



Université Blaise Pascal



LABORATOIRE DES SCIENCES ET MATERIAUX POUR
L'ELECTRONIQUE,
U.F.R. Sciences Exactes et Naturelle et D'AUTOMATIQUE
(U.M.R.6602 du CNRS)

MEMOIRE

Présenté par

Chabha Boussamet

En vue de l'obtention du

MASTER STIC : Informatique MSIR
« RPM »

Commande optimale pour la navigation sûre
d'un robot mobile

le 29/06/2009

Encadré par: **Lounis Aduane**

Table des matières

| | |
|---|----|
| <i>Introduction générale</i> | 1 |
| Robotique mobile..... | 2 |
| <i>I- Problématique de la commande optimale</i> | 4 |
| I-1 Introduction _____ | 4 |
| I-2 Position du problème _____ | 4 |
| I-3 Principe d’optimalité de Bellman _____ | 5 |
| I-4 Principe du minimum de Pontriaguine _____ | 5 |
| I-5 Conclusion _____ | 7 |
| <i>II- Commande optimale en robotique mobile</i> | 8 |
| II-1 Introduction _____ | 8 |
| II-2 Plus courts chemins de Dubins _____ | 8 |
| II-3 Plus courts chemins de Reeds et Shepp _____ | 9 |
| II-4 Commande optimale en temps et en énergie _____ | 11 |
| II-5 Combinaison platitude et commande optimale _____ | 13 |
| II-5-1 Définition d’un système plat..... | 13 |
| II-5-2 Modèle du robot..... | 15 |
| II-5-3 Définition des B-splines..... | 15 |
| II-5-4 Propriétés des B-splines..... | 15 |
| II-6 Commande prédictive _____ | 17 |
| II-7 Conclusion _____ | 20 |
| <i>III- Commande optimale d’un robot mobile avec une architecture hybride</i> | 21 |
| III-1 Introduction _____ | 21 |
| III-2 Architecture hybride _____ | 21 |
| III-3 Commande hybride pour la coordination d’actions en robotique mobile __ | 22 |
| III-4 Commande optimale pour les systèmes dynamiques hybrides _____ | 24 |
| III-5 Conclusion _____ | 25 |

| | |
|--|-----------|
| <i>IV- Commande hybride utilisée</i> | 26 |
| IV-1 Introduction | 26 |
| IV.2 Architecture proposée | 26 |
| IV-3 Contrôleur d'attraction vers la cible | 27 |
| IV-4 Contrôleur d'évitement d'obstacles | 27 |
| IV-5 Optimalité du cycle limite | 29 |
| IV-6 Longueur de la trajectoire du cycle limite | 30 |
| IV-6-1 Définition de la longueur des courbes | 30 |
| IV-6-2 Calcul de la longueur du cycle limite | 31 |
| IV-7 Résultats de simulation | 35 |
| IV-7-1 Optimisation de μ | 37 |
| IV-7-2 Exemple de simulation | 39 |
| IV-8 Conclusion | 42 |
| <i>Conclusion générale et perspectives</i> | 43 |
| <i>Bibliographie</i> | 44 |

Introduction générale

Dans ce stage, le travail que nous proposons s'inscrit dans le cadre de la commande optimale pour la navigation d'un robot mobile dans un environnement contraint, en utilisant une architecture de commande hybride. Nous nous sommes intéressés aux problèmes de la caractérisation et du calcul du plus court chemin lors de la phase d'évitement d'obstacles. Plus spécifiquement, on minimise la longueur de la trajectoire à suivre pendant l'évitement d'obstacles, à faire suivre par le robot pour atteindre sa cible, et cela en trouvant le paramètre μ qui modifie l'angle d'incidence vers le cercle limite de l'obstacle.

L'organisation du rapport de stage est comme suit :

Le chapitre I de ce rapport introduit le problème de la commande optimale. Les concepts de base tels que le principe d'optimalité de Bellman et le principe du minimum de Pontriaguine sont développés.

Le chapitre II présente l'application de la commande optimale en robotique mobile. Un état de l'art sur les travaux proposés dans la littérature est réalisé.

Le chapitre III est un prolongement du chapitre III dans le cadre de la commande optimale en robotique mobile avec une architecture hybride. Après avoir décrit les architectures de commande hybride, on présente quelques travaux réalisés en commande optimale pour ces architectures.

Dans le chapitre IV, nous proposons dans le cadre de l'évitement d'obstacles par la méthode des cycles limites, de trouver la trajectoire optimale à faire suivre par le robot pour atteindre sa cible. L'efficacité de cette stratégie de commande est illustrée par des exemples en simulation.

Ce rapport se termine par une conclusion générale où l'accent sera mis sur la contribution de la caractérisation du plus court chemin lors de l'évitement d'obstacles. Par ailleurs, des perspectives sur les travaux s'inscrivant dans le prolongement de ce stage et à mener ultérieurement sont proposées.

Robotique mobile

Le robot peut être vu comme l'expression d'un désir intemporel de l'homme : celui de donner naissance à un clone qui lui obéirait sans ressentiment ni souffrance. La robotique a été une muse pour le cinéma où l'homme laisse libre cours à son imagination, tels que les voitures volantes ou la capacité de doter les robots d'une conscience. Les progrès effectués au cours du temps sont considérables mais toujours pas de machine intelligente à l'horizon. Dans ce chapitre un bref historique sur l'évolution de la robotique au cours du temps est proposé.

Historique

L'histoire des robots remonte à l'Antiquité, selon la mythologie grecque, Pygmalion, roi de Chypre, s'était épris d'une statuette d'ivoire qu'il avait sculptée. Il demanda à la déesse Aphrodite de l'animer et put alors l'épouser. Le conte pour enfants de Pinocchio raconte l'histoire d'un pantin articulé qui devient un vrai petit garçon. Et l'histoire imaginaire de la créature de Frankenstein est un autre exemple d'une entité créée par la main de l'homme.

L'une des premières réalisations est due au Français Jacques de Vaucanson qui présenta, en 1730, un canard articulé en cuivre capable de boire, manger, cancaner et digérer comme un véritable animal. Mais ce n'est que dans la deuxième partie du XXe siècle que sont apparues les premières formes utiles des robots. En 1950, l'Anglais Grey Walter réalisa une machine ayant un comportement autonome. Il s'agissait d'une sorte de tortue mécanique capable de se déplacer. Cette tortue se dirigeait vers toute source de lumière et, en l'absence de lumière, se déplaçait de manière aléatoire. Ce type de comportement est caractéristique d'un animal peu évolué, ou mieux d'un insecte. En 1960, un projet intéressant fut développé à l'université John Hopkins, aux Etats-Unis. Il s'agissait d'une plate-forme autonome équipée d'un sonar et d'un œil, dont le seul but était de maintenir ses batteries chargées. Le monde, pour ce robot, se divisait en deux parties : les prises de courant et tout le reste, pour la première fois, un robot semblait percevoir le monde.

On distingue deux groupes de robots : les robots fixes et les robots mobiles. Les robots fixes bien plus généralisés dans l'industrie, sont utilisés pour réaliser des tâches dangereuses (exemple : soudure du châssis ou peinture de la carrosserie dans l'industrie), les robots mobiles pour transporter des charges (depuis les chaînes de fabrication jusqu'aux zones de stockage) ou encore pour transporter le courrier dans les bureaux, ou pour intervenir dans des milieux hostiles. Bien évidemment, nous sommes de nos jours de plus en plus exigeant quand aux tâches que doit accomplir le robot.

Tous les efforts des roboticiens vont alors consister à mettre en place les outils permettant de faire évoluer le robot dans son environnement de manière satisfaisante, qu'il s'agisse de suivre un chemin connu ou d'aller d'un point à un autre en réagissant à une modification de l'environnement ou à la présence d'obstacles.

Chapitre I

Problématique de la commande optimale

I-1 Introduction

Dans ce chapitre, on introduit le problème de la commande optimale. Les concepts de base tels que le principe d'optimalité de Bellman et le principe du minimum de Pontriaguine sont développés. Les problèmes de commande optimale se rencontrent dans la vie de tous les jours: comment arriver à destination le plus rapidement possible, comment minimiser sa consommation... Pour un système dynamique donné et dont les équations sont connues, le problème de commande optimale consiste alors à trouver la commande minimisant un critère donné. Une des grandes applications de la commande optimale a été l'application au lanceur Apollo dans les années 1960. Notons néanmoins que les difficultés soulevées par ce genre de problème sont loin d'être complètement résolues. La commande optimale reste donc un sujet de recherche d'actualité.

I-2 Position du problème

Soit un système à temps continu de représentation d'état :

$$\dot{x} = f(x, u, t) \quad (1)$$

et de condition initiale $x(t_0) = x_0$, où $t \in \mathbb{R}$, $u \in \mathbb{R}^m$ et $x \in \mathbb{R}^n$. Les signaux u et x sont des fonctions de \mathbb{R} vers respectivement \mathbb{R}^m et \mathbb{R}^n . Pour la condition initiale x_0 et la commande u , l'équation d'état (1) définit une trajectoire unique x pour l'état sur $[t_0, t_f]$. Celle-ci est fonction de la condition initiale x_0 et de la commande u sur $[t_0, t_f]$.

Soit un critère :

$$J(x_0, t_0, u) = \theta(x_f, x_f) + \int_{t_0}^{t_f} \phi(x, u, t) dt \quad (2)$$

avec $x_f = x(t_f)$. Les fonctions θ et ϕ ainsi que les instants t_0 et t_f étant données, ce critère ne dépend que de x_0 et de u sur $[t_0, t_f]$. L'application qui au signal de commande u associe le critère scalaire $J(x_0, t_0, u)$ est une fonctionnelle.

