TU 1487

THESE

présentée à

l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

pour obtenir le grade de

DOCTEUR spécialité INFORMATIQUE

Arrêté ministériel du 23 novembre 1988

—

PLANIFICATION DE MOUVEMENT POUR MOBILE NON-HOLONOME EN ESPACE DE TRAVAIL DYNAMIQUE

—

Thierry FRAICHARD

—

Thèse soutenue le 22 avril 1992

Composition du jury :

Président :	M. Jean-Pierre Verjus
Rapporteurs :	M. Jean-Claude Latombe
	M. Jean-Paul Laumond
Examinateurs :	M. Jean-Daniel Boissonnat
	M. Yves Demazeau
	M. Christian Laugier
	M. Dominique Meizel
	Mme Marie de Saint Blancard

Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle
46, av. Félix Viallet. 38031 Grenoble Cedex

Remerciements

Je remercie, tout d'abord, les membres du jury qui ont accepté de juger ce travail et d'y apporter leur caution :

- M. Jean-Pierre Verjus, Professeur à l'INPG.
- M. Jean-Claude Latombe, Professeur à l'Université de Stanford, Californie.
- M. Jean-Paul Laumond, Chargé de Recherche CNRS.
- M. Jean-Daniel Boissonnat, Directeur de Recherche INRIA.
- M. Dominique Meizel, Professeur à l'Université de Technologie de Compiègne.
- Mme Marie de Saint Blancard, Ingénieur de Recherche à la Direction des Recherches et Affaires Scientifiques de Peugeot SA.

Parmi les membres du jury, je tiens à remercier tout spécialement les rapporteurs, MM. Jean-Claude Latombe et Jean-Paul Laumond, dont les commentaires ont permis l'amélioration substantielle du manuscrit original.

Je remercie, ensuite, M. Yves Demazeau, Chargé de Recherche CNRS, et tout particulièrement, M. Christian Laugier, Directeur de Recherche INRIA, pour leur encadrement scientifique.

Je remercie, enfin, tous ceux qui, à titres divers, ont contribué à ce travail et dont le soutien anonyme m'a permis de le mener à bien.

Th. Fraichard
Grenoble, septembre 1992

Table des matières

Introduction 1

1 La planification de mouvement 5
- 1.1 Présentation générale . . . 5
- 1.2 La planification de chemin . . . 6
  - 1.2.1 Le paradigme du déménageur de piano . . . 6
  - 1.2.2 L'espace des configurations . . . 7
  - 1.2.3 La complexité de la planification de chemin . . . 8
  - 1.2.4 Les grandes approches . . . 9
    - 1.2.4.1 Les méthodes de décomposition cellulaire . . . 10
    - 1.2.4.2 Les méthodes de type 'squelette' . . . 11
    - 1.2.4.3 Les méthodes de type 'potentiel' . . . 12
- 1.3 Les contraintes cinématiques . . . 13
  - 1.3.1 Les types de contraintes cinématiques . . . 14
    - 1.3.1.1 Les contraintes holonomes . . . 14
    - 1.3.1.2 Les contraintes non-holonomes . . . 15
  - 1.3.2 Caractérisation de l'holonomie . . . 17
  - 1.3.3 Non-holonomie et problème de décision . . . 18
  - 1.3.4 Non-holonomie et problème complet . . . 19
    - 1.3.4.1 Chemins avec manœuvre . . . 19
    - 1.3.4.2 Chemins sans manœuvre . . . 21
- 1.4 La planification de trajectoire . . . 22
  - 1.4.1 La notion de trajectoire . . . 22
  - 1.4.2 La complexité de la planification de trajectoire . . . 22
  - 1.4.3 Les grandes approches . . . 23
    - 1.4.3.1 Les obstacles mobiles . . . 23
    - 1.4.3.2 Les contraintes dynamiques . . . 24

2 Planification de chemin et non-holonomie 27
- 2.1 Introduction . . . 27
- 2.2 Formulation du problème . . . 28
  - 2.2.1 Le mobile . . . 28
    - 2.2.1.1 Modélisation . . . 28

2.2.1.2 Non-holonomie de $\mathcal{A}$ 29
2.2.1.3 Contrôlabilité de $\mathcal{A}$ 30
2.2.1.4 Chemins exécutables par $\mathcal{A}$ 30
2.2.2 L'espace de travail W 31
2.2.3 Le squelette S 31
2.2.4 La planification d'un chemin sans manœuvre pour $\mathcal{A}$ . . . 32
2.3 L'algorithme de planification 33
2.3.1 Le principe général 33
2.3.2 Le traitement d'un virage 34
2.3.2.1 La caractérisation d'un virage 34
2.3.2.2 L'évitement de collision dans un virage 35
2.3.2.3 L'espace des centres de giration CCS 36
2.3.3 La définition de CCS: cas général 37
2.3.3.1 La caractérisation des domaines D, D^* et $K(D^*)$ 37
2.3.3.2 Les obstacles inclus dans D 38
2.3.3.3 Les conditions d'existence de Γ 40
2.3.3.4 La définition finale de CCS 40
2.3.4 La définition de CCS: premier et dernier virage 42
2.3.4.1 Le premier virage 42
2.3.4.2 Le dernier virage 43
2.3.5 L'exploration de CCS 43
2.3.6 L'enchaînement des virages et la génération du chemin Υ . 44
2.3.6.1 L'arbre T associé à la génération de Υ 44
2.3.6.2 L'exploration de T 45
2.4 Implantation et résultats expérimentaux 46
2.4.1 Espaces de travail structurés 47
2.4.2 Espaces de travail quelconques 49
2.5 Conclusion 51

3 La planification de trajectoire **53**
3.1 Introduction 53
3.2 Formulation du problème 54
3.2.1 La caractérisation des chemins adjacents 54
3.2.2 La structure de W 54
3.2.3 Les obstacles mobiles 55
3.2.4 La caractérisation de $\mathcal{A}$ et de ses mouvements 56
3.2.4.1 Le déplacement latéral et les chemins fictifs . . . 56
3.2.4.2 Le déplacement longitudinal 57
3.2.5 La planification d'une trajectoire pour $\mathcal{A}$ 57
3.3 L'algorithme de planification 59
3.3.1 Le principe général 59
3.3.2 Le graphe états-temps 60
3.3.3 L'exploration du graphe états-temps 61

3.3.3.1 L'algorithme 61
3.3.3.2 La fonction de coût 63
3.3.3.3 Le développement d'un nœud 64
3.4 Implantation et résultats expérimentaux 65
3.4.1 Le test de sûreté 65
3.4.2 Le changement de chemin 65
3.4.3 Résultats expérimentaux 67
3.5 Conclusion 68

4 Un système de planification de mouvement **71**
4.1 Présentation du projet PROMETHEUS 71
4.2 Le planificateur de mouvement 72
4.2.1 Architecture générale 72
4.2.2 Les entrées du système 73
4.2.3 La planification d'itinéraire 74
4.2.4 La planification de chemins adjacents 74
4.3 Planification et contrôle 76

Conclusion **77**

Bibliographie **79**

Table des figures **86**

Introduction

La robotique autonome

A la lecture de la littérature récente consacrée à la robotique, il apparaît que l'objectif principal de cette discipline est de créer des *robots autonomes*[1]. Cet objectif posé, il reste à préciser les termes robot et autonome.

Par robot, on entend tout système mécanique équipé de capteurs, d'actionneurs et d'un système de contrôle de type ordinateur. La capacité de décision du robot réside dans son système de contrôle. Les capteurs lui permettent de percevoir son espace de travail tandis que les actionneurs lui permettent d'agir sur cet espace de travail.

Un tel robot sera dit autonome si, moyennant une spécification externe 'de haut-niveau' de la tâche que l'on veut lui faire réaliser, il est capable de la mener à bien sans autre intervention. Autrement dit, on veut pouvoir indiquer au robot ce qu'il doit faire puis lui laisser le soin de déterminer comment le faire.

Selon les cas, il sera plus ou moins facile à un robot donné de prétendre au titre de robot autonome: l'autonomie d'un bras manipulateur opérant au sein d'une cellule robotique en contexte industriel (milieu restreint, connu et quasi-statique) est un objectif plus facile à atteindre que l'autonomie d'un véhicule d'exploration planétaire (milieu virtuellement illimité, mal connu, dynamique et incertain).

Le développement d'un robot autonome va poser un certain nombre de problèmes importants dans nombre de domaines très différents (perception, raisonnement automatique, contrôle, etc.). Le lecteur pourra, par exemple, se référer à [9] pour une présentation générale de ces différents aspects. Cependant, il est un problème qui se posera toujours: celui de la *planification de mouvement.* Ce problème est fondamental en robotique. En effet, quel que soit le robot et l'espace de travail considéré, le robot doit agir sur cet espace de travail et il ne peut le faire qu'au travers de mouvements. Pour lui, action est synonyme de mouvement. Un robot incapable de planifier ses propres mouvements ne saurait être autonome. Dans ce mémoire. nous allons aborder ce problème de la planification de mouvement.

[1] 'One of the ultimate goals in Robotics is to create *autonomous robots.*' [40, p.ix].

Le problème abordé

La problématique générale abordée dans ce mémoire est celle de la planification de mouvement dans un espace de travail dynamique, i.e. un espace de travail qui comporte des obstacles fixes ainsi que des obstacles mobiles. Cette problématique classique est enrichie des deux facteurs suivants :

- les contraintes cinématiques et dynamiques du mobile dont on veut planifier les mouvements sont prises en compte. En particulier, le type de mobile que l'on considère est soumis à une contrainte cinématique particulière dite *non-holonome* (une roue est un exemple de mobile soumis à une contrainte non-holonome, elle ne peut se déplacer que dans une direction perpendiculaire à son axe de rotation). Une telle contrainte, bien que couramment répandue dans les robots mobiles terrestres, n'a été abordée que récemment en robotique (1986).

- L'espace de travail est supposé connu, aux incertitudes près, dans ses aspects statiques. Il n'en est pas de même dans ses aspects dynamiques que l'on suppose partiellement connus. Plus précisément, le comportement des obstacles mobiles n'est connu que sur un intervalle de temps donné, un *horizon temporel*. La conséquence majeure de cet horizon temporel est de borner supérieurement le temps dont on dispose pour planifier un mouvement (aspect temps réel dans son acception habituelle de temps contraint).

Le problème particulier que nous allons traiter est celui de la planification des mouvements d'un mobile $\mathcal{A}$ de type voiture qui se déplace dans un espace de travail $\mathcal{W}$ plan. $\mathcal{A}$ est soumis à une contrainte cinématique non-holonome qui l'oblige à se déplacer dans une direction perpendiculaire à l'axe de ses roues. De plus, le rayon de braquage, l'accélération et la vitesse de $\mathcal{A}$ sont limitées. Une application représentative de ce travail de recherche est celle de la planification des mouvements d'une voiture se déplaçant dans le réseau routier (Projet PROMETHEUS[2]). Dans ce cas, l' espace de travail $\mathcal{W}$, i.e. la chaussée, est assimilé à un plan. Il est encombré d'obstacles fixes (limites de la chaussée) et mobiles (véhicules, piétons, etc.). Le type de mouvement que l'on désire planifier correspond aux situations de conduite normale, i.e. sans manœuvre (sans inversion de la direction de déplacement), et à des vitesses relativement élevées. Pour planifier les mouvements de $\mathcal{A}$, on a besoin de connaître le comportement des obstacles mobiles. Ces comportements seront estimés au début du processus de planification. L'horizon temporel correspond alors à la durée de validité des hypothèses que l'on fait sur le comportement des obstacles mobiles.

[2]projet EUREKA 'PROgraMme for a European Traffic with Highest Efficiency and Unprecedented Safety'.

Ainsi que nous l'avons souligné précédemment, la planification de mouvement est fondamentale en robotique. A ce titre, elle a fait l'objet de nombreux travaux de recherche tant sur le plan théorique qu'appliqué. De l'ensemble de ces travaux, il ressort que le problème général de la planification de mouvement est complexe. Cette complexité rend illusoire la recherche d'un algorithme universel de planification de mouvement qui soit utilisable en pratique. Au contraire, il est préférable de rechercher des algorithmes efficaces bien adaptés à des problèmes précis. Notre travail se situe dans cette perspective.

Notre contribution

Nous rappelons que nous abordons un problème de planification de mouvement pour un mobile $\mathcal{A}$ soumis à des contraintes cinématiques et dynamiques et se déplaçant dans un espace de travail $\mathcal{W}$ dynamique.

La thèse défendue dans ce mémoire est la suivante: ce problème peut être résolu de façon efficace[3] lorsqu'il existe une *structuration de $\mathcal{W}$ naturelle pour $\mathcal{A}$*, i.e. lorsqu'il est possible de structurer $\mathcal{W}$ en un ensemble de *zones libres* à l'intérieur desquelles $\mathcal{A}$ peut se déplacer sans collision avec les obstacles fixes de $\mathcal{W}$. Dans ce cas, le problème considéré peut être abordé suivant deux directions complémentaires:

- La *planification de chemin* qui prend en compte les obstacles fixes de $\mathcal{W}$ et les contraintes cinématiques de $\mathcal{A}$.
- La *planification de trajectoire* qui prend en compte les obstacles mobiles de $\mathcal{W}$ et les contraintes dynamiques de $\mathcal{A}$.

L'hypothèse sur la structuration de $\mathcal{W}$ ne s'applique naturellement pas à tout type d'espace de travail. Cependant, elle est valide dans un certain nombre de cas (intérieur de bâtiments, espace maritime, etc.) et notamment dans le cas du réseau routier. Les zones libres sont alors définies par les voies de circulation. Ces voies peuvent se croiser mais aussi être disposées de manière adjacente les unes par rapport aux autres. La structuration automatique de $\mathcal{W}$ constitue à elle seule un problème complexe qui sort du domaine de recherche abordé dans ce mémoire mais qui devra être abordé par la suite. Nous allons supposer que cette structure est connue a priori.

Nous allons présenter un système de planification de mouvement qui permet de résoudre efficacement le problème considéré en intégrant des techniques de résolution propres à chacune des deux planifications mentionnées ci-dessus.

Au préalable, nous allons proposer une technique de résolution pour chacun des deux sous-problèmes de planification précédents. Ces techniques sont inspi-

[3] En terme de temps de calcul.

rée d'approches existantes que nous avons adaptées de façon à tirer parti des caractéristiques propres à notre problème, ceci dans un souci d'efficacité :

- Planification de chemin : on se limite ici à l'aspect géométrique du mouvement. Etant donné une séquence de voies, on détermine un chemin sans collision avec les obstacles fixes de $\mathcal{W}$ et exécutable par $\mathcal{A}$ compte tenu de ses contraintes cinématiques. La méthode proposée est basée sur le concept *d'espace des centres de giration* initialement introduit par Laumond [45].

- Planification de trajectoire : étant donné un ensemble de chemins adjacents dont l'un amène $\mathcal{A}$ à son but, et sachant que $\mathcal{A}$ peut passer d'un chemin à un autre, on détermine la trajectoire de $\mathcal{A}$ le long de ces chemins afin d'éviter toute collision avec les obstacles mobiles de $\mathcal{W}$ tout en respectant les contraintes dynamiques de $\mathcal{A}$ (vitesse et accélération limitée). La méthode proposée s'inspire des travaux portants sur la génération de trajectoire le long d'un chemin donné et plus particulièrement des travaux de Canny et *al.* [13].

Notre contribution s'apparente aux approches basées sur le principe de la *décomposition chemin-vitesse.* Ce principe, introduit par Kant et Zucker [38], consiste à planifier un mouvement en indexant temporellement un chemin calculé au préalable. Cependant, le fait que $\mathcal{A}$ puisse quitter son chemin de référence pour passer sur un autre chemin afin d'éviter des obstacles mobiles démarque résolument notre contribution de ces approches.

La structure du mémoire

Tout d'abord, le chapitre 1 présente la problématique générale de la planification de mouvement, puis passe en revue les différents travaux pertinents dans l'optique de ce mémoire. Ensuite, dans les chapitres 2 et 3, nous présentons les deux techniques de planification, respectivement de chemin et de trajectoire, dont la combinaison va permettre de résoudre de façon efficace le problème considéré. Cette combinaison est illustrée au chapitre 4 dans le cadre d'un système de planification de mouvement pour une automobile se déplaçant dans le réseau routier.

Chapitre 1

La planification de mouvement

1.1 Présentation générale

En toute généralité et de manière intuitive, le problème de la planification de mouvement pour un robot peut se formuler de la façon suivante :

> Etant donné un robot $\mathcal{A}$ quelconque placé dans un espace de travail $\mathcal{W}$ tout aussi quelconque. Etant donné p_i la position[1] initiale de $\mathcal{A}$ et p_f la position finale où l'on désire amener $\mathcal{A}$. Trouver un mouvement qui permette de faire passer $\mathcal{A}$ de p_i à p_f. Ce mouvement doit être *exécutable* par $\mathcal{A}$ et *sans collision* avec les éventuels obstacles de $\mathcal{W}$.

Il est clair que planifier un mouvement ne se limite pas à la prise en compte du seul aspect géométrique de l'évitement de collision entre le robot et les obstacles de son espace de travail. Cela peut recouvrir des aspects aussi divers que :

- La prise en compte de la dimension temporelle pour l'évitement d'obstacles mobiles.
- La coordination des mouvements de plusieurs robots dans l'hypothèse où $\mathcal{A}$ est constitué de plusieurs sous-systèmes indépendants.
- La prise en compte des caractéristiques physiques de $\mathcal{A}$ et de $\mathcal{W}$ (contraintes cinématiques et dynamiques, masse, gravité, frottement, etc.).
- L'optimisation d'un certain critère (distance, temps, énergie, etc.).
- Le raisonnement sur les incertitudes concernant l'espace de travail qui peut amener à planifier des mouvements basés sur des données perceptives.

[1] Le terme *position* doit être pris dans un sens très large, sens dépendant tout naturellement de l'espace de travail ainsi que du système robotique considérés. Dans le cas d'un robot mobile en déplacement sur un plan, la position du robot peut être spécifiée par les coordonnées dans le plan d'un point de référence du robot ainsi que son orientation.

- La prise en compte d'un espace de travail partiellement connu, voire inconnu, qui peut rendre nécessaire l'alternance entre planification et exécution dans le but d'obtenir les informations manquantes.

La diversité de ces aspects se reflète dans la diversité des approches et des hypothèses qui existent dans les recherches en planification de mouvement. Il est cependant possible de faire une distinction entre ce que nous appellerons la *planification de chemin* et la *planification de trajectoire*[2]. Historiquement, la planification de chemin correspond aux premiers travaux de recherche effectués dans le domaine de la planification de mouvement. Elle se limite à l'aspect géométrique du problème. La planification de trajectoire, quant à elle, introduit la dimension temporelle. Cette simple extension permet de traiter un spectre beaucoup plus large de problèmes de planification de mouvement, Cela permet de considérer des obstacles mobiles et les contraintes de nature dynamique auxquelles peut être soumis le robot (force, accélération, vitesse).

Dans le cadre de ce mémoire, il nous est impossible de passer en revue tous les problèmes qui sont liés à la planification de mouvement dans toute sa généralité. Aussi nous limiterons nous à l'étude des aspects pertinents compte tenu du problème particulier que l'on se pose. La section 1.2 commence par présenter la planification de chemin qui, malgré son caractère restrictif, est intéressante à plusieurs titres. Tout d'abord, elle a permis de mieux cerner le problème de la planification de mouvement, de le formaliser et surtout d'en montrer la complexité intrinsèque. Par ailleurs, plusieurs techniques développées dans ce cadre se sont révélées adaptées pour la prise en compte d'autres aspects. Ensuite, la section 1.3 aborde le problème posé par les contraintes cinématiques en général et les contraintes non-holonomes en particulier. Enfin, la section 1.4 se penche sur les problèmes que posent les obstacles mobiles et les contraintes de nature dynamique dans le cadre de la planification de trajectoire.

1.2 La planification de chemin

1.2.1 Le paradigme du déménageur de piano

Le paradigme dominant la planification de chemin est celui du *déménageur de piano*. L'idée maîtresse de ce paradigme est que le déplacement du robot considéré n'est contraint que par la géométrie des obstacles présents dans son espace de travail. On s'intéresse donc au au problème du déplacement d'une forme géométrique parmi d'autres formes géométriques. Ce problème peut se formuler de la façon suivante :

> Soit un robot $\mathcal{A}$ constitué de un ou plusieurs corps rigides placés dans un espace de travail $\mathcal{W}$ encombré d'un ensemble de corps rigides

[2] D'après les termes anglo-saxons 'path planning' et 'trajectory planning'.

$\mathcal{B}_1,\ldots,\mathcal{B}_n$ tous considérés comme des obstacles. L'espace de travail est géométriquement décrit dans un référentiel absolu ; il est supposé statique. On appelle *placement* de $\mathcal{A}$ la spécification de la position et de l'orientation de chacun des constituants de $\mathcal{A}$ par rapport au référentiel absolu.

Etant donné un placement initial p_i et un placement final p_f. Le problème consiste à trouver un chemin, i.e. une séquence continue de placements, permettant à $\mathcal{A}$ de se déplacer de p_i à p_f en évitant les collisions avec les $\mathcal{B}_i$.

On remarquera qu'il n'est posé aucune contrainte sur le type de déplacements que peut effectuer le robot. La dimension temporelle est écartée de même que toute considération d'ordre dynamique. En règle générale, il existe une infinité de solution à un problème de planification de chemin donné. Dans la majorité des cas et dans la mesure où le temps n'est pas pris en compte, on recherche la solution qui minimise la distance parcourue par $\mathcal{A}$.

Avant de donner quelques résultats sur la complexité intrinsèque du problème du déménageur de piano et de présenter les principales techniques de résolution de ce problème, nous allons introduire le concept *d'espace des configurations* qui est à la base d'une formulation mathématique précise de la planification de chemin.

1.2.2 L'espace des configurations

A notre connaissance, le concept d'espace des configurations apparaît pour la première fois en robotique en 1977 chez Udupa [86]. Cependant, il ne sera véritablement introduit qu'en 1979 par Lozano-Perez et Wesley [55, 54]. Le lecteur désireux de trouver une formalisation complète de l'espace des configurations pourra se reporter à [47] ou bien encore à [40].

Considérons le problème de la planification des mouvements de $\mathcal{A}$ dans $\mathcal{W}$, le principe de base du concept d'espace des configurations est de reformuler ce problème dans un nouvel espace, l'espace des configurations de $\mathcal{A}$, dans lequel $\mathcal{A}$ est représenté par un point. Le problème se ramène ainsi à la planification des mouvements d'un point dans cet espace des configurations.

$\mathcal{W}$ est un sous-ensemble de l'espace euclidien $\mathbf{R}^2$ ou $\mathbf{R}^3$. Un obstacle $\mathcal{B}_i$ ayant été défini comme un corps rigide constitue donc un sous-ensemble connexe et compact, i.e. fermé et borné, de $\mathcal{W}$. On se limite au cas où $\mathcal{B}_i$ est régulier, i.e. au cas où la fermeture de l'intérieur de $\mathcal{B}_i$ est égale à $\mathcal{B}_i$.

On appelle *configuration* de $\mathcal{A}$ un m-uplet de paramètres indépendants caractérisant un placement de $\mathcal{A}$ dans $\mathcal{W}$. L'*espace des configurations* de $\mathcal{A}$ est l'espace noté $\mathcal{C}$ de toutes les configurations possibles de $\mathcal{A}$. Soit q une configuration de $\mathcal{A}$, on note $\mathcal{A}(q)$ le sous-ensemble de $\mathcal{W}$ occupé par $\mathcal{A}$ dans la configuration q. Chaque obstacle $\mathcal{B}_i$ a une image dans $\mathcal{C}$ que l'on nomme *$\mathcal{C}$-obstacle*, que l'on note

$\mathcal{CB}_i$ et que l'on définit par : $\mathcal{CB}_i = \{q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{B}_i \neq \emptyset\}$. Une configuration q de $\mathcal{A}$ est dite :

- *libre* si $\mathcal{A}(q) \cap \bigcup_{i=1}^{n} \mathcal{B}_i = \emptyset$.
- *de contact* si $\mathcal{A}(q) \cap \bigcup_{i=1}^{n} \mathcal{B}_i \neq \emptyset$ et $\text{int}(\mathcal{A}(q)) \cap \bigcup_{i=1}^{n} \text{int}(\mathcal{B}_i) = \emptyset$[3].
- *admissible* si elle est soit libre, soit de contact.
- *de collision* si elle n'est ni libre, ni de contact.

Ces différents types de configurations permettent en particulier de définir l'*espace des configurations admissibles* et l'*espace des configurations libres* que l'on note respectivement $\mathcal{AC}$ et $\mathcal{FC}$

Moyennant la définition d'une distance de $\mathcal{C}\times\mathcal{C}$ vers R, on peut doter $\mathcal{C}$ de la topologie métrique induite par cette distance. Un *chemin* entre deux configurations q_i et q_f est alors défini comme étant une fonction continue Υ de $[0,1]$ vers $\mathcal{C}$ et telle que $\Upsilon(0) = q_i$ et $\Upsilon(1) = q_f$. Un chemin est dit *admissible*, i.e. sans collision (resp. *libre*, i.e. sans contact) si : $\forall i \in [0,1]$, $\Upsilon(i) \in \mathcal{AC}$ (resp. $\mathcal{FC}$).

Dans ce cadre, la planification de chemin peut se décomposer en deux sous-problèmes :

> Etant donné un robot $\mathcal{A}$, son espace de travail $\mathcal{W}$, une configuration initiale q_i et une configuration finale q_f.
>
> 1. *Problème de décision.*
>
>   Existe-t-il un chemin admissible (resp. libre)[4] entre q_i et q_f ?
> 2. *Problème complet.*
>
>   S'il existe au moins un tel chemin alors le trouver.

Le problème de décision revient à déterminer l'existence d'une composante connexe dans $\mathcal{AC}$ (resp. $\mathcal{FC}$) qui contiennent q_i et q_f. Il reste alors à déterminer un chemin contenu dans cette composante afin de résoudre le problème complet.

1.2.3 La complexité de la planification de chemin

Dans cette section, nous étudions la complexité algorithmique de la planification de chemin. Nous nous contentons de citer quelques résultats importants et significatifs et nous renvoyons le lecteur à [84] ou à [40] pour une présentation plus complète des nombreux résultats théoriques qui ont été obtenu à ce jour.

Un premier résultat fondamental montre que le problème du déménageur de piano est décidable pour un système robotique quelconque et qu'il peut être résolu en temps polynômial pour un système robotique donné :

[3] int(X) dénote l'intérieur de l'espace topologique X.

[4] Selon que les contacts avec les obstacles sont autorisés ou non.

> Etant donné un espace des configurations de dimension m dont l'espace des configurations libres est un ensemble semi-algébrique défini par n contraintes polynômiales de degré au plus égal à d. Un chemin libre peut être calculé en un temps polynômial en n et d ('complexité de l'environnement') et exponentiel en m ('complexité du robot') .

Ce résultat a d'abord été établi par Schwartz et Sharir [74] qui donnent un algorithme qui utilise la technique de décomposition cylindrique d'ensemble semi-algébrique de Collins et qui est doublement exponentiel en m. Depuis ces travaux, Canny [12] a proposé un algorithme plus efficace qui est simplement exponentiel en m. L'intérêt de ces algorithmes n'est cependant que théorique, ils exhibent un majorant de la complexité du problème et le fait qu'ils soient d'une complexité exponentielle en m les rend inutilisables en pratique même sur des exemples triviaux [56].

Nous allons maintenant donner quelques résultats sur la borne inférieure de la complexité du problème. Historiquement, le premier résultat est du à Reif qui, en 1979, a montré que la planification d'un chemin libre pour un robot composé d'un nombre arbitraire de corps polyédriques parmi des obstacles polyédriques est un problème PEspace-dur [70]. Depuis, des résultats similaires ont été démontrés pour des systèmes plus simples. Ainsi Hopcroft, Joseph et Whitesides ont établi que le problème est PEspace-dur pour un système articulé plan [30]. De même, Hopcroft et Wilfong montre le caractère PEspace-complet de la coordination des mouvements de plusieurs rectangles dans une boite rectangulaire [31].

Les quelques résultats exposés précédemment suggèrent la complexité intrinsèque de la planification de chemin. En règle générale, on constate que cette complexité s'accroît très rapidement avec la dimension de l'espace des configurations et la complexité de l'environnement. Face à cette constatation, de nombreux travaux de recherche se sont attaqué à des cas particuliers du problème général avec comme résultat, des algorithmes de complexité moindre. Ainsi en ce qui concerne le problème de planification de chemin dans un espace de travail bi-dimensionnel encombré d'obstacles polygonaux, il existe des algorithmes de complexité polynômiale tout à fait utilisables en pratique (cf. le récapitulatif établi dans [47]).

1.2.4 Les grandes approches

Il existe un grand nombre de techniques de résolution du problème de la planification de chemin. Certaines sont générales alors que d'autres ne s'appliquent qu'à certains cas particuliers. Cependant, en dépit de certaines différences externes, on peut les regrouper en trois grandes catégories. A l'instar de Latombe [40], nous distinguerons les méthodes dites de *décomposition cellulaire*, de type *squelette* et enfin de type *potentiel*. Ce sont les principes généraux de ces trois types de méthodes que nous allons passer en revue dans les sections suivantes.

1.2.4.1 Les méthodes de décomposition cellulaire

Toutes ces méthodes reposent sur un principe très simple. Il consiste à décomposer l'espace des configurations libres (resp. admissibles)[5] du robot en un ensemble de régions appelées *cellules* telles qu'il soit facile de trouver un chemin entre deux configurations quelconques situées à l'intérieur de la même cellule. On construit alors un graphe non orienté, le *graphe d'adjacence*, qui représente la relation d'adjacence entre les cellules. Les nœuds de ce graphe sont les cellules issues de la décomposition et deux nœuds sont reliés par une arête si les deux cellules correspondantes sont adjacentes. Trivialement, on a la propriété suivante : il existe un chemin entre deux configurations q_i et q_f si et seulement si q_i et q_f appartiennent à des cellules membres d'une même composante connexe du graphe d'adjacence. La planification d'un chemin se fait alors en quatre étapes :

1. Décomposition cellulaire de l'espace des configurations libres (resp. admissibles) considéré et construction du graphe d'adjacence.
2. Localisation de la cellule c_i (resp. c_f) contenant q_i (resp. q_f).
3. Recherche dans le graphe d'adjacence d'une séquence de cellules entre c_i et c_f (réponse au problème de décision).
4. Construction du chemin à partir de la séquence de cellules ainsi obtenue (réponse au problème complet).

Les méthodes de décomposition cellulaire existantes sont variées et se différencient d'abord par le type de cellule utilisée. Cependant, on peut les classer en deux catégories : les méthodes de décomposition dites *exactes* —où l'union fermée des cellules est égale à l'espace des configurations libres (resp. admissibles)— et les méthodes dites *approchées* —où l'union fermée des cellules est strictement incluse dans l'espace des configurations libres (resp. admissibles). Ce sont ces deux catégories que nous détaillons dans les sections suivantes.

Les méthodes exactes : le principe de telles approches est de partitionner $\mathcal{FC}$ (resp. $\mathcal{AC}$) en un ensemble de cellules dont l'union est exactement égale à $\mathcal{FC}$ (resp. $\mathcal{AC}$). Les méthodes de décomposition cellulaire exactes sont complètes, i.e. que si un chemin existe alors on a la garantie de le trouver (ceci est vrai si l'algorithme d'exploration du graphe d'adjacence est complet).

