite les données issues des caméras IR pour calculer les positions et orientations des différents marqueurs présents dans leurs champ de vision. La station qui héberge le logiciel Dtrack est donc équipée d'une carte permettant d'y connecter les caméras IR pour d'une part les synchroniser pour l'émission des flash IR et, d'autre part récupérer leur données.

Mise en œuvre du système de la capture de mouvement D'une part le système de capture de mouvement et sa mise en œuvre dans les applications de planification de trajectoire interactive permettent de tenir compte du point de vue de l'opérateur pour l'affichage de la scène (en l'équipant de lunettes munies d'un marqueur infrarouge). Cette prise en compte de son point de vue en temps réel permet à l'opérateur d'améliorer sa perception de la profondeur en réalisant de petits changements de point de vue sur la scène.

D'autre part, le *flystick* associé au système de capture de mouvement est utilisé pour fournir à l'opérateur différents contrôles sur l'application de RV. Par exemple, le *joystick* analogique présent sur le *flystick* est utilisé pour permettre à l'opérateur de changer plus radicalement son point de vue sur la scène.

5.1.1.3 Interface haptique

Système haptique Le bras Virtuose 6D35-45 de la société Haption (figure 5.3) est une interface bidirectionnelle. Elle permet à l'utilisateur d'agir sur l'environnement virtuel en attachant des objets virtuels à l'organe terminal manipulé par l'utilisateur (les mouvements de l'organe terminal sont appliqués à l'objet virtuel). Le système peut aussi fournir un retour sensoriel de l'application interactive à l'utilisateur (retour haptique). Ce retour sensoriel peut être utilisé pour un rendu physique (simulation du contact entre une pièce manipulée et un élément de l'environnement), ou pour mettre en œuvre des métaphores sensorielles (force artificielle pour de guidage de l'opérateur).



FIGURE 5.3 – Bras haptique Virtuose 6D35-45.

Logiciel IPSI Le logiciel *Interactive Physics Simulation Interface* (IPSI) fourni par Haption est un moteur physique pour la manipulation d'objets rigides. Il intègre une interface vers les périphériques haptiques de la société Haption et permet ainsi de contrôler le retour haptique. Pour cela, il intègre des briques logicielles pour la détection de collision et la simulation de déplacement, et réalise la simulation numérique temps-réel de l'environnement virtuel.

Mise en œuvre de l'interface haptique Dans les applications de planification de trajectoire interactive, l'interface haptique permet à l'opérateur humain de manipuler les objets virtuels en 6D avec un retour sensoriel (contact). Elle permet aussi d'exploiter l'assistance 6D générée par le planificateur de trajectoire interactif pour guider l'opérateur dans ses manipulations. C'est donc à travers cette interface que l'opérateur est assisté en 6D selon le mode de partage d'autorité décrit dans le chapitre 4 (*H-mode*).

5.1.2 Logiciels

Les applications de RV sont développées à l'aide de Virtools Dev, un logiciel dédié au développement d'applications interactives. Cette section présente :

- le logiciel Virtools Dev utilisé au LGP,
- la manière d'y exploiter les différents périphériques présentés,
- l'intégration des développements réalisés pour la planification de trajectoire interactive proposée,
- la construction d'une application 3D immersive et interactive avec ces éléments.

5.1.2.1 Virtools Dev

Virtools Dev 4.1 a été initialement développé pour la création de jeux vidéo. L'utilisation de ce logiciel par la communauté de la RV a amené le développement du module *VR Pack* facilitant la prise en charge de la visualisation 3D et intégrant le protocole VRPN [VRPN 14]. Ce protocole permet d'exploiter les périphériques gérés par des stations dédiées à travers un réseau TCP/IP en garantissant l'échange temps-réel des données.

L'interface de Virtools Dev est composée de 3 volets (figure 5.4) :

1. Le volet en haut à droite regroupe l'ensemble des ressources disponibles pour le développement de l'application (objets 3D, comportements élémentaires,...).
2. Le volet en haut à gauche, permet de visualiser l'environnement 3D (la scène) de l'application développée et d'y importer les éléments (objets 3D, caméras, repères locaux).
3. Le volet du bas permet de définir les scripts définissant les comportements des éléments de l'application de RV.

Ces scripts consistent en l'interconnexion de blocs implantant des comportements élémentaires (appelés *Building Blocks* (BB)) comme illustré dans la figure 5.5. Ces BB présentent des entrées/sorties comportementales (horizontalement) et des entrées/sorties paramétriques (verticalement). Pour simplifier la lecture des scripts, un ensemble de BB définissant un comportement peuvent être agrégés dans un *Behaviour Graphs* (BG). Le développement d'applications interactives est réalisé de manière graphique ce qui le rend abordable pour des utilisateurs non initiés.

Le *Software Development Kit* (SDK) fourni avec Virtools permet d'enrichir Virtools avec des fonctionnalités développées en C++ à partir des classes et fonctionnalités de Virtools rendues accessibles à travers le SDK.

Mentionnons particulièrement la possibilité de développer des *managers*. Les *managers* Virtools sont des classes permettant de stocker les objet instanciés. Ils sont initialisés lors du lancement de l'application interactive et accessibles tout au long de son exécution.

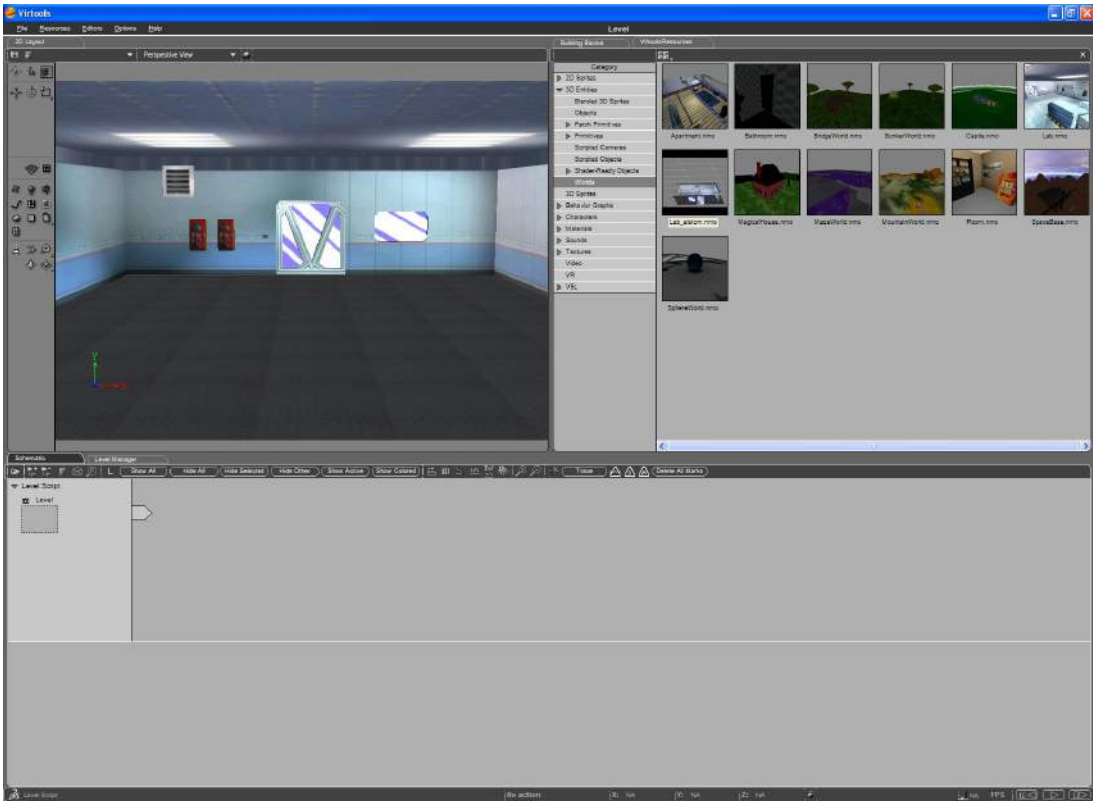


FIGURE 5.4 – Environnement de développement Virtools.

Le SDK de Virtools Dev permet aussi de développer des BB permettant de définir de nouveaux comportements élémentaires. Pour réaliser ces comportements élémentaires, les fonctionnalités du BB peuvent accéder aux objets définis dans les différents *managers* de l'application.

5.1.2.2 Intégration des interfaces comportementales

La visualisation 3D est directement gérée par Virtools Dev grâce au VR Pack qui met à disposition du développeur d'applications de RV des BB permettant de configurer la mise en stéréo de l'affichage.

La mise en œuvre de la capture de mouvement est réalisée à l'aide des fonctionnalités *Virtual Reality Peripheral Network* (VRPN) fournies dans le VR Pack de Virtools Dev. En effet, la station hébergeant le logiciel Dtrack diffuse les données sur le réseau *Transmission Control Protocl/Internet Protocol* (TCP/IP) à l'aide du protocole VRPN, il suffit donc, dans l'application de RV, de développer un script avec les BB fournis pour récupérer les informations qui nous intéressent dans les trames IP échangées sur le réseau ethernet.

La mise en œuvre du bras haptique à travers l'IPSI se fait grâce à l'*Interactive Physic Pack* (IPP). L'IPP est un ensemble de fonctionnalités, fournies par la société Haption, permettant de gérer la simulation IPSI depuis une application interactive développée avec Virtools Dev. L'IPP se présente sous Virtools Dev comme un ensemble de BB (figure 5.6). Ces BB permettent d'initialiser la simulation IPSI (chargement de corps rigides) et d'y transposer les opérations réalisées dans l'application de RV (déplacement des corps rigides). Les BB fournis dans IPP fournissent aussi des fonctionnalités plus complexes pour la manipulation d'objets (saisie et

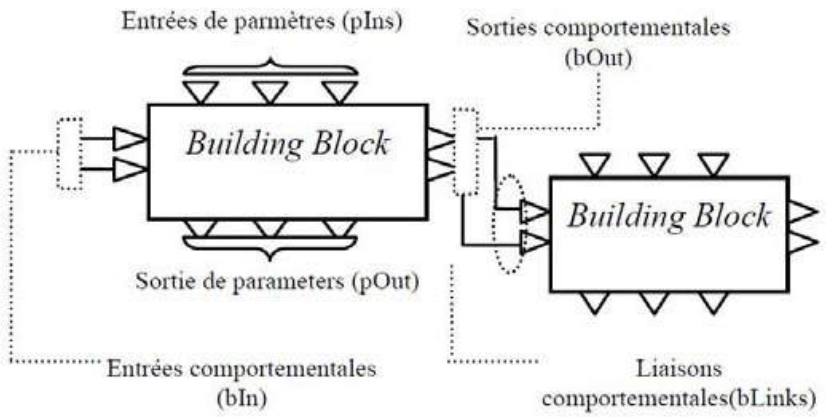


FIGURE 5.5 – Script comportemental Virtools.

relâchement d'objet, détection des collisions, interdiction des interpénétrations, utilisation de guides virtuels, application d'un effort sur l'objet manipulé...).

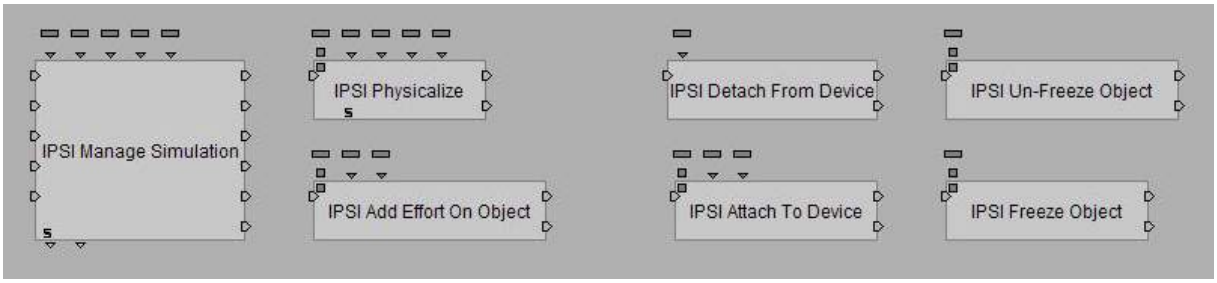


FIGURE 5.6 – Exemple de blocs comportementaux fournis avec IPP.

5.1.2.3 Intégration logicielle de la planification de trajectoire proposée

Le modèle de l'environnement présenté dans le chapitre 2, la planification de trajectoire décrite dans le chapitre 3 et la planification interactive proposée dans le chapitre 4 ont été compilés dans une *dynamic link library* (dll) chacune. Ces deux premières dlls ont été développées de manière à être indépendantes de Virtools Dev afin de pouvoir les exploiter à partir d'autres environnements de développement. Une troisième dll a donc été développée pour accéder aux fonctionnalités des deux premières depuis Virtools Dev.

Cette dll nous a permis de définir un *manager* pour stocker les différents objets de notre modèle de l'environnement et de notre planification de trajectoire. Nous avons ensuite défini différents BB permettant d'accéder à leurs fonctionnalités. Les principaux BB développés sont introduits dans la figure 5.7. Il permettent la construction du modèle de l'environnement, la gestion de l'information sémantique (visualisation, modification), la planification de trajectoire et l'assistance de l'opérateur.

Nous allons maintenant décrire les BB des différents groupes identifiés dans le figure 5.7 :

Environnement Pour modéliser l'environnement, chaque phase est réalisée par un BB dédié :

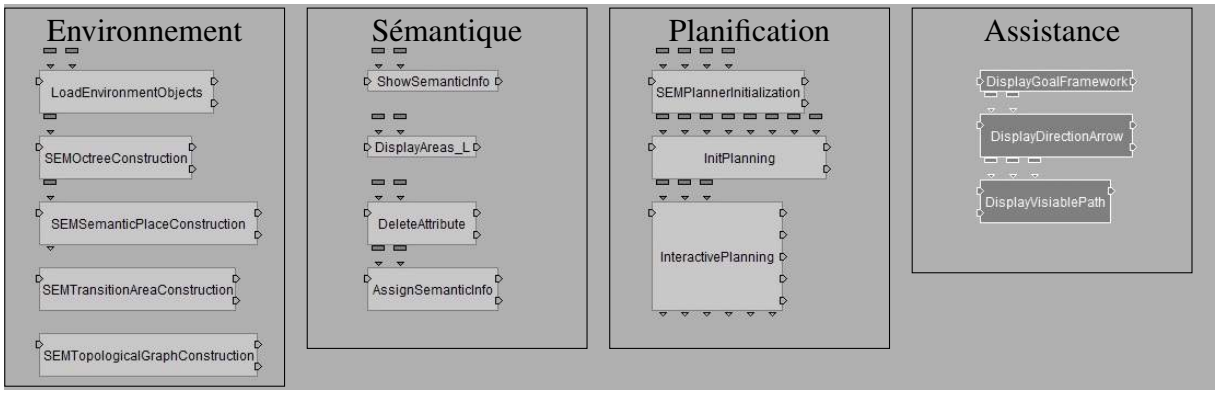


FIGURE 5.7 – BB pour la planification interactive de trajectoire.

1. un BB charge les `Corps rigides`,
2. un autre réalise la décomposition octree de l'espace de travail (espace de travail spécifié en paramètre),
3. un autre identifie les `Lieux`,
4. un autre identifie les `Frontieres`,
5. un dernier construit le `Graphe topologique`.

Comme nous avons pu le voir dans algorithme 2.1, ces différents BB sont mis en œuvre dans l'ordre dans lequel il sont décrits puisque chaque BB exploite les éléments construits par le précédent pour réaliser ses opérations.

Sémantique Pour gérer l'`Information sémantique`, nous avons fourni au développeur d'applications de RV la possibilité d'accéder à l'`Information sémantique` des différents `Lieux`, mais aussi d'afficher ces `Lieux`. Ces deux fonctionnalités sont complétées par la possibilité de supprimer et d'ajouter des attributs sémantiques aux différents `Lieux`.

Planification La planification de trajectoire interactive est faite par trois BB. Le premier permet de paramétrer le planificateur de trajectoire automatique (stratégie de planification employée, technique de planification géométrique mise en œuvre, nombre de jets aléatoires de l'algorithme RRT). Le second permet de définir la requête de planification (objet manipulé et configuration objectif) et définit une première trajectoire. Le dernier est actif tout au long du processus et gère l'ensemble de l'interaction (calcul de l'assistance et partage d'autorité ainsi que re-planification et prédiction d'intention).

Assistance Enfin, trois autres BB spécialement dédiés à l'assistance sont disponibles. Le premier permet de visualiser la requête de planification en affichant un "hologramme" (objet vert transparent) de l'objet manipulé dans sa configuration objectif. Le second permet d'afficher, sur l'objet manipulé, une flèche montrant la direction de l'assistance (en 3D). Le dernier affiche l'ensemble de la trajectoire 3D définie par le planificateur automatique de trajectoire.

5.1.2.4 Script de planification interactive de trajectoire

Dans cette section, nous décrivons comment l'ensemble des BB disponibles sont mis en œuvre pour la planification de trajectoire interactive. Nous ne détaillons ici que le script permettant de modéliser l'environnement et d'exécuter la planification de trajectoire. Les scripts dédiés à l'affichage stéréoscopique, à la capture de mouvement et à la gestion du point de vue sur la scène ne sont pas détaillés.

La figure 5.8 montre le script développé pour la planification de trajectoire interactive. Les BB précédemment décrits y sont instrumentés et l'ensemble des comportements sont synchronisés avec ceux décrits dans les autres scripts de l'application (gestion de la visualisation, du point de vue et pilotage de la planification de trajectoire). Pour alléger la lecture de ce script, les principales fonctions sont réalisées dans des BG agrégeant plusieurs BB.

Dans la figure 5.8, les fonctionnalités encadrées en bleu sont construites autour des BB développés au cours de nos travaux, les fonctionnalités encadrées en rouge traitent la simulation IPSI et sont basées sur les BB de l'IPP et la fonctionnalité en violet met en œuvre à la fois nos BB et ceux de l'IPP.