En plus de l'équation d'état qui lie les trajectoires de u et de x , d'autres contraintes peuvent intervenir (sans pour autant remettre en cause le critère choisi). Typiquement :

- l'instant final peut être imposé ou libre ;
- la commande peut appartenir à un ensemble $u \in U \neq \mathbb{R}^m$;
- des contraintes peuvent exister sur l'état final : $x_f \in X$.

Le problème de la commande optimale consiste alors à trouver la commande \tilde{u} minimisant $J(x_0, t_0, u)$:

$$\tilde{u} = \min_{u \in U} J(x_0, t_0, u) \quad (3)$$

I-3 Principe d'optimalité de Bellman

Soit le critère défini en (2), La trajectoire optimale sur $[t_0, t_f]$ est \tilde{u} et le critère optimale :

$$\tilde{u} = \min_{u \in U} J(x_0, t_0, u) \quad (4)$$

Soit $t_1 \in [t_0, t_f]$. Le principe d'optimalité de Bellman énonce que la trajectoire optimale sur $[t_0, t_f]$ contient la trajectoire optimale sur $[t_1, t_f]$ avec comme condition initiale $x_1 = x(t_1)$. Autrement dit :

$$\tilde{J}(x_0) = \min_{u[t_0, t_1], x_1} \left(\int_{t_0}^{t_1} \phi(x, u, t) dt + \tilde{J}(x_1) \right) \quad (5)$$

Bien que les développements suivants ne s'appuient pas directement sur ce principe, mais sur le principe du maximum, ce principe est un résultat classique de la commande optimale et se trouve souvent utilisé dans la littérature. Il permet d'obtenir une solution optimale en découpant l'intervalle et en résolvant un problème récursif.

I-4 Principe du minimum de Pontriaguine¹

Soit le système d'équation d'état (1) et le critère de performance défini en (2) :

On définit l'Hamiltonien du système :

$$H(x, u, p, t) = \phi(x, u, t) + p^T f(x, u, t) \quad (6)$$

¹ **Lev Semionovitch Pontriaguine** (en russe : Лев Семёнович Понтрягин; ISO 9 : Lev Semënovič Pontrâgin), né le 3 septembre 1908 à Moscou et mort le 3 mai 1988, était un mathématicien soviétique. Il a fait des découvertes majeures dans plusieurs domaines des mathématiques, en particulier en topologie.

où p est appelé état-adjoint² (vecteur de co-état). Le principe du minimum de Pontriaguine énonce que la trajectoire optimale minimise l'hamiltonien du système (Pontriaguine, 1974). Autrement dit :

$$H(\tilde{x}, \tilde{u}, \tilde{p}) \leq H(\tilde{x}, u, \tilde{p}) \quad \forall u \in U \quad (7)$$

Le long de la trajectoire optimale, on dispose d'un certain nombre d'équations permettant de résoudre le problème de commande optimale. Ces équations sont généralement établies en utilisant le calcul des variations.

L'extrémalité de la solution conduit à un jeu d'équations, appelées équations canoniques de Hamilton, qui régissent les dynamiques de l'état d'une part et de l'état adjoint d'autre part :

➤ état

$$\frac{\partial H}{\partial p} = \dot{x} \quad (8)$$

➤ état adjoint

$$\frac{\partial H}{\partial x} = -\dot{p} \quad (9)$$

Les équations provenant des conditions dites terminales, en t_0 d'une part et en t_f d'autre part sont appelées équations de transversalité :

➤ à l'origine

$$\left(-H(t_0) + \frac{\partial \theta}{\partial t_0}\right) \delta t_0 + \left(p(t_0) + \frac{\partial \theta}{\partial x_0}\right)^T \delta x_0 = 0 \quad (10)$$

➤ à l'arrivée

$$\left(H(t_f) + \frac{\partial \theta}{\partial t_f}\right) \delta t_f + \left(-p(t_f) + \frac{\partial \theta}{\partial x_f}\right)^T \delta x_f = 0 \quad (11)$$

Enfin, selon la nature du problème, on aura encore certaines relations additionnelles:

➤ si aucune contrainte (de type saturation) n'est imposée sur $u(t)$ à l'instant t , on a :

$$\frac{\partial H}{\partial u}(t) = 0 \quad (12)$$

➤ si H n'est pas une fonction explicite du temps, on a :

² En Anglais : costate vector

$$\frac{dH}{dt} = \frac{\partial H}{\partial t} = 0 \quad (13)$$

I-5 Conclusion

Dans ce chapitre nous avons introduit le problème de commande optimale, et les deux principes d'optimalité de Bellman et Pontriaguine.

Chapitre II

Commande optimale en robotique mobile

II-1 Introduction

Dans le chapitre précédant nous avons vu le problème du contrôle optimal et quelques principes d'optimisation. Ce chapitre présente l'application de la théorie de la commande optimale en robotique mobile, les chercheurs se sont principalement intéressés au problème de la planification de trajectoire vérifiant des critères d'optimalité tels que le temps ou la longueur du chemin. Le problème a été abordé en un premier temps du point de vue de la géométrie et du calcul différentiel, et plus récemment, du point de vue de la théorie de la commande optimale.

II-2 Plus courts chemins de Dubins

Dubins fut le premier à caractériser, en l'absence d'obstacles, les trajectoires de longueur minimale pour un robot sujet à une contrainte sur le rayon de courbure (Dubins, 1957). Dubins a montré que le plus court chemin pour le robot de Dubins entre deux points orientés du plan est une courbe C^1 par morceaux. Cette courbe est la concaténation de segments de droites et d'arcs de cercle, et peut s'écrire sous la forme CSC ou CCC où S est un segment de droite et C est un arc de cercle de rayon minimal, c'est-à-dire de rayon 1. Chaque arc du type S ou C peut être dégénéré, c'est-à-dire de longueur nulle. Le type d'un chemin peut être spécifié de façon plus précise : en notant R (resp. L) un arc de cercle parcouru dans le sens des aiguilles d'une montre³, le type d'un chemin peut être RSR, par exemple (voir figure II.1).

Entre deux configuration il existe au plus six chemins de Dubins qui sont :

- de type CCC réunissant les deux chemins de types LR_uL et RL_uR où u représente la longueur de l'arc intermédiaire, vérifiant $u > \pi$, ou
- de type CSC réunissant les quatre chemins de type LSL, LSR, RSL, RSL.

³ .Un arc de cercle parcouru dans le sens des aiguilles d'une montre est parcouru en tournant à droite. La notation R et L correspond aux initiales de right et left.

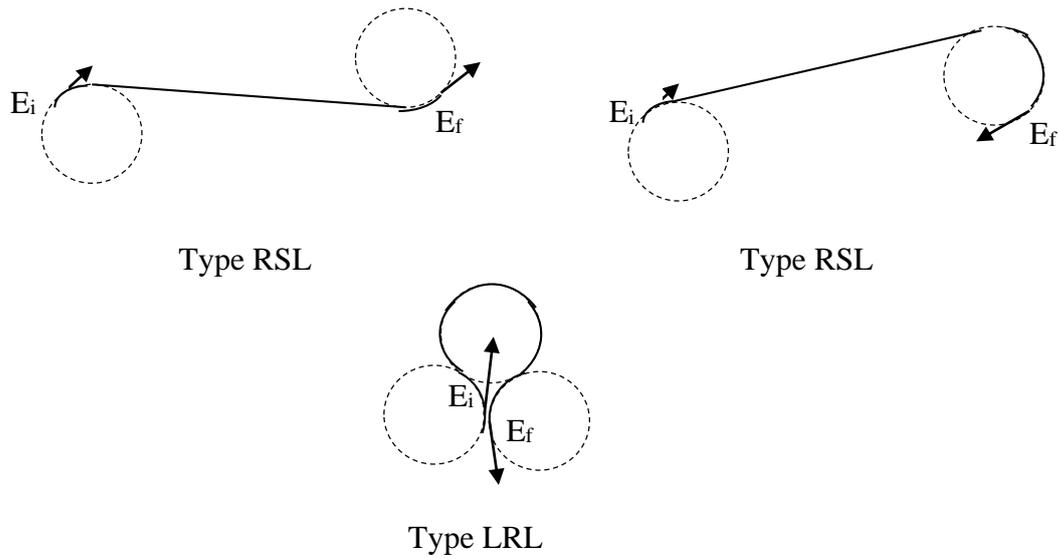


Fig.II.1- Exemples de plus courts chemins de Dubins.