Une méthode générale de décomposition cellulaire exacte a déjà été évoquée précédemment ; il s'agit de la technique de décomposition cylindrique d'ensemble semi-algébrique en cellules de Collins due à Schwartz et Sharir [74]. Il en existe d'autres adaptées à des instances particulières du problème de la planification de chemin. Citons par exemple, les travaux de [75] ou [50] pour le cas d'une barre

[5] Dans certains cas, c'est l'espace de travail qui est décomposé en cellules [16, 17].

dans un environnement polygonal ; $\mathcal{FC}$ est décomposé en régions dites *non critiques* séparées par des courbes dites *critiques* et qui correspondent à l'ensemble des configurations pour lesquelles la structure des obstacles de configurations subit un changement qualitatif. Dans le cas d'un espace des configurations bi-dimensionnel et polygonal, on peut citer les techniques classiques de décomposition en cellules trapézoïdales ou en cellules polygonales convexes.

Une autre variante de ces approches consiste à décomposer la frontière de $\mathcal{FC}$. Cela nécessite tout d'abord une procédure qui permette de relier par un chemin dans $\mathcal{FC}$ les différentes composantes connexes de $\delta(\mathcal{FC})$. Ensuite, il faut une procédure qui permette de relier par un chemin dans $\mathcal{FC}$ la configuration initiale (resp. finale) à une configuration de $\delta(\mathcal{FC})$. Ce type d'approche a été développée dans [3] pour un polygone en environnement polygonal.

Les méthodes approchées : le principe de telles approches est de partitionner $\mathcal{FC}$ (resp. $\mathcal{AC}$) en un ensemble de cellules dont l'union est strictement incluse dans $\mathcal{FC}$ (resp. $\mathcal{AC}$). Les cellules auront une forme prédéfinie relativement simple (des rectangloïdes par exemple) qui, en général, ne permettent pas de représenter exactement $\mathcal{FC}$ (resp. $\mathcal{AC}$), d'où l'approximation. La plupart des méthodes de décomposition cellulaire approchées opèrent de façon hiérarchique ce qui permet d'adapter localement la taille des cellules jusqu'à un niveau de résolution donné. On obtient ainsi une description arborescente de $\mathcal{FC}$ (resp. $\mathcal{AC}$). Fortes de ces principes de base, les méthodes existantes vont différer d'une part par le type de cellules employées mais aussi par l'algorithme de décomposition utilisé. Comme travaux représentatifs de cette diversité, on peut citer [29, 43, 22, 11, 80] et [89].

A l'opposé des méthodes exactes, les méthodes approchées peuvent ne pas être complètes. Cependant, la plupart d'entre elles sont 'complète pour une résolution donnée' dans le sens où s'il existe un chemin, la méthode le trouvera moyennant une résolution suffisamment petite. Il est évident que ceci se fera au détriment du temps de calcul. En effet, la complexité algorithmique de ces méthodes est polynômiale en fonction de la résolution et exponentielle en la dimension de l'espace des configurations. Le principal avantage de ce type d'approches est d'être utilisable en pratique comme le démontrent certains travaux ayant donné lieu à des implantations opérationnelles, e.g. les systèmes de programmation automatique HANDEY [57], NNS [1] et SHARP [42].

1.2.4.2 Les méthodes de type 'squelette'

La méthode de type squelette consiste à réduire le problème de l'exploration de $\mathcal{FC}$ (resp. $\mathcal{AC}$) en un problème d'exploration d'un sous-espace uni-dimensionnel, le *squelette* que l'on note $\mathcal{S}$, qui conserve cependant toutes les propriétés de connexité de l'espace initial. $\mathcal{S}$ est structuré en un graphe ; les arêtes du graphe étant des portions de courbe et les nœuds leurs extrémités. La planification d'un chemin se fait alors en trois étapes :

1. Construction du squelette $\mathcal{S}$ associé à $\mathcal{FC}$ (resp. $\mathcal{AC}$) et structuration de $\mathcal{S}$ en un graphe.

2. Détermination d'un chemin entre la configuration initiale q_i (resp. la configuration finale q_f) et un point r_i (resp. r_f) de $\mathcal{S}$.

3. Recherche dans $\mathcal{S}$ d'un chemin entre r_i et r_f.

Les méthodes de type squelette existantes vont se différencier essentiellement par le type de squelette utilisé. A ce jour, on peut distinguer quatre grands type de squelette : les *graphes de visibilité*, les *rétractions*, les *freeways* et les *roadmaps*.

- le graphe de visibilité : cette méthode s'applique essentiellement aux espaces des configurations bi-dimensionnels avec obstacles polygonaux. Les nœuds de $\mathcal{S}$ sont les sommets des obstacles tandis que les arêtes sont les segments joignant deux à deux les sommets 'visibles' [61].

- Les rétractions : une rétraction, au sens topologique du terme, est une fonction continue de $\mathcal{FC}$ (resp. $\mathcal{AC}$) vers un réseau de courbes uni-dimensionnelles. La restriction de la rétraction à ce réseau est l'identité. Cette rétraction constitue $\mathcal{S}$. Un exemple classique de rétraction est le diagramme de Voronoï [64, 51].

- Les 'freeways': cette méthode introduite par Brooks [10] s'applique essentiellement aux espaces des configurations bi-dimensionnels avec obstacles polygonaux. L'espace libre est partitionné en un ensemble de cônes généralisés dont les axes constituent les 'freeways' que le robot va pouvoir suivre.

- les 'roadmaps' : introduites par Canny [12], ce sont aussi des sous-espaces uni-dimensionnels de $\mathcal{C}$. Nous renvoyons le lecteur à [12] pour la définition de ces roadpmaps. Notons cependant que nous avons déjà évoqué ces travaux précédemment ; en effet, c'est cette méthode qui permet la résolution du problème du déménageur de piano avec une complexité simplement exponentielle en la dimension de $\mathcal{C}$ (cf. §1.2.3).

1.2.4.3 Les méthodes de type 'potentiel'

Le principe des approches décrites dans les deux sections précédentes était de capturer la connexité globale de l'espace libre dans un graphe qui est ensuite exploré. Les approches présentées dans cette section relèvent d'un tout autre principe que l'on peut énoncer comme suit : Le robot est assimilé à une particule plongée dans un champ de potentiel artificiel. Typiquement, ce champ de potentiel est défini comme la combinaison d'un potentiel qui attire le robot vers son but et de potentiels qui l'écartent des différents obstacles. Les méthodes de type potentiel

sont par nature itérative. A chaque itération, la force induite par le champ de potentiel détermine le déplacement du robot.

A l'origine, ces méthodes furent développées pour l'évitement d'obstacles en ligne, le robot n'ayant pas de modèle a priori des obstacles mais les découvrant au fur et à mesure à l'aide de ses capteurs. On parle alors et à juste titre de méthodes *locales* par opposition aux méthodes précédentes dites *globales* et qui imposent la connaissance a priori de l'ensemble des obstacles. Dans le domaine des méthodes locales, Khatib fait figure de pionnier avec les travaux exposés dans [39]. Depuis, de nombreux travaux sont venus enrichir ce domaine. Citons, parmi d'autres, [83, 23] et [26].

L'avantage principal des méthodes de type potentiel réside dans leur efficacité en terme de temps de calcul qui permet de les utiliser en ligne. Elles ont cependant un défaut majeur : le champ de potentiel peut contenir des minima locaux autres que le but dans lesquels le robot restera bloqué. Pour sortir d'une situation de blocage, on a généralement recours à des heuristiques mais sans garantie de convergence vers le but.

C'est pour pallier le problème des minima locaux tout en conservant l'efficacité en terme de temps de calcul de la méthode que l'on a eu l'idée de combiner l'idée du champ de potentiel avec l'hypothèse de la connaissance a priori de l'ensemble des obstacles. Naturellement, ce genre de méthode ne peut prétendre au titre de méthode locale. Ainsi Rimon et Koditschek s'attaquent au problème de la définition de fonctions potentielles dont le seul minimum local est le but [72]. Par ailleurs, Barraquand et Latombe ont proposé dans [5] une approche combinant parcours de grille et champ de potentiel. L'espace des configurations est modélisé par une grille régulière que l'on explore de façon heuristique à l'aide d'un champ de potentiel. Lorsque l'on rencontre un minimum local, on en sort au moyen de mouvements browniens. En pratique, cette approche donne de bons résultats même dans des cas complexes.

1.3 Les contraintes cinématiques

Dans le paradigme du déménageur de piano, l'existence d'un chemin entre deux configurations q_i et q_f est caractérisée par l'existence d'une composante connexe de l'espace des configurations libres (resp. admissibles) qui contiennent q_i et q_f. Ainsi toute séquence continue de configurations libres (resp. admissibles) entre q_i et q_f constitue un chemin. Naturellement, cette caractérisation n'est correcte que si l'on considère un robot entièrement libre de ses mouvements. En règle générale cependant, un robot est soumis à ce que l'on appelle des *contraintes cinématiques*, i.e. des équations ou inéquations portant sur ses paramètres de configuration et leurs dérivées, qui peuvent limiter ses capacités de mouvement. En présence de telles contraintes, la question se pose alors de savoir si le paradigme du déménageur de piano reste valide.

L'objet de la présente section est d'étudier l'influence des contraintes cinématiques sur la planification de chemin. La section 1.3.1 présente les grands types de contraintes cinématiques ainsi que leurs principales propriétés. Dans les sections 1.3.2 et1.3.3, nous nous intéressons aux problèmes posés par les contraintes cinématiques dites *non-holonomes*. Enfin la section 1.3.4 passe en revue les principaux travaux existants dans le domaine de la planification de chemin en présence de contraintes non-holonomes.

1.3.1 Les types de contraintes cinématiques

Soit un robot $\mathcal{A}$ dont l'espace des configurations $\mathcal{C}$ est de dimension m. $\mathcal{A}$ peut être soumis à des contraintes cinématiques, i.e. des liaisons au premier ordre sur les paramètres de configuration du système. Les trois sections suivantes ont pour objet de présenter les deux grands types de contraintes dites *holonomes* et *non-holonomes* ainsi que leurs influences sur la planification de chemin.

1.3.1.1 Les contraintes holonomes

Supposons que s'appliquent au mouvement de $\mathcal{A}$ des contraintes de la forme:

$$F(q,t) = F(q_1,\ldots,q_m,t) \begin{bmatrix} = \\ < \\ \leq \end{bmatrix} 0 \qquad (1.1)$$

avec $q \in \mathcal{C}$ et où t dénote le temps. Une relation de la forme (1.1) constitue une contrainte *holonome* d'égalité ou d'inégalité selon le cas. Lorsque F est différentiable et de dérivée non nulle, une contrainte holonome d'égalité de la forme (1.1) définit un sous-espace de $\mathcal{C}$ de dimension $m-1$ qui est l'ensemble des configurations admissibles pour $\mathcal{A}$. Dans le cas d'une contrainte holonome d'inégalité, l'ensemble des configurations admissibles pour $\mathcal{A}$ est un sous-ensemble de $\mathcal{C}$.

Exemple: les contraintes engendrées par les articulations prismatiques ou rotoïdes d'un bras manipulateur sont des exemples typiques de contraintes holonomes. Considérons le cas très simple où $\mathcal{A}$ est un bras manipulateur planaire réduit à un seul corps mobile autour d'une articulation rotoïde (figure 1.1). Une configuration de $\mathcal{A}$ est définie par le triplet (x_R, y_R, θ) de $\mathbf{R}^2 \times \mathbf{S}^1$ —$\mathbf{S}^1$est le cercle unité— avec $(x_R,\ y_R)$ les coordonnées d'un point de référence R situé, par exemple, sur l'axe de rotation de l'articulation, et θ l'orientation de $\mathcal{A}$. L'articulation impose que R soit fixe par rapport au référentiel de $\mathcal{W}$. Ceci se traduit par les deux contraintes holonomes d'égalité suivantes:

$$x_R + \alpha_1 = 0 \quad \text{et} \quad y_R + \alpha_2 = 0$$

avec α_1 et α_1 deux constantes. Ces deux contraintes sont indépendantes du temps. Du fait de ces deux contraintes, une configuration ne se définit plus que par θ

ainsi le véritable espace des configurations $\mathcal{C}$ de $\mathcal{A}$ est $\mathbf{S}^1$. On peut en outre imaginer que la rotation de l'articulation soit limitée mécaniquement. Ceci se traduira alors par des contraintes holonomes d'inégalité du type $\theta_1 \leq \theta \leq \theta_2$. Ce genre de contrainte ne réduit pas la dimension de $\mathcal{C}$. Mais elle réduit l'ensemble des configurations admissibles tout comme le ferait un obstacle de $\mathcal{W}$. □

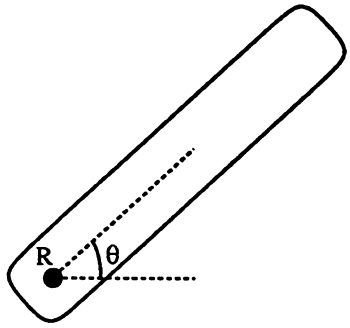


Figure 1.1 : un bras planaire mono-corps

En ce qui concerne la planification de chemin, les contraintes holonomes ne soulèvent donc aucun problème nouveau. La prise en compte de contraintes holonomes va simplement entraîner une redéfinition de $\mathcal{C}$ sans remettre en cause la caractérisation de l'existence d'un chemin.

1.3.1.2 Les contraintes non-holonomes

Supposons maintenant que s'appliquent au mouvement de $\mathcal{A}$ des contraintes de la forme :

$$G(q, \dot{q}, t) = G(q_1, \ldots, q_m, \dot{q_1}, \ldots, \dot{q_m}, t) \begin{bmatrix} = \\ < \\ \leq \end{bmatrix} 0 \qquad (1.2)$$

avec $q \in \mathcal{C}$ et où t dénote le temps. $\dot{q}$ est le vecteur vitesse associé à q, i.e. un vecteur de $\mathcal{T}_q(\mathcal{C})$, l'espace tangent à $\mathcal{C}$ en q ; $\dot{q_i} = dq_i/dt$ pour $i = 1, \ldots, m$. Lorsque G est différentiable, une relation d'égalité de la forme (1.2) qui ne peut pas se ramener par intégration à la forme (1.1) constitue une contrainte *non-holonome* d'égalité. $\mathcal{A}$ est qualifié de non-holonome s'il possède au moins une contrainte non-holonome.

Pour chaque configuration q, une contrainte non-holonome d'égalité définit un sous-espace de $\mathcal{T}_q(\mathcal{C})$ de dimension $m - 1$ qui est l'ensemble des vitesses, i.e. des déplacements instantanés, autorisées pour $\mathcal{A}$ en q. Dans le cas d'une contrainte d'inégalité, l'ensemble des vitesses admissibles pour $\mathcal{A}$ en q est un sous-ensemble de $\mathcal{T}_q(\mathcal{C})$.

Un robot $\mathcal{A}$ non-holonome n'est plus totalement libre de ses mouvements. Cette perte de liberté de mouvement se reflète dans le nombre de *degrés de liberté* de $\mathcal{A}$. Le nombre de degrés de liberté d'un système est le nombre de paramètres de configurations (après prise en compte des éventuelles contraintes holonomes d'égalité) moins le nombre de contraintes non-holonomes d'égalité. Un robot holonome a autant de degrés de liberté que de paramètres de configurations, il peut effectuer tout déplacement instantané ce qui n'est pas le cas pour un robot qui a moins de degrés de liberté que de paramètres de configurations.

Exemple: des contraintes non-holonomes apparaîssent pour des robots de type avion et pour les mains articulées (au point de contact entre l'extrémité des doigts et les objets manipulés). Généralement, la non-holonomie survient lorsqu'il y a contact avec roulement entre deux corps. Considérons le cas très simple où $\mathcal{A}$ est une roue qui se déplace sur un plan en tournant autour de son axe (figure 1.2). $\mathcal{A}$ est modélisé schématiquement par un disque rigide sans épaisseur ; nous avons donc un contact ponctuel entre $\mathcal{A}$ et $\mathcal{W}$. Une configuration de $\mathcal{A}$ est définie par un triplet (x_R, y_R, θ) de $\mathbf{R}^2 \times \mathbf{S}^1$ avec (x_R, y_R) les coordonnées d'un point de référence R situé, par exemple, au centre du disque représentant $\mathcal{A}$ et θ l'orientation de $\mathcal{A}$. Dans l'hypothèse d'un contact sans glissement entre $\mathcal{A}$ et $\mathcal{W}$, $\mathcal{A}$ se déplace dans une direction perpendiculaire à celle de son axe de rotation ce qui se traduit par les deux relations suivantes :

$$\dot{x}_R = \lambda \cos\theta \quad \text{et} \quad \dot{y}_R = \lambda \sin\theta$$

avec λ une constante. En combinant ces relations et en éliminant λ, on obtient :

$$-\dot{x}_R \sin\theta + \dot{y}_R \cos\theta = 0 \tag{1.3}$$

Nous montrerons dans la section 2.2.1.2 que cette contrainte cinématique d'égalité indépendante du temps n'est pas intégrable, c'est donc une contrainte non-holonome. Elle traduit le fait que $\mathcal{A}$ doit toujours se déplacer dans une direction tangente à son orientation θ. Cette contrainte réduit $\mathcal{T}_q(\mathcal{C})$ à un plan. $\mathcal{A}$ n'a que deux degrés de liberté □

Une séquence continue de configurations libres (resp. admissibles) ne constitue pas forcément un chemin qui soit *exécutable* par un robot non-holonome, i.e. vérifiant les contraintes cinématiques du robot. Les contraintes non-holonomes soulèvent donc un double problème en ce qui concerne la planification de chemin :

1. *Problème de décision.*

   La non-holonomie remet-elle en cause le principe selon lequel l'existence d'un chemin entre deux configurations q_i et q_f est caractérisée par l'existence d'une composante connexe de l'espace des configurations libres (resp. admissibles) qui contiennent q_i et q_f ?

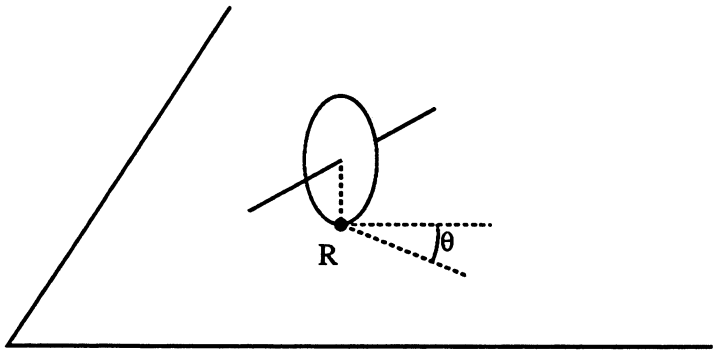


Figure 1.2 : une roue

2. *Problème complet.*

   Comment déterminer un chemin exécutable par $\mathcal{A}$?

Nous allons maintenant présenter quelques éléments de réponse à ces deux problèmes dans les sections 1.3.3 (problème de décision) et 1.3.4 (problème complet).

1.3.2 Caractérisation de l'holonomie

Lorsque l'on veut planifier un chemin pour un robot soumis à une contrainte d'égalité de la forme (1.2), il est important de déterminer si cette contrainte est véritablement non-holonome. Si ce n'est pas le cas, on peut la réécrire en une contrainte holonome de la forme (1.1) ce qui permet de poser la planification de chemin dans le cadre du paradigme du déménageur de piano. Il est donc important de caractériser formellement l'holonomie.

Cette caractérisation est essentiellement due à Barraquand et Latombe [4, 6]. Ces auteurs se sont appuyés sur des notions de géométrie différentielles et, en particulier, sur le théorème d'intégrabilité de Frobenius pour caractériser l'holonomie (et la non-holonomie) de contraintes d'égalités de la forme (1.2). Nous ne présenterons ici qu'un seul résultat que nous utiliserons dans la section 2.2.1.2 pour montrer la non-holonomie d'un mobile de type voiture. Ce résultat, extrait de [4], concerne les contraintes d'égalités linéaires en $\dot{q}$, i.e. les contraintes qui peuvent s'écrire de la façon suivante :

$$G(q, \dot{q}) = \omega(q).\dot{q} = \sum_{i=1}^{m} \omega_i(q).\dot{q}_i \tag{1.4}$$

avec ω une 1-forme différentielle de q. Dans ce cas, la caractérisation de l'holonomie est donnée par le corollaire suivant :

Corollaire 1 *Une contrainte cinématique de la forme* $G(q,\dot{q}) = \omega(q).\dot{q} = \sum_{i=1}^{m}\omega_i(q).\dot{q}_i$ *est holonome si et seulement si l'égalité :*

$$A_{ijk} = \omega_i(\frac{\partial\omega_k}{\partial q_j} - \frac{\partial\omega_j}{\partial q_k}) + \omega_j(\frac{\partial\omega_i}{\partial q_k} - \frac{\partial\omega_k}{\partial q_i}) + \omega_k(\frac{\partial\omega_j}{\partial q_i} - \frac{\partial\omega_i}{\partial q_j}) = 0 \qquad (1.5)$$

est vérifiée pour tout $i,j,k \in [1,m]$ *tels que* $1 \leq i < j < k \leq m$.

1.3.3 Non-holonomie et problème de décision

Dans le cas d'un robot holonome, l'existence d'un chemin libre entre deux configurations est caractérisée par l'existence d'une composante connexe de l'espace des configurations libres qui contienne ces deux configurations. Que devient cette propriété lorsque le robot est soumis à des contraintes de la forme (1.2) ? Cette question est directement reliée à la notion de contrôlabilité et de contrôlabilité locale issue de la théorie du contrôle : un système est *localement contrôlable* si et seulement si l'ensemble des configurations que le système peut atteindre à partir d'une configuration quelconque q tout en restant dans un voisinage donné de q est un voisinage de q. Un système qui est localement contrôlable est aussi *contrôlable*, i.e. que toute configuration peut être atteinte depuis toute autre configuration.

Pour un robot contrôlable, l'existence d'un chemin libre entre deux configurations implique donc l'existence d'un chemin libre exécutable entre ces deux configurations. Il est, par conséquent, important de pouvoir déterminer si un robot donné est contrôlable ou non. En effet, si c'est le cas alors le problème de décision est résolu.

Le premier résultat de contrôlabilité obtenu en robotique est du à Laumond [44] pour un mobile de type voiture. La démonstration repose sur le fait que tout chemin libre inclus dans un ouvert de l'espace des configurations libres $\mathcal{FC}$ peut être transformé en un chemin libre exécutable et complètement inclus dans l'ouvert en question. Ensuite, plusieurs auteurs ont montré que certains résultats de la théorie du contrôle des systèmes non-linéaires peuvent être utilisés pour démontrer la contrôlabilité de robots soumis à des contraintes de la forme (1.2). Sans entrer dans les détails, nous préciserons simplement que ces robots sont assimilés à des systèmes de contrôle non-linéaires et que leur contrôlabilité est démontré à l'aide du résultat suivant : si l'algèbre de Lie associé au champ de vecteurs engendrés par les contrôles du robot considéré a la même dimension que l'espace des configurations du robot alors celui-ci est contrôlable. Ainsi Li et Canny [53] prouvent la contrôlabilité d'une boule en déplacement dans le plan. De façon indépendante, Laumond et Siméon [48] et Barraquand et Latombe [4] prouvent la contrôlabilité d'une voiture tractant une remorque. Enfin, Barraquand et Latombe [6] généralisent les résultats obtenus précédemment à des contraintes d'égalités et d'inégalités pouvant être linéaires ou non-linéaires en la vitesse.

1.3.4 Non-holonomie et problème complet

La planification de chemin avec contraintes non-holonomes s'est constituée en un véritable domaine de recherche en robotique avec les travaux de Laumond sur un robot de type voiture en 1986 [44]. Depuis, ce domaine a fait l'objet de nombreux travaux et l'on peut supposer que cela va continuer car le planificateur général pour un système non-holonome est encore loin d'exister. Par ailleurs, on constate que la majorité des travaux porte sur les contraintes non-holonomes que l'on rencontre pour des robots de type véhicule terrestre en déplacement dans le plan[6] ; l'archétype d'un tel robot étant naturellement la voiture. Ce ne sont cependant pas les seuls exemples de robots non-holonomes comme le montrent les travaux présentés dans [53] (main articulée) ou dans [59] (robot spatial).

En ce qui concerne les contributions à la planification de chemin ; nous distinguerons les approches visant à planifier un chemin avec manœuvre et celles visant à planifier un chemin sans manœuvre. Ces dernières sont plus particulièrement intéressante pour notre problème.

1.3.4.1 Chemins avec manœuvre

Comme nous venons de le mentionner, Laumond [44] démontre qu'un robot de type voiture est contrôlable. La démonstration repose sur le fait que tout chemin libre inclus dans un ouvert de l'espace des configurations libres $\mathcal{FC}$ peut être transformé en un chemin libre exécutable et complètement inclus dans l'ouvert en question. La preuve constructive de cette propriété a conduit à la réalisation par Laumond et *al.* [49] du premier planificateur de chemin pour un robot de type voiture. Ce planificateur s'appuie sur une discrétisation approximante de $\mathcal{FC}$ en parallélépipèdes. Il recherche la séquence de parallélépipèdes permettant de passer de la configuration initiale à la configuration finale puis un chemin exécutable composé d'une séquence de manœuvres élémentaires est calculé pour traverser chaque parallélépipède. Le problème de cette approche réside dans le fait que l'on ne peut pas contrôler le nombre de manœuvres.

La méthode proposée par Barraquand et Latombe [4] consiste à discrétiser régulièrement $\mathcal{FC}$ en rectangloïdes que l'on structure en un graphe. Les arcs du graphe sont établis à partit d'une discrétisation des paramètres de contrôle de $\mathcal{A}$. L'exploration de ce graphe est basée sur une stratégie de coût visant à minimiser le nombre de manœuvres. Ce planificateur produit donc un chemin comportant un nombre minimum de manœuvre compte tenu de la résolution choisie. Du fait de la décomposition de $\mathcal{FC}$ qu'elle entraîne, cette méthode est limitée à des robots dont $\mathcal{FC}$ est de faible dimension. Elle a été testée avec succès pour un robot de type voiture et étendue au cas où ce robot tracte une remorque.

Remarquons que le cas d'un robot de type voiture tractant une remorque avait déjà été abordé par Laumond et Siméon dans [48]. Ils y montrent la contrôlabilité

[6] On commence à envisager les espaces de travail tri-dimensionnels [81, 77].

d'un tel système. Cependant, la preuve constructive de cette propriété ne conduit pas à un algorithme de planification de chemin efficace.

Pommier propose dans [68] une méthode heuristique qui comporte les trois étapes suivantes :

1. Recherche du squelette du chemin à l'aide d'un graphe de visibilité. Cette recherche ne tient compte ni des contraintes cinématiques de $\mathcal{A}$ ni de sa géométrie.

2. Pour chaque sommet du squelette, détermination d'une 'cible', i.e. d'un centre de rotation, qui permette à $\mathcal{A}$ de contourner le sommet en question. Cette étape prend en compte l'encombrement et les contraintes cinématiques de $\mathcal{A}$. Le calcul du centre de rotation utilise le concept d'*espace des centres de giration* [45].

3. Génération du chemin final en faisant évoluer les centres de rotation de $\mathcal{A}$ de cible en cible. Cette étape permet la réalisation de manœuvres simples ainsi que l'inclusion d'arcs de clothoïdes dans le chemin final.

Une autre méthode heuristique est proposée par Tournassoud dans [85]. Elle consiste à décomposer l'espace de travail en un ensemble de *cônes* très similaires aux 'freeways' de Brooks [10]. Deux cônes successifs définissent un *corridor*. La traversée d'un corridor par $\mathcal{A}$ s'effectue au moyen de chemins de contact qui ne laissent qu'un seul degré de liberté à $\mathcal{A}$.

Enfin et surtout, Jacobs et *al.* [35] présente un algorithme qui peut s'appliquer à tout robot contrôlable. Cet algorithme comporte trois étapes :

1. Recherche d'un chemin libre pour $\mathcal{A}$ sans tenir compte de la non-holonomie (utilisation d'une méthode de type déménageur de piano).

2. On divise récursivement ce chemin libre en tronçons jusqu'à ce que chaque tronçon puisse être remplacé par un chemin libre exécutable de longueur minimale.

3. On optimise le chemin libre exécutable ainsi obtenu de façon aléatoire afin de réduire sa longueur totale.

Les points clés de cet algorithme résident d'une part dans le calcul d'un chemin libre exécutable de longueur minimale et d'autre part dans une propriété qui garantisse la convergence du processus récursif de division. Le premier point est résolu dans le cas d'un robot de type voiture grâce aux travaux de Reeds et Shepp [69][7] sur la forme des plus court chemins pour un mobile de type voiture en mouvement dans le plan euclidien sans obstacles. Quant au deuxième point,

[7] Ces travaux ont été repris dans [8].

Taix montre dans [82] une propriété topologique locale qui garantit la convergence en un temps fini.

Cet algorithme a été implanté pour un robot de type voiture avec deux planificateurs holonomes différents [82, 41] et il fonctionne de manière très efficace. Notons que cet algorithme est complet si le planificateur holonome l'est.

1.3.4.2 Chemins sans manœuvre

La planification de chemin sans manœuvre, i.e. sans passage de la marche avant à la marche arrière et réciproquement, est un problème qui semble plus compliqué que le cas où les manœuvres sont permises. En effet, l'existence d'un chemin avec manœuvre pour $\mathcal{A}$ n'implique nullement l'existence d'un chemin sans manœuvre.

On trouve une réponse au problème de décision quant à la planification de chemin sans manœuvre dans [24]. Fortune et Wilfong y décrivent un algorithme qui permet de déterminer l'existence d'un chemin sans manœuvre pour un mobile ponctuel non-holonome à rayon de giration minoré. Cet algorithme est exact mais il ne permet pas de construire un chemin. De plus, il est d'une complexité exponentielle en fonction de la complexité de l'environnement.

Le problème très particulier de la planification de chemin sans manœuvre pour un robot de type voiture a été abordé dès 1957 par Dubins. Il montre qu'en l'absence d'obstacles, les plus courts chemins pour un tel robot sont constitués de segments et d'arcs de cercle de rayon minimum [20]. Trente ans plus tard, Laumond [45] reprend le même problème mais en présence d'obstacles. Il considère un robot circulaire de type voiture au milieu d'obstacles définis par des courbes fermées. Il introduit notamment le concept d'*espace des centres de giration* que nous reprendrons dans ce mémoire. L'algorithme présenté planifie des chemins sans manœuvre constitués de segments et d'arcs de cercle de rayon minimum. Malheureusement, cet algorithme n'a fait l'objet d'aucune implantation et certains points délicats de l'algorithme présentés laissent peu d'espoir d'implantation utilisable en pratique.

Jacobs et Canny [32] proposent une méthode approchée qui s'appuient sur la définition d'un ensemble de sous-chemins 'canoniques'. Le chemin final étant obtenu pas concaténation de ces sous-chemins canoniques avec des sous-chemins de contact. Les auteurs montrent que s'il existe une solution, il est possible de la trouver en ne considérant que ce type de chemins.

Enfin, Wilfong [87] présente un algorithme qui, bien qu'abordant le problème d'une manière très simplifiée, nous intéresse dans la mesure où il se place dans le cas où $\mathcal{W}$ est structuré. Etant donné un ensemble de 'voies de circulation' rectilignes qui peuvent se croiser, l'algorithme détermine tous les arcs de cercle qui permettent de passer directement d'une voie à une autre. $\mathcal{A}$ est contraint de suivre une voie et de ne la quitter que pour emprunter un arc de cercle qui lui permet de passer sur une autre voie.