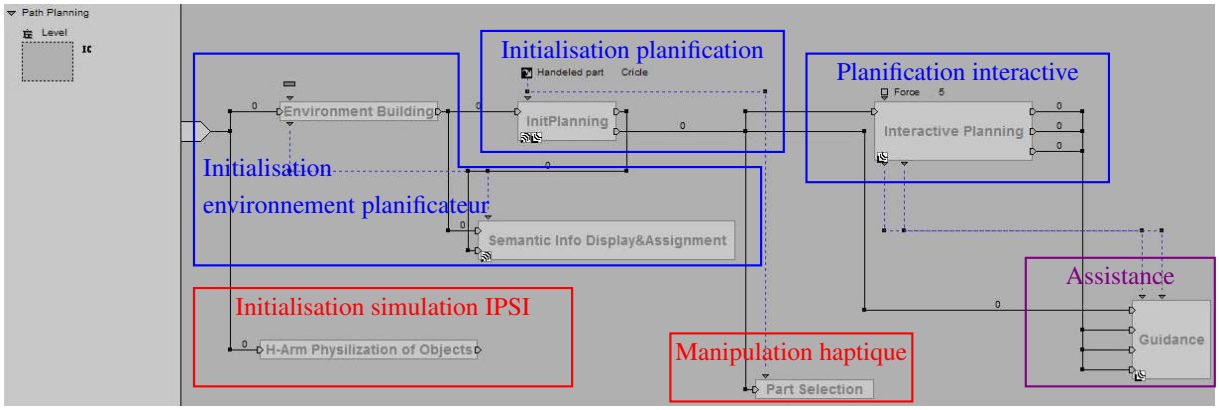


FIGURE 5.8 – Script de planification interactive de trajectoire en RV.

Ainsi, on voit dans la figure 5.8 que l'**Initialisation de l'environnement du planificateur** de trajectoire est réalisée dans deux BG :

1. le premier construit le modèle de l'environnement et met en œuvre pour cela les cinq BB que nous avons développés qui y sont consacrés.

2. Les BB de gestion de l'`Information sémantique` sont ensuite utilisés pour permettre à l'utilisateur de vérifier et corriger les `Informations sémantiques` associées aux différents `Lieux` créés.

En parallèle de cette initialisation de l'environnement de planification, l'**Initialisation de la simulation IPSI** est réalisée à l'aide des BB fournis dans l'IPP.

Ensuite, l'**Initialisation de la planification** interactive de trajectoire à proprement parler est réalisée :

- paramétrage du planificateur,

- planification d'une première trajectoire.

Cette initialisation de la planification interactive de trajectoire est réalisée alors que, dans la simulation IPSI, la **Manipulation haptique** est configurée en attachant l'objet manipulé à l'interface haptique.

Enfin, et tout au long de l'exécution de la tâche, le fonctionnement **Planification interactive** est organisé autour du BB *InteractivePlanning* afin de fournir l'**Assistance** à l'opérateur :

- affichage de la trajectoire 3D déterminée par le planificateur automatique de trajectoire,
- affichage de la flèche indiquant la direction de progression 3D,
- application d'un effort de guidage 6D sur l'objet manipulé (ressenti par l'utilisateur à travers l'interface haptique).

5.2 Applications de planification

Cette section décrit les différentes applications sur lesquelles l'architecture de modélisation de l'environnement 3D, de planification automatique ou interactive de trajectoire a été validé. Ces descriptions portent sur les scènes 3D dans lesquelles la planification de trajectoire est réalisée et détaillent les informations associées modélisées dans l'architecture multi-niveaux de l'environnement. Les détails introduits permettent d'interpréter finement les résultats de la planification de trajectoire présentés dans la section 5.3 ainsi que les éléments d'interaction et de partage de contrôle abordés dans la section 5.4.

5.2.1 Première application : instance 3D de l'exemple illustratif

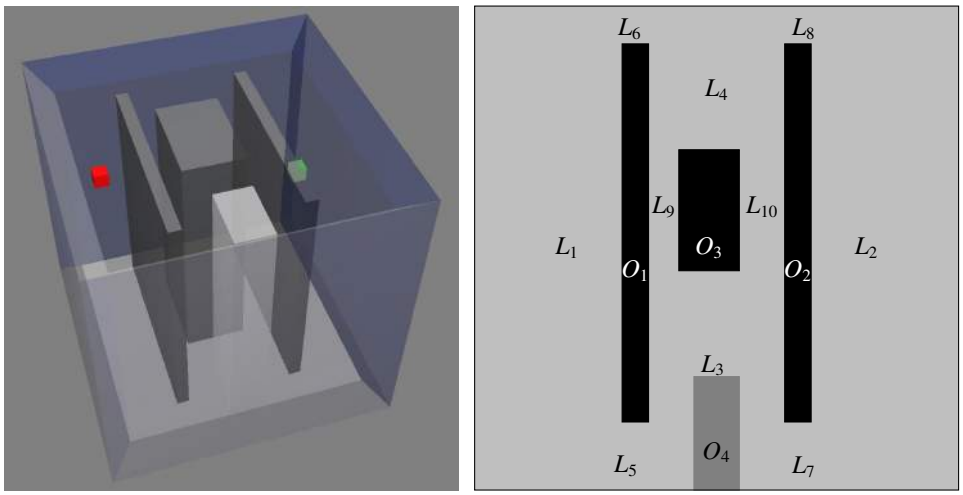
Cette application met en œuvre un environnement qui nous a permis de vérifier le bien-fondé de notre proposition de planification interactive. De plus, cette application nous a permis de réaliser les réglages des différents paramètres liés à la planification de trajectoire et à l'interaction donnés dans les sections 5.3 et 5.4.

L'environnement (figure 5.9(a)) mis en œuvre est une instance 3D de l'exemple utilisé dans les chapitres précédents pour illustrer les concepts du modèle de l'environnement et du planificateur de trajectoire proposés (figure 5.9(b)).

Différentes déclinaisons de cet environnement ont été réalisées pour vérifier le contrôle de la planification automatique de trajectoire à l'aide de l'`Information sémantique`. Une description générale du modèle de l'environnement est tout d'abord donnée (section 5.2.1.1) avant d'en détailler les variations dans les différentes déclinaisons (section 5.2.1.2).

5.2.1.1 Description générale

L'environnement de planification de cette première application est constitué d'un espace de travail cubique dans lequel se trouvent 4 obstacles (O_1 à O_4). Trois d'entre eux sont fixes (O_1 à



(a) Environnement d'expérimentation.

(b) Modèle 2D de l'environnement d'expérimentation.

FIGURE 5.9 – Environnement d'expérimentation inspiré de l'exemple illustratif 2D.

O_3), et un quatrième est mobile (O_4). La requête de planification, dans cet environnement, vise à déplacer un cube d'une position initiale à une position finale respectivement rouge et verte dans la figure 5.9(a).

L'obstacle (O_4) n'est pas pris en compte lors de la construction de niveau topologique du modèle de l'`Espace libre`, mais vient enrichir de l'attribut "encombré" l'`Information sémantique` associée au `Lieu` dans lequel il se trouve.

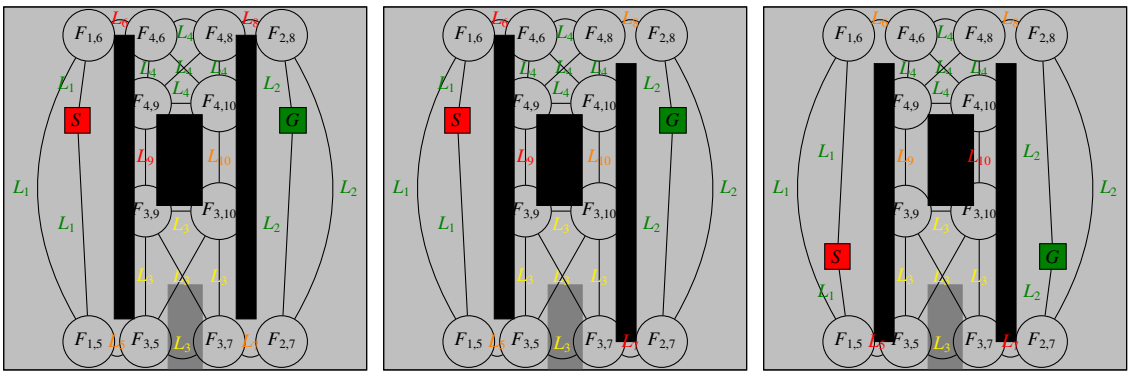
Les obstacles fixes (O_1 à O_3) permettent d'identifier dix `Lieux` (L_1 a L_{10}) dont six représentent des passages complexes (L_5 à L_{10}). Une complexité de "2" est associé à trois d'entre eux. Les trois autres, plus étroits, ont une complexité de "3" (figure 5.9(b)).

5.2.1.2 Déclinaisons

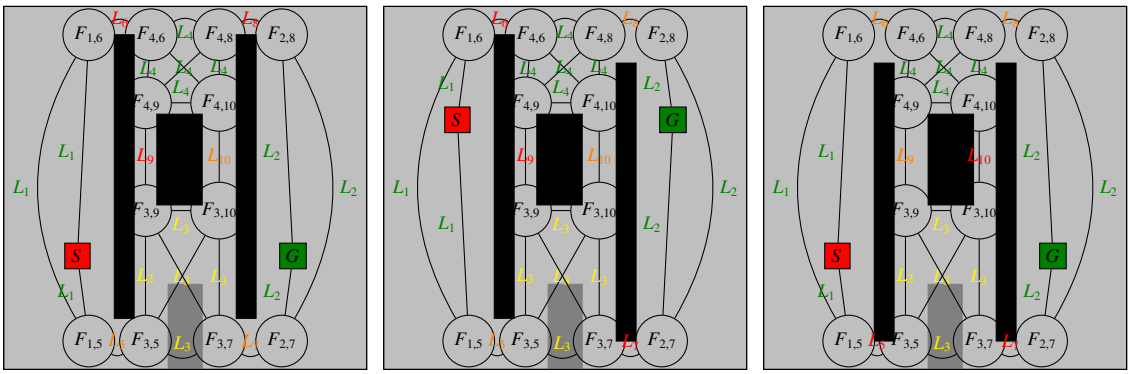
Pour valider le contrôle de la planification de trajectoire à l'aide de l'information sémantique, trois déclinaisons de cet environnement ont été testées (Environnement 1, Environnement 2 et Environnement 3). Ces déclinaisons sont obtenues en faisant varier les positions des obstacles définis comme fixes (O_1 à O_3) afin d'aménager différentes configurations des `Lieux` de complexité 2 et 3. Les obstacles O_1 et O_2 sont déplacés verticalement et l'obstacle O_3 est déplacé horizontalement.

Dans chacun de ces trois environnements, deux requêtes de planification (a et b) sont traitées. Dans la requête de planification a, les configurations initiale et finale sont proches des `Lieux` L_6 et L_8 respectivement, alors que dans la requête de planification b, elles sont proches des `Lieux` L_5 et L_7.

La figure 5.10 montre la projection 2D des différentes déclinaisons étudiées. Les `Graphes topologiques` correspondants à chacune des déclinaisons sont aussi donnés, les `Lieux` y apparaissent en couleur (couleur liée à l'`Information sémantique` : vert pour "complexité=1", orange pour "complexité=2", rouge pour "complexité=3" et jaune pour "encombré").



(a) Déclinaison 1a. (b) Déclinaison 2a. (c) Déclinaison 3a.



(d) Déclinaison 1b. (e) Déclinaison 2b. (f) Déclinaison 3b.

FIGURE 5.10 – Projection 2D des six déclinaisons de l'application inspirée de l'exemple illustratif.

Dans tous les environnements, les Lieux L_1, L_2, L_3 et L_4 ont une complexité de 1 et L_3 est encombré. La différence entre ces environnements porte donc sur les Lieux de complexité 2 et 3. Les Informations sémantiques des différentes déclinaisons de l'environnement sont synthétisées dans la table table 5.1.

		L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
Encombrement				✓							
Compl.	Environnement 1	1	1	1	1	2	3	2	3	3	2
	Environnement 2	1	1	1	1	2	3	3	2	3	2
	Environnement 3	1	1	1	1	3	2	3	2	2	3

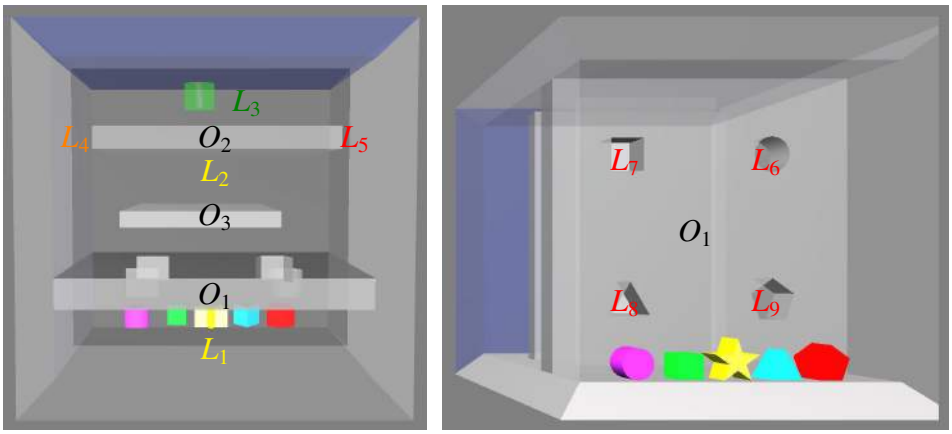
Tableau 5.1 – Information sémantique des Lieux de l'instance 3D de l'exemple illustratif.

5.2.2 Seconde application : jeu d'encastrement

Cette deuxième application confronte notre proposition à un cas difficile de planification de trajectoire. Elle fait apparaitre des passages fortement complexes qui tiennent les techniques de planification de trajectoire traditionnelles en échec. L'expertise liée à cette application est simple à modéliser à travers le niveau sémantique du modèle de l'Espace libre et permet de pointer les passages complexes. Cette application a permis de régler les paramètres liés à la relaxation de la contrainte de non-collision pour le contrôle du temps de planification nécessaire dans le cadre

d'une utilisation interactive. L'intervention d'un opérateur humain pour franchir les passages difficiles est donc indispensable.

L'environnement utilisé pour tester et mettre au point le planificateur interactif de trajectoire met en œuvre des notions inspirées des jeux d'encastrement utilisés pour le développement cognitif et psycho-moteur de jeunes enfants. Cet environnement est illustré dans la figure 5.11.



(a) Autres Lieux de l'espace de travail. (b) Passages à travers O_1.

FIGURE 5.11 – Environnement inspiré des jeux d'encastrement.

5.2.2.1 Description générale

Cet environnement est lui aussi constitué d'un espace de travail cubique encombré par trois Corps rigides (O_1 à O_3). Deux de ces Corps rigides (O_1 et O_2) sont fixes et divisent la scène en trois Lieux principaux L_1, L_2 et L_3 de complexité 1 (figure 5.11(a)). Le dernier Corps rigide (O_3) encombre le Lieu L_2.

O_1 et O_2 peuvent être franchis pour naviguer parmi les Lieux principaux. Pour cela, O_1 dispose de trous ayant des formes spécifiques (rond, carré, triangle et pentagone) inspirées des jeux d'encastrement (figure 5.11(b)). Chacun de ces trous constitue un Lieu (L_6, L_7, L_8 et L_9). Le second obstacle peut être contourné de deux côtés (L_4 et L_5 sur la figure 5.11(a)), un des passages étant plus étroit (complexité de L_5=3) que l'autre (complexité de L_4=2).

Ainsi, en plus des trois Lieux principaux (L_1 à L_3), cet environnement dispose de six Lieux de passage (L_4 à L_9). La description sémantique de ces Lieux de passage comprend leur complexité, mais aussi, pour les trous de l'obstacle O_1, leur forme géométrique ("rond" pour L_6,"carré" pour L_7,"triangle" pour L_8= et "pentagone" pour L_9). L'Information sémantique de cet environnement est synthétisée dans la table 5.2.

La requête de planification de trajectoire à traiter dans cette simulation consiste à déplacer les Corps rigides manipulables colorés en position initiale dans L_1 à une position finale verte dans L_3 (figure 5.11(a)).

Attribut\ Lieu	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9
Complexité	1	1	1	2	3	3	3	3	3
Encombrement	✓	✓							
Forme						○	□	△	⬠

Tableau 5.2 – Information sémantique attachée aux Lieux de l'application inspirée des jeux d'encastrement.

5.2.2.2 Déclinaisons

Dans cet environnement, les différentes requêtes de planification à traiter varient par la forme (le profil) et la position initiale des Corps rigides auxquels il faut faire franchir les obstacles. Des profils rond, carré, triangle, pentagone et étoile sont disponibles (figure 5.11(b)). La forme du profil de ces différents objets est décrite dans l'Information sémantique associée à chacun d'entre eux.

Les trous de O_1 illustrés dans la figure 5.11(b) et les différents Corps rigides manipulables ont été dimensionnés de manière à pouvoir, pour certains Corps rigides, emprunter différents trous de O_1. La compatibilité obtenue est décrite dans la table 5.3. Le symbole "✓" y exprime la possibilité de faire passer un objet donné dans un trou donné. Lorsque "✓" est accompagné de "*", l'objet comme le Lieu porte le même Attribut de forme.

Cette compatibilité trou/objet permet d'une part au planificateur de trajectoire de proposer des Chemins topologiques pertinents pour les objets manipulés lorsque ceux-ci portent un Attribut de forme identique que celui d'un Lieu associé à un trou, et d'autre part, les Chemins topologiques alternatifs dus à la compatibilité de certains objets avec des trous de formes différentes permettent à l'opérateur humain de mettre en œuvre l'ensemble des modes d'interaction et de partage de contrôle pour réaliser la tâche.

Objet\ Lieu	L_6	L_7	L_8	L_9
○	✓*	✓		✓
□	✓	✓*		✓
△		✓	✓*	✓
⬠				✓*
☆				✓

Tableau 5.3 – Compatibilité objet/trou.

5.3 Validation de la planification de trajectoire non-interactive

L'objectif est ici d'évaluer l'intérêt de l'implication des différentes natures d'information dans le processus de planification de trajectoire.

Pour cela, différentes stratégies de planification exploitant différents éléments du modèle de l'environnement proposé sont comparées. La configuration du planificateur de trajectoire pour chacune d'elles a été faite de manière à privilégier la définition d'une trajectoire sans collision au temps de définition de cette trajectoire. Pour résumer, l'utilisation non interactive du planificateur de trajectoire nous a amené à ne relaxer la contrainte de non-collision qu'après un grand nombre d'essais aléatoires.