II-3 Plus courts chemins de Reeds et Shepp

En 1990, Reeds et Shepp (Shepp, 1990) caractérisèrent, en l'absence d'obstacles, les trajectoires de longueur minimale pour un robot sujet à une contrainte sur le rayon de courbure mais capable d'effectuer des manœuvres (marches arrière). Pour le robot de Reeds et Shepp, les trajectoires optimales présentent des points de rebroussement où le robot passe de la marche avant à la marche arrière. Un plus court chemin pour ce robot vérifie que la vitesse angulaire du robot prend des valeurs extrêmes ou nulles. De plus, Reeds et Shepp ont pu montrer un résultat similaire à celui de Dubins, en l'absence d'obstacle, il existe un plus court chemin joignant deux configurations qui est constitué au plus de cinq arcs (arcs de cercle unitaire ou des segments de droite), et au maximum de deux points de rebroussement. Ceux-ci sont notés par le signe « | » dans le mot représentant le type d'un chemin. Plus précisément, ce plus court chemin est d'un appartenant à l'une des neuf familles suivantes

-CSC

-C|C|C

-CC|C ou C|CC

-CC_u|C_uC ou C|C_uC_u|C avec $u < \frac{\pi}{2}$

-C|C _{$\pi/2$} SC ou CSC _{$\pi/2$} C

$-C|C_{\pi/2}SC_{\pi/2}|C$

Les types de trajectoires sont toutes les combinaisons possibles avec les arcs R et L, sachant que l'on ne peut avoir ni R|L ni L|R. (figure II.2)

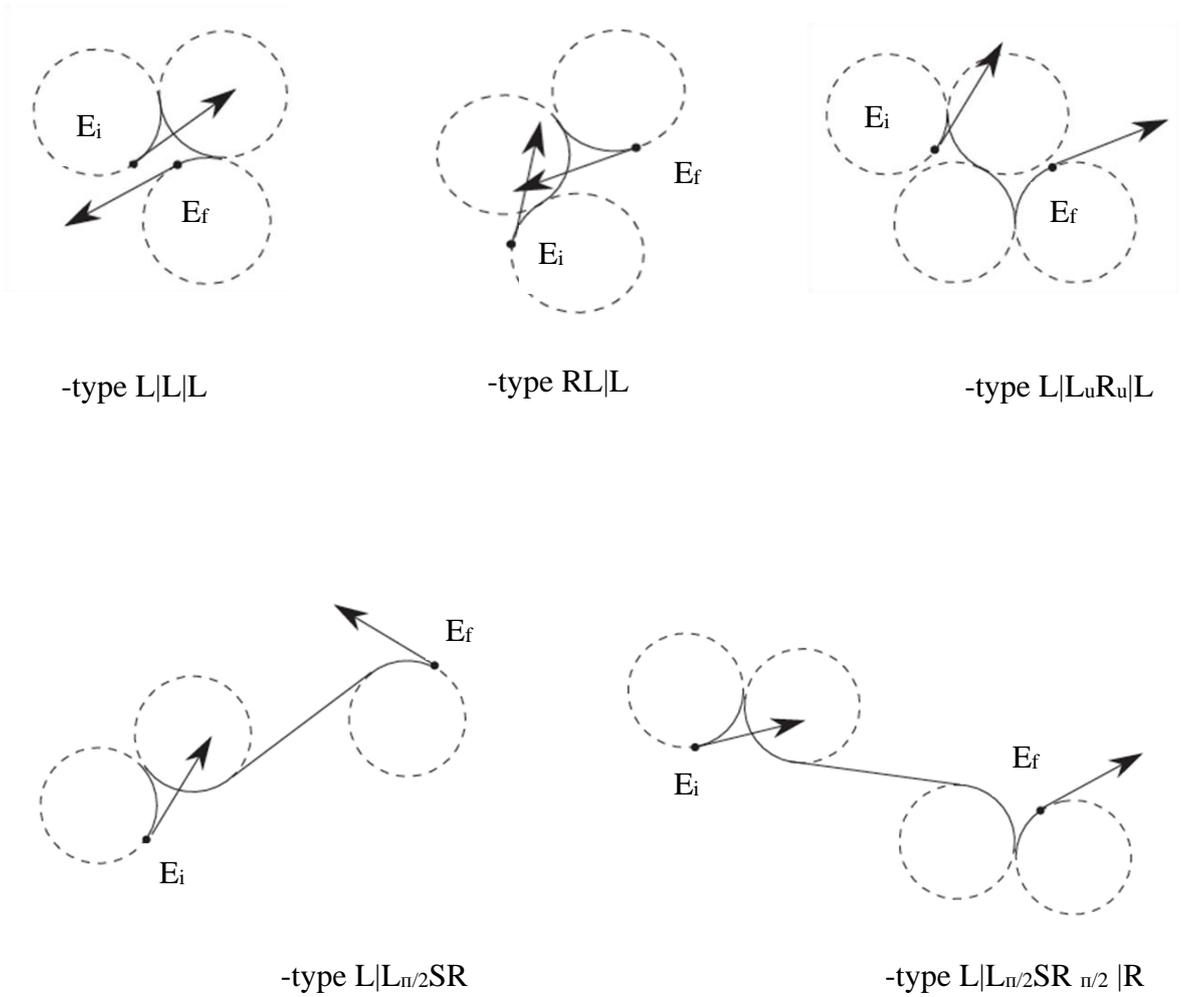


Fig.II.2- Exemples de plus court chemins de Reeds et Shepp.

En 1993, ces résultats furent redémontrés par P.Souères dans sa thèse (Souères, 1993). Ces nouvelles démonstrations utilisèrent le cadre formel de la théorie de la commande optimale.

Ce travail s'appuie sur une famille réduite de trajectoires déjà déterminés (chemins de Dubbins, chemins de Reeds et Shepp) suffisamment riche pour toujours contenir une solution. L'auteur utilise ce résultat en couplant une méthode géométrique avec les conditions nécessaires du principe du maximum de Pontriaguine (PMP).

II-4 Commande optimale en temps et en énergie

Dans le cadre de la commande optimale on peut citer d'autres travaux plus récents (Ancenay, 2004) qui au lieu de s'intéresser à la caractérisation des plus courts chemins, fait une étude sur l'optimalité en terme de temps et d'énergie. La trajectoire de chaque robot est complètement définie par son état initial (position et vitesse) et les séquences ultérieures d'accélération. Les variables P , V et A indiquent respectivement le vecteur position, le vecteur de la vitesse et le vecteur d'accélération. La trajectoire est discrétisée en n étapes de durée Δ .

$$\begin{cases} V^i = V^0 + \Delta \sum_{j=1}^i A^j \\ P^i = P^0 + \Delta \sum_{j=1}^i V^j \end{cases} \quad (14)$$

La tâche à accomplir étant de ramener le robot d'un état initial à état final tout en satisfaisant les différentes contraintes. Deux problèmes à résoudre :

1- Détermination du nombre minimal d'étape n_0 tel que la mission puisse être achevée en respectant les limites physiques du robot (vitesse et accélération maximal du robot). Pour estimer le nombre minimal d'étapes n_0 , une recherche binaire est utilisée. (Fig.II.3)

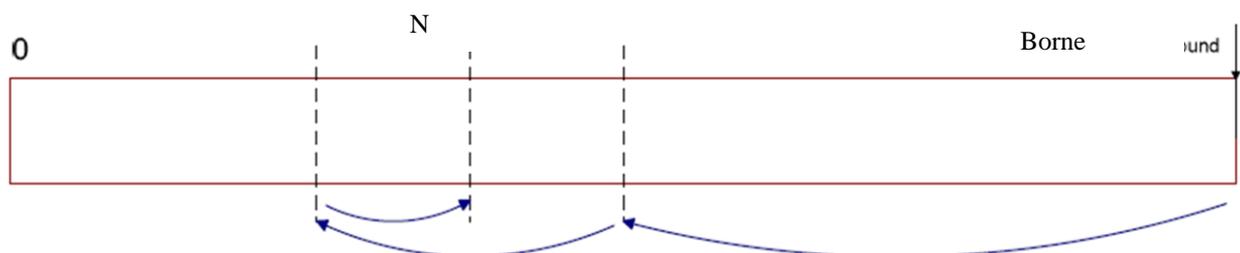


Fig.II.3 – Recherche binaire du minimum n_0

2- Détermination de n tel que $n \geq n_0$, qui donne la séquence des accélérations qui minimise l'énergie consommée.

À fin de simplifier le problème 2D, la résolution c'est faite en 1D car le chemin 2D est une combinaison de deux chemins 1D comme cela est expliqué en figure II.4 et II.5, mais il faut avoir le même nombre d'étapes $n_x = n_y$ pour pouvoir revenir au problème 2D une fois la résolution faite en 1D. (Figure II.6)

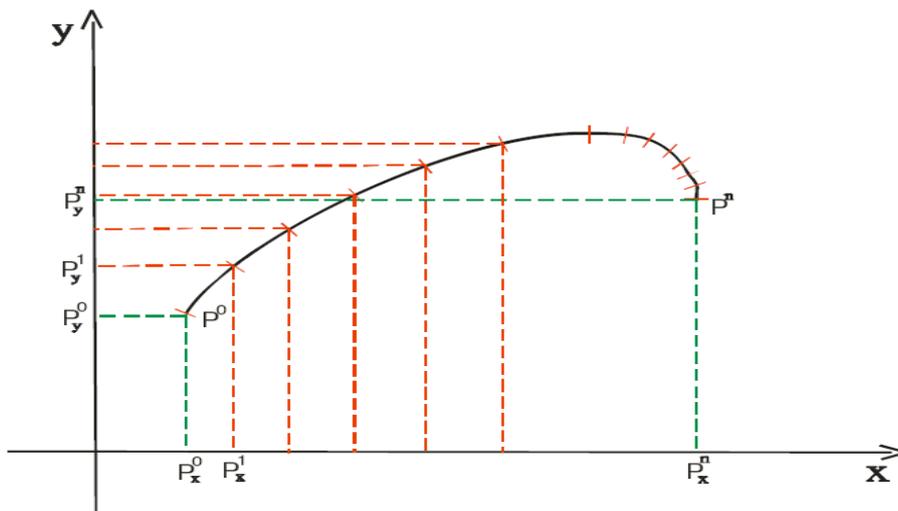


Fig.II.4- Chemin 2D est une combinaison de deux chemins 1D.