On peut remarquer que toutes les méthodes que l'on vient de présenter plani-

fient des chemins sans manœuvre constitués de segments de droite reliés par des arcs de cercle de rayon minimum. On peut objecter que les discontinuité de l'accélération rendent ces chemins irréalisables en pratique. Certains auteurs ont donc étudié des types de chemins à accélération continue comme, par exemple, les clothoïdes [36], les spirales cubiques [37] ou encore certains type de polynômes [60]. Cependant aucun de ces travaux ne se place dans le contexte de la planification de chemin, ils ne considèrent que l'aspect description d'un chemin. Par contre, on trouve dans [34] les premiers éléments d'un planificateur de chemin de forme clothoïdale avec manœuvres pour un robot non-holonome; la structure globale du planificateur étant la même que celle présentée dans [35].

1.4 La planification de trajectoire

1.4.1 La notion de trajectoire

Jusqu'à présent, il n'a été question que de planification de chemin; un chemin étant une courbe continue de l'espace des configurations du robot. On ne s'est pas préoccupé de l'indexation temporelle de cette courbe, i.e. de la façon dont le robot allait se déplacer au cours du temps le long de ce chemin. Ce problème particulier est abordé dans le cadre de la planification de trajectoire. Une trajectoire est une fonction continue du temps qui spécifie à chaque instant la configuration du robot. L'introduction de la dimension temporelle permet de traiter un spectre beaucoup plus large de problèmes de planification de mouvement. En particulier, ceci permet de considérer un espace de travail encombré d'obstacles mobiles et ensuite, cela autorise la prise en compte de contraintes de nature dynamique auxquelles peut être soumis le robot (force, accélération, vitesse). Dans le cadre de la planification de trajectoire, il est, en général, plus pertinent d'optimiser la durée d'une trajectoire plutôt que sa longueur. Dans la majorité des cas, on recherche donc la solution qui est optimale en temps et non en distance comme dans la planification de chemin.

1.4.2 La complexité de la planification de trajectoire

Nous avons indiqué dans la section 1.2.3 la complexité de la planification de chemin. Dans la mesure où la planification de trajectoire prend en compte une dimension supplémentaire, le temps, et des contraintes supplémentaires, on peut s'attendre à ce que planifier une trajectoire soit un problème au moins aussi complexe que planifier un chemin. Les quelques exemples que nous citons ci-après montrent que c'est le cas.

Ainsi Reif et Sharir [71] ont montré que le problème de la planification d'une trajectoire pour un disque dans un espace de travail tri-dimensionnel encombré d'un nombre arbitraire d'obstacles en rotation est PEspace-dur si la vitesse du

disque est bornée et NP-dur sinon. Canny [12] montre que planifier une trajectoire pour un robot ponctuel dont la vitesse est bornée et qui se déplace dans un plan au milieu de polygones en translation est NP-dur. Canny et Reif [15] montrent encore que planifier une trajectoire pour un robot ponctuel dont l'accélération et la vitesse sont bornées et qui se déplace dans un espace de travail tri-dimensionnel est NP-dur.

1.4.3 Les grandes approches

Il y a de très nombreux travaux qui abordent le problème de la planification de trajectoire. Nous concentrerons notre présentation sur les deux aspects qui nous intéressent à savoir, le traitement des obstacles mobiles et la prise en compte de contraintes dynamiques.

1.4.3.1 Les obstacles mobiles

La présence d'obstacles mobiles dans l'espace de travail d'un robot est un facteur qui rend nécessaire la prise en compte de la dimension temporelle lors de la planification d'un mouvement. En effet, une configuration qui est libre à l'instant t peut très bien être occupée à l'instant t' par un obstacle mobile.

Soit un robot $\mathcal{A}$ placé dans un espace de travail $\mathcal{W}$ encombré d'un ensemble d'obstacles mobiles $\mathcal{B}_1,\ldots,\mathcal{B}_n$ et soit $\mathcal{C}$ l'espace des configurations de $\mathcal{A}$. La principale approche qui est utilisée pour planifier une trajectoire pour $\mathcal{A}$ consiste à ajouter la dimension temporelle à l'espace des configurations du robot. On obtient ainsi un nouvel espace $\mathcal{CT} = \mathcal{C}\times[0,\infty[$ que l'on appelle *espace-temps des configurations*[8]. Soit m la dimension de $\mathcal{C}$, la dimension de $\mathcal{CT}$ est égale à $m+1$. Un point de $\mathcal{CT}$ représente une configuration q du robot à un instant t donné. Chaque obstacle mobile $\mathcal{B}_i$ a une image dans $\mathcal{CT}$ que l'on nomme *$\mathcal{CT}$-obstacle*, que l'on note $\mathcal{CTB}_i$ et que l'on définit par : $\mathcal{CTB}_i = \{(q,t) \in \mathcal{CT} \mid \mathcal{A}(q) \cap \mathcal{B}_i(t) \neq \emptyset\}$.

La définition de $\mathcal{CT}$ permet de ramener le problème de la planification des mouvements de $\mathcal{A}$ en présence d'obstacles mobiles au problème de la planification des mouvements d'un point dans un espace statique. Ceci permet donc l'utilisation des différentes méthodes qui ont été présentées dans la section 1.2. Cependant, ces méthodes doivent être légèrement adaptées pour tenir compte d'une particularité de la dimension temporelle de $\mathcal{CT}$ à savoir le fait qu'il est impossible de remonter dans le temps. Parmi les travaux existants dans ce domaine, on peut distinguer les travaux qui s'appuient sur une extension du concept de graphe de visibilité [21, 71, 28] et ceux qui s'appuient sur une décomposition cellulaire de $\mathcal{CT}$ [27, 76].

Une autre approche qui est utilisée pour planifier un mouvement en présence d'obstacles mobiles est la 'décomposition chemin-vitesse'. Cette approche a été

[8] Dans la littérature anglo-saxonne, on rencontre les deux termes 'Configuration-time space' et 'Configuration space-time'.

introduite par Kant et Zucker [38]. Elle consiste à résoudre le problème en deux étapes : (a) on planifie un chemin qui évite les collisions avec les obstacles fixes de l'espace de travail et (b) on planifie la vitesse le long de ce chemin afin d'éviter les collisions avec les obstacles mobiles. Le problème posé dans l'étape (a) est un problème très classique de planification de chemin, les méthodes présentées dans la section 1.2 peuvent être utilisées pour le résoudre. Le problème posé dans l'étape (b) est essentiellement un problème de planification de mouvement en présence d'obstacles mobiles. C'en est cependant une instance très simple puisque l'espace de travail considéré est uni-dimensionnel.

La décomposition chemin-vitesse est donc plus efficace dans la mesure où elle remplace un problème compliqué par deux problèmes plus simples. Cependant, elle souffre d'un défaut majeur : elle est intrinsèquement incomplète. En effet, il peut arriver qu'il n'existe aucun profil de vitesse le long du chemin trouvé dans l'étape (a) qui garantisse l'absence de collision avec les obstacles mobiles. Dans ce cas la méthode échoue alors que l'on aurait pu trouver une solution en choisissant un autre chemin. Pour tenter de lever cette incomplétude, on peut soit itérer le processus, soit effectuer la planification de vitesse pour un ensemble de chemins [66].

1.4.3.2 Les contraintes dynamiques

La prise en compte de contraintes de nature dynamique est un problème très complexe. Il y a très peu de résultats sur la planification d'une trajectoire optimale en temps pour un robot soumis à des contraintes dynamiques. Nous en avons cités deux dans la section 1.4.2 et encore ne concernent-ils que le cas relativement simple de la planification de trajectoire optimale en temps pour un robot ponctuel dont la vitesse et l'accélération sont bornées [63, 14]. Par contre, il existe de nombreux travaux qui visent à obtenir une solution approchée à ce problème. On parle alors de planification de trajectoire sous-optimales en temps. Les principaux outils pour la recherche d'une trajectoire sous-optimale en temps sont : (a) la restriction à une certaine classe de trajectoire (trajectoire 'bang-bang' par exemple) et (b) la discrétisation soit de l'espace de travail, soit de l'espace des configurations, soit de l'espace des états[9] du robot dans le but de ramener le problème de la planification de trajectoire à un problème de recherche dans un graphe. On peut distinguer deux grandes classes d'approches :

1. Celles qui visent à déterminer une trajectoire optimale ou sous-optimale en temps le long d'un chemin donné [7, 79]. Cette classe d'approche découle du découpage traditionnel entre la planification et l'exécution de mouvement. Les travaux correspondants relèvent plus du contrôle optimal que de la planification de mouvement proprement dite. Cette classe d'approche souffre

[9] Aussi appelé *espace des phases.*

de deux défauts : (a) la trajectoire optimale en temps le long d'un chemin donné fut-il le plus court ne correspond pas forcément à la trajectoire optimale en temps et (b) le fait de ne pas considérer les contraintes dynamiques lors de la recherche du chemin peut rendre problématique le déplacement du robot le long de celui-ci.

2. Celles qui visent effectivement à déterminer une trajectoire optimale ou sous-optimale en temps. Cette classe d'approche a l'avantage de considérer non seulement des contraintes dynamiques mais aussi des contraintes statiques (obstacles). Nous pouvons citer quelques travaux représentatifs dans ce domaine : [78] définit une grille dans l'espace de travail du robot. Le meilleur chemin est obtenu par recherche hiérarchique, il est ensuite optimisé pour intégrer les contraintes dynamiques du robot. [73] propose une méthode très similaire mais qui s'appuie sur la discrétisation de l'espace des configurations. Ces deux méthodes déterminent donc une trajectoire sous-optimale en temps pour un robot de type bras manipulateur. Cependant, elles ont un défaut : leur complexité en terme de temps de calcul est exponentielle en la taille de la grille. Ce défaut est levé dans [13] qui présente une méthode de planification de trajectoire sous-optimale en temps pour un robot ponctuel dont la vitesse et l'accélération sont bornées. Cette méthode s'appuie sur une discrétisation de l'espace des états du robot. Sa complexité est polynômiale en la taille de la grille. Les travaux de [33] et [19] étendent ce travail au cas d'un bras manipulateur.

Enfin, parmi ces méthodes visant à obtenir une solution approchée, on peut remarquer celles qui garantissent que la solution trouvée est sous-optimale en temps à un facteur ϵ près, i.e. que la durée de la solution trouvée ne dépassera pas la durée de la trajectoire optimale de plus de ϵ. Cette propriété est obtenue grâce au concept de trajectoires *sûres* [13, 33] et [19]. Une trajectoire est dite sûre s'il existe une zone sans collision qui enveloppe la trajectoire. Il est alors possible de trouver dans cette enveloppe une trajectoire qui approche la trajectoire de référence d'aussi près qu'on le souhaite..

Chapitre 2

Planification de chemin et non-holonomie

Ce chapitre présente une technique de planification de chemin sans manœuvre pour un mobile de type voiture. Cette technique s'applique à un espace de travail statique et structuré en un ensemble de voies rectilignes pouvant se croiser. Etant donné une séquence de voies, on détermine un chemin sans collision avec les obstacles fixes de l'espace de travail et exécutable par le mobile compte tenu de ses contraintes cinématiques. L'algorithme présenté engendre un chemin composé de segments et d'arcs de cercle tangents. Il s'appuie sur la notion de 'virage', i.e. une transition d'une voie à une autre. Chaque virage est négocié au moyen d'un arc de cercle dont le centre est recherché dans un espace particulier dérivé de *l'espace des centres de giration* introduit par J-P. Laumond en 1987.

2.1 Introduction

En 1986, Laumond [44] aborde le problème de la prise en compte de contraintes non-holonomes dans le domaine de la planification de chemin en étudiant le cas d'un robot mobile de type voiture. Depuis, de nombreux travaux se sont penchés sur ce problème précis (cf. §1.3.4). D'autre part, si l'on excepte les travaux très spécifiques de Dubins [20] en 1957, c'est une fois de plus Laumond [45] qui, en 1987, aborde le problème particulier de la planification de chemin sans manœuvre, i.e. sans inversion de la direction de déplacement, pour un mobile de type voiture. A ce jour, on ne dispose que de très peu de résultats sur ce problème précis (cf. §1.3.4). Notons que, dans le cas avec manœuvre comme dans le cas sans manœuvre, les méthodes existantes[1] ont en commun de planifier un chemin composé de segments de droite et d'arcs de cercle. Ceci découle des travaux de Dubins [20] puis de Reeds et Shepp [69][2] sur la forme des plus court chemins pour

[1] A l'exception du planificateur de chemin qui est esquissé dans [34] et qui utilise des clotoïdes.
[2] Ces travaux ont été repris dans [8].

un mobile de type voiture en mouvement dans un espace de travail plan et sans obstacles.

Dans ce chapitre, nous présentons une méthode de planification de chemin sans manœuvre pour un mobile de type voiture. Le chemin que nous planifions est composé de segments de droite et d'arcs de cercle. Notre contribution est basée sur le concept *d'espace des centres de giration*, i.e. le domaine dans lequel rechercher le centre des différents arcs de cercle qui composent le chemin. Ce concept a été initialement introduit par Laumond [45] dans le cadre d'un algorithme de planification de chemin sans manœuvre pour un mobile de type voiture de forme circulaire. Malheureusement, cet algorithme n'a fait l'objet d'aucune implantation. En ce qui nous concerne, nous considérons un mobile de forme quelconque et l'algorithme que nous proposons a été implanté et est utilisable en pratique.

D'autre part, à l'instar de Wilfong [87], nous nous plaçons dans le cas d'un espace de travail structuré. Cette structure consiste en un ensemble de voies rectilignes et pouvant se croiser. Afin d'atteindre son but, le mobile doit suivre une certaine séquence de voies qui définit le *squelette* du chemin à planifier. Chez Wilfong, le mobile est contraint de suivre une voie et de ne la quitter que pour emprunter un arc de cercle qui lui permet de passer sur une autre voie. Nous allégeons cette contrainte en planifiant un chemin *topologiquement équivalent* au squelette. Autrement dit, il suffit que les obstacles conservent la même disposition, topologiquement parlant, vis-à-vis du squelette et du chemin.

2.2 Formulation du problème

2.2.1 Le mobile

2.2.1.1 Modélisation

Soit $\mathcal{A}$ un mobile de type voiture. Il évolue dans un espace de travail bi-dimensionnel $\mathcal{W} = \mathbf{R}^2$. $\mathcal{A}$ est modélisé schématiquement comme un corps rigide bi-dimensionnel de forme rectangulaire. On définit une configuration q de $\mathcal{A}$ par le triplet (x_R, y_R, θ) de $\mathbf{R}^2 \times \mathbf{S}^1$—$\mathbf{S}^1$ est le cercle unité— où (x_R, y_R) sont les coordonnées du point médian R de l'axe de roues arrières et θ est l'orientation de $\mathcal{A}$, i.e. l'angle entre l'axe des x du référentiel associé à $\mathcal{W}$ et l'axe principal de $\mathcal{A}$ (figure 2.1). L'espace des configurations $\mathcal{C}$ de $\mathcal{A}$ est donc de dimension trois.

$\mathcal{A}$ est muni de deux roues arrières à orientation fixe et de deux roues avants orientables. Une roue est assimilée à un disque rigide sans épaisseur ; nous avons donc un contact ponctuel entre chaque roue et le sol. Dans l'hypothèse de roulement sans glissement, une roue se déplace dans la direction perpendiculaire à celle de son axe. Le mouvement d'un solide dans le plan est soit une translation pure, soit une rotation pure autour d'un centre de rotation donné (on remarque que la translation est un cas particulier de rotation avec un centre de rotation à l'infini). Dans le cas d'une rotation, les axes des quatre roues se croisent en un

même point G qui est le centre de rotation instantané de la voiture [2]. G est naturellement situé sur la droite passant par l'axe des roues arrières. On notera que, lors d'un virage, les roues avants de $\mathcal{A}$ ont chacune une orientation propre qui satisfait cette condition [52]. On appelle *angle de braquage* de $\mathcal{A}$, l'angle, noté ϕ, entre l'axe principal de $\mathcal{A}$ et la perpendiculaire de la droite passant par G et le point médian de l'axe des roues avants F=(x_F, y_F). La distance entre les points R et G représente le rayon de courbure instantané ρ_R de la courbe tracée par R:

$$\rho_R = \frac{l_w}{\tan\phi}$$

où l_w est l'empattement de $\mathcal{A}$, i.e. la distance entre F et R. ρ_R est aussi appelé *rayon de giration* de $\mathcal{A}$.

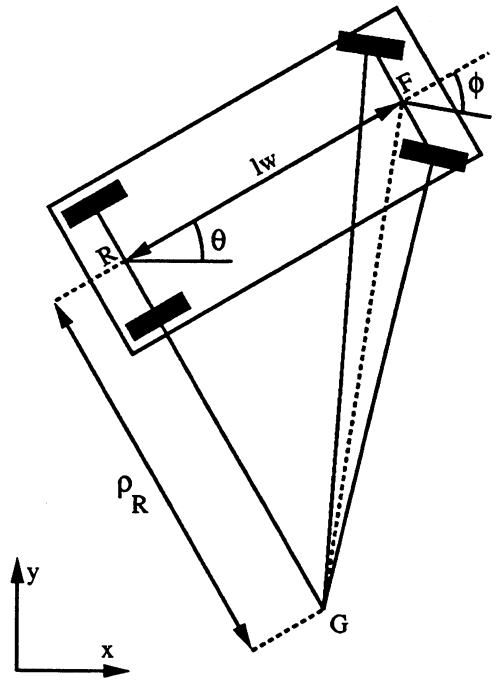


Figure 2.1 : un mobile de type voiture

2.2.1.2 Non-holonomie de $\mathcal{A}$

De par notre hypothèse de contact sans glissement entre les roues et le sol, la vitesse du point R reste parallèle à l'axe principal de $\mathcal{A}$, ce qui se traduit par les deux relations suivantes :

$$\dot{x}_R = \lambda\cos\theta \quad \text{et} \quad \dot{y}_R = \lambda\sin\theta$$

où λ est une constante. En combinant ces relations et en éliminant λ, on obtient :

$$-\dot{x}_R\sin\theta + \dot{y}_R\cos\theta = 0 \tag{2.1}$$

Cette relation est une contrainte cinématique d'égalité, elle peut se mettre sous la forme $g(q,\dot{q}) = \sum_{i=1}^{3} \omega_i(q).\dot{q}_i$ avec $\omega_1 = -\sin(\theta)$, $\omega_2 = \cos(\theta)$ et $\omega_3 = 0$. on calcule alors l'unique coefficient a_{123} qui est égal à -1 et par application du corollaire **1** que l'on a présenté dans la section 1.3.2, on en déduit que la contrainte (2.1) est non-holonome, $\mathcal{A}$ est donc un système non-holonome.

Par ailleurs, de même que l'angle de braquage d'une voiture est limité pour des raisons mécaniques, on suppose que l'angle de braquage ϕ de $\mathcal{A}$ est limité. Cela se traduit par la relation suivante :

$$|\phi| \leq \phi_{max} < \frac{\pi}{2}$$

où ϕ_{max} est l'angle de braquage maximum. Cette contrainte a pour effet de minorer ρ_R. Ainsi on doit toujours avoir :

$$\rho_R \geq l_w / \tan \phi_{max} = \rho_R^{min} \tag{2.2}$$

où ρ_R^{min} est le rayon de giration minimum de $\mathcal{A}$.

Remarque : ρ_R est aussi limité pour des raisons dynamiques. En effet, lorsque $\mathcal{A}$ est en mouvement, il est soumis à une certaine accélération centrifuge qui ne doit pas dépasser un certain seuil g_{max}. Cette contrainte dynamique a pour effet de limiter ρ_R de la façon suivante :

$$\rho_R \geq \frac{v^2}{g_{max}} = \rho_R^{min}(v)$$

où v est la vitesse instantanée de $\mathcal{A}$. □

2.2.1.3 Contrôlabilité de $\mathcal{A}$

La contrôlabilité de $\mathcal{A}$ a été démontrée tout d'abord dans [44] puis, par la suite, dans [6] à l'aide d'outils de la théorie du contrôle des systèmes non-linéaires. Ce résultat formel confirme l'intuition expérimentale qu'a tout conducteur, à savoir que, moyennant un certain nombre de manœuvres, il est toujours possible d'amener une voiture d'une configuration à une autre.

2.2.1.4 Chemins exécutables par $\mathcal{A}$

Un chemin pour $\mathcal{A}$ est une séquence continue de configurations, i.e. une courbe continue $\Upsilon_{\mathcal{C}} : [0,1] \longrightarrow \mathcal{C}$. Ce chemin se projette dans le plan xy, autrement dit $\mathcal{W}$, en une courbe $\Upsilon_{\mathcal{W}} : [0,1] \longrightarrow \mathcal{W}$ qui représente la trace laissée par R dans $\mathcal{W}$. La contrainte non-holonome (2.1) peut se réécrire sous la forme :

$$\tan\theta = \frac{\dot{y}_R}{\dot{x}_R} \tag{2.3}$$

Il en résulte que le coefficient angulaire de la tangente en tout point de $\Upsilon_{\mathcal{W}}$ est égale à θ. En d'autres termes, il y a bijection entre $\Upsilon_{\mathcal{C}}$ et $\Upsilon_{\mathcal{W}}$ et, par extension, $\Upsilon_{\mathcal{W}}$ constitue aussi un chemin pour $\mathcal{A}$. De fait, le chemin que nous allons chercher à déterminer pour $\mathcal{A}$ est une courbe $\Upsilon_{\mathcal{W}} : [0,1] \longrightarrow \mathcal{W}$. Ceci nous amène à étudier les caractéristiques de $\Upsilon_{\mathcal{W}}$.

Soit v le module de la vitesse de R, cette vitesse est positive ou négative. Lorsque $v > 0$ (resp. $v < 0$) alors $\mathcal{A}$ avance (resp. recule). Lorsque $\phi > 0$ (resp. $\phi < 0$) alors $\mathcal{A}$ tourne à gauche (resp. à droite). Du fait de la contrainte (2.2), $\Upsilon_{\mathcal{W}}$ ne comporte pas de points anguleux (un tel point correspond à une rotation sur place). Cependant, il peut comporter des points de rebroussement qui correspondent à des changements de la direction de déplacement de $\mathcal{A}$, i.e. des changements du signe de v. Dans chaque intervalle où v ne change pas de signe, la relation (2.2) contraint $\Upsilon_{\mathcal{W}}$ à être de classe C^1 —une courbe est dite de classe C^n si elle est différentiable n fois et que sa dérivée $n^{\text{ième}}$ est continue. En outre, la courbure en tout point de $\Upsilon_{\mathcal{W}}$ (quand elle est définie) doit être inférieure à l'inverse du rayon de giration minimum de $\mathcal{A}$. En résumé, $\Upsilon_{\mathcal{W}} : [0,1] \longrightarrow \mathcal{W}$ représente un chemin exécutable par $\mathcal{A}$ si et seulement si il satisfait aux deux conditions suivantes :

- $\Upsilon_{\mathcal{W}}$ est de classe C^1 par morceaux et les points de discontinuité de la direction tangente correspondent à des points de rebroussement.
- La courbure en tout point de $\Upsilon_{\mathcal{W}}$ (quand elle est définie) est inférieure à $1/\rho_R^{min}$.

Par ailleurs, $\Upsilon_{\mathcal{W}}$ est sans manœuvre si :

- $\Upsilon_{\mathcal{W}}$ est de classe C^1.

2.2.2 L'espace de travail $\mathcal{W}$

$\mathcal{A}$ évolue dans dans un espace de travail plan $\mathcal{W}$ encombré par un ensemble d'obstacles fixes $\mathcal{B}=\{o_1,\ldots,o_n\}$. Chaque obstacle est modélisé par un polygone généralisé, i.e. un polygone dont les arêtes sont soit des segments de droite, soit des portions d'arc de cercle [46].

2.2.3 Le squelette $\mathcal{S}$

Nous nous plaçons dans le cas d'un espace de travail structuré. Cette structure consiste en un ensemble de voies rectilignes et pouvant se croiser. Afin d'atteindre son but, le mobile doit suivre une certaine séquence de voies qui définit ce que l'on appelle le *squelette* du chemin. Soit $\mathcal{S}$ ce squelette, c'est une ligne brisée que l'on représente par une suite ordonnée de points $(p_1, p_2, \ldots, p_S)$ avec $p_i \in \mathcal{W}$ et $p_i = (x_i, y_i)$. On suppose que la configuration initiale de $\mathcal{A}$ est $q_i = (x_1, y_1, \theta_1)$

avec θ_1 l'orientation du segment p_1p_2. La configuration finale que $\mathcal{A}$ cherche à atteindre est $q_f = (x_S, y_S, \theta_S)$ avec θ_S l'orientation du segment $p_{S-1}p_S$. Chaque segment p_ip_{i+1} de $\mathcal{S}$ représente une portion de voie que $\mathcal{A}$ doit pouvoir suivre sans entrer en collision avec les obstacles de $\mathcal{W}$.

Ainsi que nous l'avons mentionné dans la section 2.1, les obstacles doivent conserver la même disposition, topologiquement parlant, vis-à-vis de $\mathcal{S}$ et de $\Upsilon_\mathcal{W}$. Formellement, $\mathcal{S}$ et $\Upsilon_\mathcal{W}$ sont **topologiquement équivalents** si et seulement si :

- $\Upsilon_\mathcal{W}$ et $\mathcal{S}$ ont les mêmes extrémités p_1 et p_S.
- $\Upsilon_\mathcal{W}$ et $\mathcal{S}$ ont les mêmes coefficients angulaires aux points p_1 et p_S.
- $\Upsilon_\mathcal{W}$ et $\mathcal{S}$ appartiennent à la même classe d'homotopie[3], i.e. il est possible de déformer de façon continue $\Upsilon_\mathcal{W}$ en $\mathcal{S}$.

Notons que l'équivalence topologique entre Υ et $\mathcal{S}$ n'implique nullement que les segments qui composent $\mathcal{S}$ doivent faire partie de Υ.

2.2.4 La planification d'un chemin sans manœuvre

Nous pouvons maintenant formaliser le problème de planification de chemin que l'on cherche à résoudre :

> *Etant donné le mobile $\mathcal{A}$, l'espace de travail $\mathcal{W}$ et le squelette $\mathcal{S}$ tels que nous les avons définis dans les sections précédentes, déterminer un chemin $\Upsilon : [0,1] \longrightarrow \mathcal{W}$ qui vérifie les propriétés suivantes :*
>
> - $\Upsilon(0) = p_1$ *et* $\Upsilon(1) = p_S$.
> - Υ *est de classe* C^1.
> - *La courbure en tout point de* Υ *(quand elle est définie) est inférieure à* $1/\rho_R^{min}$.
> - Υ *est sans collision avec les éléments de* $\mathcal{B}$.
> - Υ *est topologiquement équivalent à* $\mathcal{S}$.

Nous allons rechercher une solution à ce problème parmi l'ensemble des chemins de classe C^1 et composés de segments de droite et d'arcs de cercle de rayon supérieur à ρ_R^{min}. Ce type de chemin est exécutable par $\mathcal{A}$ et il est possible de montrer que l'existence d'un chemin solution quelconque implique l'existence d'un chemin de ce type (généralisation de la propriété établie dans [45] pour des chemins de classe C^1 et composés de segments de droite et d'arcs de cercle de rayon égal à ρ_R^{min}).

[3]Soit f et g, deux fonctions continues de $[0,1]$ dans $\mathcal{W}$. f et g appartiennent à la même classe d'homotopie si et seulement si il existe une fonction continue ϕ de $[0,1] \times [a,b]$ dans $\mathcal{W}$ telle que $\phi(t,a) = f(t)$ et $\phi(t,b) = g(t)$.

2.3 L'algorithme de planification

2.3.1 Le principe général

Notre algorithme de planification du chemin Υ s'appuie sur le squelette $\mathcal{S}$ et sur ses propriétés. Soit $p_{i-1}p_i$ et p_ip_{i+1} deux segments consécutifs de $\mathcal{S}$. Chacun de ces segments représente une voie que $\mathcal{A}$ est capable de suivre. Dans notre cas, i.e. celui de la planification de chemin sans manœuvre, on considère que le passage de $p_{i-1}p_i$ à p_ip_{i+1} se fait tout simplement au moyen d'un arc de cercle qui fait passer $\mathcal{A}$ de $p_{i-1}p_i$ à p_ip_{i+1}[4]. La paire de segments $p_{i-1}p_i$ et p_ip_{i+1} constitue ce que l'on appelle un *virage* et que l'on dénote par le triplet (p_{i-1}, p_i, p_{i+1}). Ainsi $\mathcal{S}$ définit (s-2) virages.

Le principe de base de notre algorithme est de rechercher, pour chaque virage de $\mathcal{S}$, un arc de cercle de rayon r qui permette de négocier ce virage sans collision avec les éléments de $\mathcal{B}$. Le centre de cet arc de cercle est recherché dans un domaine particulier de $\mathcal{W}$ appelé *espace des centres de giration*[5]. La détermination de cet arc de cercle définit le chemin associé au virage considéré. Le chemin Υ complet est obtenu en traitant séquentiellement tous les virages de $\mathcal{S}$ du premier jusqu'au dernier et en utilisant le mécanisme d'enchaînement suivant : soit (p_{i-1}, p_i, p_{i+1}) le triplet associé au i^{eme} virage de $\mathcal{S}$ et soit A_i l'arc de cercle de rayon r qui permet de négocier ce virage. L'enchaînement entre le i^{eme} et le $(i+1)^{eme}$ virage est réalisé en associant au $(i+1)^{eme}$ virage non pas le triplet (p_i, p_{i+1}, p_{i+2}) mais le triplet (p, p_{i+1}, p_{i+2}) avec p le point commun à A_i et à la tangente séparatrice de A_i et p_{i+1}. Ce mécanisme d'enchaînement est illustré dans la figure 2.2.

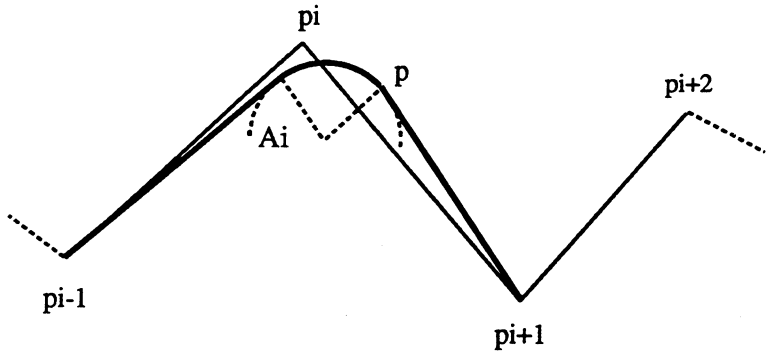


Figure 2.2 : l'enchaînement des virages

Le rayon r de l'arc de cercle qui va permettre de négocier un virage donné sera a priori égal à ρ_R^{min}, i.e. le rayon de giration minimum de $\mathcal{A}$. Cependant, si

[4]Le fait de se restreindre à un unique arc de cercle pour négocier un virage donné conduit à ignorer les solutions nécessitant l'enchaînement de plusieurs arcs de cercle.

[5]L'espace des centres de giration dérive du 'space of centers of curvature' introduit dans [46].

l'on dispose d'une estimation de la vitesse v à laquelle on devra négocier le virage considéré alors on peut choisir r égal à $\max(\rho_R^{min}, \rho_R^{min}(v))$ (cf. §2.2.1.2).

2.3.2 Le traitement d'un virage

2.3.2.1 La caractérisation d'un virage

Soit Λ le virage de $\mathcal{S}$ dont on désire planifier le chemin correspondant. Λ est caractérisé par le triplet ($\widehat{n_0n_1}$, n_1n_2, n_2n_3). $\widehat{n_0n_1}$ est l'arc de cercle de centre c_p et de rayon r_p associé au virage qui précéde Λ[6]. n_1n_2 et n_2n_3 sont les deux segments associés à Λ. Ces trois éléments sont connectés de la manière illustrée par la figure 2.3 et constituent le squelette Γ_n associé à Λ. Λ est un virage à droite si $\overrightarrow{n_1n_2} \wedge \overrightarrow{n_2n_3} < 0$ et un virage à gauche sinon.