Chacun des critères considérés a été évalué pour toutes les stratégies sur chacune des déclinaisons des deux applications de planification de trajectoire. Pour cela, l'ensemble des simulations a été évaluée statistiquement en effectuant plusieurs essais de chaque déclinaison. Ces différents essais ont été menés sur une station bi-processeur 2,6 GHz disposant de 32 Go de mémoire vive. Chacun des processeurs de la station a 6 cœurs pouvant chacun exécuter 2 *threads* en parallèle (24 *threads* en tout).

Dans cette section, nous allons tout d'abord détailler les différentes stratégies de planification comparées (section 5.3.1). Nous allons ensuite décrire les critères d'évaluation retenus (section 5.3.2) avant de détailler les résultats obtenus (section 5.3.3).

5.3.1 Stratégies de planification comparées

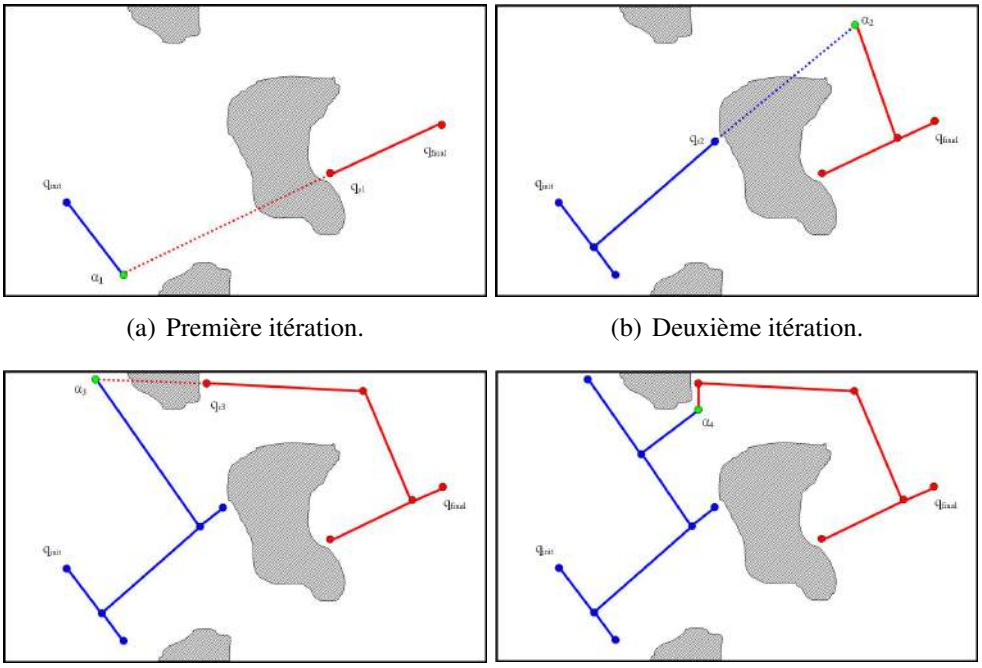
Pour évaluer l'intérêt de l'utilisation des différentes natures d'information (géométrique, topologique et sémantique) dans le processus de planification de trajectoire, trois stratégies de planification ont été définies. La première de celles-ci (stratégie G) est une stratégie de planification de référence de la littérature exploitant uniquement le niveau géométrique du modèle des `Corps rigides` (section 5.3.1.1). La seconde stratégie (stratégie GT) exploite les niveaux géométrique et topologique du modèle de l'environnement (section 5.3.1.2). La dernière stratégie (stratégie GTS) est celle proposée dans le chapitre 3 et exploite donc l'ensemble de la modélisation de l'environnement (section 5.3.1.3).

5.3.1.1 Stratégie G : géométrie seule

La planification de trajectoire n'exploitant que le niveau géométrique du modèle de l'environnement est une planification de trajectoire probabiliste avec une approche locale. Le planificateur implémenté met en œuvre un algorithme RRT bidirectionnel avec un échantillonnage uniforme de l'espace 6D limité à l'espace 3D de la scène (cube limitant l'espace de travail).

Cet algorithme illustré en 2D dans la figure 5.12 essaie en premier lieu d'inter-connecter directement les configurations initiale et finale. Ensuite, il entre dans une boucle itérative où, à chaque itération, il développe deux arbres (un premier depuis la configuration initiale q_{init} et l'autre depuis la configuration finale q_{final}).

Pour cela, à chaque itération, une configuration est aléatoirement choisie (α_1, α_2, α_3 et α_4), chaque arbre est étendu dans la direction de cette configuration depuis la configuration de l'arbre qui en est le plus proche. Ce processus est réalisé en boucle jusqu'à ce que les deux arbres développés depuis q_{init} et q_{final} soient connectés. La trajectoire obtenue est alors la série de nœuds et d'arcs des deux arbres permettant de relier q_{init} et q_{final}.



(a) Première itération. (b) Deuxième itération.

(c) Troisième itération. (d) Quatrième itération.

FIGURE 5.12 – RRT bidirectionnel en 2D.

Pour borner le temps de définition d'une trajectoire, nous avons limité le nombre d'itérations de l'algorithme à 10^6. Cette limite correspond, sur la station utilisée, à une heure trente minutes de calcul pour les environnements étudiés (le nombre d'`Corps rigides` de la scène influe sur le temps de calcul).

5.3.1.2 Stratégie GT : géométrie et topologie

La stratégie de planification permettant d'exploiter à la fois les informations topologiques et géométriques de l'environnement est construite à partir du processus de planification en deux phases (grossière et fine) décrit dans le chapitre 3. L'absence d'information sémantique nous oblige à déterminer de nouvelles modalités de définition des coûts du `Graphe topologique` pour la planification grossière, et des `Stratégies de planification` pour la planification fine.

Pour cette stratégie de planification, les coûts du `Graphe topologique` sont portés par les arcs du graphe et sont proportionnels à la distance entre les `Frontieres` que ceux-ci connectent. Une fois le `Chemin topologique` défini, les jalons géométriques dans les `Frontieres` qui le composent sont déterminés aléatoirement par un jet uniforme 6D dans l'espace 3D de chaque `Frontiere`.

De plus, la `Stratégie de planification` définie pour chaque `Étape topologique` est l'exécution d'un algorithme RRT (similaire à celui de la stratégie G) dans le `Lieu` concerné.

Toujours pour borner le temps de définition d'une trajectoire, certaines limites ont été fixées au cours du processus de planification :

- Lors de la planification grossière, le nombre de jets aléatoires utilisés pour définir les jalons géométriques dans les `Frontieres` empruntées par le `Chemin topologique` est

limité à 2000 pour chaque Frontiere. Si aucun jalon sans collision n'est trouvé après ces 2000 essais, un jalon est arbitrairement défini au centre de la Frontiere sans considérer le problème de la collision de ce jalon avec les obstacles de l'environnement.

- Le nombre maximal d'itérations de l'algorithme RRT est défini à 10^5 du fait de la simplification des requêtes de planification obtenue par le jalonnement de la trajectoire et la limitation de l'espace de travail considéré par l'algorithme RRT (espace limité au Lieu à traverser).

Ces deux limitations se traduisent par des temps de calcul maximaux de 5 secondes pour chaque Frontiere dans laquelle un jalon géométrique doit être défini, et de 10 minutes pour l'algorithme RRT de chaque Étape topologique. Ces temps de calcul ont été déterminés sans paralléliser aucun de ces processus (l'ensemble des ressources de la station est dédié au processus évalué).

5.3.1.3 Stratégie GTS : géométrie, topologie et sémantique

La stratégie de planification de trajectoire permettant d'exploiter l'ensemble des informations du modèle de l'environnement proposé est celle définie dans le chapitre 3. Mis à part les coûts du Graphe topologique et les Stratégies de planification, elle est donc similaire à la stratégie GT.

Les coûts du Graphe topologique sont ici définis en fonction de l'Information sémantique des différents Lieux concernés. Pour ces coûts, nous considérons deux types d'Attributs :

Complexité : cet attribut nous permet de définir des coûts de 0.5, 1 et 5 pour les complexités de 1, 2 et 3 respectivement.

Forme géométrique : cet attribut nous permet de définir un coût de 0 lorsqu'il n'est pas renseigné, de 0.5 si sa valeur est identique à celle de l'attribut de forme géométrique de l'objet manipulé, et de 5 si ce n'est pas le cas.

Pour chaque Lieu i ces deux coûts sont sommés dans $C_{semantique_i}$. Ils sont ensuite utilisés pour déterminer les coûts $C_{n_{i,j}}$ et $C_{a_{i_{j,k}}}$ correspondant respectivement au coût du nœud associé à la Frontiere $F_{i,j}$ et au coût de l'arc liant les Frontieres $F_{i,j}$ et $F_{i,k}$ à travers le Lieu L_i comme indiqué dans l'équation 5.1.

$$\begin{aligned} C_{n_{i,j}} &= \frac{C_{semantique_i} + C_{semantique_j}}{2} \\ C_{a_{i_{j,k}}} &= d_{i_{j,k}}, C_{semantique_i} \end{aligned} \tag{5.1}$$

En ce qui concerne la Stratégie de planification de chaque Étape topologique, l'attribut d'encombrement est évalué pour n'exécuter l'algorithme RRT que dans les Lieux encombrés.

5.3.2 Critères d'évaluation

La qualité de la stratégie de planification de trajectoire est évaluée en fonction du succès des expérimentations menées. Ce succès est mesuré de la manière suivante :

Stratégie G : Le taux de succès pour la définition d'une trajectoire illustre la qualité de la stratégie employée. Ce taux de succès est défini comme le nombre de trajectoires (sans collision) définies divisé par le nombre d'essais menés.

Stratégie GT et GTS : Le `Chemin topologique` est défini de manière déterministe lors de la planification grossière. Le fait que ce `Chemin topologique` soit réalisable (il existe au moins une trajectoire 6D permettant de parcourir le `Chemin topologique`) est un premier indicateur binaire. De plus, lorsque le `Chemin topologique` est réalisable, la capacité du planificateur à y définir une trajectoire sans collision constitue un indicateur de réussite complémentaire. Le taux de succès est dans ce cas défini comme le nombre de trajectoires définies sans collision divisé par le nombre de trajectoires avec collision.

Pour évaluer les performances des différentes stratégies de planification, l'indicateur retenu est le nombre de configurations aléatoires nécessaires pour définir les trajectoires. Comme les trois stratégies de planification considérées s'appuient sur l'utilisation de configurations aléatoirement définies pour calculer une trajectoire, le nombre de ces configurations nécessaires constitue une métrique unifiée pour les comparer. Malgré tout, pour les stratégies GT et GTS, nous distinguerons les configurations aléatoires utilisées au cours de la planification grossière (jalons définis dans les `Frontieres`) de celles utilisées pour la planification fine (algorithme RRT).

5.3.3 Résultats

5.3.3.1 Première application : instance 3D de l'exemple illustratif

Dans cette section, nous présentons et interprétons les résultats obtenus pour les planifications de trajectoire réalisées sur les différentes déclinaisons de l'instance 3D de l'exemple illustratif. Les indicateurs obtenus ont été extraits d'échantillons de plus de 100 essais pour la stratégie G, et de plus de 4000 essais pour les stratégies GT et GTS.

Pour cette application, toutes les stratégies de planification de trajectoire permettent, pour toutes les déclinaisons considérées, de converger sur une trajectoire réalisable (sans collision). Les indicateurs de succès obtenus sont résumés dans la table 5.4.

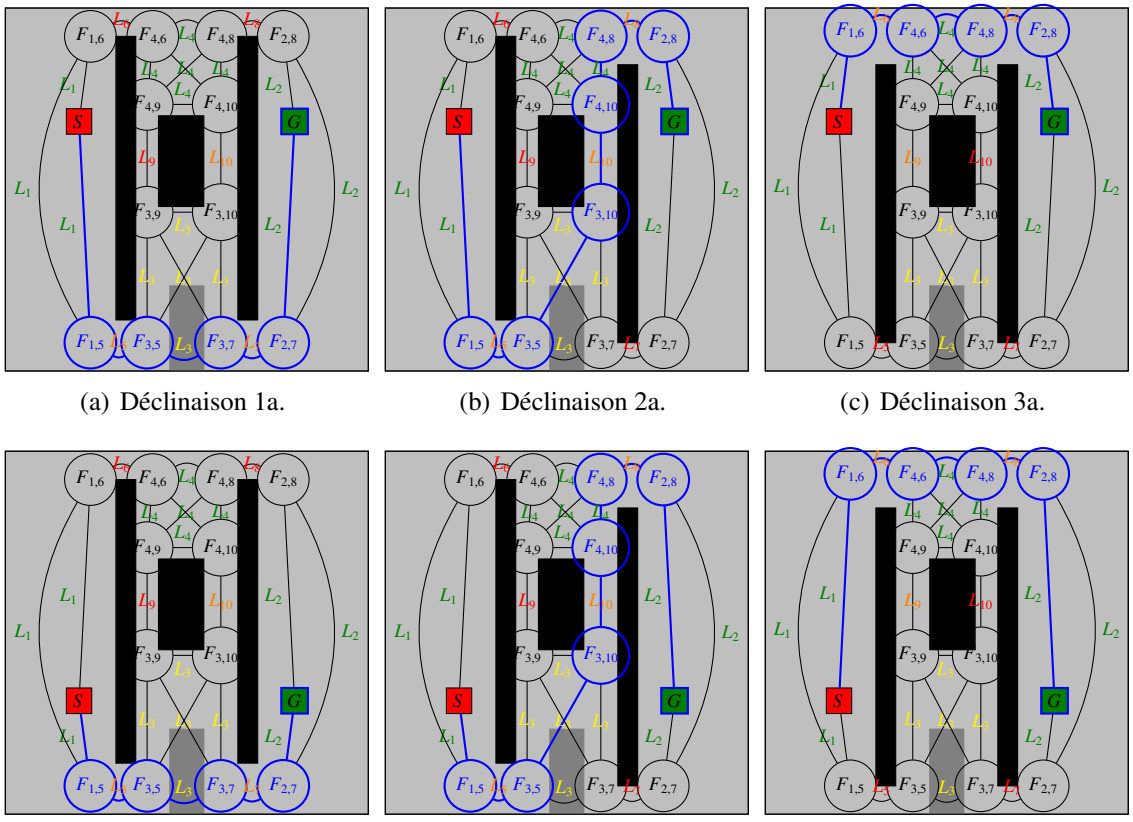
Ces indicateurs ne permettent donc pas de juger de la qualité relative des stratégies puisque toutes permettent ici de définir une trajectoire réalisable. Il est toutefois intéressant de comparer les trajectoires obtenues.

D'une part, la comparaison des stratégies GT et GTS permet de valider le contrôle de la planification grossière par l'`Information sémantique`. En effet, les `Chemins topologiques` trouvés par la stratégie GT minimisent la distance à parcourir en empruntant les `Lieux` L_1, L_6,

Stratégie	Critère	Décl 1a	Décl 1b	Décl 2a	Décl 2b	Décl 3a	Décl 3b
Stratégie G	Taux de succès	100%	100%	100%	100%	100%	100%
Stratégie GT	Faisabilité Chemin topologique	✓	✓	✓	✓	✓	✓
	Taux de succès	100%	100%	100%	100%	100%	100%
Stratégie GTS	Faisabilité Chemin topologique	✓	✓	✓	✓	✓	✓
	Taux de succès	100%	100%	100%	100%	100%	100%

Tableau 5.4 – Qualité des stratégies de planification de trajectoire (exemple illustratif).

L_4, L_8 et L_2 pour la requête de planification *a* et les Lieux L_1, L_5, L_3, L_7 et L_2 pour la requête de planification *b*. La stratégie GTS, elle, privilégie, pour chaque déclinaison, les Lieux dont la difficulté est moindre. Les Chemins topologiques obtenus pour chacune des déclinaisons sont illustrés dans la figure 5.13 où ils apparaissent en bleu.



(a) Déclinaison 1a. (b) Déclinaison 2a. (c) Déclinaison 3a.

(d) Déclinaison 1b. (e) Déclinaison 2b. (f) Déclinaison 3b.

FIGURE 5.13 – Chemins topologiques retenus par la stratégie GTS (exemple illustratif).

D'autre part, lorsqu'on projette les trajectoires obtenues avec la stratégie G sur le niveau topologique du modèle de l'Espace libre, on constate que dans la majorité des essais, pour chaque déclinaison, la trajectoire définie avec l'information géométrique seule emprunte le Chemin topologique retenu avec la stratégie GTS (avec le contrôle par l'Information sémantique). Ce constat tend à montrer que la stratégie probabiliste mise en œuvre trouve et emprunte plus

facilement les passages larges. Ainsi, pour faciliter la définition de trajectoires géométriques lorsque le processus est scindé en deux phases (grossière sur la topologie et fine sur la géométrie), il semble pertinent d'orienter l'exploration de la topologie vers les passages les plus larges pour faciliter la seconde phase de planification réalisée sur la géométrie de l'environnement, ce que nous faisons grâce aux coûts du `Graphe topologique` définis en fonction de la complexité de chaque `Lieu`.

Les trois stratégies étudiées permettant de définir une trajectoire sans collision, afin de les comparer plus finement, nous allons nous attacher à comparer leurs performances au regard du nombre de configurations aléatoirement définies utilisées.

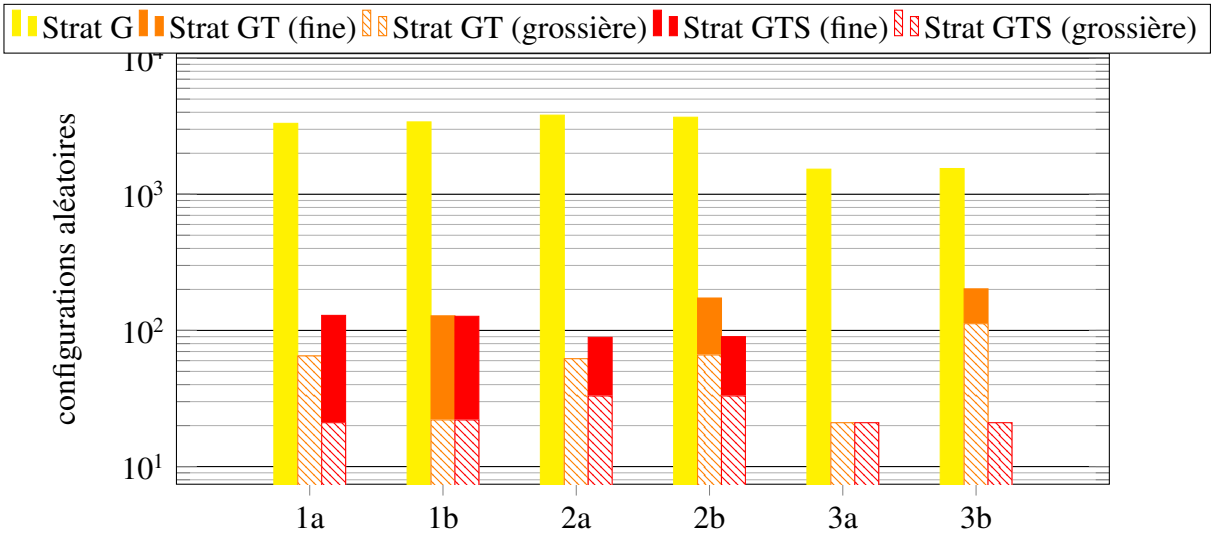


FIGURE 5.14 – Performances des différentes stratégies de planification (exemple illustratif).