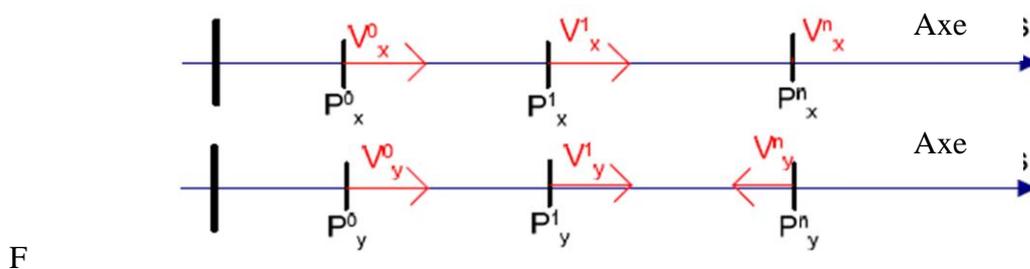


Fig.II.5- Les deux chemins 1D correspondant au chemin 2D de Fig.III.4.

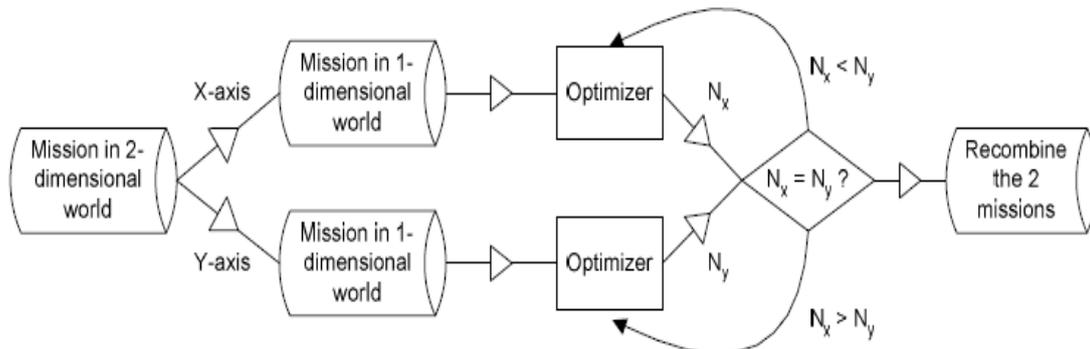


Fig.II.6- Obtention de la solution 2D à partir de la solution 1D.

On doit atteindre l'état final en n étapes. Puis pour minimiser l'énergie consommée, il faut trouver une séquence d'accélération qui minimise la fonction coût $\sum_{k=1}^n \|A^k\|^2$ en $n > n_0$.

Une première méthode a été proposée en utilisant la programmation quadratique, bien que cette méthode a été introduite pour le contrôle de robot, la contribution est dans la réduction de la mission 2D en une mission 1D.

Dans la seconde méthode, la recherche de séquence d'accélération optimales est réduite au problème du plus court chemin dans un graphe. Le problème d'optimisation est résolu avec la programmation dynamique en utilisant l'algorithme Dijkstra (algorithme polynomial).

II-5 Combinaison platitude et commande optimale

La notion de platitude a été introduite en 1992 par Michel Fliess en collaboration avec Jean Lévine, Philippe Martin et Pierre Rouchon a donné lieu au cours de la dernière décennie à de nombreuses applications, dont certaines sur site industriel. C'est la seule théorie non linéaire appliquée industriellement.

II-5-1 Définition d'un système plat

On dit que le système défini par :

$$\dot{x} = f(x, u), x \in \mathbb{R}^n, u \in \mathbb{R}^m \quad (15)$$

est plat s'il existe une application $h: \mathbb{R}^n * (\mathbb{R}^m)^{r+1} \rightarrow \mathbb{R}^m$; une application $\phi: (\mathbb{R}^m)^r \rightarrow \mathbb{R}^n$ et une application $\varphi: (\mathbb{R}^m)^{r+1} \rightarrow \mathbb{R}^m$ telles qu'on puisse écrire :

$$\begin{cases} y = h(x, u, \dot{u}, \dots, u^{(r)}) \\ x = \emptyset(y, \dot{y}, \dots, y^{(r-1)}) \\ u = \varphi(y, \dot{y}, \dots, y^{(r-1)}, y^{(r)}) \end{cases} \quad (16)$$

Cela revient à dire qu'on a résumé tout le comportement dynamique du système par le comportement de sa sortie plate : toutes les trajectoires sont de la forme

$$\begin{aligned} x(t) &= \emptyset(y(t), \dot{y}(t), \dots, y^{(r)}(t)) \\ u(t) &= \varphi(y(t), \dot{y}(t), \dots, y^{(r+1)}(t)) \end{aligned} \quad (17)$$

Où r est un entier.

Il est à noter que :

1. la dimension de la sortie plate est égale au nombre de commandes du système.
2. il n'y a pas unicité des sorties plates. Autrement dit le paramétrage n'est pas unique.
3. on peut souvent trouver des sorties plates possédant une interprétation physique.

Exemple II.5.1: Le pendule de Huygens

On considère un solide de masse m dans un plan vertical dont on sait commander les accélérations en un point A distinct du centre d'inertie G . Soit α l'angle entre (AG) et la verticale et x et y les coordonnées de A , on a :

$$\begin{cases} \ddot{x} = u \\ \ddot{z} = v \\ a\ddot{\alpha} = g \sin \alpha + u \cos \alpha + v \sin \alpha \end{cases} \quad (18)$$

avec $a = \frac{I}{md}$ ou $d = AG$ et I est le moment d'inertie du solide en G .

Une sortie plate est le point de coordonnées $(x - a \sin \alpha, z + a \cos \alpha)$ situé sur la droite (AG) à une distance a de A . Ce point remarquable est connu des mécaniciens depuis Huygens sous le nom de *centre d'oscillation*.

Cette théorie a été utilisée dans la littérature pour la commande des robots mobiles par la génération de trajectoire optimale, parmi ces travaux citons ceux de Defroot.

Defroot dans sa thèse s'est intéressé en premier lieu au cas mono-robot, ou une architecture de planification de trajectoire basée sur la platitude différentielle et l'optimisation est proposée (Defroot, 2007). L'auteur a travaillé sur le calcul d'une trajectoire admissible avec évitement d'obstacles statiques pour un robot unicycle entre une configuration de départ et une

configuration d'arrivée donnée, et ce en minimisant le temps de parcours. La planification est faite en combinant la commande optimale et la théorie de la platitude.

II-5-2 Modèle du robot

Soit l'abscisse et l'ordonnée (x, y) du milieu de l'axe des deux roues motrices, θ l'orientation du robot. Où v est la vitesse linéaire du robot et ω est la vitesse angulaire du robot.

En prenant comme sortie $z = [x, y]^T$, le système (19) peut se réécrire sous la forme d'un système plat. En effet, on a :

$$\begin{cases} \theta = \arctan \frac{\dot{y}}{\dot{x}} \\ v = \sqrt{\dot{x}^2 + \dot{y}^2} \\ \omega = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2} \end{cases} \quad (19)$$

Afin de transformer le problème de commande optimale en un problème de programmation non linéaire, il est nécessaire de paramétriser la sortie plate. Le choix c'est porté sur les B-splines. La courbe spline spécifiant la sortie plate, est construite comme une combinaison linéaire de fonctions de base appelées B-splines. Une fonction B-spline correspond au raccordement de courbes de Bézier.

II-5-3 Définition des B-splines

Soit une suite non décroissante de réels positifs $\{\text{noeud}_0 \leq \dots \leq \text{noeud}_M\}$, appelé suite de nœuds, qui contient l'ensemble des points de raccordement. Les B-splines d'ordre $d = 1$ sont définies par :

$$\forall j = \{0, \dots, M-1\},$$

$$B_{j,1}(t) = \begin{cases} 1 & \text{si } \text{noeud}_j \leq t \leq \text{noeud}_{j+1} \\ 0 & \text{sinon} \end{cases} \quad (20)$$

et les B-splines d'ordre $d \geq 2$ ($d \in \mathbb{N}$) par la relation de récurrence : $\forall j = \{0, \dots, M-d\}$,

$$B_{j,d}(t) = \frac{t - \text{noeud}_j}{\text{noeud}_{j+d} - \text{noeud}_j} B_{j,d-1}(t) + \frac{\text{noeud}_{j+d+1} - t}{\text{noeud}_{j+d+1} - \text{noeud}_{j+1}} B_{j+1,d-1}(t) \quad (21)$$

II-5-4 Propriétés des B-splines

Avec les notations ci-dessus, on montre les propriétés suivantes :

1. $B_{j,d}(t) = 0, \forall t \notin [\text{noeud}_j, \text{noeud}_{j+d}]$. De plus $B_{j,d}(t) \geq 0$ sur $[\text{noeud}_j, \text{noeud}_{j+d}]$.

2. Les B-splines d'ordre d sont des fonctions polynômiales par morceaux de degré au plus $d-1$.

Si les nœuds sont tous distincts, ce sont des fonctions de classe C^{d-2} . Lorsqu'un nœud est de multiplicité $1 \leq r \leq d-1$, la B-spline n'est que $d-r-1$ fois continûment dérivable en ce point.

Si un nœud est de multiplicité d , il apparaît alors une discontinuité dans le graphe de la B-spline. A partir des relations de récurrence, on a :

$$\dot{B}_{j,d}(t) = (d-1) \frac{B_{j,d-1}(t)}{\text{noeud}_{j+d-1} - \text{noeud}_j} - \frac{B_{j+1,d-1}(t)}{\text{noeud}_{j+d} - \text{noeud}_{j+1}} \quad (22)$$

qui permet d'obtenir les dérivées successives des B-splines en tenant compte de la multiplicité des nœuds.

Après avoir exprimé les trajectoires de la sortie plate à l'aide des B-splines, le problème de commande optimale peut facilement se transformer en un problème de programmation non linéaire.