Remarque : $\widehat{n_0n_1}$ et n_1n_2 sont des éléments du chemin associé au virage qui précède Λ dans $\mathcal{S}$. De ce fait, ils constituent un chemin sans collision avec les éléments de $\mathcal{B}$. □

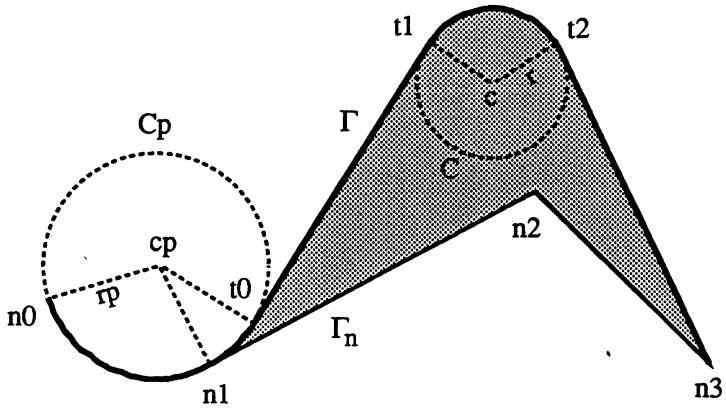


Figure 2.3 : le virage Λ

Le squelette Γ_n n'est évidemment pas exécutable tel quel par $\mathcal{A}$ du fait de la discontinuité de la direction tangente au point n_2. Nous allons donc rechercher un chemin Γ qui vérifie les différentes contraintes exposées dans la section 2.2.1.4. Γ est composé de quatre éléments ($\widehat{n_0t_0}$, t_0t_1, $\widehat{t_1t_2}$, t_2n_3), reliés de manière à former une courbe de classe C^1. $\widehat{n_0t_0}$ est un arc de cercle de centre c_p et de rayon r_p. $\widehat{t_1t_2}$ est un arc de cercle de rayon r tournant à droite ou à gauche selon que Λ est un virage à droite ou à gauche. t_0t_1 et t_2n_3 sont deux segments de droite reliant respectivement $\widehat{n_0t_0}$ à $\widehat{t_1t_2}$ et $\widehat{t_1t_2}$ à n_3. Il est important de noter que Γ est totalement défini par le centre c du cercle C qui supporte l'arc de cercle $\widehat{t_1t_2}$.

Remarque : Au niveau de Λ, la contrainte d'équivalence topologique se traduit

[6]Dans le cas général, r_p peut être différent de r, ceci afin de permettre de négocier les virages avec des rayons de giration différents (fonction de la vitesse, par exemple).

par le fait qu'il ne doit y avoir aucun obstacle dans le domaine définit par le sextuplet ($\widehat{n_0t_0}$, t_0t_1, $\widehat{t_1t_2}$, t_2n_3, n_2n_3, n_1n_2), i.e. la zone grisée de la figure 2.3. □

2.3.2.2 L'évitement de collision dans un virage

Il y a collision lorsque un élément de $\mathcal{B}$, i.e. un obstacle, coupe la région qui est balayée par $\mathcal{A}$ quand celui-ci se déplace le long d'un chemin Υ. En d'autres termes, Υ est sans collision avec $o \in \mathcal{B}$ si et seulement si la distance entre Υ et o demeure supérieure à une certaine valeur d qui dépend d'une part de la géométrie de $\mathcal{A}$ et d'autre part de la forme de Υ.

Prenons le cas du mobile $\mathcal{A}$ illustré dans la figure 2.4. Si Υ est une ligne droite alors $d = w/2$ avec w la largeur de $\mathcal{A}$. Si Υ est un cercle de rayon r alors la distance d varie selon que l'obstacle o se trouve à l'intérieur ou bien à l'extérieur du cercle. Si o se trouve à l'intérieur alors $d = w/2$ sinon d est égale à :

$$\mathcal{F}(r, l_f, w) = (\sqrt{(r + w/2)^2 + {l_f}^2} - r)$$

où l_f est la distance entre l'axe des roues arrières et la face avant de $\mathcal{A}$.

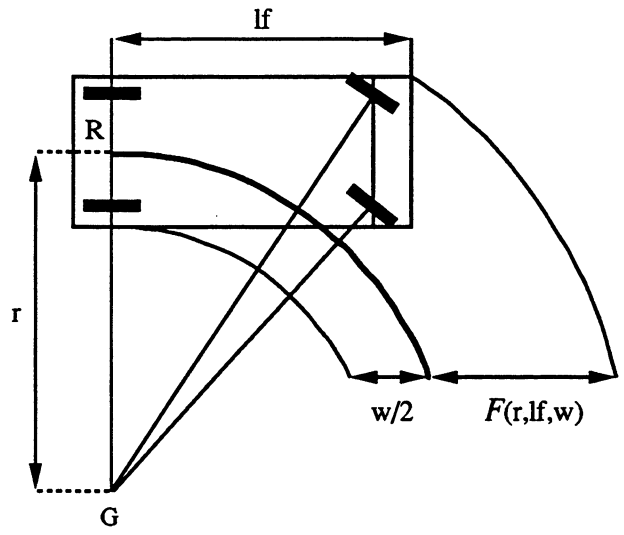


Figure 2.4 : la région balayée par $\mathcal{A}$ lors d'un virage

Considérons maintenant le cas où Υ est un chemin de classe C^1 composé d'une demi-droite, d'un arc de cercle de rayon r et d'une demi-droite. Υ partitionne $\mathcal{W}$ en deux domaines dont l'un est convexe et l'autre concave (figure 2.5). Trivialement, une condition suffisante pour que Υ soit sans collision avec les éléments de $\mathcal{B}$ est qu'il ne coupe pas les obstacles du domaine convexe grossis isotropiquement de $w/2$ et les obstacles du domaine concave grossis isotropiquement de $\mathcal{F}(r, l_f, w)$. Notons qu'une condition nécessaire pour que Υ soit sans collision est : $\forall\, o \in \mathcal{B}$, Υ ne coupe pas $\mathcal{G}(o, w/2)$ où $\mathcal{G}(o, d)$ est l'obstacle o grossi isotropiquement de

la valeur d. Cette remarque nous amène à définir l'ensemble $\mathcal{B}'$ des obstacles grossis isotropiquement de $w/2$, ensemble que nous utilisons par la suite pour la recherche d'un chemin sans collision.

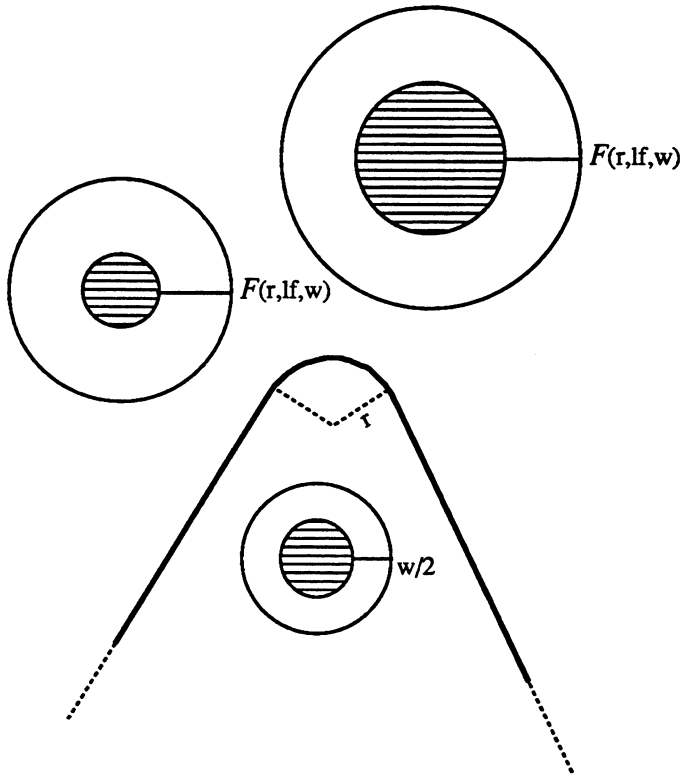


Figure 2.5 : la non-collision du chemin Υ

Remarque : Nous avons déterminé les différentes valeurs de d en supposant $\mathcal{A}$ de forme rectangulaire. Toutefois, la notion de région balayée et partant l'algorithme peut s'appliquer à un mobile de forme quelconque. □

2.3.2.3 L'espace des centres de giration CCS

Ainsi que nous l'avons mentionné dans le paragraphe 2.3.2.1, le chemin Γ que l'on cherche à planifier pour le virage Λ est complètement défini par le centre c du cercle C qui supporte l'arc de cercle $\widehat{t_1 t_2}$. Compte tenu de cette propriété, notre principe de planification du chemin Γ pour le virage Λ est de choisir c dans $\mathcal{W}$ tout en garantissant que le chemin Γ qui résulte de ce choix est sans collision et respecte l'équivalence topologique.

Ce principe de résolution nous amène à définir *l'espace des centre de giration* associé au virage Λ et noté CCS. CCS est un domaine de $\mathcal{W}$ défini de sorte que si $c \in CCS$ alors le chemin Γ qui en résulte vérifie les contraintes de non-collision et d'équivalence topologique.

Dans les deux sections suivantes, nous définissons l'espace des centre de giration CCS associé à un virage. Tout d'abord, nous définissons CCS dans le cas général (§2.3.3). Le cas général est celui d'un virage tel qu'il a été défini au paragraphe 2.3.2.1. Ceci correspond à un virage qui n'est ni le premier, ni le dernier virage de $\mathcal{S}$. Ensuite, nous définissons CCS dans le cas du premier ou du dernier virage de $\mathcal{S}$ (§2.3.4). Ce cas diffère du cas général dans la mesure où des contraintes additionnelles apparaîssent sur l'orientation de $\mathcal{A}$ au début ou à la fin du virage.

2.3.3 La définition de CCS : cas général

La définition de l'espace des centres de giration CCS associé au virage Λ se fait en trois étapes. Tout d'abord, on définit le domaine D support de l'ensemble des chemins Γ solutions. De façon duale, on définit $K(D^*)$, le domaine support de l'ensemble des centres c solutions, i.e. le domaine support de CCS. Ensuite, on traite les obstacles qui sont inclus dans D. Enfin, on prend en compte certaines conditions d'existence de Γ.

2.3.3.1 La caractérisation des domaines D, D^* et $K(D^*)$

Nous rappelons que Γ_n est le squelette associé au virage Λ, i.e. le triplet ($\widehat{n_0n_1}$, n_1n_2, n_2n_3) et que l'on recherche un chemin sans collision Γ de la forme ($\widehat{n_0t_0}$, t_0t_1, $\widehat{t_1t_2}$, t_2n_3) (figure 2.3).

Soit Γ_s le plus court[7] chemin de type Γ qui soit topologiquement équivalent à Γ_n et qui ne coupe pas les éléments de $\mathcal{B}'$. Γ_s se compose d'un arc de cercle de centre c_p et de rayon r_p, d'un segment S_1, d'un arc de cercle de rayon r et d'un segment S_2 (figure 2.6a).

Soit γ_s le sous-chemin de Γ_s composé de S_1, A et S_2 et soit γ le sous-chemin correspondant de Γ : $\Gamma=\widehat{n_0t_0}\cup\gamma$ avec $\gamma=(t_0t_1, \widehat{t_1t_2}, t_2n_3)$. Du fait de l'équivalence topologique, γ doit appartenir à la même classe d'homotopie que γ_s. Laumond a montré dans [45] qu'un chemin appartenant à la même classe d'homotopie que (S_1, A, S_2) est nécessairement inclus dans le domaine concave D définit par A et les deux demi-droites support de S_1 et S_2. En conséquence, le sous-chemin γ du chemin Γ doit être inclus dans le domaine D. Formellement, D est défini de la façon suivante :

$$D = \bigcup_{\lambda\in[\lambda_1,\lambda_2]} P_\lambda$$

où P_λ est le demi-plan définit par la tangente à A dont le coefficient angulaire est λ et qui ne contient pas A. λ_1 (resp. λ_2) est le coefficient angulaire de S_1

[7]Plus court selon la distance euclidienne.

(resp. S_2) .

Remarque: Par construction, γ_s évite les obstacles de $\mathcal{B}'$. γ_s constitue donc un chemin sans collision en ce qui concerne les obstacles inclus dans $\bar{D}$, i.e. le complémentaire de D. Tout chemin γ inclus dans D vérifie aussi cette propriété. □

Le dual d'un chemin constitué de segments et d'arcs de cercle alternés est défini comme étant l'ensemble des lignes brisée reliant les centres des différents arcs de cercle dont la courbure est de même signe. A chaque changement de signe de la courbure, une nouvelle ligne brisée commence. Parfois, une ligne peut se réduire à un simple point. C'est le cas pour γ^*, le dual de γ, qui se réduit au centre de l'unique arc de cercle qui compose γ. Dans [45], Laumond a montré que γ appartient à la même classe d'homotopie que γ_s si et seulement si :

1. γ^* est inclus dans le domaine $D^* = \bigcup_{\lambda\in[\lambda_1,\lambda_2]} P^*_\lambda$ avec P^*_λ le translaté de P_λ suivant le vecteur $(r.\cos\lambda, -r.\sin\lambda)$ (figure 2.6b).

2. γ^* coupe le domaine $K(D^*) = D^* \cap (\bigcap_{\lambda\in[\lambda_1,\lambda_2]} P^*_\lambda)$ (figure 2.6c).

Dans notre cas, γ^* se réduit au point c, centre de $\widehat{t_1t_2}$. La condition 2 oblige c à faire partie de $K(D^*)$ et l'on peut donc utiliser $K(D^*)$ pour caractériser l'ensemble des chemins de type γ qui appartiennent à la même classe d'homotopie que γ_s. Si l'on choisit le point c dans $K(D^*)$ alors on peut construire un chemin Γ qui est topologiquement équivalent à Γ_s et par conséquent à Γ_n. De plus Γ est sans collision avec les obstacles inclus dans $\bar{D}$.

2.3.3.2 Les obstacles inclus dans D

Dans la section précédente, nous avons défini le domaine D dans lequel doit se trouver Γ, mais nous l'avons fait sans tenir compte des obstacles qui se trouvent dans ce domaine. Soit $o \in \mathcal{B}$ un tel obstacle. Du fait de l'équivalence topologique, o ne doit pas être situé dans le domaine délimité par Γ_s et Γ. o se situe donc dans le domaine concave défini par γ. Compte tenu des remarques faites dans la section 2.3.2.2, γ est sans collision avec o si les trois conditions suivantes sont satisfaites :

1. $\widehat{t_1t_2}$ ne coupe pas l'obstacle $f(o)$=$\mathcal{G}(o, \mathcal{F}(r, l_f, w))$.

   Cette condition est satisfaite si l'on ampute $K(D^*)$ du domaine $\mathcal{G}(f(o), r)$. En effet, un arc de cercle de rayon r ne coupe pas un obstacle si la distance entre cet obstacle et le centre de cet arc est supérieure à r.

2. t_0t_1 ne coupe pas $\mathcal{G}(o, w/2)$.

   Cette condition est satisfaite si l'on ampute $K(D^*)$ du domaine $\mathcal{G}(g_a(o), r)$ où $g_a(o)$ est l'ensemble des points $p \in \mathcal{W}$ tels que le segment reliant tangentiellement p à C_p coupe $\mathcal{G}(o, w/2)$ (figure 2.7).

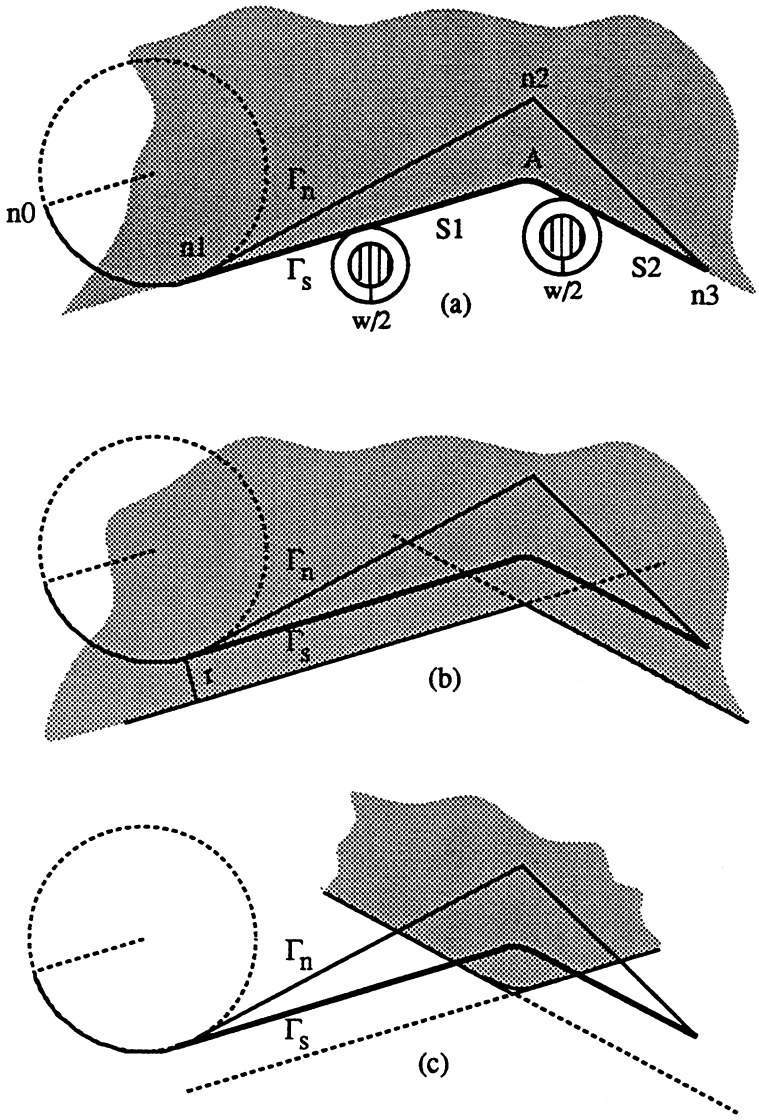


Figure 2.6 : (a) D, (b) D^* et (c) $K(D^*)$

3. t_2n_3 ne coupe pas $\mathcal{G}(o, w/2)$.

   Cette condition est satisfaite si l'on ampute $K(D^*)$ du domaine $\mathcal{G}(g_b(o), r)$ où $g_b(o)$ est l'ensemble des points $p \in \mathcal{W}$ tels que le segment reliant p à n_3 coupe $\mathcal{G}(o, w/2)$ (figure 2.7).

On sait maintenant comment choisir c pour garantir la non-collision de γ avec les obstacles inclus dans D. Il ne reste plus qu'à assurer l'équivalence topologique. Pour cela, o doit être situé dans le domaine concave défini par γ. L'analyse des différents placements possibles de γ par rapport à o montre que γ ne doit pas couper l'enveloppe convexe de $g_a(o)$ et $g_b(o)$. La condition d'équivalence topolo-

gique est satisfaite si l'on ampute $K(D^*)$ de $\mathcal{G}(g(o), r)$ où $g(o)$ est l'enveloppe convexe de $g_a(o)$ et $g_b(o)$.

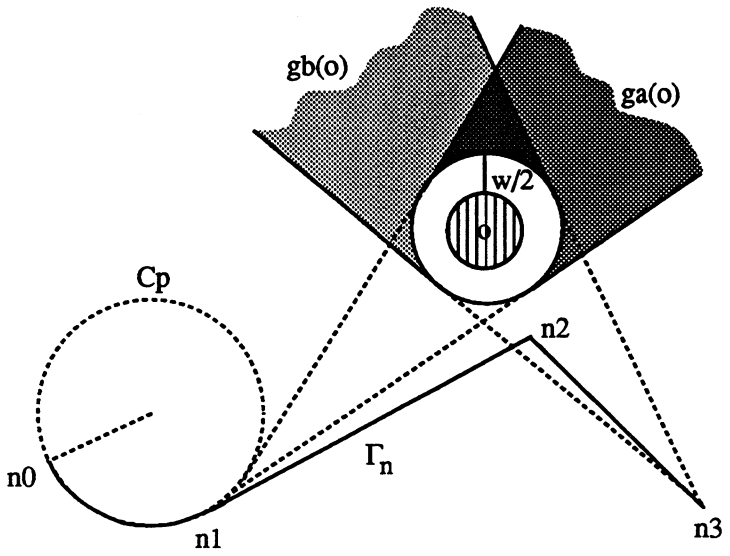


Figure 2.7 : $g_a(o)$ et $g_b(o), o \subset D$

2.3.3.3 Les conditions d'existence de Γ

Le choix du point c permet d'abord de définir le cercle C support de l'arc de cercle $\widehat{t_1 t_2}$ ce qui permet ensuite de définir le sous-chemin γ. Cependant, une condition nécessaire à l'existence de γ, et donc de Γ, est l'existence des deux segments $t_0 t_1$ et $t_2 n_3$ (fut-ce avec une longueur nulle).

En ce qui concerne l'existence de $t_0 t_1$, il y a deux situations possibles. Si le cercle C_p support de l'arc de cercle $\widehat{n_0 n_1}$ est situé dans le domaine convexe défini par $n_1 n_2$ et $n_2 n_3$, i.e. si C_p est 'à l'extérieur' de Λ, alors $t_0 t_1$ existe quel que soit le point c choisi. Sinon, t_1 doit être situé à l'extérieur de C_p. Cette condition est satisfaite si l'on ampute $K(D^*)$ du domaine $\mathcal{G}(C_p, r)$. En ce qui concerne $t_2 n_3$, il existe si n_3 est situé à l'extérieur du cercle C. Cette condition est satisfaite si l'on ampute $K(D^*)$ du domaine $\mathcal{G}(n_3, r)$.

2.3.3.4 La définition finale de CCS

Formellement l'espace des centres de giration CCS du virage Λ est défini par :

$$\begin{aligned} CCS \;=\; K(D^*) \;&-\; \bigcup_{o \in \mathcal{B}, o \subset D} \{\mathcal{G}(f(o), r), \mathcal{G}(g(o), r)\} \\ &-\; \mathcal{G}(n_3, r) \\ &-\; \mathcal{G}(C_p, r) \text{ si } C_p \text{ est 'à l'extérieur' de } \Lambda \end{aligned}$$

La figure 2.8 montre un exemple simple d'espace des centres de giration pour un virage Λ de squelette Γ_n=($\widehat{n_0 n_1}$, $n_1 n_2$, $n_2 n_3$). $\mathcal{B}$ comprend trois obstacles circulaires o_1, o_2 et o_3. Le mobile $\mathcal{A}$ est représenté dans la fenêtre inférieure gauche, l'arc de cercle représente son rayon de giration r. Les différents domaines dont on ampute $K(D^*)$ sont visualisés dans la fenêtre principale. La ligne noire épaisse représente la frontière de $K(D^*)$ tandis que la région pointillée est l'espace CCS des centres de giration de Λ. Si le centre de l'arc de cercle de giration est choisi dans CCS alors le chemin résultant Γ est sans collision et topologiquement équivalent à Γ_n. Naturellement, une condition suffisante d'existence pour Γ est que $CCS \neq \emptyset$.

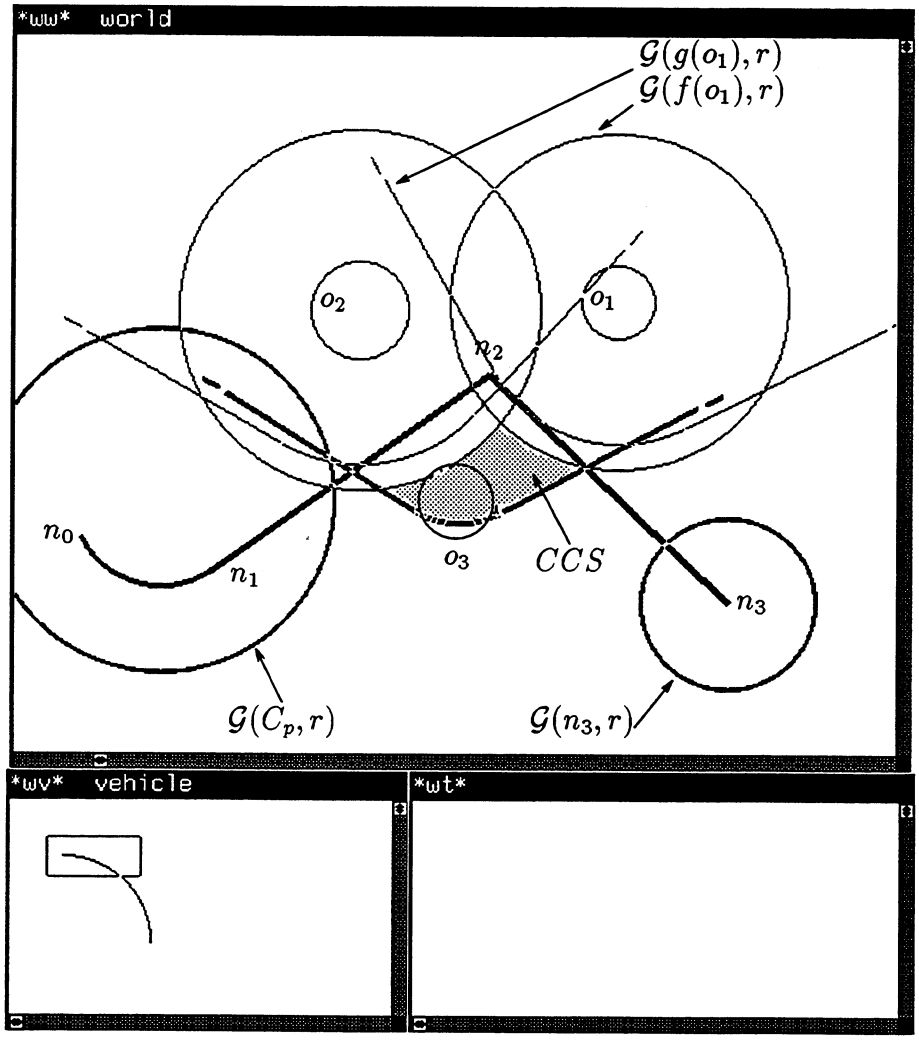


Figure 2.8 : un exemple d'espace des centres de giration

2.3.4 La définition de CCS : premier et dernier virage

2.3.4.1 Le premier virage

Le premier virage de $\mathcal{S}$ est caractérisé par le couple $(n_1n_2,\ n_2n_3)$ —il n'y a pas de virage précédent. $\mathcal{A}$ part de n_1 avec comme orientation initiale l'orientation du segment n_1n_2. la génération du chemin Γ associé au premier virage se fait en deux étapes :

1. Détermination d'un chemin à l'aide de la méthode générale en considérant que $c_p = n_1$ et que $r_p = 0$. Le résultat est un chemin $(n_1t_1,\ \widehat{t_1t_2},\ t_2n_3)$.

2. Remplacement du segment n_1t_1 par la séquence de classe C^1 composée de l'arc de cercle $\widehat{n_1t_0}$ de rayon r tangent à n_1n_2 au point n_1 et du segment t_0t_1 reliant cet arc à $\widehat{t_1t_2}$ (figure 2.9).

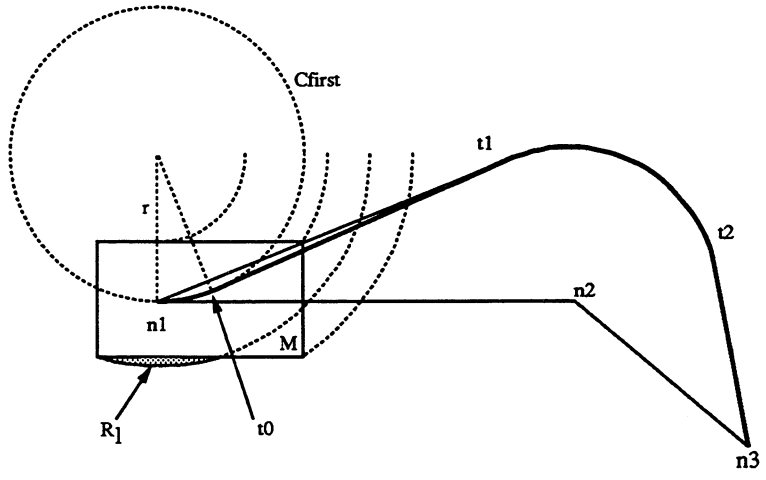


Figure 2.9 : le premier virage

La seconde étape permet de satisfaire la contrainte sur l'orientation initiale de $\mathcal{A}$ mais elle soulève deux problèmes : le remplacement est-il toujours possible et si oui, la nouvelle séquence est-elle sans collision ? Considérons tout d'abord la non-collision : la figure 2.9 présente la région balayée par $\mathcal{A}$ durant le parcours de la séquence $(\widehat{n_1t_0},\ t_0t_1)$ lorsque $\mathcal{A}$ commence à tourner vers la gauche. Par construction, il n'y a aucun obstacle dans le domaine compris entre le chemin nominal et le chemin obtenu à l'issue de la première étape. Par hypothèse, nous savons que $\mathcal{A}$ peut se déplacer le long de n_1n_2 sans collision. En conséquence, une collision ne peut se produire que si un obstacle coupe le domaine R_l. Lorsque $\mathcal{A}$ commence à tourner vers la droite, la situation est symétrique. Une collision ne peut se produire que si un obstacle coupe le domaine R_r, domaine symétrique de R_l par rapport à l'axe principal de $\mathcal{A}$. Sans perte de généralité, nous supposons

que la configuration initiale de $\mathcal{A}$ est telle que aucun obstacle ne coupe le domaine $R_l \cup R_r$. Ainsi nous avons l'assurance que le chemin obtenu est bien sans collision.

Considérons ensuite la possibilité ou non de remplacer n_1n_2 par la séquence $(\widehat{n_1t_0}, t_0t_1)$. Lorsque $\mathcal{A}$ commence à tourner dans la même direction que le virage, le remplacement est toujours possible. Par contre, lorsque $\mathcal{A}$ commence à tourner dans la direction opposée à celle du virage, le remplacement est possible si et seulement si $\widehat{t_1t_2}$ est situé à l'extérieur du cercle C_{first} de rayon r tangent à n_1n_2 au point n_1 et situé dans le domaine concave défini par n_1n_2 et n_2n_3. Cette condition est satisfaite si l'on ampute $K(D^*)$ du domaine $\mathcal{G}(C_{first}, r)$. L'espace des centres de giration CCS_{first} pour le premier virage de $\mathcal{S}$ est alors défini par :

$$\begin{aligned} CCS_{first} \;=\; K(D^*) \;&-\; \bigcup_{o\in\mathcal{B}\wedge o\subset D}\{\mathcal{G}(f(o),r),\mathcal{G}(g(o),r)\} \\ &-\; \mathcal{G}(n_3,r) \\ &-\; \mathcal{G}(C_{first},r) \end{aligned}$$

2.3.4.2 Le dernier virage

Comme dans le cas général, le dernier virage de $\mathcal{S}$ est caractérisé par le triplet $(\widehat{n_0n_1}, n_1n_2, n_2n_3)$. Cependant, $\mathcal{A}$ doit atteindre n_3 avec comme orientation finale l'orientation du segment n_2n_3. Cette situation est très proche de celle du premier virage et la génération du chemin Γ associé à Λ se fait aussi en deux étapes. La deuxième étape consiste à remplacer t_2n_3 par une séquence de classe C^1 composée d'un segment et d'un arc de cercle de rayon r. Les conditions de non-collision et d'existence de la nouvelle séquence sont similaires à celles du cas du premier virage et conduisent à la définition suivante de CCS_{last}, l'espace des centres de giration pour le dernier virage de $\mathcal{S}$:

$$\begin{aligned} CCS_{last} \;=\; K(D^*) \;&-\; \bigcup_{o\in\mathcal{B},o\subset D}\{\mathcal{G}(f(o),r),\mathcal{G}(g(o),r)\} \\ &-\; \mathcal{G}(n_3,r) \\ &-\; \mathcal{G}(C_p,r) \text{ si } C_p \text{ est 'à l'extérieur' de } \Lambda \\ &-\; \mathcal{G}(C_{last},r) \end{aligned}$$

avec C_{last} le cercle de rayon r tangent à n_2n_3 au point n_3 et situé dans le domaine concave défini par n_1n_2 et n_2n_3.