La figure 5.14 synthétise ces performances pour les 3 stratégies de planification dans les différentes déclinaisons. Pour les stratégies GT et GTS, les configurations aléatoirement définies utilisées lors de la planification grossière pour définir les jalons dans les `Frontieres` du `Chemin topologique` (en hachuré) sont distinguées des configurations aléatoirement définies utilisées lors de la planification fine par l'algorithme RRT (en plein) :

- On constate, que l'introduction de données topologiques et le partitionnement du processus de planification en deux phases permettent, pour cette application, de diviser par 10 le nombre de configurations aléatoires utilisées.

- On constate aussi que l'exploitation de données sémantiques (stratégie GTS par rapport à stratégie GT) n'apporte pas nécessairement un gain de performance du point de vue du nombre de configurations aléatoires utilisées (requête de planification *a*). Les bonnes performances de la stratégie GT obtenues pour traiter la requête de planification *a* sont dues à un jalonnement particulièrement adapté de la trajectoire lors de la planification grossière. En effet, pour cette requête de planification, une fois le `Chemin topologique` L_1, L_6, L_4, L_8 et L_2 jalonné, la planification fine ne nécessite aucune configuration supplémentaire.

Par ailleurs, les performances de la planification grossière sont systématiquement améliorées lorsque la stratégie GTS est utilisée. Pour traiter la requête de planification *a*, les performances

globales de la stratégie GTS sont inférieures à celles de la stratégie GT du fait de l'utilisation de configurations supplémentaires pour la planification fine dans le Lieu L_3 lorsque celui-ci figure dans le Chemin topologique. Pour la requête de planification *b*, la tendance est inversée puisque dans ce cas, la stratégie GT retient un Chemin topologique qui traverse L_3 alors que la stratégie GTS ne le parcourt qu'en partie. Pour les déclinaisons 1b et 3a, les performances des stratégies GT et GTS sont identiques puisque les Chemins topologiques retenus par chacune de ces stratégies sont identiques.

La requête de planification *b* permet d'illustrer l'aspect circonstanciel des bonnes performances de la stratégie GT pour la requête de planification *a*. Elle permet de mettre en évidence l'intérêt des résultats obtenus grâce aux raisonnements réalisés sur des données symboliques. En effet, la stratégie GTS, grâce à la mise en œuvre de la sémantique, retient le même Chemin topologique pour les requêtes de planification *a* et *b*.

Pour aller plus loin dans l'analyse des performances des stratégies GT et GTS, les boîtes à moustaches de la figure 5.15 donnent la distribution (par quartiles) du nombre de configurations aléatoires utilisées dans les essais menés sur chaque déclinaison. On constate avec ces distributions que le fait de cibler les passages larges avec la stratégie GTS conduit à une dispersion moindre du nombre de configurations aléatoires utilisées. De plus, même pour la requête de planification *a* où la stratégie GT dispose d'un avantage circonstanciel (inutilité de la planification fine), la stratégie GTS présente des performances comparables avec une dispersion plus faible.

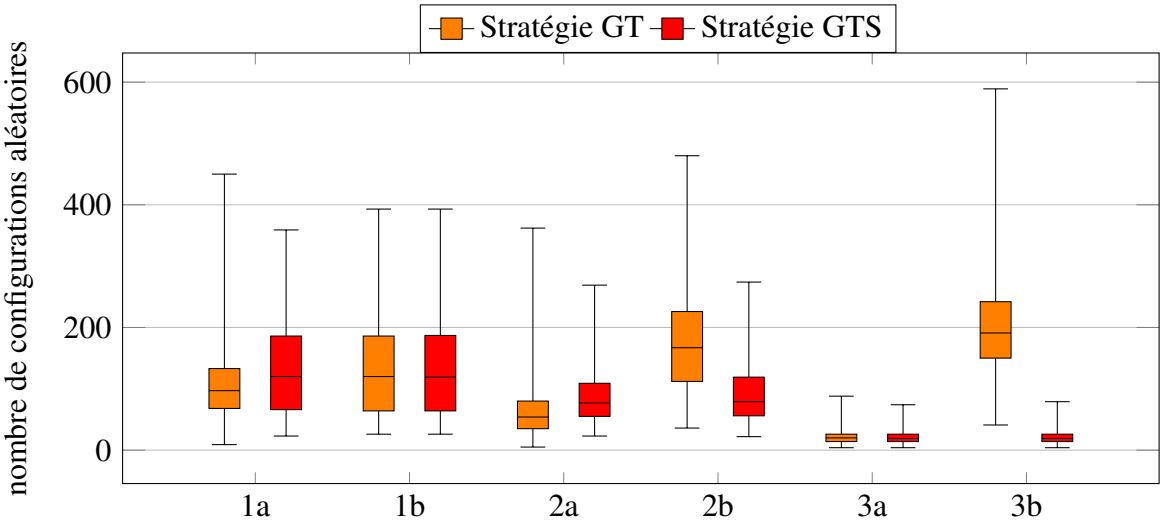


FIGURE 5.15 – Distribution du nombre de configurations aléatoires utilisées par les stratégies GT et GTS (exemple illustratif).

5.3.3.2 Seconde application : jeu d'encastrement

Dans cette section, nous présentons et interprétons les résultats obtenus pour les planifications de trajectoire réalisées sur l'application inspirée des jeux d'encastrement. Les indicateurs obtenus ont été extraits d'échantillons d'une centaine d'essais pour la stratégie G (sauf lorsqu'une centaine d'essais ont suffi à mettre en évidence l'incapacité de la technique à définir une trajectoire sans collision).

Pour cette application, dans chaque déclinaison (chaque objet manipulé), la qualité des trajectoires fournies par les différentes stratégies de planification varie. Les résultats obtenus sont synthétisés dans la table 5.5.

Stratégie	Critère	○	□	△	⬠	☆
Stratégie G	Taux de succès	98,6%	100%	90,2%	0%	0%
Stratégie GT	Faisabilité Chemin topologique			✓	✓	
	Taux de succès	∅	∅	99,0%	10,4%	∅
Stratégie GTS	Faisabilité Chemin topologique	✓	✓	✓	✓	
	Taux de succès	63,6%	91,6%	97,9%	22,8%	∅

Tableau 5.5 – Qualité des stratégies de planification de trajectoire (jeu d'encastrement).

Nous allons analyser ces résultats pour chacune des stratégies.

Stratégie G : La stratégie G n'a pas été en mesure de traiter toutes les requêtes de planification. Aucune trajectoire n'a pu être planifiée pour le pentagone et l'étoile. En revanche, pour le rond, le carré et le triangle, on observe un taux de succès supérieur à 90%. Pour ces trois objets présentant un taux de succès considérable, les trajectoires planifiées par la stratégie G empruntent le trou pentagonal (Lieu L_9) qui est le trou le plus large.

Stratégie GT : Dans cette application, les positions initiales des objets, la position de l'objectif et le critère de distance utilisé seul par la stratégie GT amènent la planification grossière à retenir le Chemin topologique empruntant les Lieux L_1, L_8 (trou triangulaire), L_2, L_4 (passage de complexité 2) et L_3 pour le rond, le carré et l'étoile. Ce chemin n'est pas réalisable pour tous ces objets (voir la compatibilité trou/objet dans la table 5.3). Pour le triangle et le pentagone, le Chemin topologique emprunte les Lieux L_1, L_9 (trou pentagonal), L_2, L_5 (passage de complexité 3) et L_3. Ce chemin est cette fois réalisable, mais on note toutefois que pour ces deux objets, les trajectoires définies peuvent comporter des jalons en collision. Les 2000 essais pour chaque Frontiere du Chemin topologique permettant d'y définir aléatoirement un jalon géométrique ne sont donc pas suffisants compte tenu des contraintes géométriques élevées pour traverser les trous de l'obstacle O_1.

Stratégie GTS : En utilisant l'Information sémantique pour contrôler la planification grossière, la stratégie GTS privilégie pour chaque objet le Chemin topologique empruntant le trou de la même forme pour traverser O_1 et le passage le moins complexe pour franchir O_2. Pour l'étoile pour laquelle la planification grossière ne dispose pas de l'information suffisante pour choisir le bon passage (aucun trou en forme d'étoile), le Chemin topologique retenu est similaire à celui retenu par la stratégie GT (L_1, L_8, L_2, L_4 et L_3). Les Chemins topologiques obtenus pour les autres objets sont : L_1, L_6, L_2, L_4 et L_3 pour le rond, L_1, L_7, L_2, L_4 et L_3 pour le carré, L_1, L_8, L_2, L_4 et L_3 pour le triangle et L_1, L_6, L_2,

L_4 et L_3 pour le pentagone. Ces quatre derniers Chemins topologiques sont tous réalisables, mais ne permettent toutefois pas d'atteindre un taux de succès de 100% à cause de la forte contrainte géométrique dans les trous de O_1 qui amène à y définir des jalons en collision. Pour chacun de ces Chemins topologiques, l'algorithme RRT est utilisé dans la première Étape topologique (L_1 encombré par les autres objets manipulables), et dans la troisième (L_2 encombré par O_3).

La figure 5.16 synthétise les performances obtenues (en terme de nombre moyen de configurations aléatoires utilisées) pour les 3 stratégies de planification et les cinq requêtes de planification (les différents objets). Cependant, pour cette application, cet indicateur est à considérer avec beaucoup de précaution car les échecs constatés, associés au nombre tronqué de configurations aléatoires utilisées introduisent un biais dans la distribution des échantillons.

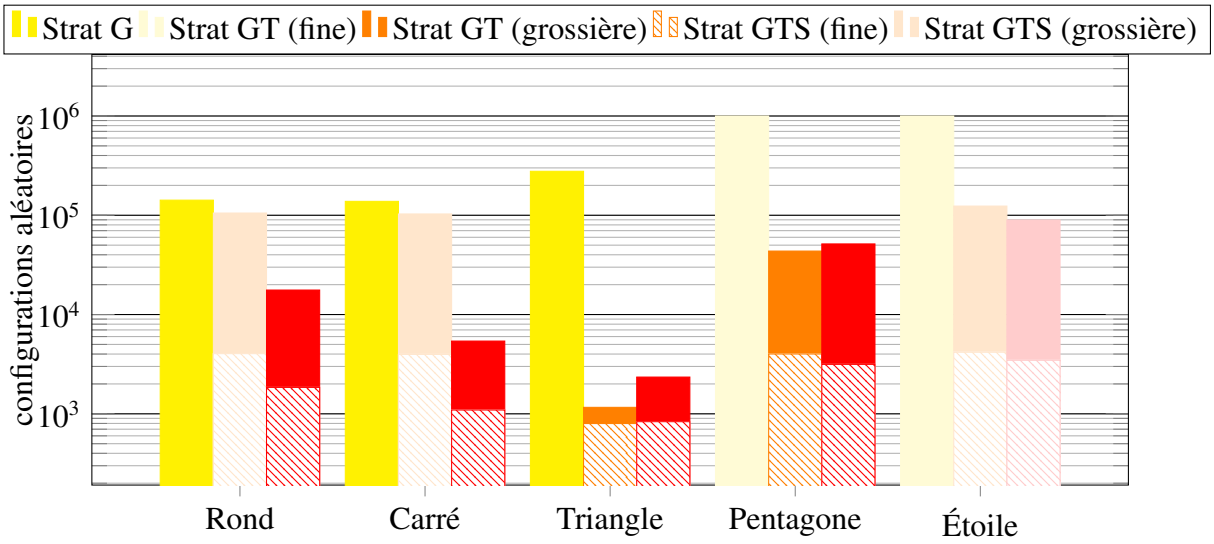


FIGURE 5.16 – Performances des différentes stratégies de planification (jeu d'encastrement).

Dans la figure 5.16 les éléments affichés en filigrane correspondent aux stratégies de planification et déclinaisons ne permettant pas de définir une trajectoire réalisable.

Ainsi, l'algorithme RRT de la stratégie G, pour le pentagone et l'étoile, a utilisé 10^6 configurations aléatoires ce qui correspond à la borne fixée entrainant l'abandon de la planification de trajectoire.

De même, pour les stratégies GT et GTS en échec, on voit qu'environ 4000 configurations ont été utilisées par la planification grossière, ce qui correspond aux deux jalons en collision définis aux Frontieres des Lieux associés aux trous de l'obstacle O_1.

En revanche, lorsque les Chemins topologiques retenus sont réalisables, le nombre de configurations aléatoires utilisées par la planification grossière tombe en dessous des 4000, ce qui correspond à un seul jalon géométrique en collision défini au niveau du trou permettant de traverser O_1. Pour ce qui est de la planification fine, lorsqu'on peut comparer les performances des stratégies GT et GTS (triangle, pentagone), on observe un phénomène circonstanciel similaire à celui observé dans l'expérimentation précédente. En effet, dans ce cas le Chemin topologique retenu par la stratégie GT sollicite la planification fine pour des portions de trajec-

toires plus simples à définir avec une stratégie probabiliste : la trajectoire entre les Frontieres $F_{2,9}$ et $F_{2,5}$ est plus courte et moins sinueuse que celle entre les Frontieres $F_{2,9}$ et $F_{2,4}$.

5.3.4 Synthèse sur la planification de trajectoire multi-niveaux

Les expérimentations menées permettent de comparer les trois stratégies de planification de trajectoire et donc de faire apparaitre les avantages et inconvénients de chacune.

La stratégie G, en ne se basant que sur le niveau géométrique du modèle de l'environnement ne met en œuvre aucun raisonnement *a priori* sur l'environnement et converge sur les trajectoires empruntant les passages les moins contraints géométriquement (chemins les plus simples). En revanche, malgré l'existence de trajectoires réalisables, celles-ci peuvent ne pas être trouvées (cas de l'étoile et du pentagone) ou nécessiter un temps de calcul interdisant toute interaction avec un opérateur humain (cas du rond, du carré et du triangle) lorsqu'elles présentent des contraintes géométriques importantes.

La stratégie GT est capable de fournir des Chemins topologiques dans un temps raisonnable (processus déterministe de la planification grossière). Cependant en ne contrôlant pas l'exploration de la topologie pour la définition des Chemins topologiques celle-ci est soumise aux positions des Lieux et des Frontieres ainsi que celles des configurations initiale et finale). La stratégie GT définit donc le Chemin topologique le plus court, mais celui-ci n'est pas nécessairement le plus simple (cas de la requête de planification *b* dans la première application, ou du pentagone dans la seconde), ou même réalisable (cas du rond, du carré et de l'étoile dans la seconde application).

Enfin, la stratégie GTS, en exploitant l'ensemble des informations (géométrique, topologique et sémantique) est capable de fournir des Chemins topologiques réalisables, même si une trajectoire 6D sans collision n'a pas pu y être déterminée.

Grâce à l'utilisation de l'Information sémantique pour le contrôle du processus de planification de trajectoire, cette stratégie est moins assujettie au contexte géométrique de la planification. En effet, le contrôle du processus de planification de trajectoire à l'aide de raisonnements réalisés sur des concepts ayant un haut niveau d'abstraction permet d'améliorer la robustesse de la planification grossière.

Cependant, le Chemin topologique qu'elle retient peut ne pas être réalisable (cas de l'étoile de la seconde application) ou ne pas être le plus simple (cas du rond, du carré, et du triangle toujours dans la seconde application) lorsque l'information sémantique disponible ou le traitement qui en est fait ne sont pas suffisants. Par exemple, dans la seconde application, de meilleures performances peuvent êtres obtenues par un meilleur contrôle de la planification grossière en modélisant l'ensemble de la compatibilité donnée dans la table 5.3. Un niveau d'expertise supérieur peut encore être modélisé en définissant des scores aux compatibilités des différents couples trou/objet.

La meilleure pertinence des Chemins topologiques définis à l'aide de l'Information sémantique nous permet de confirmer l'intérêt de la stratégie GTS. Les Chemins topologiques sont suffisamment adaptés à la requête de planification à traiter pour y relaxer localement la

contrainte de non-collision des processus probabilistes (définition des jalons dans les Frontieres et algorithme RRT). Cette possibilité de relaxer la contrainte de non-collision après un nombre déterminé de configurations aléatoires permet de limiter le temps de définition d'une trajectoire. On peut ainsi fournir une trajectoire suffisante pour assister un opérateur lorsqu'il effectue des tâches de manipulation dans un délai permettant l'interaction temps-réel entre le processus de planification et l'opérateur. Nous allons voir dans la section suivante les expérimentations de planification interactive de trajectoire menées avec cette stratégie de planification.

5.4 Validation de la planification de trajectoire interactive

L'objectif est ici d'évaluer la capacité du système à collaborer avec un opérateur humain pour définir une trajectoire.

Pour cela, la stratégie GTS de planification de trajectoire est utilisée sur une station permettant l'utilisation du bras haptique. Cette station dispose de 2Go de mémoire vive et est équipée d'un processeur bi-cœurs (2 thread par cœur) cadencé à 3.7 GHz. Pour mener les expérimentations de planification de trajectoire sur cette station, les paramètres de la stratégie GTS sont adaptés pour que celle-ci puisse être utilisée de manière interactive.

Nous allons voir comment, dans les applications étudiées, un opérateur humain peut, grâce aux modes d'interaction et de partage de contrôle, apporter son expertise pour la définition d'une trajectoire.