Le calcul se fait en premier lieu par un algorithme hors ligne résumé ci-dessous :

- Le problème de commande optimale est transformé en un problème d'optimisation de paramètres, en utilisant les fonctions B-splines pour spécifier les trajectoires des sorties plates.
- Calcul des points de contrôle optimaux en utilisant une routine d'optimisation de type CFSQP (Constrained Feasible Sequential Quadratic Programming) Ces routines utilisent un algorithme d'optimisation basé sur les méthodes de programmation séquentielle quadratique avec un test de faisabilité. La programmation séquentielle quadratique est une technique itérative dans laquelle le critère est remplacé par une approximation quadratique et les contraintes par des approximations linéaires. Les routines CFSQP permettent de déterminer rapidement une solution vérifiant les contraintes, puis une solution quasi-optimale est calculée durant le temps alloué au calcul d'optimisation.
- Calcul des commandes associées en utilisant la propriété de platitude du système.

En deuxième lieu, utilisation d'un algorithme en ligne :

Le calcul d'une trajectoire complète de la configuration initiale à la configuration finale en une seule étape se révèle généralement impossible du fait de la complexité des calculs, mais aussi

l'environnement n'est que partiellement connu et doit être exploré au fur et à mesure que le robot effectue sa mission. Ainsi, la trajectoire doit être calculée graduellement au cours du temps. Ceci peut être réalisé par l'utilisation d'une stratégie basée sur un horizon glissant, dans laquelle des trajectoires sont calculées en résolvant des problèmes d'optimisation sur un horizon limité.

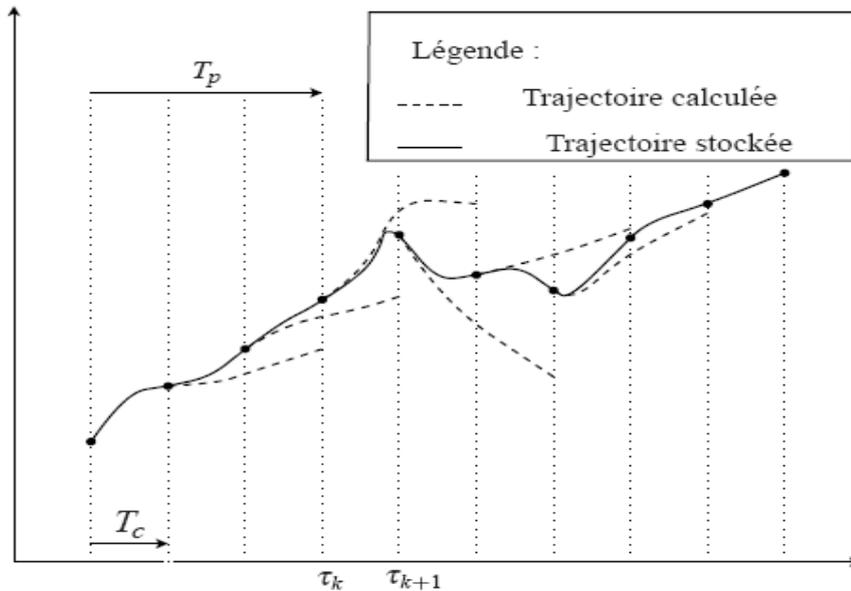


Fig.II.7-Horizons de calculs et de planification.

$T_p (> 0)$: horizon de planification

$T_c (> 0)$: horizon de calculs

$\tau_k (k \in \mathbb{N}, \tau_k = t_{\text{initial}} + k T_c)$: mises à jour

La résolution du problème de planification de trajectoire sur un horizon glissant s'effectue en deux étapes :

- une phase d'initialisation avant que le robot ne se déplace,
- une phase de calculs itératifs pendant le déplacement du robot.

II-6 Commande prédictive

La commande prédictive c'est la résolution répétée à chaque pas de temps T (T spécifiée par l'utilisateur) d'un problème de commande optimale : "comment aller de l'état actuel à un

objectif de manière optimale en satisfaisant des contraintes”. Pour cela il faut connaître à chaque itération l’état du système et utiliser un outil de résolution numérique.

On peut définir les grandes des lignes de la commande prédictive MPC (Model Predictive Control) :

1- Utilisation explicite d'un modèle pour prédire le comportement futur du système

2-Calcul d'une séquence d'échantillons futurs de commande minimisant une fonction de coût sur un horizon fuyant.

Définition d'un horizon fuyant

Seul le 1er élément de la séquence « optimale » précédente est appliqué sur le système. Tous les autres éléments peuvent être oubliés car à la période d'échantillonnage suivante, les séquences sont décalées, une nouvelle sortie est mesurée et la procédure complète est répétée. Ce procédé repose sur le principe de l'Horizon fuyant.

3-Seul le premier échantillon de commande est réellement appliqué au système. Toute la séquence est recalculée à chaque pas.

L'utilisation de la commande prédictive pour la conception des lois de commande est très attractive dû à la simplicité de son concept, son réglage intuitif, Prise en compte explicite des contraintes et son indépendance du modèle mathématique. La commande prédictive peut s'adapter à tout type de systèmes, des plus simples aux plus complexes (systèmes instables, avec retards, non minimum de phase, très peu amortis, multi-variables, non linéaires, variant).

Un problème qui n'est pas traité par la théorie de la commande prédictive mais qui a une importance pratique est celui de l'influence des temps de calculs. A un instant donné, on calcule la commande optimale partant du point actuel, ou plutôt on commence à calculer cette commande. En réalité elle ne sera disponible qu'une fois le calcul terminé. Entre temps on pourra utiliser l'optimum précédent. Mais cette commande n'est pas adaptée au problème. Si les temps de calculs (qui sont en outre variables) ne sont pas négligeables par rapport à δ temps d'application, on risque d'engendrer une instabilité dans le schéma de commande prédictive. Cette situation est résumée sur la figure II.8.

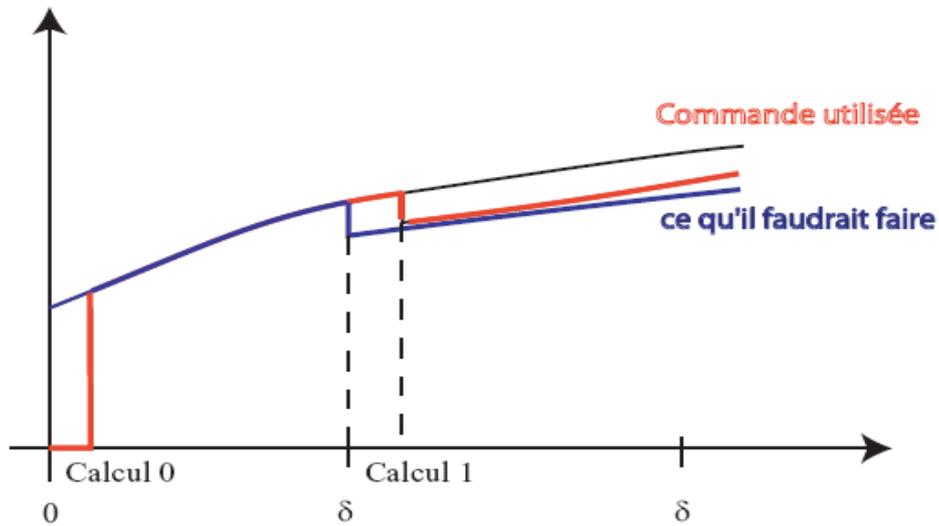


Fig.II.8 – Le problème des temps de calculs.

Il faut donc utiliser pour δ un temps qui soit grand devant les temps de calculs et court devant les constantes de temps du système. Ces deux contraintes se contredisent dans le cas des systèmes rapides. Dans (Alamir, 2007), où une étude brève sur les principes et les résultats de la commande prédictive non linéaire pour les systèmes à dynamique rapide est exposée. Cet article relate quelques pistes proposées par la communauté scientifique afin de tenir compte des contraintes de mises en œuvre en temps réel. Ceci explique le grand succès de la commande prédictive dans le domaine du génie des procédés et son succès plus que modeste dans le domaine de l'aéronautique (ceci est en train de changer avec l'arrivée de calculateurs embarqués réellement puissants).

Dans (Kühne, 2005), la commande prédictive est utilisée pour la commande d'un robot mobile. En un premier temps une étude est faite en utilisant une approche de commande prédictive non-linéaire. En second temps, il utilise la commande linéaire prédictive. L'idée consiste à utiliser une approche de linéarisation successive. Une comparaison entre les deux méthodes montre qu'il n'y a pas une grandes différence quand aux performances. Cependant, l'approche linéaire permet de réduire l'effort de calcul nécessaire pour résoudre les problèmes d'optimisation en temps réel.

II-7 Conclusion

Ce chapitre traite les différentes méthodes de commande optimale pour les robots mobile non holonome. Nous avons abordé la caractérisation des plus courts chemins, l'optimalité en termes de temps et d'énergie, les concepts de platitude, et enfin la commande prédictive.

Chapitre III

Commande optimale d'un robot mobile avec une architecture hybride

III-1 Introduction

Après avoir abordé dans le chapitre précédant les différentes méthodes de commande optimale pour la robotique mobile. Ce chapitre traite les stratégies d'optimisation utilisées en robotique mobile avec les architectures hybrides. Beaucoup de systèmes rencontrés en pratique impliquent l'utilisation de systèmes continus couplés à des éléments discrets. Ces systèmes aux dynamiques continues et discrètes coexistent et interagissent sont appelés systèmes hybrides. Une valve commandée va s'ouvrir et se fermer alternativement, les roues d'une voiture qui tantôt glissent tantôt roulent sur une route verglacés sont des exemples typiques de systèmes hybrides.