2.3.5 L'exploration de CCS

Dans les sections précédentes, nous avons défini CCS, l'espace des centres de giration pour un virage quelconque de $\mathcal{S}$ et nous avons montré qu'une condition suffisante pour obtenir un chemin Γ exécutable et sans collision est que le point c appartiennent à CCS. Se pose maintenant le problème de l'existence et surtout du choix de c.

CCS est inclus dans $\mathcal{W}$, c'est un ensemble qui n'est pas nécessairement connexe et dont les composantes connexes ne sont pas nécessairement simplement connexes puisqu'elles peuvent contenir des trous. Cependant, si l'on observe

la définition de CCS, on note que celui-ci est constitué d'un domaine support, $K(D^*)$, que l'on ampute d'un certain nombre de domaines. Tous ces domaines sont simplement connexes et d'une géométrie très simple. Aussi, plutôt que de représenter explicitement CCS, nous ne représentons explicitement que $K(D^*)$ et l'ensemble des domaines dont on l'ampute. Le test de l'appartenance d'un point quelconque p de $\mathcal{W}$ à CCS peut alors se faire par vérification de l'appartenance de p à $K(D^*)$ et de la non-appartenance de p aux différents domaines dont on ampute $K(D^*)$. Ainsi on dispose d'un moyen opératoire efficace pour décider si un point quelconque de $\mathcal{W}$ appartient à CCS. Le défaut de cette approche réside naturellement dans le fait que la réponse au problème de l'existence d'un point c nécessite, dans le cas où $CCS = \emptyset$, l'exploration complète de $K(D^*)$.

Afin de résoudre ce problème, nous utilisons une stratégie heuristique de recherche du point c qui s'appuie sur la remarque suivante: la façon la plus 'naturelle' de négocier un virage Λ quelconque est de suivre l'arc de cercle tangent aux segments n_1n_2 et n_2n_3. Soit g, le centre de cet arc de cercle, c est recherché dans le voisinage de g. Plus précisément, la recherche de c est restreinte au disque C_g de centre g et dont le rayon est fonction de r (e.g. $r_g = 2r$) (figure 2.10). Concrètement, C_g est discrétisé en un ensemble de points $\mathcal{E}=\{\varepsilon_1,\ldots,\varepsilon_e\}$ qui constitue l'espace de recherche de Λ. Si $CCS \cap \mathcal{E} = \emptyset$ alors on décide qu'il n'existe pas de chemin solution Γ pour le virage considéré.

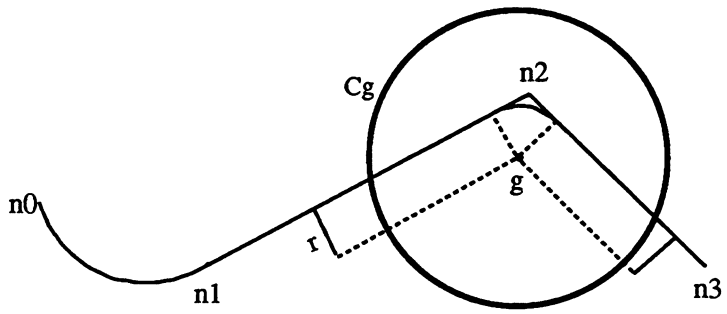


Figure 2.10: C_g

Remarque: Cette stratégie n'est naturellement pas complète puisque CCS n'est pas exploré dans sa totalité. Cependant, les tests que nous avons réalisés nous ont permis de la valider expérimentalement. □

2.3.6 L'enchaînement des virages et la génération du chemin Υ

2.3.6.1 L'arbre $\mathcal{T}$ associé à la génération de Υ

La définition puis l'exploration de l'espace des centres de giration d'un virage Λ de $\mathcal{S}$ permet de construire un chemin solution Γ pour Λ. Par construction, Γ

s'enchaîne convenablement avec le chemin associé au virage qui précède Λ. De même, le traitement du virage qui suit Λ va produire un chemin qui s'enchaîne convenablement avec Γ. Ainsi le traitement séquentiel de tous les virages de $\mathcal{S}$ du premier jusqu'au dernier détermine le chemin complet Υ recherché.

Λ étant caractérisé par le triplet $(\widehat{n_0 n_1}, n_1 n_2, n_2 n_3)$, il dépend du chemin qui a été choisi pour le virage qui précède Λ. De même, lors du traitement de Λ, le choix de c parmi l'un des éléments de $\mathcal{E}$ définit le prochain virage. Il y a donc au plus e possibilités en ce qui concerne ce prochain virage. Chacune de ces possibilités engendre à son tour e possibilités pour le virage suivant et ainsi de suite.

Il est donc possible de formuler le problème de la génération du chemin complet Υ comme un problème d'exploration d'un certain arbre $\mathcal{T}$. On définit un nœud N_j de $\mathcal{T}$ par le couple $\{\Lambda_j, d_j\}$ avec Λ_j un virage donné et d_j le rang correspondant à ce virage dans $\mathcal{S}$, $d_j \in \{1, \ldots, (s-2)\}$. Soit CCS_j et $\mathcal{E}_j$ l'espace des centres de giration et l'espace de recherche associés à Λ_j, N_j est dit *valide* si et seulement si $CCS_j \cap \mathcal{E}_j \neq \emptyset$, i.e. s'il existe un chemin solution Γ pour le virage Λ_j. Le nœud N_k a pour parent le nœud N_j si et seulement si :

1. $d_k = d_j + 1$, i.e. si Λ_k est un virage qui suit immédiatement Λ_j.

2. $\exists \varepsilon_i \in \mathcal{E}_j, \varepsilon_i \in CCS_j$ et ε_i est le centre de l'arc de cercle $\widehat{n_0 n_1}$ associé à Λ_k.

Chaque arête de $\mathcal{T}$ est donc caractérisée par un point ε_i qui définit simultanément le chemin solution Γ pour le nœud parent et le virage associé pour le nœud fils. $\mathcal{T}$ est un arbre e-aire de profondeur (s-2).

Le premier virage de $\mathcal{S}$ constitue la racine de $\mathcal{T}$. Un nœud de profondeur (s-2) est un nœud dont le virage associé correspond au dernier virage de $\mathcal{S}$. Tout chemin dans $\mathcal{T}$ reliant la racine et un nœud de profondeur (s-2) valide détermine un chemin complet Υ.

2.3.6.2 L'exploration de $\mathcal{T}$

La détermination de Υ passe donc par l'exploration de $\mathcal{T}$. Pour ce faire, on dispose d'un certain nombre de techniques classiques [62]. Dans la mesure où l'on cherche un chemin entre la racine de $\mathcal{T}$ et un nœud quelconque de profondeur donnée, une simple exploration en profondeur d'abord de $\mathcal{T}$ permet de résoudre le problème. Schématiquement, l'exploration s'effectue de la manière suivante :

algorithme

L'algorithme utilise une liste OPEN des nœuds de $\mathcal{T}$ dont on n'a pas encore calculé les successeurs. N contient le nœud en cours de traitement : $\mathrm{N} = \{\Lambda, d\}$. CCS et $\mathcal{E}$ dénotent respectivement l'espace des centres de giration et l'espace de recherche de Λ.

1. Initialisation de OPEN avec la racine de $\mathcal{T}$.

2. Si OPEN est vide alors 'échec'.
3. N ← tête de OPEN.
4. Calcul de CCS et $\mathcal{E}$.
5. Si d=(s-2) et si N est valide alors 'succès'.
6. Calcul des successeurs de N : il y a un successeur pour chaque $\varepsilon_i \in \mathcal{E}$ tel que $\varepsilon_i \in CCS$. Les successeurs sont installés en tête de OPEN.
7. Aller à 2.

$\mathcal{T}$ a au plus $\sum_{u=1}^{s-2} e^u$ arêtes/nœuds. Dans le pire des cas, cet algorithme examine chacun des nœuds. Le traitement d'un nœud impliquant le calcul d'un espace des centres de giration, il est important pour l'efficacité de l'algorithme de minimiser le nombre de nœuds examinés. Pour ce faire, il faut, lors du calcul des successeurs de N, être capable de les classer selon un critère donné et de placer le 'meilleur' successeur en tête de OPEN ; 'meilleur' dans le sens où il ne provoque pas d'échec lors du traitement des virages ultérieurs. Le successeur de Λ dépendant directement du point ε_i choisi, se pose alors le problème de l'ordonnancement des ε_i ou plus précisément le problème de l'évaluation du risque d'échec ultérieur entraîné par le choix de ε_i comme point g de Λ. Il n'y a malheureusement pas de réponse définitive à cette question. Le choix du 'meilleur' ε_i dépend non seulement de la disposition particulière des obstacles mais aussi de la forme du squelette. Cependant et suite à la remarque faite dans la section 2.3.5, nous avons défini les deux stratégies alternatives suivantes :

1. Stratégie générale : on choisit en priorité le point $\varepsilon_i \in CCS$ qui est le plus proche de g.
2. Stratégie pour virages 'proches' : cette stratégie est appliquée dans le cas suivant : soit ($\widehat{n_0 n_1}$, n_1n_2, n_2n_3) le triplet caractérisant le virage courant Λ. Le virage qui suit Λ est caractérisé par la paire (n_2n_3, n_3n_4). Soit P le demi-plan qui contient n_3 et dont la frontière est la bissectrice des deux segments n_1n_2 et n_3n_4 (figure 2.11). Si $g \in P$ alors on dit que les deux virages sont proches. Dans ce cas, on choisit en priorité le point $\varepsilon_i \in CCS \wedge \varepsilon_i \notin P$ qui est le plus proche de g.

2.4 Implantation et résultats expérimentaux

L'algorithme de planification de chemin sans manœuvre présenté dans la section précédente a fait l'objet d'une implantation. Un prototype, développé en Common Lisp nous a permis d'évaluer l'algorithme. Dans les deux sections suivantes, nous présentons quelques résultats d'expérimentations réalisées dans des espaces de travail tout d'abord structurés puis quelconques.

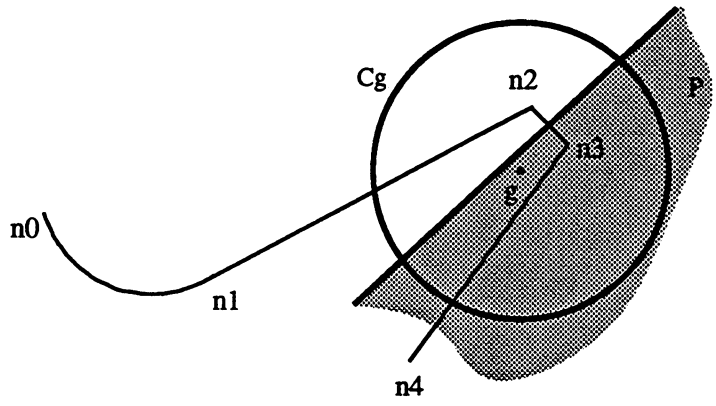


Figure 2.11 : deux virages 'proches'

2.4.1 Espaces de travail structurés

Tout d'abord, dans la mesure où nous n'avons pas abordé le problème de la structuration de l'espace de travail, nous avons choisi, pour tester notre algorithme, des espaces de travail explicitement structurés tel que le réseau routier. Ce choix permet, d'une part, d'éliminer le problème de l'extraction des voies qui constituent le squelette du chemin à planifier et d'autres part, de tester l'algorithme dans des situations réalistes.

Nous avons donc considéré des mobiles dont les caractéristiques (dimension, rayon de giration) sont celles de véhicules réels (Renault 5, Citroën XM, etc.) et nous les avons placés dans des espaces de travail de type réseau routier et plus précisément de type intersection : une intersection représente le point de jonction entre plusieurs routes, i.e. l'endroit où les voies vont se croiser entre elles ce qui permet la mise en œuvre de notre algorithme. Il existe un certain nombre de règles qui président à l'aménagement du réseau routier en général et des intersections en particulier [18, 58, 67]. Parmi cet ensemble de règles, nous avons retenu en particulier celles concernant la dimension des voies de circulation (de 3 à 3,50 mètres pour des véhicules d'environ deux mètres de large). La définition d'un espace de travail se fait donc en définissant un réseau quelconque de routes. Chaque route se compose d'une ou de plusieurs voies rectilignes adjacentes et aux dimensions voulues, ces réseaux peuvent être simples (figure 2.12) ou plus complexes (figure 2.13 et 2.14). Ce réseau de routes définit la chaussé, i.e. l'espace de travail réel du mobile. Le complémentaire de cette chaussée constitue les différents obstacles.

Les figures 2.12, 2.13 et 2.14 montrent quelques résultats d'expérimentations réalisées dans des espaces de travail de ce type. Les obstacles sont représentés par des polygones et le squelette $\mathcal{S}$ par une fine ligne brisée. Le chemin final Υ est représenté par un trait noir épais (les rayons des arcs de cercle qui compo-

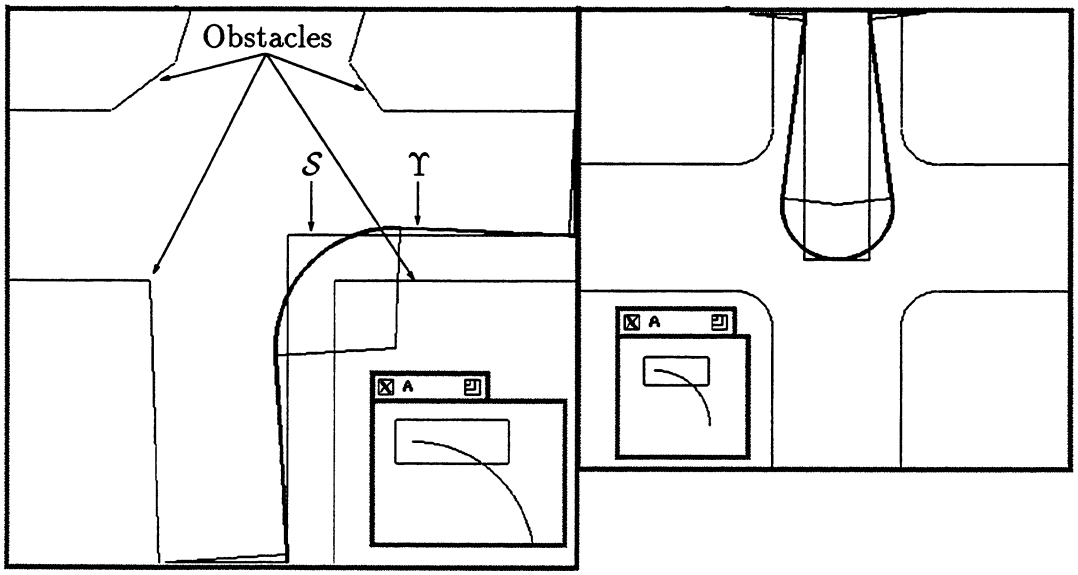


Figure 2.12 : a figure 2.12: b

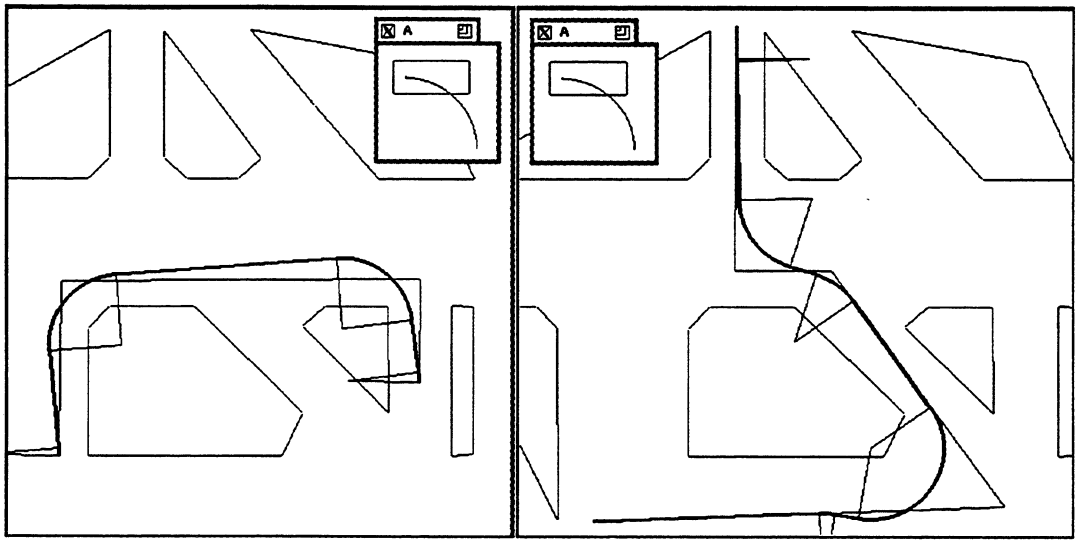


Figure 2.13 : a figure 2.13: b

sent Υ sont affichés). Le mobile $\mathcal{A}$ considéré est visualisé dans une sous-fenêtre avec son rayon de giration minimum. Ces expérimentations se sont révélées très concluantes. Elles ont montré que :

1. L'ensemble des solutions pour un virage donné est relativement réduit et est en général localisé dans le voisinage du point g. Autrement dit, l'heuristique introduite dans la section 2.3.5 conduisant à restreindre la recherche de c

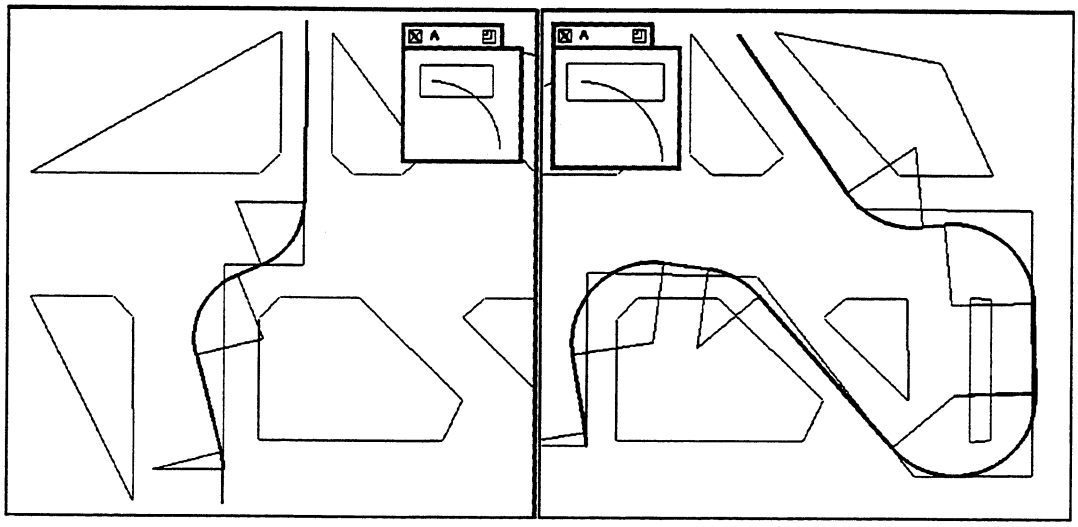


Figure 2.14 : a figure 2.14: b

au disque C_g de centre g est valide.

2. L'utilisation combinée des deux stratégies d'exploration du disque C_g (cf. §2.3.6.2) permet de converger rapidement vers une solution globale. En pratique, l'algorithme a toujours convergé *directement* vers la solution, même dans le cas très contraint du demi-tour (figure 2.12b) ou bien de l'enchaînement rapide de deux virages (figure 2.14a).

3. L'algorithme est efficace. Tout d'abord en raison de la remarque précédente et ensuite, parce que le temps de traitement d'un virage (construction et exploration de CCS est faible (de l'ordre de la seconde pour une implantation réalisée sans souci d'efficacité).

2.4.2 Espaces de travail quelconques

Nous avons commencé quelques expérimentations en espaces de travail quelconques du type de celui qui est présenté dans les figures 2.15, 2.16 et 2.17. Nous avons emprunté cet exemple à Jacobs et Canny [32]. Cet exemple présentait le plus court chemin sans manœuvre, entre deux configurations données, pour un mobile ponctuel de type voiture. Nous avons alors défini, en nous basant simplement sur le graphe de visibilité, un squelette topologiquement équivalent à ce plus court chemin (figure 2.15a), puis, nous avons fourni ce squelette à notre algorithme. Celui-ci a très facilement déterminé un chemin sans manœuvre et ceci malgré la difficulté supplémentaire introduite par le fait que le mobile n'était plus un simple point mais un rectangle (figure 2.15b). Les figures 2.16 et 2.17 présentent deux autres exemples de planification de chemin à partir de squelettes établis grâce au graphe de visibilité.

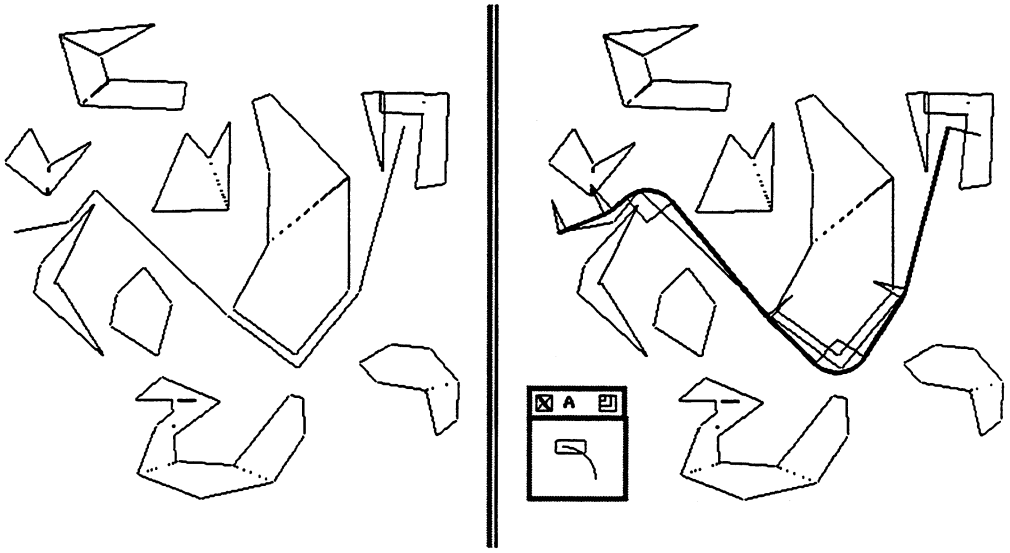


Figure 2.15 : a

figure 2.15: b

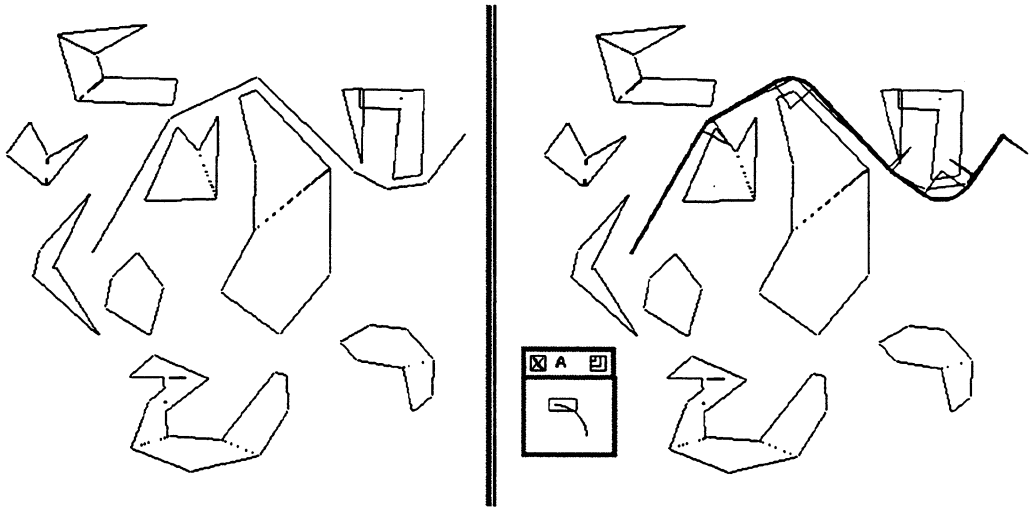


Figure 2.16 : a

figure 2.16: b

Toutefois, l'évaluation de notre algorithme en espace de travail quelconque est délicate. En effet, lorsqu'un échec survient lors de la planification d'un chemin, que pouvons-nous en conclure ? Cet échec peut être dû à l'incomplétude de l'algorithme lui-même. Mais il peut aussi être dû au squelette qui lui a été fourni en entrée. Enfin, il se peut, tout simplement, qu'il n'existe pas de chemin sans manœuvre topologiquement équivalent au squelette. En d'autres termes, l'évaluation de notre algorithme en espace de travail quelconque ne peut être faite indépendamment du processus d'extraction de voies. Par ailleurs, pour pouvoir évaluer précisément le couple (processus d'extraction de voies, algorithme de planification de chemin), il faudrait disposer d'un résultat d'existence de chemin sans

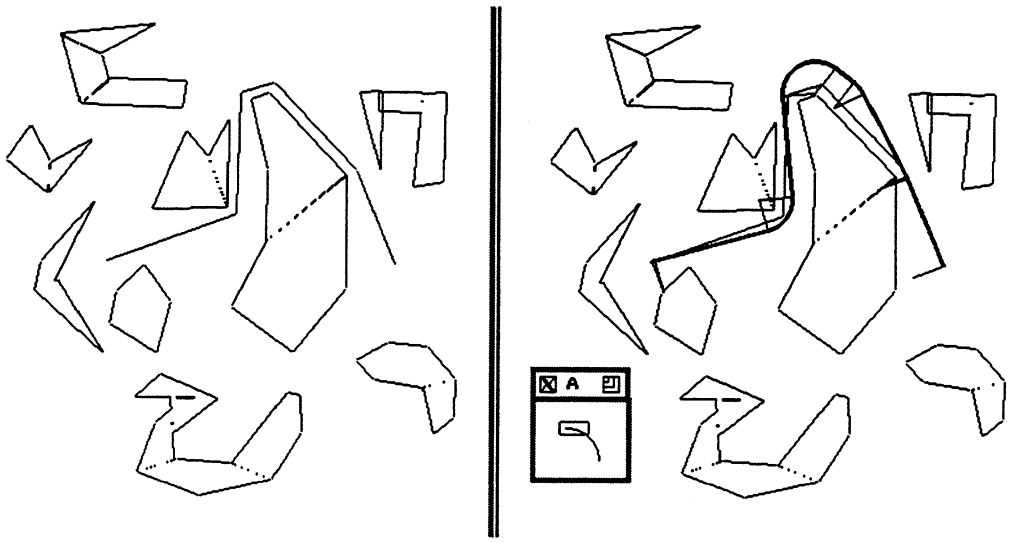


Figure 2.17 : a figure 2.17: b

manœuvre similaire au résultat de contrôlabilité pour les chemins avec manœuvre (cf. §1.3.3). Malheureusement, on ne dispose pas de ce résultat à ce jour.

2.5 Conclusion

Dans ce chapitre, nous avons présenté une technique de planification de chemin sans manœuvre pour un mobile de type voiture. Cette technique s'applique à un espace de travail statique et structuré en un ensemble de voies rectilignes pouvant se croiser. Etant donné une séquence de voies, on détermine un chemin sans collision avec les obstacles fixes de l'espace de travail et exécutable par le mobile compte tenu de ses contraintes cinématiques. L'algorithme présenté engendre un chemin composé de segments et d'arcs de cercle tangents. Il s'appuie sur la notion de 'virage', i.e. une transition d'une voie à une autre. Chaque virage est négocié au moyen d'un arc de cercle dont le centre est recherché dans un espace particulier dérivé de *l'espace des centres de giration* introduit en 1987 par J-P. Laumond.

L'algorithme présenté a été implanté et testé. Les tests réalisés ont permis de démontrer l'efficacité en terme de temps de calcul de l'algorithme. Cette efficacité découle de la mise en œuvre d'une heuristique de recherche des centres des différents arcs de cercle qui compose le chemin planifié. Notons l'incomplétude de l'algorithme, incomplétude toute relative puisque les différentes expérimentations qui ont été menées dans le cas d'espaces de travail réalistes se sont révélées concluantes.

Les extensions et améliorations futures qu'il serait souhaitable d'apporter à la technique que l'on vient de présenter pourraient porter sur les points suivants :

- Généralisation de la structure, i.e. prise en compte de voies non-rectilignes

et en particulier circulaire.

- Jusqu'à présent, les voies définissant la structure de l'espace de travail étaient supposées connues a priori. Ceci nous amenait à considérer des espaces de travail naturellement structurés, i.e. pour lesquels l'extraction de la structure était immédiate (e.g. espaces de travail de type réseau routier). La question à laquelle il faut maintenant répondre est de savoir comment obtenir automatiquement cette structure. La réponse à cette question permettrait d'étendre notre approche à des espaces de travail quelconques.

- Passage à des espaces de travail tri-dimensionnels, en particulier des espaces de travail sous-marin ou aérien.

Chapitre 3

La planification de trajectoire

Ce chapitre présente une technique de planification de trajectoire sans collision pour un mobile de type voiture dont la vitesse et l'accélération sont limitées. La technique présentée s'applique à un espace de travail dynamique. Etant donné un ensemble de chemins adjacents dont l'un amène le mobile à son but, et sachant que le mobile peut passer d'un chemin à un autre, on détermine la trajectoire du mobile le long de ces chemins afin d'éviter toute collision avec les obstacles mobiles de l'espace de travail tout en respectant les contraintes dynamiques du mobile (vitesse et accélération limitée). L'algorithme présenté discrétise puis explore *l'espace-temps des états* du mobile (cet espace-temps des états est obtenu en rajoutant la dimension temporelle à l'espace des états du mobile).

3.1 Introduction

Au chapitre précédent, nous avons abordé le problème de la planification d'un chemin sans manœuvre pour un mobile de type voiture. Nous avons présenté un algorithme capable de déterminer un chemin qui soit exécutable par le mobile et sans collision avec les obstacles fixes de l'espace de travail du mobile.

Dans ce chapitre, nous présentons une méthode de planification de trajectoire pour un mobile de type voiture dont la vitesse et l'accélération sont limitées et qui se déplace dans un espace de travail dynamique. A ce jour, on ne dispose que de très peu de résultats sur ce problème précis (cf. §1.4.3). Notre contribution se trouve en prolongement direct des travaux entrepris par Canny et *al.* [13] et poursuivis par Donald et Xavier [19] qui présentent un algorithme de planification de trajectoire pour un mobile dont la vitesse et l'accélération sont limitées mais qui se déplace dans un espace de travail statique. Le principe de leur approche consiste en une exploration de l'espace des états du mobile, cet espace des états ayant été au préalable discrétisé. Nous reprenons ce principe et l'étendons pour prendre en compte des obstacles mobiles ce qui nous amène à définir *l'espace des états-temps* du mobile, i.e. l'espace des états du mobile augmenté de la dimension

temporelle.

D'autre part, nous nous inspirons du principe de la *décomposition chemin-vitesse* pour réduire la complexité de notre problème. Ce principe introduit par Kant et Zucker [38] consiste à planifier une trajectoire en indexant temporellement un chemin calculé au préalable (cf. §1.4.3.1). Nous allons donc déterminer la vitesse du mobile le long d'un chemin (calculé avec une méthode telle que celle présentée au chapitre précédent). Cependant, nous apportons une extension majeure à ce principe en autorisant le mobile à quitter son chemin pour passer sur un chemin adjacent (l'adjacence entre chemins est définie formellement au paragraphe 3.2.1).