Dans cette section, nous allons tout d'abord décrire les paramètres du planificateur interactif de trajectoire (section 5.4.1). Pour cela, nos détaillerons d'une part les réglages apportés à la stratégie GTS pour une utilisation interactive, et d'autre part, les réglages des différents modes d'interaction et de partage de contrôle. Nous détaillerons ensuite les différents scénarios de manipulation testés (section 5.4.2).

5.4.1 Réglages du planificateur interactif de trajectoire

Cette section présente les paramètres du planificateur interactif de trajectoire retenus pour les expérimentations. Ces paramètres peuvent nécessiter de nouveaux réglages pour de nouvelles applications de planification interactive. En effet, leur réglage est soumis à l'environnement 3D dans lequel la planification interactive est réalisée (dimension des Corps rigides, lourdeur des modèles).

5.4.1.1 Planification de trajectoire

La stratégie de planification de trajectoire retenue est la stratégie GTS décrite dans la section précédente. Cette stratégie met en œuvre les données sémantiques, topologiques et géométriques de l'environnement. Elle permet, lorsque l'information sémantique est suffisante, de fournir une trajectoire définie sur un Chemin topologique particulièrement adapté à la requête de planification.

Le point clé de la mise en œuvre de la planification de trajectoire interactive est la mise à disposition d'un `Chemin topologique` jalonné géométriquement dans un délai suffisamment court pour permettre l'interaction en temps-réel. En effet, il est plus important ici d'intégrer les manipulations de l'opérateur en temps-réel pour être en mesure de l'assister dans les manipulations restantes que de définir une trajectoire complète sans collision. Ainsi, nous avons revu les limites des processus probabilistes de la planification de trajectoire pour cette utilisation :

- la contrainte de non-collision des jalons définis dans les `Frontieres` du `Chemin topologique` lors de la planification grossière est abandonnée après 500 configurations aléatoires testées,

- l'algorithme RRT utilisé pour la planification fine des `Étape topologique` traversant des `Lieux` encombrés est limité à 3000 configurations aléatoires.

5.4.1.2 Modes d'interaction

Les techniques de partage de contrôle qui enrichissent les différents modes d'interaction entre le planificateur de trajectoire et l'opérateur humain s'appuient fortement sur la direction instantanée du mouvement (section 4.2.2). Pour le calcul de cette direction instantanée du mouvement, le filtre décrit par l'équation 4.4 a été réalisé avec un coefficient $\alpha_m = 0.95$ pour une fréquence d'échantillonage voisine de 60Hz.

Assistance et partage d'autorité L'assistance proposée à l'opérateur par le planificateur de trajectoire est générée grâce à la configuration du prochain jalon sur la trajectoire. Un jalon est considéré atteint lorsque la distance entre la position courante de l'objet manipulé et celle du jalon est inférieure à 0.02 [1]. On rappelle de plus qu'au niveau topologique, l'atteinte de la `Frontiere` objectif d'une `Étape topologique` permet de passer à l'`Étape topologique` suivante avant que le jalon géométrique ne soit atteint.

L'assistance produite est filtrée sur 30 échantillons, soit, dans nos applications, sur une demi seconde.

Les efforts de l'assistance proposée sont modulés (*H-mode* section 4.2.2.1) entre 100 et 10% de 3N et 3N.m.

Prédiction d'intention et re-planification Pour la prédiction d'intention, les réglages concernent le test permettant de vérifier si l'opérateur suit l'assistance proposée. Ce test est composé de deux vérifications.

D'une part, l'amplitude du vecteur mouvement est contrôlée. Elle doit être supérieure à 0,003 pour garantir qu'il s'agit bien d'un mouvement volontaire de l'opérateur et non d'un bruit de mesure de la position du bras haptique.

1. Le Logiciel Virtools Dev n'utilise pas d'unité de longueur. Dans notre application, la taille de l'espace de travail (arête du cube) mesure 1.

Le second test porte sur l'écart angulaire entre le mouvement et l'assistance. Pour cela, le produit scalaire des deux vecteurs normalisés est réalisé. Si celui-ci est inférieur à 0.8 (angle d'environ 36°), on considère que l'opérateur suit l'assistance.

Pour la re-planification de trajectoire, une fois les probabilités associées à toutes les `Frontieres`, la re-planification de trajectoire est réalisée avec les coûts définis dans l'équation 4.10 où le coefficient k est défini à 15 afin de fournir des coûts comparables aux coûts définis à l'aide de l'`Information sémantique`. Les coûts ainsi définis permettent d'influencer le résultat obtenu par la planification grossière.

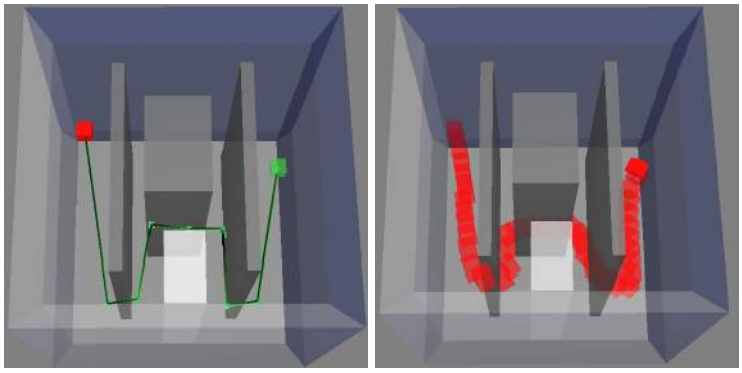
5.4.2 Résultats

Les réglages du planificateur de trajectoire interactif étant décrits, nous allons maintenant voir, dans les deux applications considérées (instance 3D de l'exemple illustratif et jeu d'encastrement), ce que les modes d'interaction et les techniques de partage de contrôle qui leur sont associées, ont permis de réaliser.

5.4.2.1 Assistance et partage d'autorité

L'assistance fournie à l'opérateur permet de l'aider à réaliser la tâche et s'adapte aux manipulations qu'il réalise. Elle le guide pour réaliser les manipulations permettant de parcourir la trajectoire planifiée tout en lui permettant de localement redéfinir la trajectoire.

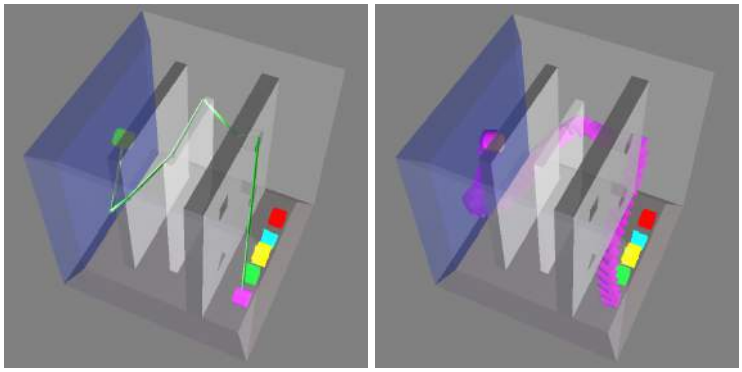
Instance 3D de l'exemple illustratif Dans cette application, l'assistance permet à l'opérateur de suivre la trajectoire planifiée. La possibilité de redéfinir localement la trajectoire est utilisée pour réaliser la tâche avec une trajectoire plus naturelle. Pour cela la redéfinition locale de la trajectoire permet de diminuer les variations d'altitude et de manipuler l'objet avec des trajectoires courbes plutôt que des lignes droites entre les jalons géométriques (figure 5.17). La redéfinition de la trajectoire est d'autant plus facile que le partage d'autorité diminue la force de l'assistance soumise à l'opérateur.



(a) Planification initiale. (b) Trajectoire planifiée.

FIGURE 5.17 – Trajectoire planifiée sans replanification dans la déclinaison 1a.

Jeu d'encastrement De même pour cette application, l'assistance permet encore une fois de guider l'opérateur sur le Chemin topologique retenu. On note que cette fois, en plus de l'usage évoqué dans l'application précédente, la redéfinition locale de la trajectoire revêt un caractère particulièrement important pour passer l'obstacle O_1. En effet, on rentre ici dans le cas où la responsabilité de diagnostiquer la faisabilité de la trajectoire planifiée revient à l'opérateur : étant donnée la limite du nombre de configurations aléatoires utilisées au cours de la planification grossière, les jalons aux extrémités des trous de O_1 sont le plus souvent définis en collision. L'opérateur doit donc, pour réaliser la tâche, modifier localement la trajectoire planifiée afin de franchir O_1 et atteindre L_3 (figure 5.18).



(a) Planification initiale. (b) Trajectoire planifiée.

FIGURE 5.18 – Trajectoire planifiée sans replanification pour le rond.

5.4.2.2 Re-planification et détection d'intention

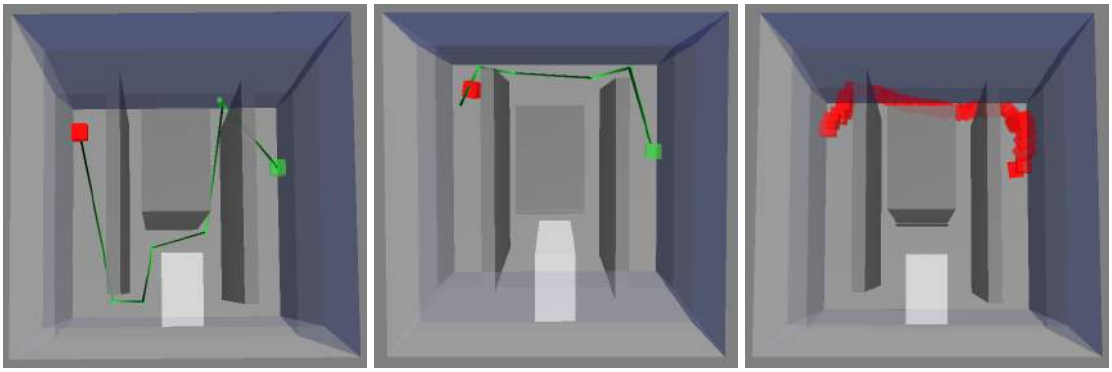
La re-planification et la détection d'intention permettent à l'opérateur de mettre en œuvre une expertise différente de celle modélisée dans les différents éléments de notre architecture traitant de sémantique. Nous allons voir ici comment cette possibilité est mise en œuvre dans les deux simulations considérées.

Instance 3D de l'exemple illustratif Dans cette application, l'opérateur peut considérer que l'encombrement du Lieu L_3 constitue un élément augmentant le coût des manipulations dans ce Lieu de telle manière que traverser des Lieux plus complexes pour l'éviter représente une alternative intéressante.

Dans ce cas, le Chemin topologique retenu dans l'environnement 3 est le plus adapté. En revanche, pour les environnements 1 et 2, la prédiction d'intention et la re-planification vont permettre au système d'intégrer les manipulations de l'opérateur au plus tôt et de re-planifier une trajectoire en conséquence. Le système sera alors en mesure d'assister l'opérateur sur cette trajectoire nouvellement définie :

Dans l'environnement 2, lorsque l'opérateur déplace le Corps rigide en direction du passage L_6 le système détecte l'intention de l'opérateur et lance un processus de re-planification. La détection d'intention permet de guider la planification grossière vers la Frontiere $F_{1,6}$.

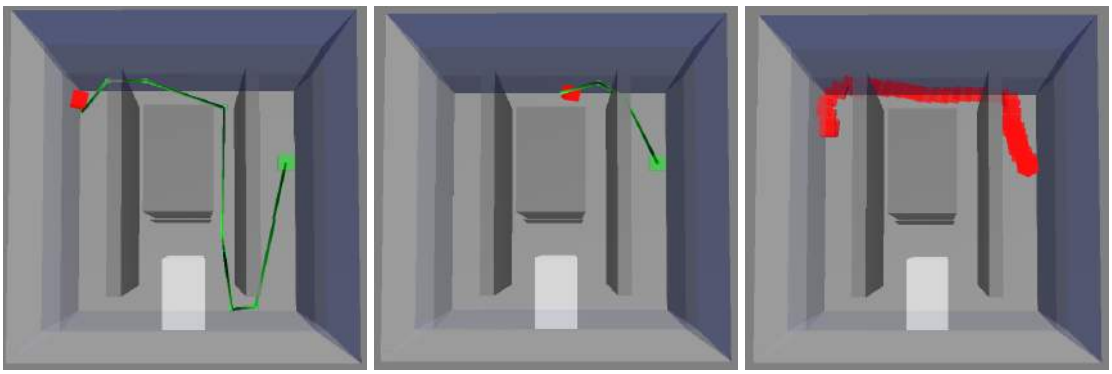
Le Chemin topologique retenu par la planification grossière lors de cette re-planification emprunte les Lieux L_1, L_6, L_4, L_8 et L_2. Dès que les jalons géométriques dans les Frontieres $F_{1,6}$, $F_{4,6}$, $F_{4,8}$ et $F_{2,8}$ sont définis, la trajectoire ainsi obtenue est utilisée pour assister l'opérateur (figure 5.19)



(a) Planification initiale. (b) Re-planification. (c) Trajectoire planifiée.

FIGURE 5.19 – Trajectoire planifiée avec replanification dans la déclinaison 2a.

Dans l'environnement 1, de même, l'opérateur peut éviter le Lieu L_3 (figure 5.20). Il doit alors déplacer le Corps rigide manipulé vers L_6, l'opérateur permet au système de définir une nouvelle trajectoire qui emprunte cette fois les Lieux L_1, L_6, L_4, L_{10}, L_3, L_7, et L_2 (figure 5.20(a)). La planification grossière emprunte ici le Lieu L_6 grâce aux coûts définis par la prédiction d'intention mais privilégie ensuite les Lieux les moins complexes. Le Lieu L_3 est donc toujours emprunté. Pour l'éviter et emprunter les Lieux L_8 et L_2, il faut donc que dans le Lieu L_4 l'opérateur manipule le Corps rigide vers la Frontiere $F_{4,8}$ pour déclencher la planification d'une nouvelle trajectoire sur le Chemin topologique L_4, L_8 et L_2 (figure 5.20(b)).



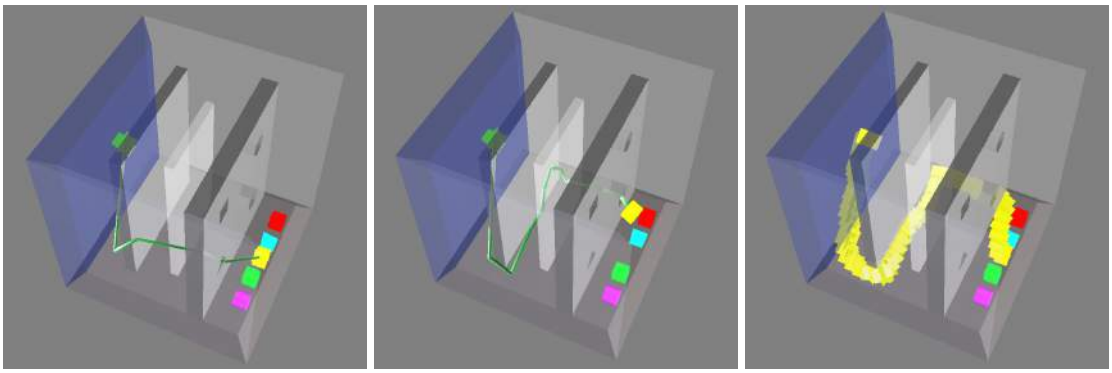
(a) Première re-planification. (b) Seconde re-planification. (c) Trajectoire planifiée.

FIGURE 5.20 – Trajectoire planifiée avec replanification dans la déclinaison 1a.

Jeu d'encastrement Dans cette application encore, la re-planification, couplée à la prédiction d'intention permet au système de bénéficier de l'expertise de l'opérateur. Cette expertise

permet à l'opérateur soit de guider le planificateur de trajectoire sur un Chemin topologique réalisable lorsqu'il y a eu un échec (cas de l'étoile), soit de le guider sur un Chemin topologique plus facile (cas du rond, du carré et du triangle).

Dans le cas de l'étoile, le Chemin topologique proposé initialement n'est en effet pas réalisable (trou en forme de triangle L_8 emprunté). L'opérateur peut donc déplacer l'étoile vers le trou en pentagone (L_9). La définition d'une nouvelle trajectoire est alors initiée. Le Chemin topologique retenu par la planification grossière emprunte les Lieux L_1, L_9, L_2, L_4 et L_3. Ce Chemin est réalisable et la trajectoire qui y est définie est utilisée pour assister l'opérateur (figure 5.21).



(a) Planification initiale. (b) Re-planification. (c) Trajectoire planifiée.

FIGURE 5.21 – Trajectoire planifiée pour l'étoile.

En ce qui concerne le rond, le carré et le triangle, pour ces trois Corps rigides, la re-planification et la détection d'intention peuvent encore être mises en œuvre (figure 5.22).

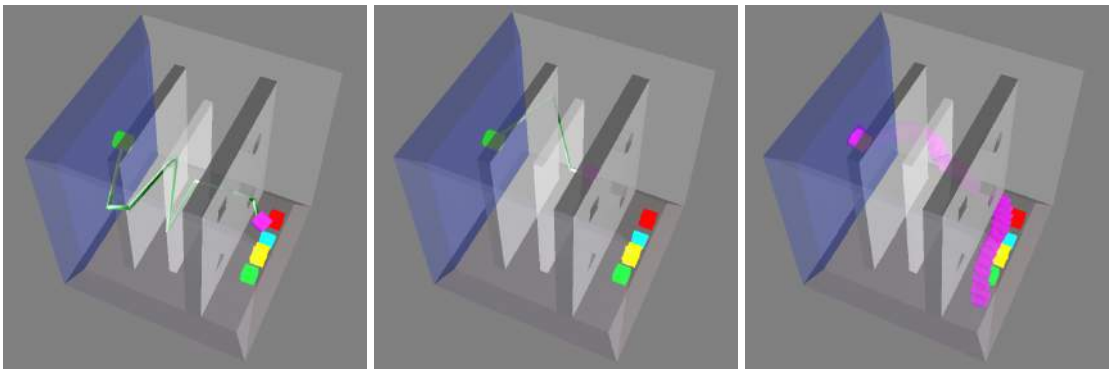
Par exemple, le trou le plus facile à emprunter (présentant le jeu le plus important) est le trou en forme de pentagone. L'opérateur peut donc faire bénéficier le planificateur de cette expertise en déplaçant ces Corps rigides vers ce trou. Le Chemin topologique L_1, L_9, L_2, L_4 et L_3 est alors à nouveau choisi lors de la planification grossière, et la trajectoire associée utilisée pour assister l'opérateur (figure 5.22(a)).