III-2 Architecture hybride

Une architecture hybride est l'assemblage de plusieurs contrôleurs, chacun doit commander le robot pour effectuer une tâche élémentaire (attraction vers la cible, évitement d'obstacles, etc). Le passage d'un contrôleur (continu) à un autre se fait suite à un événement discret. Ceci s'inscrit dans le cadre des architectures de contrôle comportementales proposées initialement par (Brooks, 1986) afin de briser la complexité de la tâche globale. Cette tâche est accomplie en coordonnant ces contrôleurs élémentaires. Deux mécanismes de coordination existent :

- Sélection d'actions (Brooks, 1986): Un seul contrôleur, sélectionné parmi les contrôleurs élémentaires, est appliqué à chaque instant.
- Fusion d'actions (Arakin, 1989): la commande appliquée au robot résulte d'une fusion de tout ou d'une partie des contrôleurs disponibles dans l'architecture de contrôle.

On s'intéresse au cas de la sélection d'actions, où un seul contrôleur va agir sur les actionneurs du robot. Plusieurs travaux de recherche se sont intéressés à construire des architectures de contrôle de type subsomption (Brooks, 1986), on peut citer :

Adouane propose une architecture multi-modes (Adouane, 2009a). Le robot doit atteindre une cible finale tout en évitant des obstacles. Cette tâche globale est décomposée en deux tâches élémentaires : Attraction vers la cible et évitement d'obstacles en utilisant les cycles limites

(Adouane, 2009b). La sélection d'un contrôleur se fait de manière hiérarchique. Cependant, ces passages d'un contrôleur à un autre en fonction de la tâche élémentaire à accomplir peuvent engendrer l'instabilité de la loi de commande si la commutation se fait de manière brusque. Pour résoudre ce problème, il est proposé d'introduire un mécanisme de contrôle adaptatif en agissant sur le gain de la loi de commande afin d'assurer la décroissance régulière d'une fonction de Lyapunov commune à tous les contrôleurs, et garantir ainsi la stabilité globale de l'architecture de contrôle.

III-3 Commande hybride pour la coordination d'actions en robotique mobile

Egerstedt (Egerstedt, 2002) parle des inconvénients des oscillations (le phénomène zéno) dû au switch secs entre les contrôleurs. Pour éviter ces inconvénients, il propose d'ajouter des nœuds dans l'automate pour les régulés, ce nœud engendre le mode glissant. La solution du mode glissant est donnée sous la forme :

$$f_s = \alpha f_{OA} + (1 - \alpha) f_{GA} \quad (23)$$

(OA : Obstacle Avoidance, GA : Goal Attraction), $\alpha \in [0 \ 1]$.

Le choix de α se fait de manière à ce que f_s soit orthogonale à f_{OA} , l'ajout de ce genre d'élément permet l'introduction du mode glissant lors de l'activation des deux contrôleurs au même temps. La méthode des champs de potentiels est appliquée pour l'évitement d'obstacle (figure III.1). L'obstacle fournit un champ répulsif tandis que l'objectif fournit un champ attractif. Au niveau de la surface de switch, qui correspond ici à la limite d'activation du contrôleur d'évitement d'obstacle, les deux champs pointent dans des sens contraires, ce qui activera le mode glissant. Cette solution constitue alors le nœud supplémentaire pour réguler l'automate et éviter ainsi le phénomène zéno (figure III.2). Cette méthode trouve sa limitation dès qu'il s'agit d'une architecture de contrôle à plus de deux contrôleurs, car l'ajout de modes glissants prenant en compte les nouveaux contrôleurs n'est pas simple à réaliser.

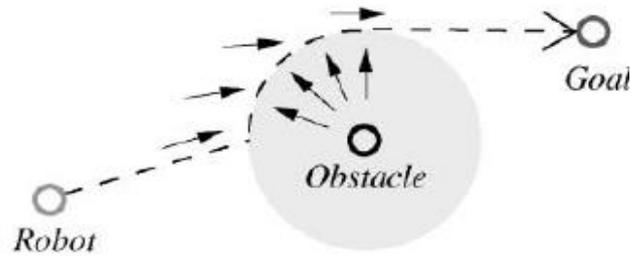


Fig.III.1- Attraction vers la cible et évitement d'obstacles par la méthode des champs de potentiels.

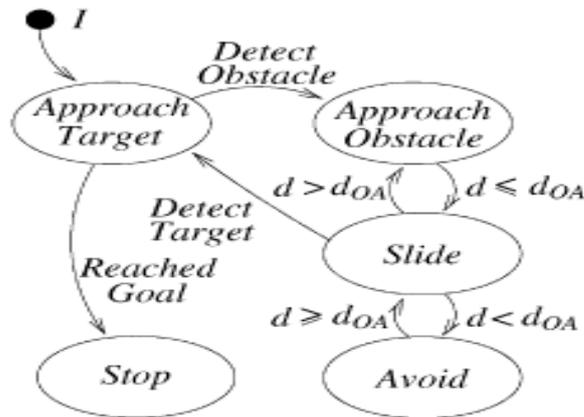


Fig.III.2- Régulation de l'automate par le mode glissant

Afin de construire une approche appropriée pour l'attraction vers la cible, il propose de faire de la planification et puis suivre cette trajectoire tout en minimisant une fonction coût :

$$J_s(u) = \min_{t \geq 0} \{(x(t) - x_{ob})^2 + (y(t) - y_{ob})^2\} \quad (24)$$

Avec $(x(t), y(t))$: position du robot en fonction du temps, (x_{ob}, y_{ob}) : position de l'obstacle à contourner, et $U_s(c)$ ensemble des lois de commande et $U_s(c) = \{u \in U : J_s(u) > c\}$.

Mais le problème est délicat à résoudre numériquement. Plutôt que de résoudre le problème de commande optimale, il minimise la courbure des trajectoires en utilisant les splines cubiques. Ainsi, il obtient une méthode en ligne pour la production de chemins à faible courbure autour des obstacles détectés.

III-4 Commande optimale en systèmes dynamiques hybrides

Boccadoro étudie dans (Boccadoro, 2005) un problème de contrôle optimal défini sur une classe de systèmes dynamiques hybrides. Le mode du système est modifié (commuté) chaque fois que la variable d'état traverse une certaine surface dans l'espace d'état, appelée surface de commutation. Ces surfaces de commutation sont paramétrées par des vecteurs de dimension finie appelé les paramètres de commutation.

Après une formulation de l'algorithme de descente du gradient dans le cas général, une application à un robot de type unicycle. Où (x,y) position du robot et θ l'angle d'orientation, v et w sont la vitesse linéaire et la vitesse angulaire respectivement. L'auteur s'est intéressé à la tâche d'atteindre une cible tout en évitant des obstacles, celle-ci est divisé en deux modes :

Mode 1 : attraction vers la cible

$$\begin{cases} v = 1 \\ w = C_1(\theta_g - \theta) \end{cases} \quad \text{Où } \theta_g = \frac{(y_g - y)}{(x_g - x)} \quad (25)$$

et C_1 est une constante positive.

Mode 2 : Evitement d'obstacle

$$\begin{cases} v = 1 \\ w = C_2(\theta - \theta_o) \end{cases} \quad \text{Où } \theta_o = \frac{(y_o - y)}{(x_o - x)} \quad (26)$$

et C_2 est une constante.

Le switch entre les deux modes se fait à chaque fois qu'une fonction g s'annule, cette fonction caractérise les surfaces de commutation, dont les paramètres de commutation sont les rayons a_1 et a_2 (Figure III.2.2).

$$g_i(x, y, a_i) = (x_0 - x)^2 + (y_0 - y)^2 + a_i^2 \quad \text{avec } i=1,2, \quad (27)$$

Ce switch se fait de manière à minimiser un critère J qui est fonction des rayons a_1 et a_2 , en pénalisant la distance entre la cible et le robot, ainsi que la proximité des obstacles.

$$J = \int_0^T ((x_g - x(t))^2 + (y_g - y(t))^2 + \beta e^{-\alpha[(x_0 - x(t))^2 + (y_0 - y(t))^2]}) dt, \quad (28)$$

Avec α et β deux constantes positives, a_1 et a_2 les rayons des cercles qui entourent l'obstacle où $a_1 \geq a_2$.

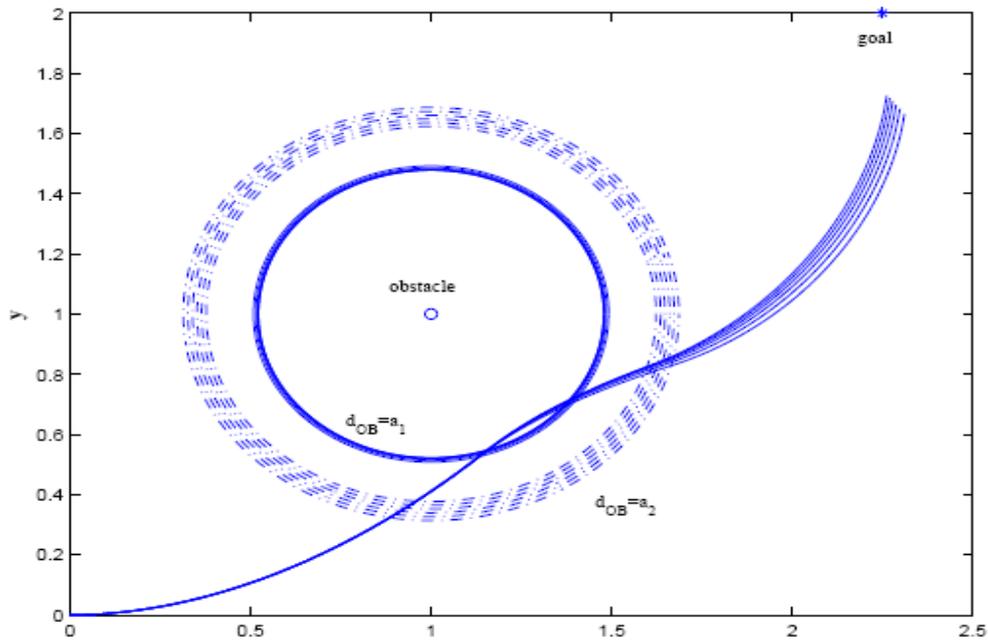


Fig.III.3- Exemple de trajectoire du robot en fonction des paramètres a_1 et a_2

La minimisation de J se fait par une optimisation numérique en utilisant l'algorithme de descente du gradient.