3.2 Formulation du problème

3.2.1 La caractérisation des chemins adjacents

Soit $\mathcal{A}$ un mobile de type voiture qui se déplace dans un espace de travail bi-dimensionnel $\mathcal{W} = \mathbf{R}^2$. Un chemin sans manœuvre exécutable par $\mathcal{A}$ est une courbe continue $\Psi : I \longrightarrow \mathcal{W}$ d'un intervalle fermé I de $\mathbf{R}$ dans $\mathcal{W}$ qui est de classe C^1 et dont la courbure (quand elle est définie) est inférieure à $1/\rho_R^{min}$ avec ρ_R^{min} le rayon de giration minimum de $\mathcal{A}$ (cf. §2.2.1.4).

Soit $\Psi : I \longmapsto \mathcal{W}$ et $\Omega : J \longmapsto \mathcal{W}$ deux chemins sans manœuvre et exécutable par $\mathcal{A}$, Ψ et Ω sont **adjacents**, i.e. disposés l'un parallèlement à l'autre avec un écart constant, si et seulement si (cf. figure 3.1) :

1. $\forall i \in I, \exists j \in J$ unique et tel que :

$$\Psi(i)\overrightarrow{\Omega}(j).\dot{\Psi}(i) = 0, \quad \Psi(i)\overrightarrow{\Omega}(j) \wedge \dot{\Psi}(i) > 0, \quad \| \Psi(i)\overrightarrow{\Omega}(j) \| = \delta L$$

2. et $\forall j \in J, \exists i \in I$ unique et tel que :

$$\Omega(j)\overrightarrow{\Psi}(i).\dot{\Omega}(j) = 0, \quad \Omega(j)\overrightarrow{\Psi}(i) \wedge \dot{\Omega}(j) < 0, \quad \| \Omega(j)\overrightarrow{\Psi}(i) \| = \delta L$$

où δL est l'écart entre Ψ et Ω. Dans ce cas, Ψ (resp. Ω) est le chemin *de droite* (resp. *de gauche*).

D'autre part, on associe au chemin Ψ son **couloir**, noté $\mathcal{R}(\Psi)$, qui est l'ensemble des points $p \in \mathcal{W}$ dont la distance à Ψ est inférieure à $\delta L/2$.

3.2.2 La structure de $\mathcal{W}$

Nous nous plaçons dans le cas où $\mathcal{W}$ est structuré en un ensemble de l chemins adjacents deux à deux avec un écart constant δL. Ces chemins sont exécutables par $\mathcal{A}$ et sans collision avec les obstacles fixes de $\mathcal{W}$. Soit Υ_i, $i = 0, 2, \ldots, l-2, l$,

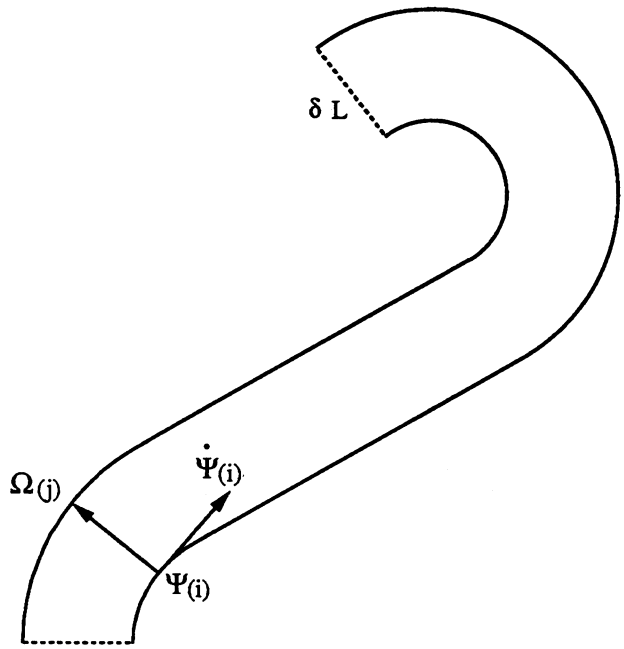


Figure 3.1 : Ψ et Ω, deux chemins adjacents

cet ensemble de chemins (la raison de cette indexation particulière apparaîtra par la suite). Υ_i est adjacent à Υ_{i+2} (resp. Υ_{i-2}) et Υ_i est le chemin de droite (resp. de gauche). On suppose que cette structure vérifie les propriétés suivantes :

1. L'ordre de grandeur de δL est celui de la largeur de $\mathcal{A}$.
2. $\forall \Upsilon_i, i = 0, \ldots, l$, il n'y a aucun obstacle fixe qui coupe $\mathcal{R}(\Upsilon_i)$.
3. $\forall \Upsilon_i, i = 0, \ldots, l$, la région balayée par $\mathcal{A}$ quand celui-ci se déplace le long de Υ_i est incluse dans $\mathcal{R}(\Upsilon_i)$.

Autrement dit, $\mathcal{A}$ peut se déplacer le long d'un chemin sans risque de collision avec les obstacles fixes de $\mathcal{W}$ et deux mobiles tels que $\mathcal{A}$ peuvent se déplacer sur deux chemins adjacents sans jamais risquer de rentrer en collision l'un avec l'autre.

3.2.3 Les obstacles mobiles

Le mobile $\mathcal{A}$ dont on veut planifier la trajectoire doit éviter toute collision avec les différents obstacles mobiles qui se déplacent dans $\mathcal{W}$. Pour simplifier, nous supposons que, lorsque $\mathcal{A}$ se déplace le long du chemin Υ_i, il doit éviter tout obstacle mobile qui coupe la région $\mathcal{R}(\Upsilon_i)$. Ce qui peut s'exprimer de manière équivalente en disant que tout obstacle mobile grossi de $\delta L/2$ qui coupe Υ_i doit être évité.

Les contraintes de non-collision induites par les obstacles mobiles vont être modélisées par un ensemble de particules ponctuelles $\mathcal{B}_j$, $j = 1, \ldots, n$, dont nous caractérisons la position à l'instant t par le couple $(\Upsilon_{\mathcal{B}_j}(t), p_{\mathcal{B}_j}(t))$ avec $\Upsilon_{\mathcal{B}_j}(t)$ le chemin occupé par $\mathcal{B}_j$ et $p_{\mathcal{B}_j}(t)$, l'abscisse curviligne le long de ce chemin. Le couple $(\Upsilon_{\mathcal{B}_j}(t), p_{\mathcal{B}_j}(t))$ indique que l'obstacle réel associé à $\mathcal{B}_j$ coupe la région $\mathcal{R}(\Upsilon_{\mathcal{B}_j})$ au voisinage de cette abscisse. Plus précisément, il indique que l'obstacle réel associé à $\mathcal{B}_j$ grossi de $\delta L/2$ coupe $\Upsilon_{\mathcal{B}_j}$ sur un intervalle $[o_{\mathcal{B}_j}(t), q_{\mathcal{B}_j}(t)]$ centré en $p_{\mathcal{B}_j}(t)$. Nous complétons la définition de $\mathcal{B}_j$ par $w_{\mathcal{B}_j} = (\max_t \mid q_{\mathcal{B}_j}(t) - o_{\mathcal{B}_j}(t) \mid)/2$, i.e. la moitié de la taille maximale de la partie du chemin $\Upsilon_{\mathcal{B}_j}(t)$ occupé par $\mathcal{B}_j$.

Remarque : Ainsi que nous l'avons mentionné dans le chapitre introductif, le comportement des obstacles mobiles n'est connu que sur un intervalle de temps donné, un *horizon temporel.* Par conséquent, les fonctions $\Upsilon_{\mathcal{B}_j}(t)$ et $p_{\mathcal{B}_j}(t)$ sont (non nécessairement partout) définies sur l'intervalle fermé $[0, t_{max}]$ où t_{max} est l'horizon temporel. □

3.2.4 La caractérisation de $\mathcal{A}$ et de ses mouvements

Le mobile $\mathcal{A}$ dont on veut planifier la trajectoire est lui aussi modélisé par une particule ponctuelle. Le type de mouvement que l'on souhaite planifier pour $\mathcal{A}$ consiste à suivre le chemin Υ_i sur lequel il se trouve. Cependant, il peut aussi effectuer un changement de chemin, i.e. passer de Υ_i vers l'un des chemins adjacents Υ_{i-2} ou Υ_{i+2} (quand ils existent). Canny [12] a montré le caractère NP-dur de la planification de trajectoire pour un robot ponctuel dont la vitesse est limitée et qui se déplace dans un plan au milieu de polygones en translation (cf. §1.4.2). Ce résultat ainsi que la nature particulière des mouvements de $\mathcal{A}$ nous a amené à simplifier le problème en décomposant les déplacements de $\mathcal{A}$ en deux composantes que l'on traite indépendamment : (a) le déplacement le long d'un chemin —*déplacement longitudinal*— et (b) le déplacement d'un chemin à un autre — *déplacement latéral.*

3.2.4.1 Le déplacement latéral et les chemins fictifs

Considérons tout d'abord le déplacement latéral, i.e. le changement de chemin tel qu'il est illustré dans la figure 3.2a. A l'instant t, $\mathcal{A}$ quitte son chemin courant Υ_i en direction du chemin adjacent Υ_{i+2}. Soit Δt l'intervalle de temps requis pour effectuer le changement de chemin. La forme de la trajectoire suivie durant le changement de chemin et la durée de Δt dépendent toutes deux des caractéristiques de $\mathcal{A}$, de Υ_i et de Υ_{i+2}. Dans la section 3.4, nous présentons une procédure de calcul de trajectoire et d'évaluation de Δt pour un mobile $\mathcal{A}$ de type voiture. Sachant que la valeur de δL est de l'ordre de la largeur de $\mathcal{A}$ alors il est raisonnable que, pendant le changement de chemin, $\mathcal{A}$ évite les collisions avec les

obstacles mobiles de chacun des chemins Υ_i et Υ_{i+2}. En conséquence, l'évaluation de la non-collision d'un changement de chemin peut être faite très simplement en testant la non-collision sur chacun des chemins Υ_i et Υ_{i+2} durant l'intervalle de temps $[t, t+\Delta t]$. Cette hypothèse rend possible la modélisation d'un déplacement latéral par l'enchaînement de trois actions successives :

1. A l'instant t, $\mathcal{A}$ passe instantanément de Υ_i à un chemin fictif Υ_{i+1}, Υ_{i+1} est adjacent à Υ_i et Υ_{i+2} avec un écart $\delta L/2$,
2. $\mathcal{A}$ se déplace le long de ce chemin fictif Υ_{i+1} durant $[t, t+\Delta t]$ (on suppose que les contraintes de non-collision imposées par les obstacles mobiles de Υ_i et Υ_{i+2} se retrouvent aussi sur Υ_{i+1})
3. A l'instant $t+\Delta t$, $\mathcal{A}$ passe instantanément de Υ_{i+1} à Υ_{i+2} (voir figure 3.2b).

Cette modélisation réduit un déplacement latéral (de nature bi-dimensionnelle) à un déplacement longitudinal le long d'un chemin fictif (de nature uni-dimensionnelle).

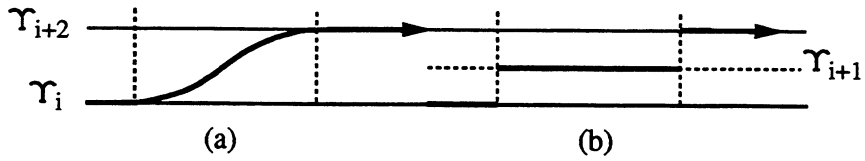


Figure 3.2 : le changement de chemin

3.2.4.2 Le déplacement longitudinal

Le déplacement longitudinal de $\mathcal{A}$ le long d'un chemin (réel ou fictif) est un mouvement uni-dimensionnel obtenu en appliquant à $\mathcal{A}$ une accélération $\ddot{p}(t)$. La vitesse $\dot{p}(t)$ et la position $p(t)$ de $\mathcal{A}$ le long de ce chemin sont définis respectivement comme la première et la seconde intégrale de $\ddot{p}(t)$ soumises à une position et une vitesse initiale. Ainsi que nous l'avons mentionné dans le paragraphe introductif, nous supposons que $\ddot{p}(t)$ et $\dot{p}(t)$ sont limitées :

$$-\ddot{p}_{max} \leq \ddot{p}(t) \leq \ddot{p}_{max} \tag{3.1}$$

$$0 \leq \dot{p}(t) \leq \dot{p}_{max} \tag{3.2}$$

3.2.5 La planification d'une trajectoire pour $\mathcal{A}$

Nous nous plaçons donc dans le cas où $\mathcal{W}$ est structuré en un ensemble de l chemins adjacents deux à deux avec un écart constant $\delta L/2$. Soit Υ_i, $i = 0, \ldots, l$,

cet ensemble de chemins, un chemin d'indice pair est un chemin réel tandis qu'un chemin d'indice impair est un chemin fictif. Dans la suite de la présentation nous confondrons un chemin et son indice.

On définit un **état** de $\mathcal{A}$ par le triplet $(\Upsilon, p, \dot{p})$ avec $\Upsilon \in \{0, \ldots, l\}$, le chemin courant de $\mathcal{A}$, p l'abscisse curviligne le long de ce chemin et $\dot{p}$ la vitesse instantanée de $\mathcal{A}$. Une **trajectoire** de $\mathcal{A}$ est définie par une fonction Γ associant à un instant $t \in [0,\ t_\Gamma]$ l'état $\Gamma(t) = (\Upsilon(t), p(t), \dot{p}(t))$. La **durée** de la trajectoire Γ est égale à t_Γ. Les deux composantes $\Upsilon(t)$ et $(p(t),\ \dot{p}(t))$ de cette trajectoire sont définies par les deux fonctions suivantes :

1. Υ: $[0,\ t_\Gamma] \longrightarrow \{0, \ldots, l\}$ qui indique le chemin courant de $\mathcal{A}$. Si Υ_t est pair alors $\mathcal{A}$ se trouve sur le chemin correspondant sinon $\mathcal{A}$ est en train d'effectuer un changement de chemin entre les chemins $\Upsilon_t - 1$ et $\Upsilon_t + 1$.

2. $\ddot{p}$: $[0,\ t_\Gamma] \longrightarrow [-\ddot{p}_{max},\ \ddot{p}_{max}]$ qui est l'accélération instantanée appliquée à $\mathcal{A}$. $p(t)$ et $\dot{p}(t)$ sont obtenues par intégration de $\ddot{p}(t)$.

En ce qui concerne l'évitement de collision, Γ est théoriquement sans collision si et seulement si $\forall t \in [0,\ t_\Gamma], p_{\mathcal{B}_j}(t) \neq p(t)$ pour chaque obstacle $\mathcal{B}_j$ tel que $\Upsilon_{\mathcal{B}_j}(t) = \Upsilon(t)$. Cependant, dans le double but d'obtenir une trajectoire qui (a) prenne en compte le fait que que $\mathcal{A}$ et les $\mathcal{B}_j$ ne sont pas des points et (b) soit 'robuste' dans le sens où elle ne rase pas les obstacles[1] alors, nous imposons que $\mathcal{A}$ évite chaque obstacle $\mathcal{B}_j$ avec une marge de sécurité $\delta_{\mathcal{B}_j}(t)$ dépendante de la taille et de la vitesse de $\mathcal{A}$ ainsi que de la taille de l'obstacle réel associé à $\mathcal{B}_j$. Formellement, une trajectoire Γ est dite **sûre** si et seulement si :

$$\forall t \in [0,\ t_\Gamma], \forall \mathcal{B}_j, j = 1, \ldots, n : \quad \Upsilon_{\mathcal{B}_j}(t) \neq \Upsilon(t) \text{ ou } \Upsilon_{\mathcal{B}_j}(t) = \Upsilon(t) \text{ et } |p(t) - p_{\mathcal{B}_j}(t)| > \delta_{\mathcal{B}_j}(t)$$

avec $\delta_{\mathcal{B}_j}(t) = w_{\mathcal{A}} + c_0 + c_1 \mid \dot{p}(t) \mid + w_{\mathcal{B}_j}$. $w_{\mathcal{A}}$ est la moitié de la taille de la partie du chemin $\Upsilon(t)$ occupé par $\mathcal{A}$, c_0 et c_1 sont deux constantes positives, c_0 représente une marge de sécurité statique.

Etant donné un état initial $s = (\Upsilon_s, p_s, \dot{p}_s)$ et un état final à atteindre $g = (\Upsilon_g, p_g, \dot{p}_g)$, une trajectoire Γ est une **solution** à ce problème si et seulement si :

1. $\Upsilon(0) = \Upsilon_s$, $p(0) = p_s$, $\dot{p}(0) = \dot{p}_s$.

2. $\exists t_\Gamma \in [0,\ t_{max}]$ tel que $\Upsilon(t_\Gamma) = \Upsilon_g$, $p(t_\Gamma) = p_g$ et $\dot{p}(t_\Gamma) = \dot{p}_g$.

3. $\forall t \in [0,\ t_\Gamma], -\ddot{p}_{max} \leq \ddot{p}(t) \leq \ddot{p}_{max}$ et $0 \leq \dot{p}(t) \leq \dot{p}_{max}$.

4. $\forall t \in [0,\ t_\Gamma], \Upsilon(t) \in \{\lambda,\ \lambda = 0, \ldots, 2l - 2\}$.

[1] Ce qui autorise les erreurs de suivi lors de l'exécution de la trajectoire.

5. Γ est sûre.

Naturellement, on recherche la **solution optimale**, i.e. la solution Γ dont la durée t_Γ soit minimum.

3.3 L'algorithme de planification

L'algorithme présenté dans ce mémoire traite le cas où tous les chemins sont rectilignes. Les chemins réels et fictifs Υ_i ont donc la même longueur p_{max} et sont disposés dans $\mathcal{W}$ de la manière illustrée par la figure 3.3.

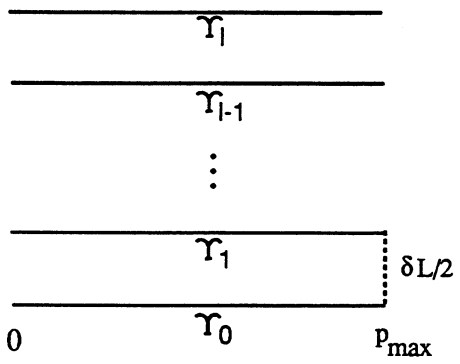


Figure 3.3 : les chemins rectilignes Υ_i, $i = 0, \ldots, l$

3.3.1 Le principe général

Dans la section précédente, nous avons défini l'ensemble des solutions de notre problème de planification de trajectoire pour $\mathcal{A}$. La complexité intrinsèque de la planification de trajectoire et la contrainte d'horizon nous ont conduit à choisir une méthode de résolution approchée. La solution que l'on cherche n'est pas la solution optimale dans l'ensemble des trajectoires solution définies précédemment mais la solution optimale dans un sous-ensemble de cet ensemble. Ce sous-ensemble est défini en discrétisant le temps —on choisit un pas de temps τ— et en choisissant l'accélération appliquée à $\mathcal{A}$ dans l'ensemble discret de valeurs $\{-\ddot{p}_{max},\ 0,\ \ddot{p}_{max}\}$. Ainsi le sous-ensemble des trajectoires que l'on considère contient toutes les trajectoires qui vérifient les contraintes suivantes :

- $\ddot{p}$ est constante par morceaux avec $\ddot{p}(t) \in \{-\ddot{p}_{max},\ 0,\ \ddot{p}_{max}\}$.
- $\Upsilon(t)$ et $\ddot{p}(t)$ ne changent de valeur qu'aux instants $t = k\tau$ pour un entier $k \geq 0$.

Une telle trajectoire est dite **bang** du fait de sa ressemblance avec les trajectoires dites 'bang-bang' de la théorie du contrôle. Le problème à résoudre consiste donc à déterminer la solution bang optimale. La complexité de ce problème va naturellement dépendre du nombre de trajectoires bangs possibles ; nombre qui dépend directement de la taille de τ, plus τ est petit et plus il existe de trajectoires bangs. D'autre part, on sent intuitivement[2] que la solution optimale bang sera d'autant plus proche de la véritable solution optimale que τ sera petit. Ainsi il est possible de mettre en balance le temps d'exécution de l'algorithme et la qualité de la solution.

Dans la section suivante, nous montrons comment transformer le problème de recherche de la solution bang optimale en un problème de recherche d'un plus court chemin dans un graphe orienté.

3.3.2 Le graphe états-temps

Un état de $\mathcal{A}$ a été défini par le triplet $(\Upsilon, p, \dot{p})$. Un **état-temps** de $\mathcal{A}$ est défini en ajoutant explicitement la dimension temporelle à un état de $\mathcal{A}$. Soit $\mathcal{TS}$ l'ensemble des états-temps, un point de $\mathcal{TS}$ est un quadruplet $s = (\Upsilon, p, \dot{p}, t)$ ou de manière équivalente $s(t) = (\Upsilon(t), p(t), \dot{p}(t))$.

Soit $s(k\tau) = (\Upsilon(k\tau), p(k\tau), \dot{p}(k\tau))$ un état-temps de $\mathcal{A}$ et soit $s((k+1)\tau)$ l'un des états-temps que $\mathcal{A}$ peut atteindre par une trajectoire bang de durée τ. $s((k+1)\tau)$ est obtenu en appliquant une accélération $\ddot{p} \in \{-\ddot{p}_{max}, 0, \ddot{p}_{max}\}$ à $\mathcal{A}$ pendant la durée τ. D'autre part, $\mathcal{A}$ peut soit rester sur son chemin courant soit effectuer un changement de chemin vers un chemin adjacent. Ainsi nous avons :

$$\begin{aligned}
\Upsilon((k+1)\tau) &= \Upsilon(k\tau) + \sigma \text{ avec } \sigma \in \{-1, 0, 1\} \\
p((k+1)\tau) &= p(k\tau) + \dot{p}(k\tau)\tau + \frac{1}{2}\ddot{p}\tau^2 \\
\dot{p}((k+1)\tau) &= \dot{p}(k\tau) + \ddot{p}\tau
\end{aligned}$$

Nous supposons que pendant l'intervalle $]k\tau, (k+1)\tau]$, $\mathcal{A}$ se trouve sur le chemin $\Upsilon((k+1)\tau)$. Par analogie avec [13], la trajectoire entre $s(k\tau)$ et $s((k+1)\tau)$ est appelée un $(\sigma, \ddot{p}, \tau)$-*bang*. Une trajectoire bang se compose donc d'une séquence de $(\sigma, \ddot{p}, \tau)$-bangs.

Soit $s(m\tau)$, $m \geq k$, un état-temps atteignable depuis $s(k\tau)$. Dans l'hypothèse où $\dot{p}(k\tau)$ est multiple de $\ddot{p}_{max}\tau$, on montre facilement que :

$$\begin{aligned}
\Upsilon(m\tau) &= \Upsilon(k\tau) + \alpha_1\sigma \\
p(m\tau) &= p(k\tau) + \alpha_2\frac{1}{2}\ddot{p}_{max}\tau^2
\end{aligned}$$

[2] Cette intuition se trouve d'ailleurs confirmée dans [13] où il est montré que, pour un choix de τ suffisamment petit, toute trajectoire sûre peut être approchée à ϵ près par une trajectoire bang sûre.

$$\dot{p}(m\tau) = \dot{p}(k\tau) + \alpha_3 \ddot{p}_{max}\tau$$

avec α_1, α_2 et α_3 entiers. Ainsi tous les états-temps atteignables depuis un état-temps donné sont disposés sur une grille régulière de $\mathcal{TS}$. Les espacements de cette grille sont de τ dans la dimension temporelle, de $\frac{1}{2}\ddot{p}_{max}\tau^2$ en position, de $\ddot{p}_{max}\tau$ en vitesse et de 1 pour les chemins.

En conséquence, il est possible de définir un graphe orienté $\mathcal{G}$ dans $\mathcal{TS}$. Les nœuds de $\mathcal{G}$ sont les points de la grille, i.e. des états-temps, et les arêtes sont les $(\sigma, \ddot{p}, \tau)$-bangs entre deux nœuds de cette grille. Ces $(\sigma, \ddot{p}, \tau)$-bangs doivent vérifier la condition (3.2) sur la vitesse et doivent être sûrs (ce point est détaillé dans la section 3.3.3.3). $\mathcal{G}$ est appelé **graphe des états-temps**, il est illustré par la figure 3.4 qui représente l'espace 'temps×position×vitesse' de deux chemins adjacents Υ_i et Υ_{i+1}. Par souci de clarté, nous n'avons représenté qu'un nœud et ses successeurs sur chacun des chemins Υ_i et Υ_{i+1} (un nœud a au plus trois successeurs par chemin). Soit A ce nœud, les états-temps atteignables depuis A par un $(\sigma, \ddot{p}, \tau)$-bang sont des points de la grille définie dans $\mathcal{TS}$, i.e. des nœuds de $\mathcal{G}$. Une arête entre A et l'un de ses successeurs représente le $(\sigma, \ddot{p}, \tau)$-bang correspondant. Une séquence d'arêtes entre deux nœuds définit une trajectoire bang. La durée d'une telle trajectoire est trivialement égale à τ fois le nombre d'arêtes dans la trajectoire. Ainsi le plus court chemin entre deux nœuds est la solution optimale entre ces deux nœuds.

Soit $s = (\Upsilon_s, p_s, \dot{p}_s)$ et $g = (\Upsilon_g, p_g, \dot{p}_g)$ l'état initial et final de $\mathcal{A}$. Sans perte de généralité, on suppose que l'état-temps $s^* = (\Upsilon_s, p_s, \dot{p}_s, 0)$ et l'ensemble des états-temps $G^* = \{(\Upsilon_g, p_g, \dot{p}_g, k\tau)$ avec $k = 0, \ldots, \lfloor \frac{t_{max}}{\tau} \rfloor\}$ appartiennent à la grille définie dans $\mathcal{TS}$. Dans ce cas, la recherche de la solution bang optimale est équivalente à la recherche d'un plus court chemin dans $\mathcal{G}$ entre s^* et un nœud de G^*. La section suivante décrit l'algorithme de recherche de cette solution.

Remarque: dans la mesure où l'on considère une région compacte de $\mathcal{TS}$, le nombre de points de la grille définie dans $\mathcal{TS}$ est fini. $\mathcal{G}$ est donc fini et la recherche pour la solution bang optimale se fait en un temps fini. □

3.3.3 L'exploration du graphe états-temps

3.3.3.1 L'algorithme

L'algorithme utilisé pour explorer $\mathcal{G}$ est du type A^* [62]. Ce type d'algorithme est intéressant dans la mesure où seule la partie de $\mathcal{G}$ qui est utile est engendrée lors de la recherche. L'algorithme construit un arbre de recherche qui recouvre progressivement $\mathcal{G}$. s^* est le nœud courant initial (racine de l'arbre de recherche). Le nœud courant est développé, i.e. on détermine tous ses successeurs. Puis on choisit le meilleur successeur (meilleur dans un sens que nous préciserons par la suite) qui devient alors le nouveau nœud courant. Ce processus est répété jusqu'à ce que l'un des nœuds buts soit atteint où bien jusqu'à ce que le graphe

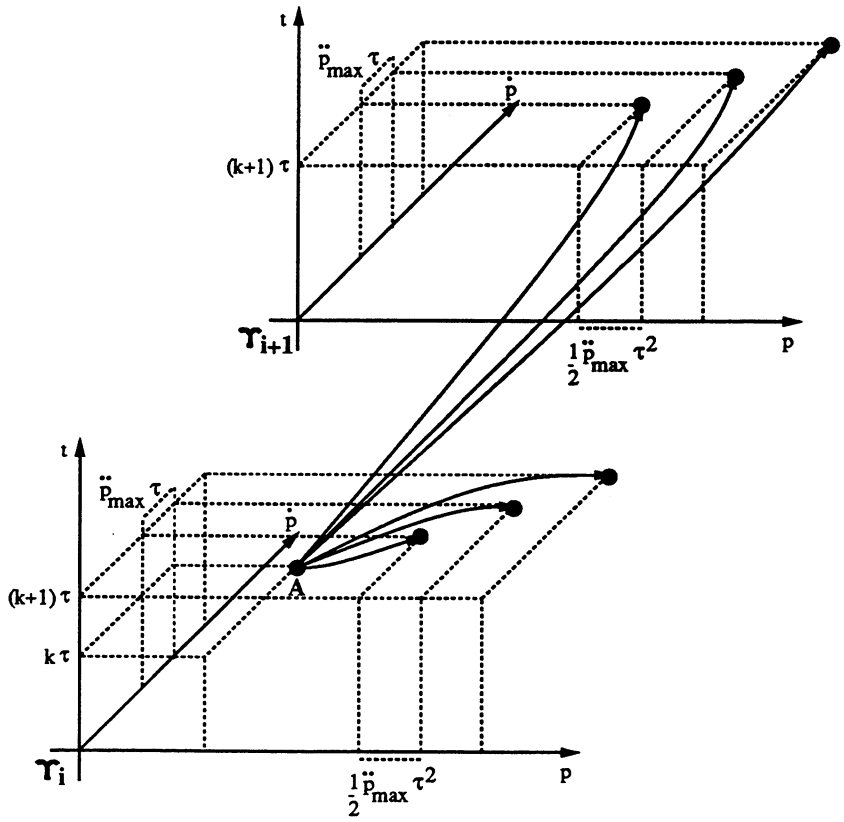


Figure 3.4 : le graphe $\mathcal{G}$

complet ait été exploré. La solution est retournée en utilisant des pointeurs. Cet algorithme est détaillé ci-après :

Algorithme

L'algorithme utilise une liste triée OPEN qui contient les feuilles de l'arbre de recherche qui n'ont pas encore été développée. elles sont triées par valeur croissante d'une certaine fonction de coût. Les nœuds déjà traités sont stockés dans un arbre équilibré CLOSE [88].

1. Initialisation de OPEN à s^* et de CLOSE à $\emptyset$.
2. Si OPEN est vide alors échec.
3. La tête de OPEN devient le nouveau nœud courant. Il est alors enlevé de OPEN et inséré dans CLOSE.
4. Calcul des nœuds atteignables depuis le nœud courant par des $(\sigma, \ddot{p}, \tau)$-bangs vérifiant les contraintes de vitesse et de sûreté. Un pointeur est établi entre chaque successeur et le nœud courant.

5. Pour chaque successeur du nœud courant, vérifier s'il appartient à G^*. Si oui alors succès, la solution bang optimale est retournée en utilisant les pointeurs. Si non et si le successeur n'appartient pas à CLOSE alors il est inséré dans OPEN en fonction de sont coût.

6. Aller à 2

Les deux sections suivantes détaillent deux points clés de l'algorithme à savoir la fonction de coût associée à chaque nœud et le développement d'un nœud.

3.3.3.2 La fonction de coût

A^* associe un certain coût $f(s)$ à chaque nœud s de $\mathcal{G}$, coût qui est utilisé afin de choisir le meilleur successeur d'un nœud. Puisque l'on recherche une trajectoire optimale en temps, $f(s)$ est une estimation de la durée de la trajectoire bang optimale qui relie s^* à un nœud de G^* en passant par s. $f(s)$ est définie de manière classique comme étant la somme de deux termes $g(s)$ et $h(s)$:

- $g(s)$ est la durée de la trajectoire bang entre s^* et s i.e. la composante temporelle de s.

- $h(s)$ est l'estimation de la trajectoire optimale entre s et un élément de G^* i.e. le temps qu'il faudrait à $\mathcal{A}$ pour atteindre g depuis son état courant avec une trajectoire 'bang-coast-bang'[3] sans tenir compte des obstacles. Dans le cas où une telle trajectoire n'existe pas, $h(s) = +\infty$.