Pour tous les objets, lorsque les Lieux L_6 ou L_9 sont empruntés pour traverser l'obstacle O_1, l'opérateur peut choisir de minimiser la distance à parcourir dans L_2 et diriger l'Corps rigide manipulé vers le Lieu L_5. Une nouvelle trajectoire est alors définie sur le Chemin topologique L_2, L_5 et L_3, elle est ensuite utilisée pour assister l'opérateur au cours de dernières manipulations (figure 5.22(b)).

5.4.3 Synthèse sur la planification de trajectoire interactive

La planification de trajectoire proposée, ainsi que les modes d'interaction et de partage de contrôle que nous lui avons associés permettent à un processus automatique et à un opérateur humain de collaborer pour la définition de trajectoires.

D'un coté, le processus de planification de trajectoire proposé permet de guider l'opérateur sur un chemin, dans la topologie de l'Espace libre, permettant de joindre les configurations



(a) Première re-planification. (b) Seconde re-planification. (c) Trajectoire planifiée.

FIGURE 5.22 – Trajectoire planifiée avec replanification pour le rond.

initiales et finales. Le système de planification est en mesure de fournir cette assistance même s'il n'a pas pu déterminer de trajectoire géométrique sans collision.

D'un autre coté, l'opérateur peut faire bénéficier le système de son expertise lors de la planification de trajectoire. Il peut ainsi corriger les tronçons de trajectoire géométrique en collision. Il peut aussi corriger plus radicalement la trajectoire en indiquant un chemin différent à emprunter dans la topologie.

Cette possibilité pour l'opérateur de mettre en œuvre son expertise pour la planification de trajectoire permet de compenser les manques dans l'information sémantique de l'environnement et dans l'usage qui en est fait par le planificateur automatique de trajectoire.

L'ensemble des interactions entre le système automatique et l'opérateur humain est réalisé dans un délai suffisamment court pour en garantir la fluidité. Cette interaction fluide permet de maintenir l'opérateur en RV en immersion dans l'environnement virtuel.

5.5 Conclusion sur les expérimentations menées

Les expérimentations présentées dans ce chapitre ont permis de valider différentes propositions introduites dans ce mémoire.

Tout d'abord, les expérimentations de planification de trajectoire hors ligne ont montré l'intérêt de l'exploitation des descriptions sémantiques et topologiques de l'environnement de planification. En effet, nous avons montré que l'introduction de données topologiques et le fractionnement du processus de planification en deux phases (dont la première, grossière, permet de définir un chemin dans la topologie de l'environnement) permettent de diviser par dix le nombre de configurations aléatoires nécessaires pour trouver une trajectoire.

Ce gain de performance est obtenu en focalisant la recherche de configurations sans collision dans les zones de l'environnement les plus géométriquement contraintes (les `Frontieres`) et en n'utilisant des techniques de planification de trajectoire complexes que dans les `Lieux` où cela est nécessaire (les `Lieux` encombrés).

De plus, le niveau sémantique du modèle de l'environnement est apparu comme indispensable pour contrôler l'exploration de la topologie de l'environnement lors de la planification

grossière. Ce niveau permet en effet d'augmenter la robustesse de cette phase de la planification en maximisant les chances que le Chemin topologique retenu soit réalisable et qu'il soit simple d'y définir une trajectoire géométrique.

Il convient toutefois de noter que la modélisation de la sémantique de l'environnement reste un point critique. En effet, alors qu'une modélisation exhaustive des concepts abstraits de l'expertise d'un humain semble impossible, il reste tout de même indispensable de modéliser suffisamment d'informations et de raisonnements associés.

Dans un second temps, les expérimentations de planification de trajectoire interactive nous ont permis de valider :

- les adaptations faites à la planification automatique de trajectoire pour que celle-ci soit mise en œuvre de manière interactive,
- les modes d'interaction et de partage de contrôle entre système automatique et opérateur humain imaginés.

La solution de planification de trajectoire interactive réalisée permet bien au système de planification de trajectoire d'assister un opérateur humain lors de la manipulation d'objets 3D virtuels, mais aussi, à l'opérateur humain d'exercer son expertise propre et de la partager avec le système automatique pour la définition de la trajectoire. Ainsi, les forces des deux entités peuvent être mises en œuvre conjointement lors de la planification de trajectoire :

Le planificateur de trajectoire réalise une exploration grossière de l'environnement en en parcourant exhaustivement sa topologie. Il explore aussi la géométrie de certains Lieux aléatoirement afin d'y définir une trajectoire permettant de les franchir.

L'opérateur humain réalise un diagnostic de la trajectoire définie par le planificateur automatique. Il la corrige localement par la manipulation. Il peut aussi solliciter la définition d'une nouvelle trajectoire par le système automatique lorsque son expertise lui permet de percevoir de meilleurs Chemins topologiques que celui proposé par le système de planification de trajectoire.

L'architecture de planification de trajectoire proposée, exploitant des données sémantiques, topologiques et géométriques de l'environnement, ainsi que les modes d'interaction et de partage de contrôle avec un opérateur humain associés présentent donc un intérêt certain pour la définition de trajectoire d'objets 3D virtuels. Cependant, alors que le fonctionnement de l'ensemble de ces éléments a pu être validé, il conviendra désormais de faire subir à cet outil des essais par un panel d'utilisateurs plus large, de profils variés.

Conclusion et perspectives

Synthèse des contributions

Les contributions des travaux présentés dans ce mémoire portent sur trois domaines liés à la planification interactive de trajectoire :

1. **La modélisation de l'environnement** proposée est un modèle multi-niveaux (géométrique, topologique et sémantique) permettant de décrire les Corps rigides et l'Espace libre.

   - Les Corps rigides sont décrits, au niveau géométrique, à l'aide de modèles polyédriques (maillage de leur surface). Au niveau sémantique, un ensemble d'attributs permet de qualifier les Corps rigides à travers leur caractère fixe ou mobile et leur forme.
   - L'Espace libre 3D est décrit au niveau géométrique par une décomposition en cellules grâce à un arbre déséquilibré (octree). Le niveau topologique modélise des Lieux et leurs Frontieres, dont la connectivité est saisie dans un Graphe topologique. Enfin le niveau sémantique permet de qualifier les Lieux de l'Espace libre pour définir, par exemple, leur difficulté à être traversés, leur encombrement par des Corps rigides mobiles et leur forme.

2. **La planification de trajectoire** proposée exploite l'ensemble des données du modèle multi-niveaux de l'environnement. Elle est réalisée dans un processus en deux phases : une planification grossière d'abord, suivie d'une planification fine.

   - La planification grossière définit un Chemin topologique dans le niveau topologique du modèle de l'Espace libre. Pour cela, elle explore le Graphe topologique à l'aide d'un algorithme de Dijkstra. Cette exploration est contrôlée par les coûts associés aux arcs et nœuds du Graphe topologique définis en fonction notamment de l'Information sémantique des Lieux et Frontieres auxquels ils sont liés. Le Chemin topologique ainsi obtenu permet un premier jalonnement géométrique de la trajectoire (un jalon géométrique est défini dans chaque Frontiere du Chemin topologique), mais aussi de partitionner la requête de planification en Étapes topologiques. Chacune des Étapes topologiques permet de traverser un Lieu et d'atteindre un frontière du Chemin topologique.

- La planification fine traite chaque Étape topologique comme une requête de planification géométrique visant à définir la trajectoire, dans un Lieu entre deux jalons géométriques définis dans deux Frontieres. Elle utilise les niveaux géométriques du modèle des Corps rigides (p. ex. avec un algorithme A*) et du modèle de l'Espace libre (p. ex. avec un algorithme RRT) pour ajouter, si nécessaire, des jalons géométriques et ainsi définir une trajectoire permettant de traverser un Lieu retenu dans le Chemin topologique. L'Information sémantique du Lieu à traverser est utilisée pour adapter la stratégie de planification de chaque Étape topologique.

Ce partitionnement de la planification de trajectoire en deux phases, ainsi que le contrôle de chacune de ces phases à l'aide de l'Information sémantique permet d'augmenter la pertinence des solutions retenues tout en réduisant le temps de calcul (division par 10 du nombre de configurations aléatoires utilisées).

3. **L'interaction et le partage de contrôle** sont organisés autour de deux modes d'interaction tirant parti de l'architecture du modèle de l'environnement multi-niveaux et du processus de planification de trajectoire en deux phases. Le premier consiste à assister l'opérateur pour la manipulation en guidant ce dernier le long d'une trajectoire pré-calculée. Le second permet à l'opérateur de déclencher, s'il le souhaite, un nouveau processus de planification de trajectoire afin que la trajectoire sur laquelle il est guidé corresponde mieux à celle qu'il souhaite emprunter.

   Ces deux modes d'interaction ont permis d'adapter le planificateur de trajectoire afin d'en contrôler le temps d'exécution. Pour cela, la contrainte de non-collision de la trajectoire est relaxée localement lorsqu'un passage trop fortement contraint géométriquement est rencontré. Une portion de trajectoire en collision est alors définie et utilisée pour assister l'opérateur. Lorsqu'en exécutant la tâche, l'opérateur rencontre une telle portion de trajectoire, il lui revient de diagnostiquer la faisabilité du Chemin topologique qui lui est proposé et de mener les actions nécessaires pour corriger la trajectoire. Par la manipulation, il peut alors :

   - soit redéfinir localement la trajectoire géométrique lorsque le Chemin topologique est réalisable,
   - soit déclencher une re-planification qui définit un nouveau Chemin topologique lorsque celui qui est proposé n'est pas réalisable.

   L'efficacité et l'ergonomie des deux modes d'interaction sont améliorées par la mise en œuvre de techniques de partage de contrôle :

   - un partage d'autorité au niveau géométrique permet de moduler l'autorité du système automatique afin de fournir une assistance prégnante lorsqu'elle est suivie par l'opérateur, et une assistance atténuée lorsque l'opérateur explore des trajectoires alternatives ;

- une prédiction d'intention au niveau géométrique associe à chaque Frontiere environnant la position courante de l'objet manipulé une probabilité d'être visée ;
- un partage d'autorité au niveau topologique exploite les probabilités définies par la prédiction d'intention du niveau géométrique pour guider la planification grossière vers le Chemin topologique préféré par l'opérateur lorsque le processus de planification a été déclenché suite aux manipulations réalisées par l'opérateur.

L'intérêt de notre architecture a été démontré à l'aide d'un cas d'étude simple pour lequel la technique de planification de trajectoire proposée a fait apparaitre un gain significatif de performance permettant une utilisation dans un cadre interactif. Une autre application, inspirée des jeux d'encastrement pour le développement psychomoteur de jeunes enfants, a montré la capacité de notre technique de planification à traiter des tâches pour lesquelles les techniques de planification automatique de trajectoire classiques échouent.

Perspectives

L'intérêt de l'approche ayant été démontré, celle-ci peut-être mise en œuvre dans une application de RV permettant la simulation du montage, du démontage ou de la maintenance d'un produit industriel. Pour traiter un tel cas, la modularité de l'architecture permet d'une part d'enrichir le niveau sémantique du modèle de l'environnement avec des Attributs permettant de modéliser l'expertise propre au cas à traiter, et d'autre part, de compléter les bibliothèques de Planificateurs géométriques et de Planificateurs locaux pour définir des Stratégies de planification adaptées aux Lieux dans lesquels la trajectoire doit être définie.

Nous souhaitons mettre en avant trois perspectives qui représentent des enjeux particuliers pour la mise en œuvre de la planification interactive de trajectoire pour des applications industrielles telles que la simulation d'assemblage et de démontage. Ces trois perspectives sont la mise en œuvre d'outils de gestion de la connaissance pour améliorer l'exploitation du niveau sémantique de l'architecture, la définition de modalités de planification de trajectoire au contact de la surface des Corps rigides de l'environnement, et l'extension des fonctionnalités du planificateur de trajectoire vers la planification de tâche.

Utilisation d'outils de représentation de la connaissance

La mise en œuvre d'outils de représentation et d'exploitation de la connaissance (outil dédié, ontologies, graphes conceptuels...) doit augmenter la modularité du niveau sémantique de notre architecture. Ce gain de modularité doit être obtenu par une façon plus conviviale d'adapter le niveau sémantique à la connaissance métier liée à la tâche à traiter. Ces outils doivent aussi faciliter la mise en œuvre de nouveaux modes d'interaction (définition des Stratégies interactives) et de partage de contrôle (prédiction d'intention avec les données topologiques, partage d'autorité et prédiction d'intention avec les données sémantiques). La mise en œuvre de

tels outils demande d'adapter les différentes parties de l'architecture de planification interactive de trajectoire :

Modèle de l'environnement : le niveau sémantique du modèle de l'environnement doit être porté par cet outil. Il convient donc d'y définir les différents Attributs. La définition des Attributs doit pouvoir s'adapter à la connaissance métier de la tâche à traiter. Il faut aussi modéliser dans cet outil les notions de Lieux et de Corps rigides afin de leur attacher l'Information sémantique.

Planification de trajectoire : Les fonctionnalités de l'Interpréteur sémantique doivent être transférées à cet outil. Il doit donc définir les coûts du Graphe topologique lors de la planification grossière, mais aussi définir la Stratégie de planification à mettre en œuvre dans chaque Étape topologique. Il faut donc modéliser dans cet outil les différentes techniques de planification implémentées à travers les Planificateurs locaux et Planificateurs géométriques et la façon de les combiner dans les Stratégies de planification.

Interaction et partage de contrôle : Cet outil, en facilitant la manipulation des données du niveau sémantique du modèle de l'environnement doit permettre de définir les modalités d'acquisition de l'expertise l'opérateur. Par exemple, il doit permettre, après plusieurs replanification, d'acquérir les préférences de l'utilisateur. Cette acquisition constitue à la fois une prédiction d'intention sur les données topologiques, et un partage d'autorité sur les traitements réalisés sur les informations sémantiques. De même, cet outil doit faciliter la définition de modes de prédictions d'intention de l'opérateur à travers les données sémantiques.

Planification de trajectoire au contact

La planification de trajectoire au contact semble être un point clé pour traiter des problématiques d'assemblage ou de démontage de produits industriels. De telles tâches ne se font pas sans mouvement au contact. Les positions finales des composants du produit assemblé présentent en effet des contacts avec d'autres composants du produit. Lors du montage (démontage) les derniers (premiers) mouvements se font donc au contact de surfaces d'autres composants du système qui peuvent de plus êtres conçues de manière à faciliter ces opérations (fonctionnalité de guidage du mouvement). La mise en œuvre de ces modalités de planification de trajectoire amène à enrichir l'ensemble des parties de l'architecture de planification interactive de trajectoire :

Modèle de l'environnement : Le modèle des Corps rigides doit être complété afin d'y disposer d'un niveau topologique (par exemple, utilisation de B-Rep). Le modèle de l'Espace libre doit être étendu pour modéliser des Lieux dédiés au mouvement au contact (Lieux au contact des surfaces des Corps rigides). De plus le niveau sémantique du modèle de l'environnement doit aussi être complété pour y faire apparaitre la fonctionnalité des surfaces et des Lieux (par exemple le guidage) dédiés au mouvement au contact.

Planification de trajectoire : Des modalités de planification de trajectoire au contact des surfaces des Corps rigides doivent être définies à travers de nouveaux Planificateurs locaux et Planificateurs géométriques. La façon de les combiner dans des Stratégies de planification en fonction de l'Information sémantique des Lieux à traverser doit aussi être définie, ainsi que l'organisation d'une planification multi-modale mettant en œuvre des techniques de planification libre et au contact.

Interaction et partage de contrôle : Les modes d'interaction et les techniques de partage de contrôle avec des trajectoires au contact doivent être vérifiés et adaptés si nécessaire. Un mode de guidage favorisant le maintien du contact doit aussi être défini. Le partage d'autorité doit être adapté à cette nouvelle contrainte (le maintien au contact doit-il, comme l'assistance, être modulé par le *H-mode* ?). De même la technique de prédiction d'intention doit être adaptée pour tenir compte de la réduction du nombre de ddl du mouvement due au contact.

Extension vers la planification de tâches

Cette perspective présente une échéance à plus long terme. L'objectif est ici de compléter les fonctionnalités du Planificateur sémantique avec de la planification de tâches. Les tâches que le Planificateur sémantique doit traiter sont des tâches de manipulation de composants de produits industriels (montage, démontage, maintenance). Pour le montage et le démontage, l'intégration de planification de tâches consiste à permettre au système de décomposer l'assemblage ou le démontage en une séquence de tâches élémentaires. Chaque tâche élémentaire permet d'assembler un composant dont la trajectoire est définie par ailleurs grâce à l'architecture proposée. Pour la maintenance, une approche plus originale peut être envisagée. On peut imaginer pour de telles tâches que la planification de tâche et la planification de trajectoire sont des processus plus entremêlés où le chemin de démontage d'un composant permet de déterminer les autres composants du système à démonter au préalable. Le partitionnement du processus de planification proposé en deux phases, ainsi que la disponibilité de données sémantiques dans le modèle de l'environnement semblent présenter un cadre approprié à de tels développements. En effet, dans ce cas, le dégagement des Lieux du Chemin topologique des Corps rigides qui les encombrent peut être considéré comme une tâche à planifier.

Bibliographie

[Aarno 05] Daniel Aarno, Staffan Ekvall & Danica Kragic. *Adaptive virtual fixtures for machine-assisted teleoperation tasks*. In ICRA, International Conference on Robotics and Automation, pages 1139–1144, Barcelona (Spain). IEEE, April 2005.

[Abbink 10] David A Abbink & Mark Mulder. *Neuromuscular analysis as a guideline in designing shared control*. Advances in haptics, vol. 109, pages 499–516, 2010.

[Abbink 12] David A Abbink, Mark Mulder & Erwin R Boer. *Haptic shared control : smoothly shifting control authority ?* Cognition, Technology & Work, vol. 14, no. 1, pages 19–28, 2012.

[Ahmadi-Pajouh 07] Mohammad Ali Ahmadi-Pajouh, Farzad Towhidkhah, Shahriar Gharibzadeh & Maysam Mashhadimalek. *Path planning in the hippocampo-prefrontal cortex pathway : An adaptive model based receding horizon planner*. Medical hypotheses, vol. 68, no. 6, pages 1411–1415, 2007.