III-5 Conclusion

Dans ce chapitre, après avoir abordé les architectures hybrides et quelques travaux qui traitent cette problématique, nous nous sommes intéressés à l'application de la théorie de la commande optimale à ces architectures.

Chapitre IV

Commande hybride utilisée

IV-1 Introduction

Après avoir abordé dans le chapitre précédent les architectures hybrides, dans ce chapitre nous présentons l'architecture de commande que nous avons utilisée et la méthode adoptée pour la caractérisation et le calcul des plus courts chemins. Enfin, l'efficacité de cette stratégie de commande est illustrée par des exemples en simulation.

IV.2 Architecture proposée

On s'intéresse à la tâche d'atteindre une cible tout en évitant les obstacles de l'environnement. Cette tâche peut être divisée en deux tâches élémentaires (Figure IV.1) :

- attraction vers la cible.
- évitement d'obstacles.

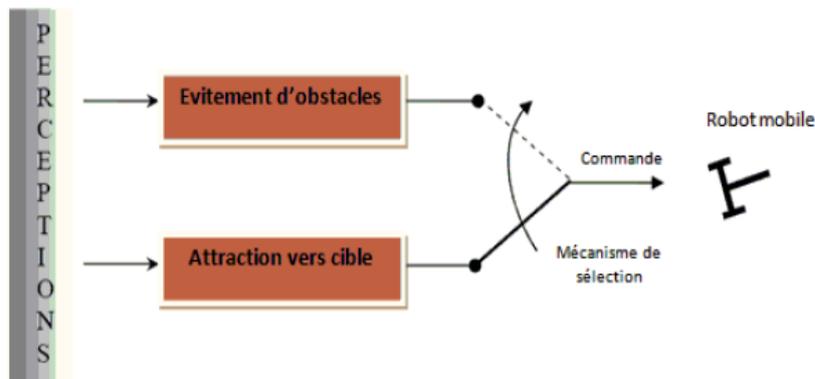


Fig.IV.1- Architecture de contrôle

Bien que notre intérêt se soit porté sur l'évitement d'obstacles, une brève description sera faite sur le contrôleur d'attraction vers la cible.

Le robot utilisé est un robot unicycle dont le modèle cinématique s'exprime dans un repère absolu (Fig.IV.2) par le système d'équation suivant :

$$\begin{aligned} \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega \end{aligned} \tag{29}$$

Où

- (x, y) sont les coordonnées du robot dans le repère absolu.
- θ est l'orientation du robot dans le repère.
- v et ω sont les vitesses linéaire et angulaire du robot respectivement.

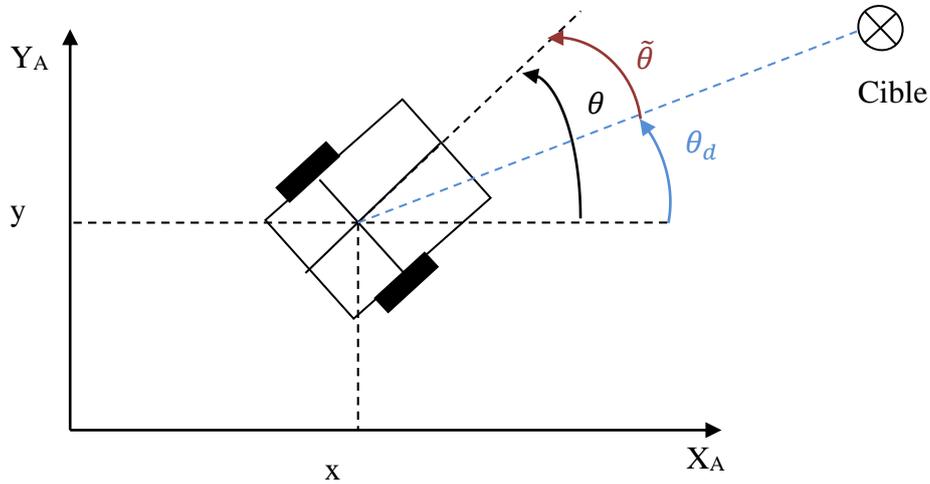


Fig.IV.2 Modèle du robot unicycle dans un repère cartésien

IV-3 Contrôleur d'attraction vers la cible

Le robot doit atteindre une cible donné de rayon R_g et de centre à coordonnées (x_g, x_g) exprimées dans le repère du robot. L'erreur angulaire $\tilde{\theta}$, telle que $\tilde{\theta} \in]\pi, \pi]$ est définit par :

$$\tilde{\theta} = \theta - \theta_d \quad (30)$$

La dérivée de l'erreur d'orientation est exprimé par :

$$\begin{aligned} \dot{\tilde{\theta}} &= \omega - \omega_d \\ \dot{\tilde{\theta}} &= \omega - v \frac{\sin \tilde{\theta}}{d} \end{aligned} \quad (31)$$

Où ω_d représente la dérivée de l'angle désiré du robot par rapport à la cible et d la distance entre le centre de l'essieu arrière du robot et la cible.

IV-4 Contrôleur d'évitement d'obstacles

L'objectif de ce contrôleur est de commander le robot pour éviter les obstacles qui gênent son attraction vers la cible (Adouane, 2009b). Ce contrôleur est basé sur des cycles limites (Kim, 2003), (Jie M.S., 2006).

La trajectoire désirée du robot est régit par le système d'équations différentielles suivant :

$$\begin{cases} \dot{x}_s = \alpha y_s + x_s(R_c^2 - x_s^2 - y_s^2) \\ \dot{y}_s = -\alpha x_s + y_s(R_c^2 - x_s^2 - y_s^2) \end{cases} \quad (32)$$

Avec $\alpha = \pm 1$ selon la direction optimale d'évitement (sens horaire ou antihoraire). (x_s, y_s) représente les coordonnées du robot par rapport au centre de l'obstacle, qui est entouré d'un cercle de rayon R_c tenant compte de la dimension du robot et d'un marge de sécurité. La figure IV.3 montre le principe d'évitement d'obstacles, et comment se fait le choix du sens α selon la position du robot par rapport au repère relatif à l'obstacle, qui est dirigé vers la cible.

Les étapes à suivre pour éviter un obstacle avec la méthode des cycles-limites, dans le repère d'évolution du robot sont les suivantes :

- définir la position de l'obstacle $(x_{obstacle}, y_{obstacle})$ ainsi que son rayon $r_{obstacle}$,
- déterminer la présence ou non d'un obstacle sur la trajectoire du robot mobile,
- chercher la position du robot dans le repère de l'obstacle,
- déterminer le cap θ_d à suivre pour éviter l'obstacle.

Le cap θ_d calculé lorsque le robot mobile rencontre un obstacle est défini par :

$$\theta_d = \tan^{-1} \frac{\dot{y}_s}{\dot{x}_s} \quad (33)$$

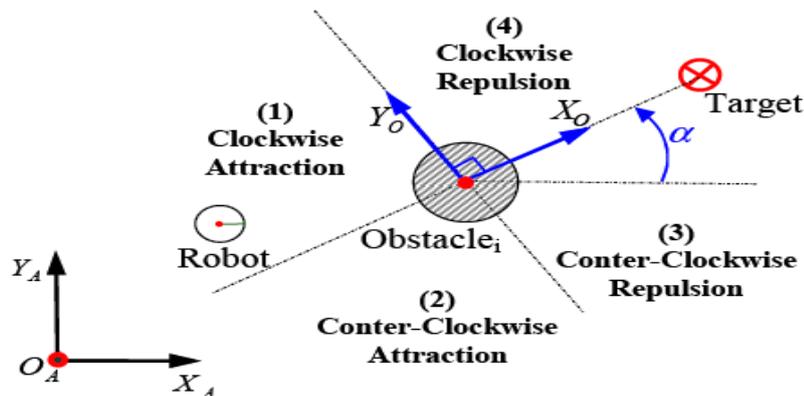


Fig.IV.3- Principe d'évitement d'obstacle

La figure IV.4 montre l'évolution possible des trajectoires du système d'équations (34). La trajectoire de tout point (x_s, y_s) inclus à l'intérieur du cercle, se déplace vers le cercle de rayon $R_c = 1$ dans le sens trigonométrique en (a) et anti-trigonométrique en (b).