La fonction heuristique $h(s)$ est trivialement admissible. L'algorithme est donc assuré de trouver la solution bang optimale si elle existe [62]. $h(s)$ est, en outre, monotone, ce qui améliore l'efficacité de l'algorithme.

Remarque : La définition de $f(s)$ donnée ci-dessus ne dépend pas de la composante 'chemin' de s. En conséquence, l'algorithme peut trouver une solution qui comporte de nombreux changements de chemins inopportuns. Pour remédier à ce problème, il suffit de modifier légèrement la définition de $f(s)$ comme suit :

$$f^*(s) = (f(s), \mid \Upsilon_g - \Upsilon \mid)$$

avec Υ le chemin de s. Autrement dit, si deux nœuds ont le même coût $f(s)$, alors le meilleur est celui dont le chemin est le plus proche du chemin but Υ_g. □

[3] I.e. accélération maximum, accélération nulle et décélération maximum.

3.3.3.3 Le développement d'un nœud

Les successeurs d'un nœud donné s de $\mathcal{G}$ sont les nœuds qui sont atteignables depuis s par un $(\sigma, \ddot{p}, \tau)$-bang avec $\sigma \in \{-1, 0, 1\}$ et $\ddot{p} \in \{-\ddot{p}_{max}, 0, \ddot{p}_{max}\}$. En conséquence, un nœud a au plus neuf successeurs. Cependant, ce nombre se réduit à trois lorsque $\mathcal{A}$ est en train d'effectuer un changement de chemin (en effet, $\mathcal{A}$ doit rester sur le chemin fictif pendant toute la durée Δt du changement de chemin).

Etant donné le $(\sigma, \ddot{p}, \tau)$-bang entre $s(k\tau)$ et $s((k+1)\tau)$, $\forall t \in [k\tau, (k+1)\tau]$, nous avons :

$$p(t) = p(k\tau) + \dot{p}(k\tau)(t - k\tau) + \frac{1}{2}\ddot{p}(t - k\tau)^2 \quad (3.3)$$

$$\dot{p}(t) = \dot{p}(k\tau) + \ddot{p}(t - k\tau) \quad (3.4)$$

Ce $(\sigma, \ddot{p}, \tau)$-bang est valide si (a) il est sûr et (b) s'il vérifie la condition (3.2) sur la vitesse. La contrainte de vitesse est testée aisément à l'aide de l'équation (3.4). En ce qui concerne la sûreté, il faut vérifier que $\forall t \in [k\tau, (k+1)\tau]$, la distance entre $p(t)$ et chaque obstacle $\mathcal{B}_j$ qui se trouve sur le chemin $\Upsilon((k+1)\tau)$ n'est jamais inférieure à $\delta_{\mathcal{B}_j}(t)$. En pratique, la sûreté de la position $p(t)$ est testée en grossissant $\mathcal{B}_j$ de $\delta_{\mathcal{B}_j}(t)$ puis en vérifiant que $p(t)$ ne coupe pas l'obstacle ainsi grossi (figure 3.5).

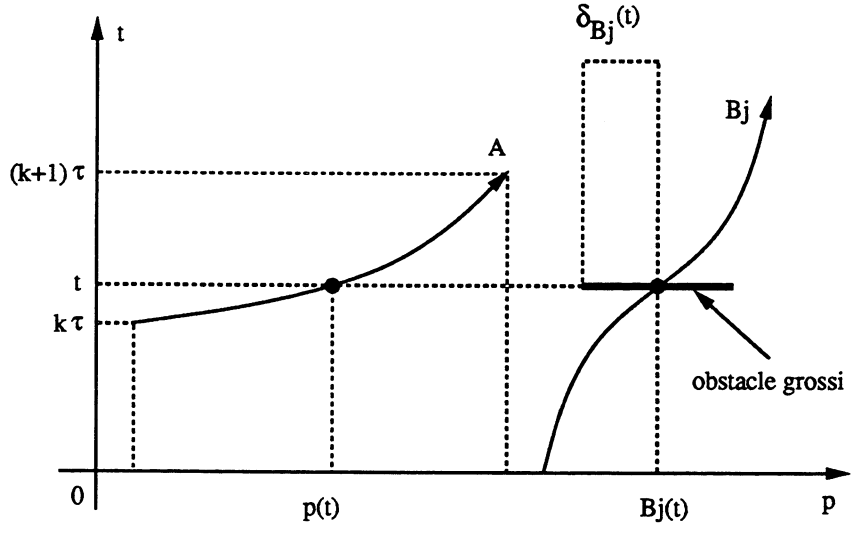


Figure 3.5 : le test de sûreté

Pour des raisons d'efficacité, le contrôle de sûreté dans l'intervalle $[k\tau, (k+1)\tau]$ entre $\mathcal{A}$ et un obstacle $\mathcal{B}_j$ est effectué en utilisant la marge de sécurité correspondant à la vitesse maximum atteinte par $\mathcal{A}$ dans cet intervalle. En d'autres termes, on vérifie que $\forall t \in [k\tau, (k+1)\tau]$, la distance entre $p(t)$ et $\mathcal{B}_j$(t) n'est

jamais inférieure à $\delta_{\mathcal{B}_j} = w_{\mathcal{A}} + c_0 + c_1 \mid \max(\dot{p}(k\tau), \dot{p}((k+1)\tau)) \mid + w_{\mathcal{B}_j}$. Ce choix permet de ne grossir $\mathcal{B}_j$ que d'un nombre fini de valeurs $\delta^{\alpha}_{\mathcal{B}_j}$ correspondant aux différentes valeurs de vitesses que peut prendre $\mathcal{A}$ aux instants $k\tau$, i.e. $\delta^{\alpha}_{\mathcal{B}_j} = w_{\mathcal{A}} + c_0 + \alpha c_1(\ddot{p}_{max}\tau) + w_{\mathcal{B}_j}$ avec $\alpha = 0, \ldots, \lfloor \frac{\dot{p}_{max}}{\ddot{p}_{max}\tau} \rfloor$. Par conséquent, en grossissant chaque $\mathcal{B}_j$ des différentes valeurs $\delta^{\alpha}_{\mathcal{B}_j}$ lors une phase de pré-traitement, il devient possible de tester la sûreté d'un $(\sigma, \ddot{p}, \tau)$-bang en un temps $O(n)$ où n est le nombre total d'obstacles.

3.4 Implantation et résultats expérimentaux

L'algorithme de planification de trajectoire présenté dans la section précédente a fait l'objet d'une implantation. Un prototype, développé en C, nous a permis de tester une version simplifiée de l'algorithme. Dans l'implantation courante, la marge de sécurité associée aux obstacles n'est pas dépendante de la vitesse de $\mathcal{A}$, elle est constante et de valeur δ. D'autre part, on suppose qu'un changement de chemin se fait à vitesse constante. Par conséquent, un nœud de $\mathcal{G}$ a au plus cinq successeurs (trois successeurs sur le même chemin mais seulement un pour chaque chemin adjacent). Ces restrictions ne remettent pas en cause le fonctionnement global de l'algorithme. Dans les deux sections suivantes, nous détaillons l'implantation de la procédure de test de sûreté d'un $(\sigma, \ddot{p}, \tau)$-bang ainsi que de la procédure de calcul de l'intervalle de temps Δt nécessaire pour effectuer un changement de chemin. Enfin, nous présentons quelques résultats expérimentaux.

3.4.1 Le test de sûreté

La marge de sécurité δ associée aux obstacles étant constantes, chaque obstacle $\mathcal{B}_j$ ne doit être grossi que de la seule valeur δ. On va utiliser un bitmap[4] 'temps×position' pour représenter la position des différents obstacles grossis sur l'intervalle de temps $[0, t_{max}]$. Il y a un bitmap de ce type pour chaque chemin (figure 3.6).

Ces bitmaps permettent de vérifier très facilement la sûreté d'un $(\sigma, \ddot{p}, \tau)$-bang donné. Tout d'abord, le $(\sigma, \ddot{p}, \tau)$-bang est discrétisé en fonction de la résolution du bitmap, puis, chaque point discret $(t, p(t))$ ainsi obtenu est testé pour la collision sur le bitmap correspondant.

3.4.2 Le changement de chemin

Ainsi que nous l'avons mentionné dans la section 3.2.4.1, la forme de la trajectoire décrite par $\mathcal{A}$ durant un changement de chemin entre deux chemins Υ_i et Υ_{i+2} et

[4] I.e. une matrice d'éléments binaires.

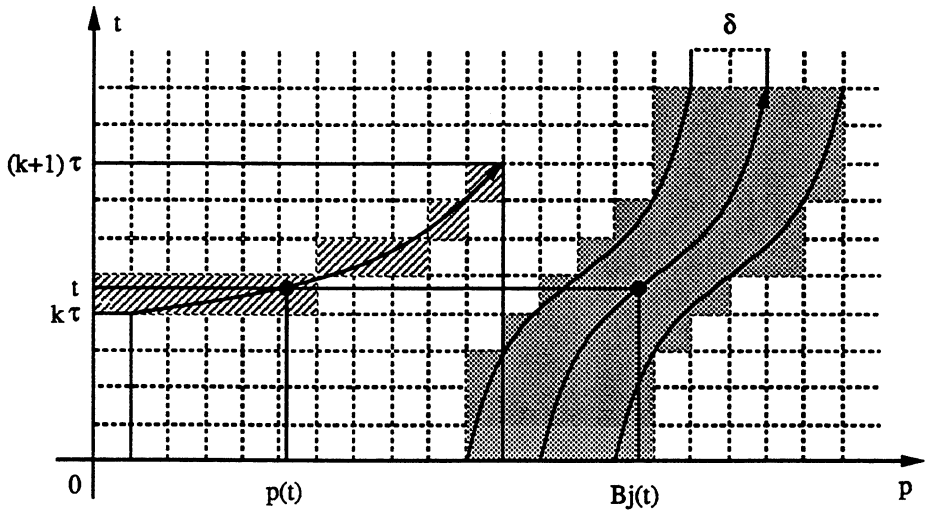


Figure 3.6 : le bitmap associé à un chemin

la durée Δt du changement de chemin dépendent toutes deux des caractéristiques de $\mathcal{A}$, de Υ_i et de Υ_{i+2}.

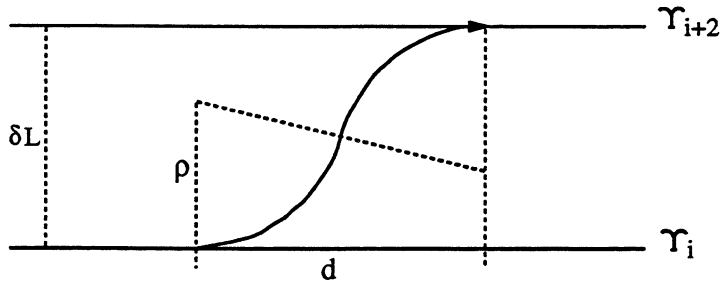


Figure 3.7 : la trajectoire décrite par $\mathcal{A}$ durant un changement de chemin

Dans l'implantation courante, la trajectoire décrite par $\mathcal{A}$ durant un changement de chemin est modélisée de façon simplifiée par deux arcs de cercle tangents de rayon ρ tels qu'illustrés par la figure 3.7. ρ est le rayon de giration minimum de $\mathcal{A}$. Nous avons vu dans la section 2.2.1.2 que ρ est limité, de façon statique, pour des raisons mécaniques. Cependant, lorsque $\mathcal{A}$ est en mouvement, il est soumis à une certaine accélération centrifuge qui ne doit pas dépasser un certain seuil g_{max}. Cette contrainte dynamique est plus forte que la contrainte statique, elle a pour effet de limiter ρ de la façon suivante :

$$\rho \geq \frac{\dot{p}^2}{g_{max}}$$

avec $\dot{p}$ la vitesse instantanée de $\mathcal{A}$. Par conséquent, si l'on connaît la valeur de g_{max} et la vitesse $\dot{p}$ de $\mathcal{A}$ au début du changement de chemin alors il est possible

de calculer ρ et partant, la distance d que $\mathcal{A}$ va parcourir le long de Υ_{i+1} :

$$\rho = \frac{\dot{p}^2}{g_{max}} \text{ et } d = \sqrt{\delta L(4\rho - \delta L)}$$

Connaissant d, il devient possible d'estimer le nombre de pas de temps τ nécessaire pour effectuer le changement de chemin. Ce nombre de pas de temps est trivialement égal à $\lceil \frac{d}{p_\tau} \rceil$.

3.4.3 Résultats expérimentaux

L'algorithme a été testé sur des exemples comportant jusqu'à quatre chemins réels (soit sept chemins au total). Pour les tests, les obstacles $\mathcal{B}_j$ sont engendrés de manière aléatoire ce qui explique qu'ils peuvent entrer en collision les uns avec les autres. En outre, on suppose qu'ils gardent une vitesse constante.

Deux exemples de planification de trajectoire comportant deux chemins réels sont présentés dans les figures 3.8 et 3.9. Dans chaque cas, deux fenêtres sont associées à chaque chemin réel : une fenêtre de trace qui montre la partie du graphe $\mathcal{G}$ qui a été effectivement exploré et une fenêtre résultat qui montre la trajectoire finale (il n'y a pas de fenêtre associé au chemin fictif). Chacune de ces fenêtres représente l'espace 'temps×position' du chemin correspondant (l'axe des positions est horizontal et l'axe des temps vertical. L'origine du repère est situé dans le coin supérieur gauche). Les segments noirs épais représentent la trace laissés par les obstacles $\mathcal{B}$j tandis que les points noirs représentent les points de la grille sous-jacente au graphe $\mathcal{G}$. On peut noter que l'espacement vertical entre les points correspond au pas de temps τ.

Dans les deux exemples, $\mathcal{A}$ part du premier chemin (chemin #0) à la position 0 (coin supérieur gauche) et avec une vitesse nulle. Son but est d'atteindre le premier chemin à la position p_{max} (bord droit de la fenêtre) avec une vitesse nulle. $\mathcal{A}$ peut dépasser des obstacles en utilisant le second chemin (chemin #1). Dans le but d'obtenir des résultats significatifs, nous avons initialiser les différentes variables du problème avec des valeurs en rapport avec l'application considérée :

$$\begin{aligned} p_{max} &= 500\text{m} \\ \delta L &= 4\text{m} \\ \dot{p}_{max} &= 72\text{km/h} \\ \ddot{p}_{max} &= 1\text{m/s}^2 \\ t_{max} &= 20\text{s} \end{aligned}$$

C'est le choix du pas de temps τ qui va déterminer le temps d'exécution moyen de l'algorithme. Pour $\tau = 1s.$, nous avons obtenu des temps d'exécution de l'ordre de la seconde.

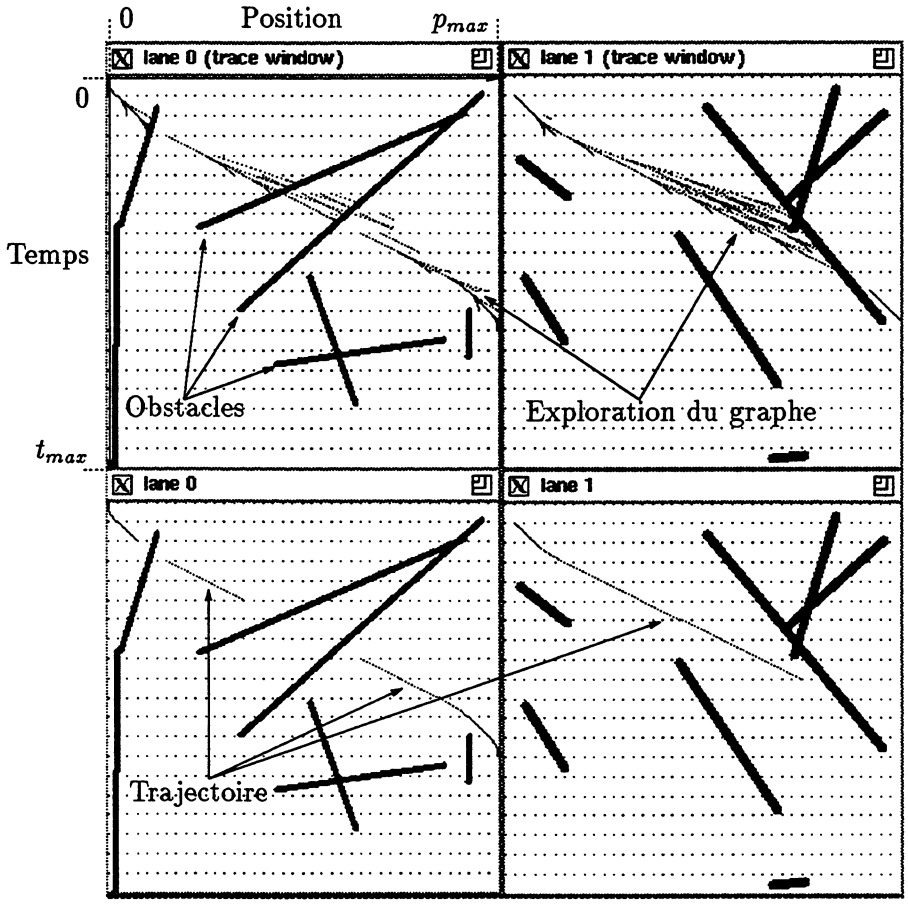


Figure 3.8 :

3.5 Conclusion

Dans ce chapitre, nous avons présenté une technique de planification de trajectoire sans collision pour un mobile de type voiture dont la vitesse et l'accélération sont limitées. La technique présentée s'applique à un espace de travail dynamique. Etant donné un ensemble de chemins adjacents dont l'un amène le mobile à son but, et sachant que le mobile peut passer d'un chemin à un autre, on détermine la trajectoire du mobile le long de ces chemins afin d'éviter toute collision avec les obstacles mobiles de l'espace de travail tout en respectant les contraintes dynamiques du mobile (vitesse et accélération limitée). L'algorithme présenté explore *l'espace-temps des états* du mobile qui est obtenu en rajoutant la dimension temporelle à l'espace des états du mobile. Cet espace des états-temps est au préalable discrétisé. La discrétisation est obtenue en définissant un pas de temps τ et en choisissant les accélérations appliquées au mobile parmi un ensemble fini de va-

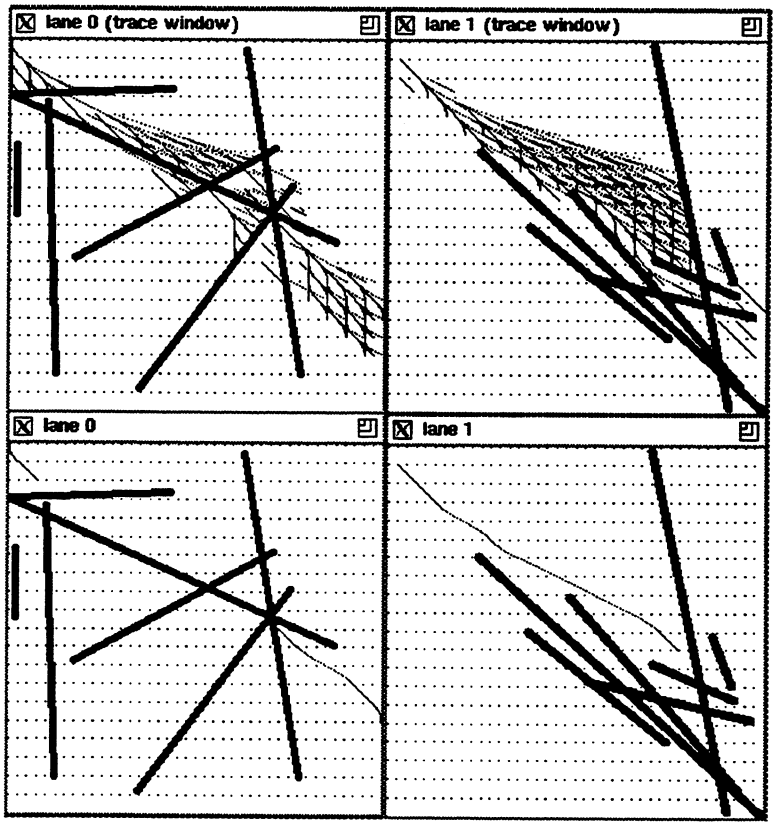


Figure 3.9 :

leurs. Le choix de τ détermine le nombre de points dans l'espace des états-temps discrétisé et partant, le temps d'exécution moyen de l'algorithme.

Le travail présenté dans ce chapitre en est encore à un stade préliminaire, il y a encore place pour de nombreuses extensions et améliorations. En particulier, les points suivants devraient être abordé :

- L'algorithme de planification de trajectoire présenté a fait l'objet d'une implantation simplifiée. Il faudrait compléter cette implantation pour intégrer tous les aspects présentés dans ce chapitre (chemins non-rectilignes, marge de sécurité dépendante de la vitesse, etc.).

- Du fait de la discrétisation de l'espace des états-temps, la trajectoire solution trouvée est sous-optimale, il serait intéressant de trouver la relation qui existe entre τ et la qualité de la solution trouvée.

- Généralisation de la méthode pour prendre en compte des contraintes véritablement dynamiques, i.e. qui dépendent de l'état courant du mobile.

Chapitre 4

Un système de planification de mouvement

Etant donné un mobile $\mathcal{A}$ qui se déplace dans un espace de travail $\mathcal{W}$, les deux chapitres précédents ont exposé respectivement une technique de planification de chemin, qui prend en compte les obstacles fixes de $\mathcal{W}$ et les contraintes cinématiques de $\mathcal{A}$, puis une technique de planification de trajectoire, qui prend en compte les obstacles mobiles de $\mathcal{W}$ et les contraintes dynamiques de $\mathcal{A}$. Dans ce chapitre, nous montrons comment combiner ces deux techniques complémentaires pour apporter une solution efficace au problème qui est le notre. i.e. la planification des mouvements d'un mobile soumis à des contraintes cinématiques et dynamiques et se déplaçant dans un espace de travail dynamique. Nous illustrons cette combinaison par la présentation d'un planificateur de mouvement pour une automobile se déplaçant dans le réseau routier. Ce planificateur, développé dans le cadre du projet européen EUREKA PROMETHEUS, est destiné à faire partie d'un système de navigation embarqué sur une automobile. Ce choix de présentation ne limite pas la portée de notre système de planification de mouvement au seul cas d'une automobile dans le réseau routier. Il peut être utilisé dans tout espace de travail structurable.

4.1 Présentation du projet PROMETHEUS

Le projet européen EUREKA PROMETHEUS est un programme de recherche qui a été lancé en octobre 1986 par les principaux constructeurs automobiles européens. Son but est d'étudier les possibilités d'amélioration du transport routier en Europe dans ses aspects sécurité, efficacité, économie, confort et impact sur l'environnement. Ces améliorations passent par la réalisation de fonctions très diverses concernant à la fois le véhicule mais aussi toute l'infrastructure du réseau routier. Au niveau d'un véhicule, ces améliorations se traduisent par une assistance au conducteur, assistance qui peut être passive (apport d'informations) ou

active (prise en charge partielle ou totale du véhicule). Le document [65] décrit très précisément les différents objectifs du projet PROMETHEUS.

Le LIFIA intervient au sein du pôle 'co-pilote' du sous-projet Pro-Art, l'un des sous-projets thématiques de PROMETHEUS. Notre objectif global est la mise au point d'un module de 'co-pilotage de manœuvres' qui doit, à court terme, assister le conducteur lors de la réalisation de certaines manœuvres (traversée d'une intersection, insertion dans le flux autoroutier, etc.). A plus long terme, ce module doit pouvoir se substituer au conducteur et prendre totalement en charge le véhicule, celui-ci devenant alors autonome (du moins durant la réalisation des manœuvres considérées).

Dans les deux cas, assistance au conducteur ou prise en charge, il est nécessaire de planifier la manœuvre, i.e. le mouvement que l'on désire effectuer. Ce mouvement doit être calculé avant l'exécution de la manœuvre désirée à partir d'informations a priori sur les obstacles fixes et mobiles présents dans l'espace de travail du véhicule. La section suivante présente l'architecture d'un planificateur de mouvement pour un tel véhicule.

4.2 Le planificateur de mouvement

4.2.1 Architecture générale

L'architecture générale du planificateur de mouvement est illustrée dans la figure 4.1. Les modules principaux de ce planificateur sont les modules de planification de chemin et de trajectoire correspondants aux deux techniques présentées dans les chapitres précédents. Cependant, en vue d'obtenir un système de planification opérationnel, ces deux modules sont complétés par deux modules secondaires dits de *planification d'itinéraire* et de *planification de chemins adjacents*:

- La planification d'itinéraire: étant donnée la structure de $\mathcal{W}$, on détermine le squelette $\mathcal{S}$ du chemin de $\mathcal{A}$, i.e. la séquence de voies que doit suivre $\mathcal{A}$ pour atteindre son but.

- La planification de chemins adjacents: étant donné le chemin Υ calculé par la planification de chemin, on détermine l'ensemble des chemins Υ_i adjacents à Υ que $\mathcal{A}$ peut emprunter, via un changement de chemin, pour atteindre son but.

Nous allons détailler le rôle de chacun de ces modules dans les sections 4.2.3 et 4.2.4. Auparavant, nous passons brièvement en revue les différentes entrées du système de planification.

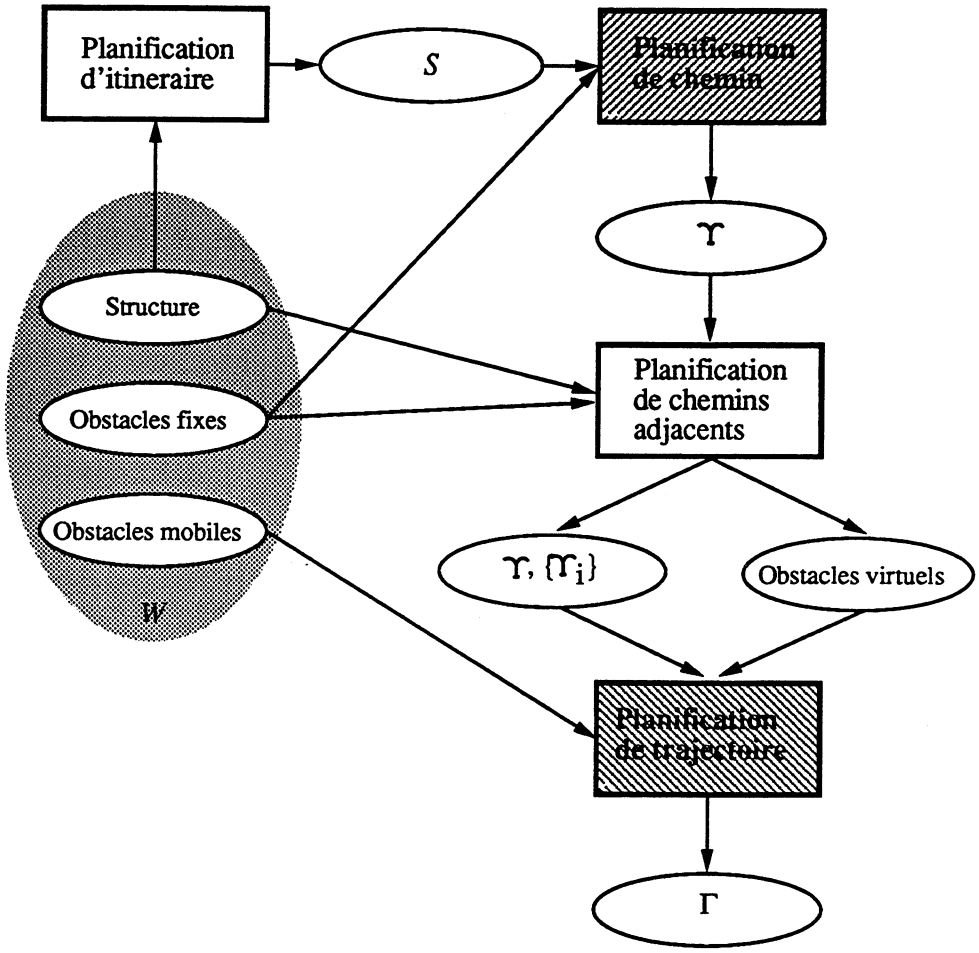


Figure 4.1 : architecture du planificateur de mouvement

4.2.2 Les entrées du système

Les informations nécessaires au planificateur de mouvement sont toutes les informations relatives à l'espace de travail $\mathcal{W}$ dans lequel évolue $\mathcal{A}$. Ces informations sont de trois types et concernent respectivement :

- Les obstacles fixes : forme et position.
- La structure de $\mathcal{W}$: géométrie et topologie des voies. Cette structure est modélisée par un graphe dont les nœuds représentent les voies et dont les arêtes représentent les relations topologiques qui existent entre les voies (adjacence ou croisement).
- Les obstacles mobiles : forme et position sur la durée de l'horizon temporel.

Dans le cadre de PROMETHEUS, il est prévu que ces informations soient issues du traitement des données obtenues au moyen d'un ensemble de capteurs

installés à bord de $\mathcal{A}$. La planification de mouvement doit être précédée d'une estimation du comportement des obstacles mobiles.. L'estimation du mouvement d'un obstacle mobile doit s'appuyer, d'une part, sur la position, la vitesse et l'accélération de cet obstacle au début de l'horizon, mais aussi, sur la nature de l'obstacle (véhicule, piéton, etc.) et sur certaines informations symboliques, e.g. présence du clignotant, etc. L'horizon temporel correspond alors à la durée de validité des hypothèses que l'on fait sur le comportement des obstacles mobiles.

4.2.3 La planification d'itinéraire

Le rôle du module de planification d'itinéraire est de déterminer le squelette $\mathcal{S}$ du chemin de $\mathcal{A}$, i.e. la séquence de voies que doit suivre $\mathcal{A}$ pour atteindre son but. $\mathcal{S}$ s'obtient très facilement en explorant le graphe qui modélise la structure de $\mathcal{W}$.

4.2.4 La planification de chemins adjacents

Le module de planification de chemin transforme le squelette $\mathcal{S}$ en un chemin Υ. Le rôle du module de planification de chemins adjacents est de déterminer l'ensemble des chemins Υ_i adjacents à Υ que $\mathcal{A}$ peut emprunter, via un changement de chemin, pour atteindre son but.

Υ est une courbe de classe C^1 composé de segments et d'arcs de cercles. La définition de l'adjacence entre chemins donnée dans la section 3.2.1 permet de déterminer très facilement le chemin Υ_r (resp. Υ_l) qui est à droite (resp. à gauche) de Υ. En répétant le processus, on peut calculer le chemin adjacent qui se trouve à droite (resp. à gauche) de Υ_r (resp. Υ_l). C'est la structure de $\mathcal{W}$ qui indique le nombre maximum de chemins adjacents qu'il faut calculer à droite et à gauche de Υ.

Après avoir calculé un chemin Υ_a adjacent à Υ, il est nécessaire de vérifier si Υ_a est valide, i.e. s'il est sans collision avec les obstacles fixes de $\mathcal{W}$ et exécutable par $\mathcal{A}$ compte tenu de ses contraintes cinématiques. En effet, si ces deux propriétés sont vérifiées pour Υ, elles ne le sont pas forcément pour Υ_a. Il y a deux raisons possibles à cela :

- Υ_a n'est plus sans collision avec les obstacles fixes de $\mathcal{W}$.
- L'un des arcs de cercle qui compose Υ_a a un rayon strictement inférieur au rayon de giration minimum de $\mathcal{A}$.