[Aigner 97] Peter Aigner & Brenan McCarragher. *Human integration into robot control utilising potential fields*. In ICRA, International Conference on Robotics and Automation, volume 1, Albuquerque (USA), pages 291–296. IEEE, April 1997.

[Alt 88] Helmut Alt & Emo Welzl. *Visibility graphs and obstacle-avoiding shortest paths*. Zeitschrift für Operations Research, vol. 32, no. 3-4, pages 145–164, 1988.

[Amato 96] Nancy M. Amato & Yan Wu. *A randomized roadmap method for path and manipulation planning*. In ICRA, International Conference on Robotics and Automation, volume 1, Mineapolis (USA), pages 113–120. IEEE, April 1996.

[Amato 98] Nancy M Amato, O Burchan Bayazit & Lucia K Dale. *OBPRM : An obstacle-based PRM for 3D workspaces*. In WAFR '98, International

Workshop on Algorithmic Foundations of Robotics, Huston (USA). Citeseer, March 1998.

[Anderson 10] Sterling J Anderson, Steven C Peters, Karl Iagnemma & James Overholt. *Semi-autonomous stability control and hazard avoidance for manned and unmanned ground vehicles*. Rapport technique, DTIC Document, 2010.

[Aurenhammer 91] Franz Aurenhammer. *Voronoi diagrams – a survey of a fundamental geometric data structure*. Computing Surveys, vol. 23, no. 3, pages 345–405, 1991.

[Bainbridge 83] Lisanne Bainbridge. *Ironies of automation*. Automatica, vol. 19, no. 6, pages 775–779, 1983.

[Barraquand 91] Jerome Barraquand & Jean-Claude Latombe. *Robot motion planning : A distributed representation approach*. The International Journal of Robotics Research, vol. 10, no. 6, pages 628–649, 1991.

[Bayazit 01] O Burchan Bayazit, Guang Song & Nancy M Amato. *Enhancing randomized motion planners : Exploring with haptic hints*. Autonomous Robots, vol. 10, no. 2, pages 163–174, 2001.

[Bennes 12] Lionel Bennes, Florence Bazzaro & Jean-Claude Sagot. *Virtual reality as a support tool for ergonomic-style convergence : multidisciplinary interaction design methodology and case study*. In VRIC, Virtual Reality International Conference, Laval (France), page 24. ACM, March 2012.

[Bennis 05] Fouad Bennis, Damien Chablat & Philippe Dépincé. *Virtual reality : A human centered tool for improving Manufacturing*. In Proceedings of Virtual Concept, Biarritz, France, November 2005.

[Bézier 66] Pierre Bézier. *Définition numérique des courbes et surfaces I*. Automatisme, vol. 11, no. 12, pages 625–632, 1966.

[Boor 99] Valérie Boor, Mark H Overmars & A Frank van der Stappen. *The Gaussian sampling strategy for probabilistic roadmap planners*. In ICRA, International Conference on Robotics and AutomationInternational Conference on Robotics and Automation, volume 2, Buffalo (USA), pages 1018–1023. IEEE, April 1999.

[Boost 14] Boost. Boost, *C++ libraries*, 2014. `http://www.boost.org`.

[Bordegoni 06] Monica Bordegoni & Umberto Cugini. *Haptic modeling in the conceptual phases of product design*. Virtual Reality, vol. 9, no. 2, pages 192–202, 2006.

[Bordegoni 09] Monica Bordegoni, Umberto Cugini, Paolo Belluco & Marcello Aliverti. *Evaluation of a haptic-based interaction system for virtual manual assembly*. In VMR, Conférence on Virtual and Mixed Reality, San Diego (USA), pages 303–312. Springer, July 2009.

[Bourdot 10] Patrick Bourdot, Thomas Convard, Flavien Picon, Mehdi Ammi, Damien Touraine & Jean-Marc Vézien. *VR-CAD integration : Multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models*. Computer-Aided Design, vol. 42, no. 5, pages 445–461, 2010.

[Brandt 07] Thorsten Brandt, Thomas Sattel & Michael Bohm. *Combining haptic human-machine interaction with predictive path planning for lane-keeping and collision avoidance systems*. In IV, Intelligent Vehicles Symposium, Istanbul (Turkey), pages 582–587. IEEE, June 2007.

[Brooks 82a] Rodney A. Brooks. *Solving the find-path problem by representing free space as generalized cones*. Rapport technique 674, MIT, 1982.

[Brooks 82b] Rodney A. Brooks & Tomas Lozano-Perez. *A subdivision algorithm in configuration space for findpath with rotation*. Rapport technique 684, MIT, 1982.

[Cailhol 13] Simon Cailhol, Philippe Fillatreau & Jean-Yves Fourquet. *Interactive multimodal Path Planning in immersion*. In ECMSM, Electronics, Control, Measurement, Signals and their application to Mechatronics, Toulouse (France), June 2013.

[Cailhol 14a] Simon Cailhol, Philippe Fillatreau, Jean-Yves Fourquet & Yingshen Zhao. *A multi-layer approach for interactive path planning control*. In ICINCO, International Conference on Informatics in Control, automation and robotics, Vienna (Austria), September 2014.

[Cailhol 14b] Simon Cailhol, Philippe Fillatreau, Jean-Yves Fourquet & Yingshen Zhao. *A multi-layer approach for path planning control in Virtual Reality simulation*. In IDMME, Integrated Design and Manufacturing in Mechanical Engineering, Toulouse (France), June 2014.

[Cailhol 15a] Simon Cailhol, Philippe Fillatreau, Yingshen Zhao & Jean-Yves Fourquet. *A hierarchic approach for path planning in Virtual Reality*.

IJIDeM, International Journal on Interactive Design and Manufacturing, pages 1–12, 2015.

[Cailhol 15b] Simon Cailhol, Philippe Fillatreau, Yingshen Zhao & Jean-Yves Fourquet. *Hierarchic interactive path planning in Virtual Reality*. LNEE, Lecture Notes in Electrical Engineering, 2015.

[Cailhol 15c] Simon Cailhol, Philippe Fillatreau, Yingshen Zhao & Jean-Yves Fourquet. *A Multi-Layer Approach of Interactive Path Planning for Assisted Manipulation in Virtual Reality (poster)*. In VR 2015, IEEE Virtual Reality, Arles (France). IEEE, March 2015.

[Cailhol a] Simon Cailhol, Philippe Fillatreau, Jean-Yves Fourquet & Yingshen Zhao. *A multi-layer approach for interactive path planning control in Virtual Reality simulation*. In ICVRV, International Conference on Virtual Reality and Visualization, Shenyang (China), August 2014, Shenyang (China).

[Canny 88] John Canny. The complexity of robot motion planning. MIT press, 1988.

[Cantamessa 12] Marco Cantamessa, Francesca Montagna & Paolo Neirotti. *An empirical analysis of the PLM implementation effects in the aerospace industry*. Computers in Industry, vol. 63, no. 3, pages 243–251, 2012.

[Choset 05] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki & Sebastian Thrun. Principles of robot motion : theory, algorithms, and implementations (intelligent robotics and autonomous agents). The MIT Press, 2005.

[Connolly 90] Christopher I. Connolly, John B Burns & Richard S. Weiss. *Path planning using Laplace's equation*. In ICRA, International Conference on Robotics and AutomationInternational Conference on Robotics and Automation, Tsushiura (Japan), pages 2102–2106. IEEE, May 1990.

[Cottet 08] Didier Cottet. *Industry perspective on multi domain simulations and virtual prototyping*. In COMPEL, Workshop on Control and Modeling for Power Electronics, Zurich (Switzerland), pages 1–6. IEEE, August 2008.

[Cottet 12] Didier Cottet, I Stevanovic, B Wunsch, D Daroui, J Ekman & G Antonini. *EM simulation of planar bus bars in multi-level power converters*. In EMC, International Symposium on Electromagnetic Compatibility, Pittsburg (USA), pages 1–6. IEEE, August 2012.

[Crandall 02] Jacob W Crandall & Michael A Goodrich. *Characterizing efficiency of human robot interaction : A case study of shared-control teleoperation*. In IROS, International Conference on Intelligent Robots and SystemsInternational Conference on Intelligent Robots and Systems, volume 2, Lausanne (Switzerland), pages 1290–1295. IEEE/RSJ, September 2002.

[Dalibard 09] Sébatien Dalibard & Jean-Paul Laumond. *Control of probabilistic diffusion in motion planning*. Algorithmic Foundation of Robotics VIII, pages 467–481, 2009.

[Dangelmaier 05] Wilhelm. Dangelmaier, Matthias Fischer, Jürgen Gausemeier, Michael Grafe, Carsten Matysczok & Bengt Mueck. *Virtual and augmented reality support for discrete manufacturing system simulation*. Computers in Industry, vol. 56, no. 4, pages 371–383, 2005.

[Dassault Systemes 14a] Dassault Systemes. *Catia*, 2014. `http://www.3ds.com/fr/produits-et-services/catia/fonctionnalites/ingenierie-mecanique/`.

[Dassault Systemes 14b] Dassault Systemes. *Delmia*, 2014. `http://www.3ds.com/fr/produits-et-services/delmia/`.

[Dassault Systemes 14c] Dassault Systemes. *SolidWorks*, 2014. `http://www.solidworks.fr/sw/products/3d-cad/cad-animation.htm`.

[De Boor 76] Carl De Boor. *Splines as linear combinations of B-splines. A survey*. 1976.

[de Winter 04] Joost CF de Winter, Max Mulder, MM van Paassen & T Yamamura. *Weighted two-dimensional longitudinal impedance for driving support system*. In International Conference on Systems, Man and Cybernetics, volume 1, pages 256–260. IEEE, 2004.

[de Winter 08] Joost CF de Winter, Max Mulder, MM Van Paassen, David A Abbink & Peter A Wieringa. *A two-dimensional weighting function for a driver assistance system*. Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics, vol. 38, no. 1, pages 189–195, 2008.

[Debus 01] Thomas Debus, Jeffrey Stoll, Robert D Howe & Pierre Dupont. *Cooperative human and machine perception in teleoperated assembly*. In Experimental Robotics VII, pages 51–60. Springer, 2001.

[Dedeoglu 99] Goksel Dedeoglu, Maja J Mataric & Gaurav S Sukhatme. *Incremental online topological map building with a mobile robot*. In Photonics

East'99, pages 129–139. International Society for Optics and Photonics, 1999.

[Delaunay 34] Boris Delaunay. *Sur la sphere vide*. Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk, vol. 7, no. 793-800, pages 1–2, 1934.

[Demiris 02] John Demiris & Gillian Hayes. *f 3 Imitation as a Dual-Route Process Featuring Predictive and Learning Components ; 4 Biologically Plausible Computational Model*. Imitation in animals and artifacts, page 327, 2002.

[Demyen 06] Douglas Demyen & Michael Buro. *Efficient triangulation-based pathfinding*. In national conference of the American Association on Artificial Intelligence, volume 6, pages 942–947, 2006.

[Descartes 64] René Descartes. Le monde de monsieur descartes ou le traité de la lumière. Michel Bobin et Nicolas le Gras, 1664.

[Di Gironimo 12] Giuseppe Di Gironimo, Giovanna Matrone, Andrea Tarallo, Michele Trotta & Antonio Lanzotti. *A virtual reality approach for usability assessment : case study on a wheelchair-mounted robot manipulator*. Engineering with Computers, pages 1–15, 2012.

[Dijkstra 59] Edsger W Dijkstra. *A note on two problems in connexion with graphs*. Numerische mathematik, vol. 1, no. 1, pages 269–271, 1959.

[Dijkstra 03] Jan Dijkstra, Jos van Leeuwen & Harry Timmermans. *Evaluating design alternatives using conjoint experiments in virtual reality*. Environment and Planning B, vol. 30, no. 3, pages 357–368, 2003.

[Dragan 13] Anca D Dragan & Siddhartha S Srinivasa. *A Policy Blending Formalism for Shared Control*. International Journal of Robotics Research, 2013.

[Dufourd 05] Delphine Dufourd. *Des cartes combinatoires pour la construction automatique de modèles d'environnement par un robot mobile*. Thèse de doctorat, Institut National Polytechnique de Toulouse, 2005.

[Eddy 02] John Eddy & Kemper E Lewis. *Visualization of Multidimensional Design and Optimization Data Using Cloud Visualization*. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, numéro 36223, pages 899–908. American Society of Mechanical Engineers, 2002.

[Eig 14] EIGEN, *C++ library for linear algebra*, 2014. http://eigen.tuxfamily.org.

[Fagg 04] Andrew H Fagg, MT Rosenstein, Robert Platt & Roderic A Grupen. *Extracting user intent in mixed initiative teleoperator control*. In American Institute of Aeronautics and Astronautics Intelligent Systems Technical Conference, 2004.

[FCL 14] FCL. FCL, F*lexible* COLLISION L*ibrary*, 2014. http://gamma.cs.unc.edu/FCL/fcl_docs/webpage/generated/index.html.

[Ferre 04] Etienne Ferre & Jean-Paul Laumond. *An iterative diffusion algorithm for part disassembly*. In International Conference on Robotics and Automation, volume 3, pages 3149–3154. IEEE, 2004.

[Filliat 03] David Filliat & Jean-Arcady Meyer. *Map-based navigation in mobile robots : : I. a review of localization strategies*. Cognitive Systems Research, vol. 4, no. 4, pages 243–282, 2003.

[Fischer 11] Xavier Fischer & Jean-Pierre Nadeau. Research in interactive design : Virtual, interactive and integrated product design and manufacturing for industrial innovation. Springer, 2011.

[Flavigné 10a] David Flavigné. *Planification de mouvement interactive : coopération humain-machine pour la recherche de trajectoire et l'animation*. Thèse de doctorat, LAAS, 2010.

[Flavigné 10b] David Flavigné & Michel Taïx. *Improving motion planning in weakly connected configuration spaces*. In International Conference on Intelligent Robots and Systems, pages 5900–5905. IEEE/RSJ, 2010.

[Flemisch 03] Frank O. Flemisch, Catherine A. Adams, Sheila R. Conway, Ken H. Goodrich, Michael T. Palmer & Paul C. Schutte. *The H-Metaphor as a guideline for vehicle automation and interaction*. Rapport technique, NASA, 2003.

[Flemisch 08a] Frank O. Flemisch, Johann Kelsch, Christian Löper, Anna Schieben & Julian Schindler. *Automation spectrum, inner/outer compatibility and other potentially useful human factors concepts for assistance and automation*. Human Factors for Assistance and Automation. Maastricht : Shaker Publishing, 2008.

[Flemisch 08b] Frank O. Flemisch, Johann Kelsch, Christian Löper, Anna Schieben, Julian Schindler & Matthias Heesen. *Cooperative control and active interfaces for vehicle assistance and automation*. In FISITA World automotive Congress, 2008.

[Flemisch 12] Frank O. Flemisch, Matthias Heesen, Tobias Hesse, Johann Kelsch, Anna Schieben & Johannes Beller. *Towards a dynamic balance between humans and automation : authority, ability, responsibility and control in shared and cooperative control situations*. Cognition, Technology & Work, vol. 14, no. 1, pages 3–18, 2012.

[Forsyth 06] Benjamin AC Forsyth & Karon E MacLean. *Predictive haptic guidance : intelligent user assistance for the control of dynamic tasks*. Transactions on Visualization and Computer Graphics, vol. 12, no. 1, pages 103–113, 2006.

[Fuchs 06a] Philippe Fuchs. Le traité de la réalité virtuelle, vol. 1 : l'homme et l'environnement virtuel. Les Presses de l'Ecole des Mines, 2006.

[Fuchs 06b] Philippe Fuchs. Le traité de la réalité virtuelle, vol. 4 : Les application de la réalité virtuelle. Les Presses de l'Ecole des Mines, 2006.

[Fuchs 06c] Philippe Fuchs, Guillaume Moreau, Sabine Coquillart & Jean-Marie Burkhardt. Le traité de la réalité virtuelle, vol. 2 : Interfaçage, immersion et interaction en environnement virtuel, volume 2. Les Presses de l'Ecole des Mines, 2006.

[Fuchs 06d] Philippe Fuchs, Guillaume Moreau, Jacques Tisseau, Thomas Convard & François d' Aubert. Le traité de la réalité virtuelle vol. 3 : Les outils et modèles informatiques des environnements virtuels). Les Presses de l'Ecole des Mines, 2006.

[Fuchs 09] Philippe Fuchs. Le traité de la réalité virtuelle, vol. 5 : Les humains virtuels, volume 5. Les Presses de l'Ecole des Mines, 2009.

[Galindo 05] Cipriano Galindo, Alessandro Saffiotti, Silvia Coradeschi, Pär Buschka, Juan-Antonio Fernandez-Madrigal & Javier González. *Multi-hierarchical semantic maps for mobile robotics*. In International Conference on Intelligent Robots and Systems, pages 2278–2283. IEEE/RSJ, 2005.

[Gerdes 01] J Christian Gerdes & Eric J Rossetter. *A unified approach to driver assistance systems based on artificial potential fields*. Journal of Dynamic Systems, Measurement, and Control, vol. 123, no. 3, pages 431–438, 2001.

[Griffiths 05] Paul G Griffiths & R Brent Gillespie. *Sharing control between humans and automation using haptic interface : primary and secondary*

task performance benefits. Human Factors : The Journal of the Human Factors and Ergonomics Society, vol. 47, no. 3, pages 574–590, 2005.

[Gude 07] Dietmar Gude, Eike Branahl, Philipp Kawalek, Andreas Prions & Wolfgang Laurig. *Evaluation of a virtual reality-based ergonomics tutorial*, 2007.

[Hart 68] Peter E Hart, Nils J Nilsson & Bertram Raphael. *A formal basis for the heuristic determination of minimum cost paths*. Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pages 100–107, 1968.

[He 09] Xuejian He & Yonghua Chen. *Haptic-aided robot path planning based on virtual tele-operation*. Robotics and computer-integrated manufacturing, vol. 25, no. 4, pages 792–803, 2009.

[Holleman 00] Christopher Holleman & EE Kavraki. *A framework for using the workspace medial axis in PRM planners*. In International Conference on Robotics and Automation, volume 2, pages 1408–1413. IEEE, 2000.