Considérons le cas illustré dans la figure 4.2 : Υ est un chemin sans collision et exécutable pour un mobile ponctuel et de rayon de giration minimum ρ_{min}. Le chemin Υ_r qui se trouve à droite de Υ n'est plus valide pour les deux raisons précédentes. Plus précisément, l'intervalle $[p_1, p_2]$ de Υ_r ne respecte plus

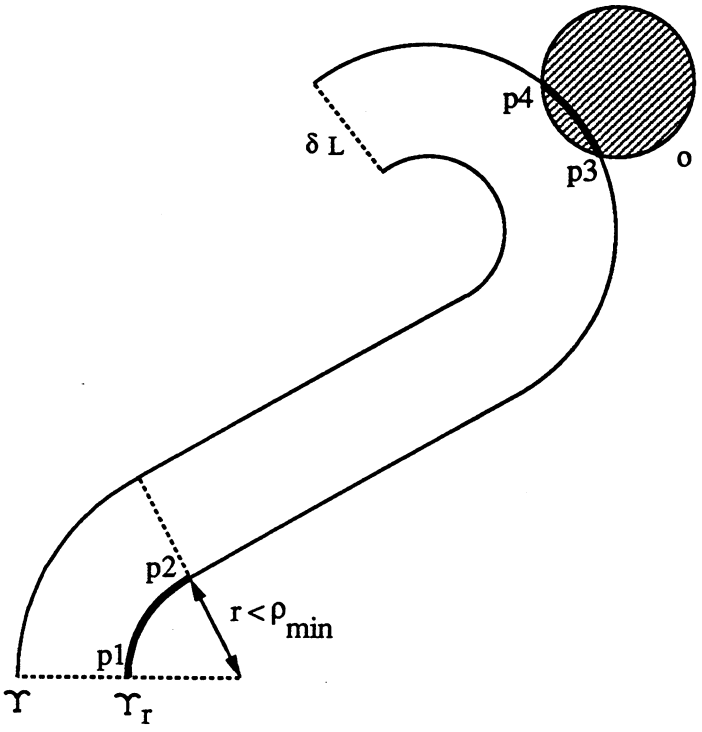


Figure 4.2: les obstacles virtuels

les contraintes cinématiques du mobile (rayon de giration minimum), tandis que l'intervalle $[p_3, p_4]$ n'est plus sans collision avec les obstacles fixes de $\mathcal{W}$..

Plutôt que de rejeter globalement un chemin adjacent qui n'est pas valide, nous n'allons invalider de ce chemin que les intervalles qui ne respectent pas les conditions de validité. De manière concrète, nous le faisons en introduisant la notion *d'obstacles virtuels.* Un obstacle virtuel est équivalent à un obstacle mobile, i.e. qu'il est modélisé par une particule ponctuelle $\mathcal{B}_j$ dont la position à l'instant t est caractérisé par le couple $(\Upsilon_{\mathcal{B}_j}(t), p_{\mathcal{B}_j}(t))$ avec $\Upsilon_{\mathcal{B}_j}(t)$ le chemin occupé par $\mathcal{B}_j$ et $p_{\mathcal{B}_j}(t)$, l'abscisse curviligne le long de ce chemin. Nous allons définir un obstacle virtuel pour chaque intervalle invalide des chemins adjacents. Un tel obstacle virtuel reste immobile et est positionné au milieu de son intervalle de définition. Son rôle est simplement d'empêcher que $\mathcal{A}$ emprunte la partie invalide du chemin correspondant. Dans l'exemple illustré par la figure 4.2, il y a deux obstacles virtuels associés au chemin Υ_r : le premier est positionné en $(p_2 - p_1)/2$ et le second en $(p_4 - p_3)/2$.

L'ensemble des obstacles virtuels est pris en compte par le module de planification de chemin au même titre que les obstacles mobiles qui se trouvent dans $\mathcal{W}$.

4.3 Planification et contrôle

Nous terminerons ce bref chapitre en insistant sur le caractère nominal de la trajectoire qui est calculée par le système de planification de mouvement que l'on vient de présenter. La trajectoire qui sera effectivement réalisée par le mobile $\mathcal{A}$ réel *ne sera pas* la trajectoire calculée par le planificateur. Il y a plusieurs raisons à cela. Il y a tout d'abord les raisons liées au mobile $\mathcal{A}$ réel :

- La position exacte de $\mathcal{A}$ ne peut être connue qu'avec une certaine précision.
- L'asservissement de $\mathcal{A}$ sur la trajectoire nominale est toujours entaché d'erreur.

Il y a ensuite une raison liée à la trajectoire nominale elle-même :

- Le chemin support de la trajectoire est composé d'une séquence de segments et d'arcs de cercle. La discontinuité de courbure qui survient lors du passage d'un segment à un arc de cercle oblige $\mathcal{A}$ à s'arrêter ce qui n'est pas envisageable dans notre cas. Pour suivre une telle trajectoire à vitesse non-nulle, $\mathcal{A}$ doit obligatoirement s'en écarter.

Enfin, il y a les raisons liées au fait qu'il y a toujours un écart entre le modèle de l'espace de travail qui est utilisé par le planificateur et l'espace de travail réel :

- Imprécision sur la forme et la position des obstacles.
- Imprécision sur le comportement des obstacles mobiles.
- Apparition d'obstacles imprévus.

On peut agir sur certains de ces points et les prendre en compte lors de la planification pour engendrer une trajectoire nominale qui est plus proche de la trajectoire effectivement réalisée, e.g. en planifiant une trajectoire sans discontinuité de courbure ou bien en grossissant les obstacles en fonction de l'incertitude en position qui leur est associé. Cependant, on ne peut pas résoudre tous ces problèmes au niveau de la planification (en particulier, on ne saurait prévoir l'imprévu !). Dans ces conditions. le *contrôle de l'exécution* de la trajectoire nominale est indispensable. La trajectoire nominale sert de base au *contrôleur d'exécution* qui peut être amené à la modifier si les circonstances l'exigent (risque de collision). Ce problème du contrôle d'exécution est aussi abordé par le LIFIA dans le cadre du projet PROMETHEUS [26, 25], cependant, nous ne l'abordons pas dans ce mémoire.

Conclusion

Dans ce mémoire, nous avons montré que le problème de la planification des mouvements d'un mobile $\mathcal{A}$ soumis à des contraintes cinématiques et dynamiques et se déplaçant dans un espace de travail $\mathcal{W}$ dynamique peut être résolu de façon efficace[1] lorsqu'il existe une *structuration de $\mathcal{W}$ naturelle pour $\mathcal{A}$*, i.e. lorsqu'il est possible de structurer $\mathcal{W}$ en un ensemble de *zones libres* à l'intérieur desquelles $\mathcal{A}$ peut se déplacer sans collision avec les obstacles fixes de $\mathcal{W}$. Dans ce cas, le problème considéré peut être abordé suivant deux directions complémentaires :

- La *planification de chemin* qui prend en compte les obstacles fixes de $\mathcal{W}$ et les contraintes cinématiques de $\mathcal{A}$.

- La *planification de trajectoire* qui prend en compte les obstacles mobiles de $\mathcal{W}$ et les contraintes dynamiques de $\mathcal{A}$.

Nous nous sommes placés dans le cadre d'une application réelle, i.e. la planification des mouvements d'une voiture dans le réseau routier. Les zones libres sont alors définies par les voies de circulation. Tout d'abord, nous avons présenté une technique de planification de chemin qui, à partir d'une séquence de voies, planifie un chemin sans manœuvre exécutable par une voiture. Cette technique, bien qu'un incomplète, donne de bons résultats en pratique. Ensuite, nous avons présenté une technique de planification de trajectoire qui détermine la trajectoire de $\mathcal{A}$ le long d'un chemin de référence. L'originalité de cette technique consiste à raisonner, non pas sur un chemin unique , mais sur une famille de chemins adjacents et à permettre à $\mathcal{A}$ de quitter son chemin de référence pour un autre chemin afin d'éviter des obstacles mobiles. Enfin, nous avons montré comment intégrer ces deux techniques au sein d'un système de planification de mouvement qui permet de résoudre efficacement le problème considéré.

A l'instar de tout travail de recherche, le travail présenté dans ce mémoire est, par nature, inachevé. En effet, il y toujours matière à recherches plus approfondies. En ce qui concerne les extensions qu'il nous semble nécessaire d'apporter à notre travail, nous ferons la distinction entre extensions 'locales' et 'globales'.

[1] En terme de temps de calcul.

Les extensions locales concernent les deux techniques de planification de chemin et de trajectoire. Nous les avons déjà présenté en conclusion des chapitres 2 et 3, nous ne les reprendrons pas ici.

Les extensions globales concernent l'ensemble de l'approche. Rappelons que notre objectif était de montrer que le problème de la planification de mouvement pour un mobile $\mathcal{A}$ dans un espace de travail $\mathcal{W}$ dynamique pouvait être résolu efficacement en s'appuyant sur une structuration de $\mathcal{W}$ et en abordant le problème suivant les deux directions complémentaires indiquées précédemment. Cet objectif a été atteint pour une voiture dans le réseau routier. Il faut maintenant considérer d'autres types de mobile et d'autres types d'espace de travail. Une telle extension pose immédiatement le problème de la structuration de $\mathcal{W}$. Dans le cas du réseau routier, ce problème ne se pose pas, En effet, le réseau routier est naturellement structuré. Considérons maintenant le cas d'un mobile $\mathcal{A}$ quelconque placé dans un espace de travail $\mathcal{W}$ tout aussi quelconque. L'approche proposée est applicable si l'on est capable de déterminer une structuration 'naturelle' de $\mathcal{W}$ pour $\mathcal{A}$, ce qui implique d'étudier les deux points suivants :

- Que signifie 'structuration naturelle' pour $\mathcal{A}$?
- Comment obtenir une telle structuration ?

Le premier point dépend naturellement des caractéristiques de $\mathcal{A}$ tandis que le second point va dépendre, lui, du type d'obstacles fixes que l'on trouve dans l'espace de travail de $\mathcal{A}$.

La réponse à ces deux questions permettra d'évaluer le champ d'application de notre approche.

Bibliographie

[1] R. Alami et H. Chochon. – Programmation et contrôle d'exécution d'une cellule flexible d'assemblage. *In : Techniques de la robotique : perception et planification*, pp. 403–433. – Hermès, Paris (F), 1988.

[2] J.C. Alexander et J.H. Maddocks. – On the maneuvering of vehicles. *SIAM J. Appl. Math.*, vol. 48, n° 1, février 1988, pp. 38–51.

[3] F. Avnaim, J-D. Boissonnat, et B. Faverjon. – A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1656–1661. – Philadelphia, PA (USA), avril 1988.

[4] J. Barraquand et J-C Latombe. – On non-holonomic mobile robots and optimal maneuvering. *Revue d'intelligence Artificielle*, vol. 3, n° 2, 1989, pp. 77–103.

[5] J. Barraquand et J-C. Latombe. – *Robot motion planning: a distributed representation approach.* – Research Report n° STAN-CS-89-1257, Stanford Univ., CA (USA), Robotics Lab, Computer Science Dept, mai 1989.

[6] J. Barraquand et J-C. Latombe. – *Controllability of mobile robots with kinematic constraints.* – Research Report n° STAN-CS-90-1317, Stanford Univ. CA (USA), Robotics Lab., Computer Science Dept, juin 1990.

[7] J.E. Bobrow, S. Dubowsky, et J.S. Gibson. – On the optimal control of robotic manipulators with actuator constraints. *In : Proc. of the American Control Conf.*, pp. 782–787. – San-Francisco, CA (USA), juin 1983.

[8] J-D. Boissonnat, A. Cérézo, et J. Leblond. – *Shortest paths of bounded curvature in the plane.* – Research Report n° 1503, Nice (F), Inst. Nat. de la Recherche en Informatique et en Automatique, juillet 1991.

[9] J.M. Brady. – *Robotics science.* – Cambridge, MA (USA), MIT Press, 1989.

[10] R. A. Brooks. – Solving the find-path problem by good representation of free space. *IEEE Trans. Systems, Man and Cybernetics*, vol. 13, n° 3, March/April 1983, pp. 190–197.

[11] R.A. Brooks et T. Lozano-Perez. – A subdivision algorithm in configuration space for findpath with rotation. *IEEE Trans. Systems, Man and Cybernetics*, vol. 15, n° 2, March/April 1985, pp. 224–233.

[12] J. Canny. – *The complexity of robot motion planning.* – Massachusetts Institute of Technology, MA (USA), Thèse de PhD, Dept of Electrical Engineering and Computer Science, 1987.

[13] J. Canny, B. Donald, J. Reif, et P. Xavier. – On the complexity of kynodynamic planning. *In : Proc. of the IEEE Symp. on the Foundations of Computer Science*, pp. 306–316. – White Plains, NY (USA), novembre 1988.

[14] J. Canny, A. Rege, et J. Reif. – An exact algorithm for kinodynamic planning in the plane. *In : Proc. of the ACM Symp. on Computational Geometry*, pp. 271–280. – Berkeley, CA (USA), 1990.

[15] J. Canny et J. Reif. – New lower bound techniques for robot motion planning. *In : Proc. of the IEEE Symp. on the Foundations of Computer Science.* – Los Angeles, CA (USA), 1987.

[16] R. Chatila. – Path planning and environmental learning in a mobile robot system. *In : Proc. of the European Conf. on Artificial Intelligence*, pp. 211–215. – Orsay (F), août 1982.

[17] J.L. Crowley. – Navigation for an intelligent mobile robot. *IEEE Trans. Robotics and Automation*, vol. 1, n° 1, mars 1985.

[18] J. Deterne. – La géométrie des carrefours et échangeurs routiers et autoroutiers en rase campagne. *Revue générale des routes*, septembre 1970.

[19] B. Donald et P. Xavier. – Provably good approximation algorithms for optimal kinodynamic planning for cartesian robots and open-chain manipulators. *In : Proc. of the ACM Symp. on Computational Geometry*, pp. 290–300. – Berkeley, CA (USA), 1990.

[20] L.E. Dubins. – On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, vol. 79, 1957, pp. 497–516.

[21] M. Erdmann et T. Lozano-Perez. – *On multiple moving objects.* – A.I. Memo n° 883, Boston, MA (USA), MIT AI Lab., mai 1986.

[22] B. Faverjon. – Obstacle avoidance using an octree in the configuration space of a manipulator. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 504–512. – Atlanta, GA (USA), 1984.

[23] B. Faverjon et P. Tournassoud. – A local-based approach for path planning of manipulators with a high number of degrees of freedom. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1152–1158. – Raleigh, NC (USA), mars 1987.

[24] S. Fortune et G. Wilfong. – Planning constrained motions. *In : Proc. of the ACM Symp. on Theory of Computer Science*, pp. 445–459. – Chicago, IL (USA), mai 1988.

[25] Th. Fraichard, M. Hassoun, et C. Laugier. – Reactive motion planning in a dynamic world. *In : Proc. of the IEEE Int. Conf. on Advanced Robotics*, pp. 1028–1032. – Pisa (I), juin 1991.

[26] Th. Fraichard et C. Laugier. – On-line reactive planning for a non-holonomic mobile in a dynamic world. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 432–437. – Sacramento, CA (USA), avril 1991.

[27] K. Fujimura et H. Samet. – A hierarchical strategy for path planning among moving obstacles. *IEEE Trans. Robotics and Automation*, vol. 5, n° 1, février 1989, pp. 61–69.

[28] K. Fujimura et H. Samet. – Motion planning in a dynamic domain. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 324–330. – Cincinnatti, OH (USA), mai 1990.

[29] L. Gouzènes. – Strategies for solving collision-free trajectories problems for mobile and manipulator robots. *Int. Journal of Robotics Research*, vol. 3, n° 4, 1984, pp. 51–65.

[30] J.E Hopcroft, D.A. Joseph, et S.H. Whitesides. – Movements problems for two-dimensional linkages. *SIAM J. Appl. Math.*, vol. 13, n° 3, 1984, pp. 610–629.

[31] J.E. Hopcroft et G.T. Wilfong. – Reducing multiple object motion plannijng to graph searching. *SIAM J. Appl. Math.*, vol. 15, n° 3, 1986.

[32] P. Jacobs et J. Canny. – Planning smooth paths for mobile robots. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 2–7. – Scottsdale, AZ (USA), mai 1989.

[33] P. Jacobs, G. Heinzinger, J. Canny, et B. Paden. – *Planning guaranteed near-time-optimal trajectories for a manipulator in a cluterred workspace.* – Research Report n° ESRC 89-20/RAMP 89-15, Berkeley, CA (USA), Engineering Systems Research Center, Univ. of California., octobre 1989.

[34] P. Jacobs, J-P Laumond, et A. Rege. – Non-holonomic motion planning for hilare-like mobile robots. *In : Intelligent Robotics*, éd. par M. Vidyasagar et M. Trivedi, pp. 338–347. – Bangalore (In), McGraw-Hill, janvier 1991. Proc. of the Int. Symp. on Intelligent Robotics.

[35] P. Jacobs, J-P. Laumond, M. Taix, et R. Murray. – *Fast and exact trajectory planning for mobile robots and other systems with non-holonomic constraints.* – Research Report n° 90318, Toulouse (F), Lab. d'Automatique et d'Analyse des Systèmes, septembre 1990.

[36] Y. Kanayama. – Trajectory generation for mobile robots. *In : Proc. of the Int. Symp. on Robotics Research*, pp. 16–23.

[37] Y. Kanayama et B.J. Hartman. – *Smooth local path planning for autonomous vehicles.* – Research Report n° TRCS88-15, Santa Barbara, CA (USA), Dept of computer science at UCSB, juin 1988.

[38] K. Kant et S. Zucker. – Toward efficient trajectory planning: the path-velocity decomposition. *Int. Journal of Robotics Research*, vol. 5, n° 3, Fall 1986, pp. 72–89.

[39] O. Khatib. – Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, vol. 5, n° 1, Spring 1986.

[40] J-C. Latombe. – *Robot motion planning.* – Kluwer Academic Press, 1990.

[41] J-C. Latombe. – A fast path planner for a car-like indoor mobile robot. *In : Proc. of the Nat. Conf. on Artificial Intelligence.* AAAI. – Anaheim, CA (USA), juillet 1991.

[42] C. Laugier. – *Raisonnement géométrique et méthodes de décision en robotique. Application à la programmation automatique des robots.* – Thèse d'etat, Inst. Nat. Polytechnique de Grenoble, décembre 1987.

[43] C. Laugier et F. Germain. – An adaptive collision-free trajectory planner. *In : Proc. of the IEEE Int. Conf. on Advanced Robotics.* – Tokyo, (Japan), septembre 1985.

[44] J-P. Laumond. – Feasible trajectories for mobile robots with kinematic and environment constraints. *Intelligent Autonomous Systems*, décembre 1986, pp. 346–354.

[45] J-P. Laumond. – Finding collision-free smooth trajectories for a non-holonomic mobile robot. *In : Proc. of the Int. Joint Conf. on Artificial Intelligence*, pp. 1120–1123. – Milan (I), août 1987.

[46] J-P. Laumond. – Obstacle growing in a non-polygonal world. *Inf. Process. Lett.*, vol. 25, avril 1987, pp. 41–50.

[47] J-P. Laumond. – L'algorithmique du mouvement en robotique. *In : Proc. of the colloquium "Géométrie discrète, géométrie algorithmique, passage du discret au continu".* – Grenoble (F), novembre 1988.

[48] J-P. Laumond et T. Siméon. – *Motion planning for a two-degrees of freedom mobile robot with towing.* – Research Report n° 89148, Toulouse (F), Lab. d'Automatique et d'Analyse des Systèmes, 1989.

[49] J-P. Laumond, T. Siméon, R. Chatila, et G. Giralt. – Trajectory planning and motion control for mobile robots. *In : Geometry and Robotics*, éd. par J-D. Boissonnat et J-P. Laumond. pp. 133–149. – Lecture Notes in Computer Science, Vol 391, Springer-Verlag.

[50] D. Leven et M. Sharir. – An efficient and simple algorithm for a ladder moving in a two-dimensional space amidst polygonal barriers. *In : Proc. of the ACM Symp. on Computational Geometry*, pp. 211–227.

[51] D. Leven et M. Sharir. – *Planning a purely translationnal motion for a convex object in two-dimensional space using generalized Voronoï diagrams.* – Research Report n° 34/85, Tel-Aviv Univ. (Israël), The Eskenasy Inst., 1985.

[52] F. Leygue. – Le comportement dynamique des véhicules routiers. *Revue générale des routes*, no444, pp. 45–57.

[53] Z. Li et J.F. Canny. – *Robot motion planning with non-holonomic constraints.* – Research Report n° UCB/ERL M89/13, Berkeley, CA (USA), College of Engineering, UCB, février 1989.

[54] T. Lozano-Perez. – Spatial planning, a configuration space approach. *IEEE Trans. Comput.*, vol. 32, n° 2, février 1983.

[55] T. Lozano-Perez et M.A. Wesley. – An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, vol. 22, n° 10, octobre 1979, pp. 560–570.

[56] J. Marchand. – The algorithm by Schwartz, Sharir and Collins on the piano mover's problem. *In : Geometry and Robotics*, éd. par J-D. Boissonnat et J-P. Laumond. pp. 49–66. – Lecture Notes in Computer Science, Vol 391, Springer-Verlag.

[57] E. Mazer. – *HANDEY: Un modèle de planificateur pour la programmation automatique des robots.* – Thèse d'etat, Institut National Polytechnique de Grenoble, décembre 1987.

[58] J. Mills. – Caractéristiques géométriques des routes. *Revue générale des routes*, juin 1970.

[59] Y. Nakamura et R. Mukherjee. – Non-holonomic path planning of space robots. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1050–1055. – Scottsdale, AZ (USA), mai 1989.

[60] W. Nelson. – Continuous curvature paths for autonomous vehicles. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1260–1264. – Scottsdale, AZ (USA), mai 1989.

[61] N.J. Nilsson. – A mobile automaton: an application of artificial intelligence techniques. *In : Proc. of the Int. Joint Conf. on Artificial Intelligence*, pp. 509–520. – Washington, DC (USA), 1969.

[62] N.J. Nilsson. – *Principles of artificial intelligence.* – Los Altos, CA (USA), Morgan Kaufmann, 1980.

[63] C. Ó'Dúnlaing. – Motion planning with inertial constraints. *Algorithmica*, vol. 2, 1987, pp. 431–475.

[64] C. Ó'Dúnlaing et C. Yap. – A retraction method for planning the motion of a disc. *Journal of Algorithms*, vol. 6, 1982, pp. 104–111.

[65] Prometheus Office. – *"Functions" or how to achieve PROMETHEUS "objectives".* – Technical report, Stuttgart (FRG), juillet 1989.

[66] T-J. Pan et R.C. Luo. – Motion planning for mobile robots in a dynamic environment with moving obstacles. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 578–583. – Cincinnati, OH (USA), mai 1990.

[67] L. Peju et R. Jeannin. – L'aménagement des carrefours urbains. *Revue générale des routes*, avril 1973.

[68] E. Pommier. – *Génération de trajectoires pour robot mobile non-holonome par gestion des centres de rotation.* – Montpellier (F), Thèse de PhD, Lab. d'Informatique, de Robotique et de Microélectronique, novembre 1991.

[69] J.A. Reeds et L.A. Shepp. – Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, vol. 145, n° 2, 1990, pp. 367–393.

[70] J. Reif. – Complexity of the movers' problem and generalizations. *In : Proc. of the IEEE Symp. on the Foundations of Computer Science*, pp. 241–247. – New-York (USA), 1979.

[71] J. Reif et M. Sharir. – Motion planning in the presence of moving obstacles. *In : Proc. of the IEEE Symp. on the Foundations of Computer Science*, pp. 144–154. – Portland, OR (USA), octobre 1985.

[72] E. Rimon et D.E. Koditschek. – Exact robot navigation in geometrically complicated but topologically simple spaces. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1937–1943. – Cincinnatti, OH (USA), mai 1990.

[73] G. Sahar et J. H. Hollerbach. – Planning of minimum-time trajectories for robot arms. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 751–758. – St Louis, MI (USA), mars 1985.

[74] J. T. Schwartz et M. Sharir. – On the piano movers' problem: II. general techniques for computing topological properties of real algebraic manifold. *Advances in Applied Mathematics*, vol. 4, 1983, pp. 298–351.

[75] J.T. Schwartz et M. Sharir. – On the piano movers' problem: I. the special case of a rigid polygonal body moving amidst polygonal barriers. *Commun. Pure Appl. Math.*, vol. 36, 1983, pp. 345–398.

[76] C.L. Shih, T.T. Lee, et W.A. Gruver. – Motion planning with time-varying polyhedral obstacles based on graph search and mathematical programming. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 331–337. – Cincinnatti, OH (USA), mai 1990.

[77] Z. Shiller et J.C. Chen. – Optimal motion planning of autonomous vehicles in three-dimensional terrains. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 198–203. – Cincinnatti, OH (USA), mai 1990.

[78] Z. Shiller et S. Dubowsky. – Global time optimal motions of robotic manipulators in the presence of obstacles. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 370–375. – Philadelphia, PA (USA), avril 1988.

[79] K.G. Shin et N.D. McKay. – Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Trans. Autom. Contr.*, vol. 30, juin 1985, pp. 531–541.

[80] T. Siméon. – *Génération automatique de trajectoires sans collision et planification de tâches de manipulation en robotique.* – Toulouse (F), Thèse de PhD, Lab. d'Automatique et d'Analyse des Systèmes, janvier 1989.

[81] T. Siméon. – Motion planning for a non-holonomic mobile robot on 3-dimensional terrains. *In : Proc. of the IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pp. 1455–1460. – Osaka (Japan), novembre 1991.

[82] M. Taix. – *Planification de mouvements pour robot mobile non-holonome.* – Toulouse (F), Thèse de PhD, Lab. d'Automatique et d'Analyse des Systèmes, janvier 1991.

[83] P. Tournassoud. – *A strategy for obstacle avoidance and its application to multi robot systems.* – Research report, Inst. Nat. de la Recherche en Informatique et en Automatique, 1986.

[84] P. Tournassoud. – *Géométrie et intelligence artificielle pour les robots.* – Paris (F), Hermès, 1988, *Traité des nouvelles technologies, série robotique.*

[85] P. Tournassoud. – Motion planning for a mobile robot with a kinematic constraint. *In : Geometry and Robotics*, éd. par J-D. Boissonnat et J-P. Laumond. pp. 150–171. – Lecture Notes in Computer Science, Vol 391, Springer-Verlag.

[86] S.M. Udupa. – Collision detection and avoidance in computer-controlled manipulators. *In : Proc. of the Int. Joint Conf. on Artificial Intelligence*, pp. 737–748. – Cambridge, MA (USA), août 1977.

[87] G. Wilfong. – Motion planning for an autonomous vehicle. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 529–533. – Philadelphia, PA (USA), avril 1988.

[88] N. Wirth. – *Algorithms and data structures.* – Englewood Cliffs, NJ (USA), Prentice Hall Int., 1986.

[89] D. Zhu et J-C. Latombe. – *New heuristic algorithms for efficient hierarchical path planning.* – Research Report n° STAN-CS-89-1279, Stanford Univ. CA (USA), Robotics Lab., Computer Science Dept, 1989.

Table des figures

1.1 un bras planaire mono-corps 15
1.2 une roue 17

2.1 un mobile de type voiture 29
2.2 l'enchaînement des virages 33
2.3 le virage Λ 34
2.4 la région balayée par $\mathcal{A}$ lors d'un virage 35
2.5 la non-collision du chemin Υ 36
2.6 (a) D, (b) D^* et (c) $K(D^*)$ 39
2.7 $g_a(o)$ et $g_b(o), o \subset D$ 40
2.8 un exemple d'espace des centres de giration 41
2.9 le premier virage 42
2.10 C_g 44
2.11 deux virages 'proches' 47
2.12 résultats expérimentaux 48
2.13 résultats expérimentaux 48
2.14 résultats expérimentaux 49
2.15 résultats expérimentaux 50
2.16 résultats expérimentaux 50
2.17 résultats expérimentaux 51

3.1 Ψ et Ω, deux chemins adjacents 55
3.2 le changement de chemin 57
3.3 les chemins rectilignes Υ_i, $i = 0, \ldots, l$ 59
3.4 le graphe G 62
3.5 le test de sûreté 64
3.6 le bitmap associé à un chemin 66
3.7 la trajectoire décrite par $\mathcal{A}$ durant un changement de chemin . . . 66
3.8 résultats expérimentaux 68
3.9 résultats expérimentaux 69

4.1 architecture du planificateur de mouvement 73
4.2 les obstacles virtuels 75

Thèse de Monsieur Thierry Fraichard

Planification de mouvement pour mobile non-holonome en espace de travail dynamique.

Résumé. Le problème abordé dans ce mémoire est celui de la planification des mouvements d'un mobile $\mathcal{A}$ soumis à des contraintes cinématiques et dynamiques et se déplaçant dans un espace de travail $\mathcal{W}$ dynamique.

Ce mémoire défend la thèse selon laquelle le problème considéré, connu pour être combinatoirement très complexe, peut être résolu de façon efficace lorsqu'il existe une *structuration de $\mathcal{W}$ naturelle pour $\mathcal{A}$*, i.e. lorsqu'il est possible de structurer $\mathcal{W}$ en un ensemble de *zones libres* à l'intérieur desquelles $\mathcal{A}$ peut se déplacer. Dans ce cas, le problème peut être abordé suivant deux directions complémentaires : 1) la *planification de chemin* qui prend en compte les obstacles fixes de $\mathcal{W}$ et les contraintes cinématiques de $\mathcal{A}$, et 2) la *planification de trajectoire* qui prend en compte les obstacles mobiles de $\mathcal{W}$ et les contraintes dynamiques de $\mathcal{A}$.

Ce mémoire traite le cas d'une voiture dans le réseau routier. Les zones libres sont alors définies par les voies de circulation. $\mathcal{A}$ est soumis à une contrainte cinématique non-holonome qui l'oblige à se déplacer dans une direction perpendiculaire à l'axe de ses roues. De plus, le rayon de braquage, l'accélération et la vitesse de $\mathcal{A}$ sont limités.

Nous commençons par présenter deux techniques de résolution propres à chacune des deux planifications mentionnées ci-dessus. Ces techniques sont bien adaptées au contexte dans lequel nous nous plaçons. Enfin, nous montrons comment intégrer ces deux techniques au sein d'un système de planification de mouvement qui permet de résoudre efficacement le problème considéré.

Mots clés. Robotique, robot mobile, planification de mouvement, planification 'ciné-dynamique', non-holonomie, obstacles mobiles.

Motion planning for a non-holonomic mobile in a dynamic workspace

Abstract. This thesis deals with motion planning for a mobile $\mathcal{A}$ which is subject to kinematic and dynamic constraints and moves within a dynamic workspace $\mathcal{W}$.

The key point of this thesis is that this problem, known to be intricate, can be solved efficiently if *$\mathcal{W}$ is naturally structured for $\mathcal{A}$*, i.e. if it is possible to define a set of *free regions* within which $\mathcal{A}$ is able to move. In this case the problem considered can be tackled along two complementary directions: 1) *Path planning* which takes into account the fixed obstacles of $\mathcal{W}$ and the kinematic constraints of $\mathcal{A}$, and 2) *Trajectory planning* which takes into account the moving obstacles of $\mathcal{W}$ and the dynamic constraints of $\mathcal{A}$.

This thesis considers a car-like mobile in the road-network. The free regions are defined by the lanes of the roadway. $\mathcal{A}$ is subject to a non-holonomic kinematic constraint which compels it to move in a direction which is perpendicular to the axle of its wheels. Besides the gyration radius and the acceleration and the velocity of $\mathcal{A}$ are bounded.

To begin with, we present two techniques in order to solve the aforementioned planning problems. These techniques are well suited to our context. Finally we show how to integrate these two techniques within a motion planner which is able to solve efficiently the problem considered.

Key words. Robotics, mobile robot, motion planning, 'kinodynamic' planning, non-holonomy, moving obstacles.