[Hsu 98] David Hsu, Lydia E Kavraki, Jean-Claude Latombe, Rajeev Motwani, Stephen Sorkin*et al.* *On finding narrow passages with probabilistic roadmap planners*. In Workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspectiv, pages 141–154, 1998.

[Hsu 05] David Hsu, Gildardo Sánchez-Ante & Zheng Sun. *Hybrid PRM sampling with a cost-sensitive adaptive strategy*. In International Conference on Robotics and Automation, pages 3874–3880. IEEE, 2005.

[Iacob 12] Robert Iacob, Diana Popescu & Peter Mitrouchev. *Assembly/Disassembly Analysis and Modeling Techniques : A Review.* Strojniski Vestnik/Journal of Mechanical Engineering, vol. 58, no. 11, 2012.

[Kavraki 96] Lydia E. Kavraki, Petr Svestka, Jean-Claude Latombe & Mark H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. Transactions on Robotics and Automation, vol. 12, no. 4, pages 566–580, 1996.

[Khatib 85] Oussama Khatib. *Real-time obstacle avoidance for manipulators and mobile robots*. In International Conference on Robotics and Automation, volume 2, pages 500–505. IEEE, 1985.

[Kieferl 03] Joachim Kieferl & Uwe Wössner. *Mixed Realities : Improving the planning process by using augmented and virtual reality*. In Proceedings of the 4th Conference on new technologies in landscape architecture, 2003.

[Kienle 12] Martin Kienle, Daniel Damböck, Heiner Bubb & Klaus Bengler. *The ergonomic value of a bidirectional haptic interface when driving a highly automated vehicle*. Cognition, Technology & Work, pages 1–8, 2012.

[Koditschek 87] Daniel E Koditschek. *Exact robot navigation by means of potential functions : Some topological considerations*. In International Conference on Robotics and Automation, volume 4, pages 1–6. IEEE, 1987.

[Kofman 05] Jonathan Kofman, Xianghai Wu, Timothy J Luu & Siddharth Verma. *Teleoperation of a robot manipulator using a vision-based human-robot interface*. Industrial Electronics, vol. 52, no. 5, pages 1206–1219, 2005.

[Koren 91] Yoram Koren & Johann Borenstein. *Potential field methods and their inherent limitations for mobile robot navigation*. In International Conference on Robotics and Automation, pages 1398–1404. IEEE, 1991.

[Kortenkamp 94] David Kortenkamp & Terry Weymouth. *Topological mapping for mobile robots using a combination of sonar and vision sensing*. In national conference of the American Association on Artificial Intelligence, volume 94, pages 979–984, 1994.

[Kuipers 91] Benjamin Kuipers & Yung-Tai Byun. *A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations*. Robotics and autonomous systems, vol. 8, no. 1, pages 47–63, 1991.

[Kunz 99] Clayton Kunz, Thomas Willeke & Illah R Nourbakhsh. *Automatic mapping of dynamic office environments*. Autonomous Robots, vol. 7, no. 2, pages 131–142, 1999.

[Kurz 95] Andreas Kurz. *ALEF : An autonomous vehicle which learns basic skills and constructs maps for navigation*. Robotics and Autonomous Systems, vol. 14, no. 2, pages 171–183, 1995.

[Ladevèze 10a] Nicolas Ladevèze. *Apport des méthodes de planification automatique dans les simulations interactives d'industrialisation et de main-*

tenance en réalité virtuelle. Thèse de doctorat, Institut National Polytechnique de Toulouse, 2010.

[Ladevèze 10b] Nicolas Ladevèze, Jean-Yves Fourquet & Bernard Puel. *Interactive path planning for haptic assistance in assembly tasks*. Computers & Graphics, vol. 34, no. 1, pages 17–25, 2010.

[Lasserre 12] Philippe Lasserre, Cyrille Duchesne & Emmanuel batista. *Technologie d'interconnexions par bumps pour semi-conducteurs de puissance*. In Electronique de Puissance du Futur, 2012.

[Latombe 91] Jean-Claude Latombe. Robot motion planning. Springer, 1991.

[LaValle 98] Steven M. LaValle. *Rapidly-Exploring Random Trees : A New Tool for Path Planning*. Rapport technique, Iowa State University, 1998.

[Lee 11] Jonghwan Lee, Soonhung Han & Jeongsam Yang. *Construction of a computer-simulated mixed reality environment for virtual factory layout planning*. Computers in Industry, vol. 62, no. 1, pages 86–98, 2011.

[Leiber 80] Heinz Leiber, Armin Czinczel & Jürgen Anlauf. *Antiblockiersystem (ABS) für Personenkraftwagen*. BOSCH TECH BER, no. 2, 1980.

[Li 03] Ming Li & Allison M Okamura. *Recognition of operator motions for real-time assistance using virtual fixtures*. In Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS., pages 125–131. IEEE, 2003.

[Lien 03] Jyh-Ming Lien, Shawna L Thomas & Nancy M Amato. *A general framework for sampling on the medial axis of the free space*. In International Conference on Robotics and Automation, volume 3, pages 4439–4444. IEEE, 2003.

[Loock 01] Achim Loock, Elmar Schömer & Im Stadtwald. *A virtual environment for interactive assembly simulation : From rigid bodies to deformable cables*. World Multiconference on Systemics, Cybernetics and Informatics, vol. 3, pages 325–332, 2001.

[Lozano-Perez 80] Tomás Lozano-Perez. *Spatial planning : A configuration space approach*. Transactions on Computers, vol. 100, no. 2, pages 108–120, 1980.

[Marayong 02] Panadda Marayong, Alessandro Bettini & Allison Okamura. *Effect of virtual fixture compliance on human-machine cooperative manipulation*. In Intelligent Robots and Systems, volume 2, pages 1089–1095. IEEE/RSJ, 2002.

[Marc 07] Jaques Marc, Nadira Belkacem & Jaques Marsot. *Virtual reality : A design tool for enhanced consideration of usability "validation elements"*. Safety science, vol. 45, no. 5, pages 589–601, 2007.

[Meagher 82] Donald Meagher. *Geometric modeling using octree encoding*. Computer graphics and image processing, vol. 19, no. 2, pages 129–147, 1982.

[Mercier 10] Stéphane Mercier, Catherine Tessier & Frédéric Dehais. *Authority management in human-robot systems*. In Analysis, Design, and Evaluation of Human-Machine Systems, volume 11, pages 410–415, 2010.

[Meyrueis 11] Vincent Meyrueis. *Modification interactive de formes en réalité virtuelle : application à la conception d'un produit*. Thèse de doctorat, École Nationale Supérieure des Mines de Paris, 2011.

[Mohellebi 09] Hakim Mohellebi, Abderrahmane Kheddar & Stéphane Espié. *Adaptive haptic feedback steering wheel for driving simulators*. Transactions on Vehicular Technology., vol. 58, no. 4, pages 1654–1666, 2009.

[Moreau 04] Guillaume Moreau, Philippe Fuchs & Panagiotis Stergiopoulos. *Applications of Virtual Reality in the manufacturing industry : from design review to ergonomic studies*. Mécanique et industries, vol. 5, no. 02, pages 171–179, 2004.

[Mulder 07] Mark Mulder. *Haptic gas pedal feedback for active car-following support*. Thèse de doctorat, Delft University of Technology, 2007.

[Mulder 08] Mark Mulder, David A Abbink & Erwin R Boer. *The effect of haptic guidance on curve negotiation behavior of young, experienced drivers*. In International Conference on Systems, Man and Cybernetics, pages 804–809. IEEE, 2008.

[Müller 94] Aua Müller, W Achenbach, E Schindler, Th Wohland & G-W MOHN. *Das neue Fahrsicherheitssystem electronic stability program von Mercedes-Benz.* ATZ. Automobiltechnische Zeitschrift, vol. 96, no. 11, pages 656–670, 1994.

[Nambisan 03] Satish Nambisan. *Information Systems as a Reference Discipline for New Product Development.* Mis Quarterly, vol. 27, no. 1, 2003.

[Nehmzow 00] Ulrich Nehmzow, David Gelder & Tom Duckett. *Automatic selection of landmarks for mobile robot navigation*. 2000.

[omp 14] OMP, *Open Multi Processing*, 2014. `http://openmp.org/wp/`.

[Overmars 92] Mark H. Overmars. A random approach to motion planning. Utrecht University, Department of Computer Science, 1992.

[Patil 10] Sachin Patil & Ron Alterovitz. *Interactive motion planning for steerable needles in 3D environments with obstacles*. In International Conference on Biomedical Robotics and Biomechatronics (BioRob), pages 893–899. IEEE RAS and EMBS, 2010.

[Picon 10] Flavien Picon. *Interaction haptique pour la conception de formes en CAO immersive*. Thèse de doctorat, Université Paris XI, 2010.

[Piegl 97] Les Piegl & Wayne Tiller. *The Nurbs Book. Monograph in Visual Communication*. Springer-Verleg, 1997.

[Pouliquen 07] Mamy Pouliquen, Alain Bernard, Jacques Marsot & Laurent Chodorge. *Virtual hands and virtual reality multimodal platform to design safer industrial systems*. Computers in Industry, vol. 58, no. 1, pages 46–56, 2007.

[Preparatat 85] Franco P Preparatat & Michael Ian Shamos. *Computational geometry : an introduction*. 1985.

[Pronobis 10] Andrzej Pronobis, O Martinez Mozos, Barbara Caputo & Patric Jensfelt. *Multi-modal semantic place classification*. The International Journal of Robotics Research, vol. 29, no. 2-3, pages 298–320, 2010.

[Pronobis 11] Andrzej Pronobis & Patric Jensfelt. *Hierarchical multi-modal place categorization*. European Conference on Mobile Robots, vol. 11, 2011.

[Reif 79] John H Reif. *Complexity of the mover's problem and generalizations extended abstract*. In 20th Annual IEEE Conference on Foundations of Computer Science, pages 421–427, 1979.

[Requicha 80] Aristides G. Requicha. *Representations for rigid solids : Theory, methods, and systems*. Computing Surveys (CSUR), vol. 12, no. 4, pages 437–464, 1980.

[Samet 84] Hanan Samet. *The quadtree and related hierarchical data structures*. Computing Surveys, vol. 16, no. 2, pages 187–260, 1984.

[Saut 06] Jean-Philippe Saut, Anis Sahbani & Véronique Perdereau. *A global approach for dexterous manipulation planning using paths in n-fingers grasp subspace*. In 9th International Conference on Control,

Automation, Robotics and Vision, ICARCV'06, pages 1–6. IEEE, 2006.

[Schoenberg 46] Isaac J. Schoenberg. *Contributions to the problem of approximation of equidistant data by analytic functions*. Quart. Appl. Math, vol. 4, no. 2, pages 45–99, 1946.

[Sheridan 78] Thomas B Sheridan & William L Verplank. *Human and computer control of undersea teleoperators*. Rapport technique, DTIC Document, 1978.

[Siemens 14a] Siemens. *KineoCam*, 2014. `http://www.plm.automation.siemens.com/en_us/products/open/kineo/industries/index.shtml`.

[Siemens 14b] Siemens. *UniGraphics*, 2014. `http://www.plm.automation.siemens.com/fr_fr/products/nx/for-simulation/motion-analysis/index.shtml`.

[Siméon 00] Thierry Siméon, J-P Laumond & Carole Nissoux. *Visibility-based probabilistic roadmaps for motion planning*. Advanced Robotics, vol. 14, no. 6, pages 477–493, 2000.

[Singh 96] Leena Singh, Harry Stephanou & John Wen. *Real-time robot motion control with circulatory fields*. In International Conference on Robotics and Automation, volume 3, pages 2737–2742. IEEE, 1996.

[Sisbot 08] Emrah Akin Sisbot. *Towards human-aware robot motions*. Thèse de doctorat, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2008.

[Smith 08] Christian Smith, Mattias Bratt & Henrik I Christensen. *Teleoperation for a ball-catching task with significant dynamics*. Neural Networks, vol. 21, no. 4, pages 604–620, 2008.

[Solomalala 07] Pierre Solomalala, José Saiz, Michel Mermet-Guyennet, Alberto Castellazzi, Mauro Ciappa, Xavier Chauffleur & Jean-Pierre Fradin. *Virtual reliability assessment of integrated power switches based on multi-domain simulation approach*. Microelectronics Reliability, vol. 47, no. 9, pages 1343–1348, 2007.

[Sowat 79] Peter Forbes Sowat. *Representing spatial experience and solving spatial problems in a simulated robot environment*. Thèse de doctorat, University of British Columbia, 1979.

[Steele 01] Micah Steele & R Brent Gillespie. *Shared control between human and machine : Using a haptic steering wheel to aid in land vehicle guidance*. In Human factors and ergonomics society annual meeting, volume 45, pages 1671–1675. SAGE Publications, 2001.

[Strandberg 04] Morten Strandberg. *Augmenting RRT-planners with local trees*. In International Conference on Robotics and Automation, volume 4, pages 3258–3262. IEEE, 2004.

[Tarault 05] Antoine Tarault, Patrick Bourdot & Jean-Marc Vézien. *Sacari : An immersive remote driving interface for autonomous vehicles*. Computational Science–ICCS 2005, pages 3–15, 2005.

[Taylor 98] Camillo J. Taylor & David Kriegman. *Vision-based motion planning and exploration algorithms for mobile robots*. Transactions on Robotics and Automation, vol. 14, no. 3, pages 417–426, 1998.

[Tching 10] Loïc Tching, Georges Dumont & Jérôme Perret. *Interactive simulation of CAD models assemblies using virtual constraint guidance. A new method to assist haptic assembly tasks*. International Journal on Interactive Design and Manufacturing, vol. 4, pages 95–102, 2010.

[Timmermans 84] Harry JP Timmermans, RECM Van der Heijden & Hans Westerveld. *Decision-making between multi-attribute choice alternatives : a model of spatial shopping behaviour using conjoint measurements*. Environment and Planning A, vol. 16, no. 3, pages 377–87, 1984.

[Torralba 03] Antonio Torralba, Kevin P. Murphy, William T. Freeman & Mark A. Rubin. *Context-based vision system for place and object recognition*. In International Conference on Computer Vision, pages 273–280. IEEE, 2003.

[Tsoi 10] Kakin K Tsoi, Mark Mulder & David A Abbink. *Balancing safety and support : Changing lanes with a haptic lane-keeping support system*. In International Conference on Systems Man and Cybernetics, pages 1236–1243. IEEE, 2010.

[Ulrich 00] Iwan Ulrich & Illah Nourbakhsh. *Appearance-based place recognition for topological localization*. In International Conference on Robotics and Automation, volume 2, pages 1023–1029. Ieee, 2000.

[uml 14] UML*, Unified Modeling Language*, 2014. `http://www.uml.org`.

[Vázquez 07] Carlos Vázquez & Jan Rosell. *Haptic guidance based on harmonic functions for the execution of teleoperated assembly tasks*. In IFAC Workshop on Intelligent Assembly and Disassembly. Citeseer, 2007.

[Viswanathan 11] Pooja Viswanathan, Tristram Southey, James Little & Alan Mackworth. *Place classification using visual object categorization and global information*. In Canadian Conference on Computer and Robot Vision, pages 1–7. IEEE, 2011.

[Vogel 04] Julia Vogel & Bernt Schiele. *A semantic typicality measure for natural scene categorization*. In Pattern Recognition, pages 195–203. Springer, 2004.

[Voronoï 08] Georges Voronoï. *Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs.* Journal für die reine und angewandte Mathematik, vol. 134, pages 198–287, 1908.

[VRPN 14] VRPN. *Virtual Reality Peripheral Network*, 2014. `http://www.cs.unc.edu/Research/vrpn/`.

[Weber 09] Carolina Weber, Verena Nitsch, Ulrich Unterhinninghofen, Berthold Farber & Martin Buss. *Position and force augmentation in a telepresence system and their effects on perceived realism*. In EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics, pages 226–231. IEEE, 2009.

[Wiener 80] Earl L. Wiener & E. Curry Renwick. *Flight-deck automation : Promises and problems*. Ergonomics, vol. 23, no. 10, pages 995–1011, 1980.

[Yamauchi 96] Brian Yamauchi & Randall Beer. *Spatial learning for navigation in dynamic environments*. Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics, vol. 26, no. 3, pages 496–505, 1996.

[Yan 13] Yu Yan, Emilie Poirson & Fouad Bennis. *Interactive and On-Line Learning System for Assembly Task Motion Planning*. In ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pages V004T05A002–V004T05A002. American Society of Mechanical Engineers, 2013.

[You 12] Erkang You & Kris Hauser. *Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input*. Robotics : Science and Systems VII, page 354, 2012.

[Yu 05] Wentao Yu, Redwan Alqasemi, Rajiv Dubey & Norali Pernalete. *Telemanipulation assistance based on motion intention recognition*. In

International Conference on Robotics and Automation, pages 1121–1126. IEEE, 2005.

[Zou 03] Xi-Yong Zou & Jing Zhu. *Virtual local target method for avoiding local minimum in potential field based robot navigation*. Journal of Zhejiang University SCIENCE A, vol. 4, no. 3, pages 264–269, 2003.

Résumé

Ce travail de recherche traite les processus de planification de trajectoire pour les tâches de manipulation comme le montage, la maintenance ou le démontage dans un contexte de Réalité Virtuelle (RV). L'approche consiste à fournir un système collaboratif associant un utilisateur immergé en RV et un processus de planification automatique de trajectoire. Elle est basée sur une modélisation sémantique, topologique et géométrique de l'environnement et sur un processus de planification en deux phases : planifications grossière et fine. Le planificateur automatique suggère une trajectoire à l'utilisateur et le guide à travers une interface haptique. L'utilisateur peut quitter la solution proposée s'il veut explorer un chemin potentiellement meilleur. Dans ce cas, le système interactif prédit les intentions de l'utilisateur et calcule en temps réel un nouveau chemin à partir des souhaits de l'utilisateur.

Abstract

This research work considers path-planning processes for manipulation tasks such as assembly, maintenance or disassembly in a virtual reality (VR) context. The approach consists in providing a collaborative system associating a user immersed in VR and an automatic path planning process. It is based on semantic, topological and geometric models of the environment and planning process split in two phases : coarse and fine planning. The automatic planner suggests a path to the user and guides him trough a haptic device. The user can escape from the proposed solution if he wants to explore a possible better way. In this case, the interactive system predicts the user's intention and computes in real-time a new path starting from the user's guess.