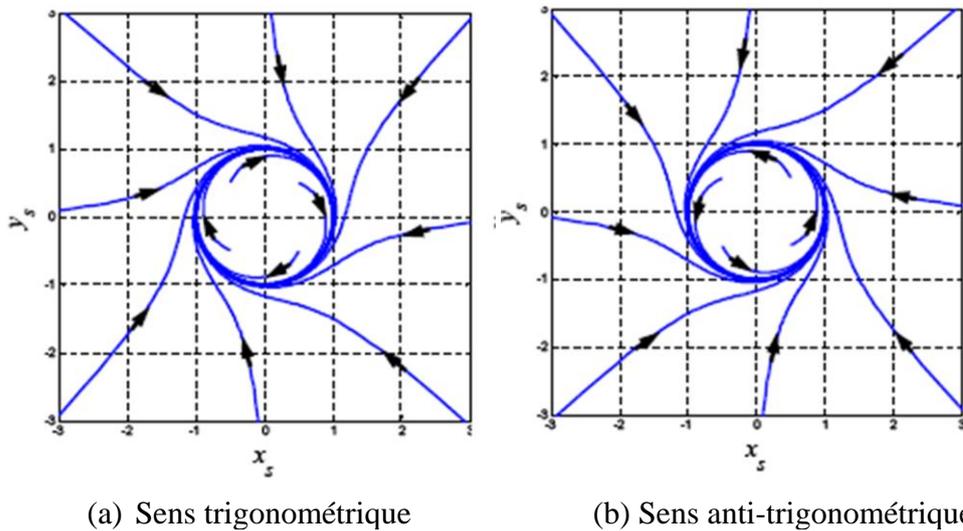


Fig.IV.4- forme possible du cycle limite en fonction de la position du robot

IV-5 Optimalité du cycle limite

Dans cette partie nous allons montrer la stratégie qu'on choisit pour la caractérisation des plus courts chemins lors de l'évitement d'obstacles.

Par la suite et afin de simplifié les calculs on prendra $\alpha = 1$, sachant que le raisonnement reste le même dans le cas où $\alpha = -1$. La seule chose qui change est le sens d'évitement, on se met dans le cas où l'évitement se fait dans le sens trigonométrique.

On a introduit à l'équation (34) un terme $\mu > 0$, qui modifie l'angle d'incidence de la trajectoire vers le cercle limite de l'obstacle.

$$\begin{cases} \dot{x}_s = \alpha y_s + \mu x_s (R_c^2 - x_s^2 - y_s^2) \\ \dot{y}_s = -\alpha x_s + \mu y_s (R_c^2 - x_s^2 - y_s^2) \end{cases} \quad (34)$$

La figure IV.5 nous donne quelques exemples sur l'influence de μ sur la variation de la courbure de la trajectoire des cycles limites, et ce en prenant comme conditions initiales :

$$x_{s0} = -5, y_{s0} = 0 \text{ et } R_c = 1.$$

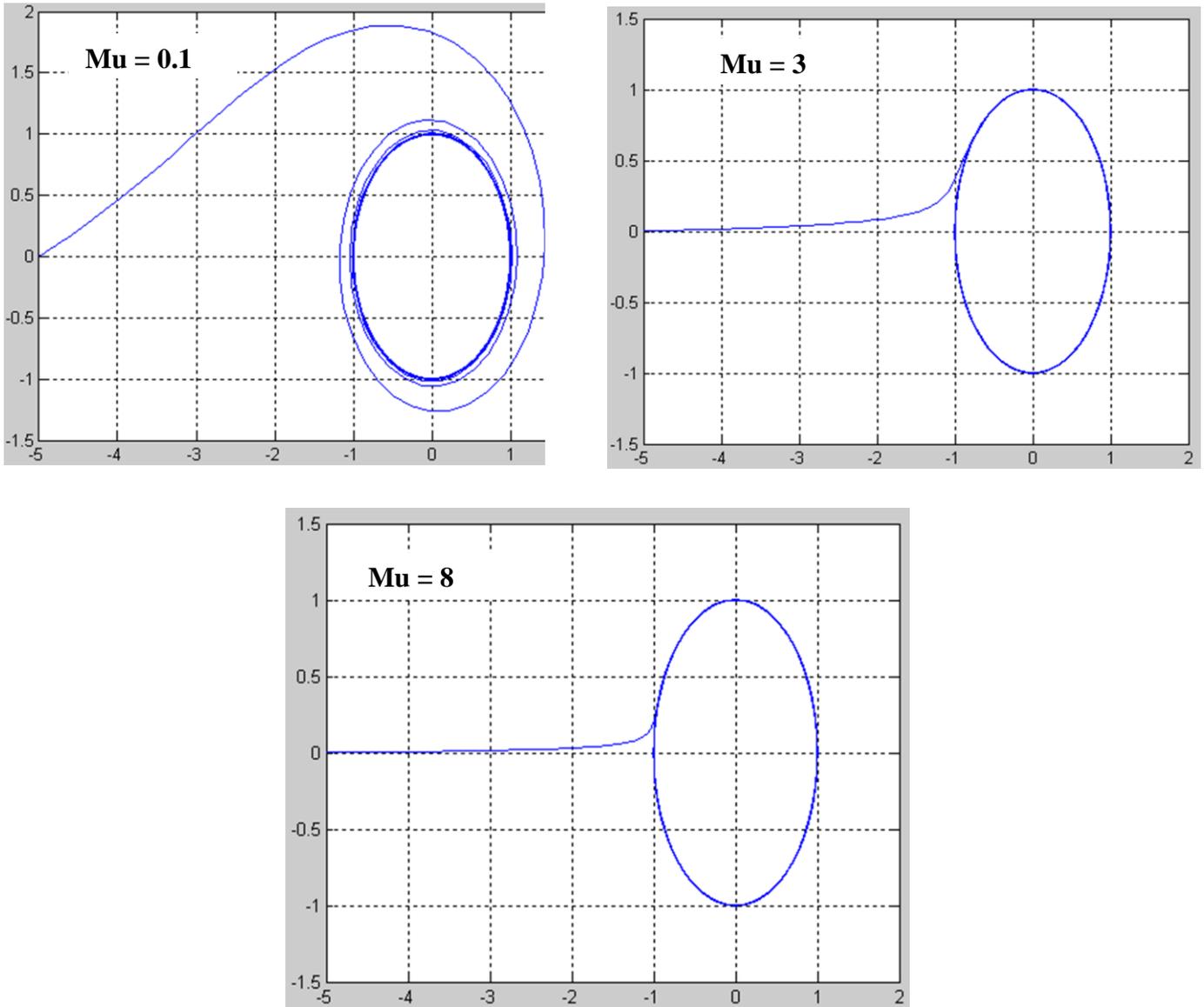


Fig.IV.5- Influence de μ sur l'incidence de la trajectoire des cycles limites

IV-6 Longueur de la trajectoire du cycle limite

L'idée est de déterminer le μ qui nous donne la trajectoire minimale en distance curviligne que doit suivre le robot lors de l'évitement d'obstacles. Pour cela il faut calculer cette longueur en fonction de μ .

IV-6-1 Définition de la longueur des courbes

La longueur d'un arc simple plan AB parcouru par le point mobile $M(t)$ lorsque "t" varie de "a" à "b" est donné par :

$$L = \int_a^b \sqrt{\dot{x}^2(u) + \dot{y}^2(u)} du \quad (35)$$

IV-6-2 Calcul de la longueur du cycle limite

Equation du cycle limite :

$$\begin{cases} \dot{X}_s = Y_s + \mu * X_s * (Rc^2 - X_s^2 - Y_s^2) \\ \dot{Y}_s = -X_s + \mu * Y_s * (Rc^2 - X_s^2 - Y_s^2) \end{cases} \quad (36)$$

Afin de simplifier les calculs, on fait un changement de variable en utilisant les coordonnées polaires :

$$\begin{cases} X_s = R * \cos \alpha \\ Y_s = R * \sin \alpha \end{cases} \quad (37)$$

En remplaçant (37) dans (36) on a :

$$\begin{cases} \dot{X}_s = R * \sin \alpha + \mu * R * \cos \alpha * (Rc^2 - R^2 * \cos^2 \alpha - R^2 * \sin^2 \alpha) \\ \dot{Y}_s = -R * \cos \alpha + \mu * R * \sin \alpha * (Rc^2 - R^2 * \cos^2 \alpha - R^2 * \sin^2 \alpha) \end{cases} \quad (38)$$

Après simplification on trouve l'expression (39) :

$$\begin{cases} \dot{X}_s = R * \sin \alpha + \mu * R * \cos \alpha * (Rc^2 - R^2) \\ \dot{Y}_s = -R * \cos \alpha + \mu * R * \sin \alpha * (Rc^2 - R^2) \end{cases} \quad (39)$$

Or en dérivant (37) on a :

$$\begin{cases} \dot{X}_s = \dot{R} * \cos \alpha - R * \sin \alpha * \dot{\alpha} \\ \dot{Y}_s = \dot{R} * \sin \alpha + R * \cos \alpha * \dot{\alpha} \end{cases} \quad (40)$$

De (39) et (40) nous avons l'expression :

$$\dot{R} * \cos \alpha - R * \sin \alpha * \dot{\alpha} = R * \sin \alpha + \mu * R * \cos \alpha * (Rc^2 - R^2) \quad (41 - a)$$

$$\dot{R} * \sin \alpha + R * \cos \alpha * \dot{\alpha} = -R * \cos \alpha + \mu * R * \sin \alpha * (Rc^2 - R^2) \quad (41 - b)$$

De (41-a) on a :

$$\begin{cases} \dot{R} * \cos \alpha = \mu * R * \cos \alpha * (Rc^2 - R^2) \\ -R * \sin \alpha * \dot{\alpha} = R * \sin \alpha \end{cases} \quad (42-a)$$

Et de (41-b) :

$$\begin{cases} \dot{R} * \sin \alpha = \mu * R * \sin \alpha * (Rc^2 - R^2) \\ R * \cos \alpha * \dot{\alpha} = -R * \cos \alpha \end{cases} \quad (42-b)$$

D'où :

$$\begin{cases} \dot{R} = \mu * R * (Rc^2 - R^2) \\ -R * \dot{\alpha} = R \end{cases} \quad (43)$$

$$(43) \Rightarrow \begin{cases} \dot{R} = \mu * R * (Rc^2 - R^2) \\ \dot{\alpha} = -1 \end{cases} \quad (44)$$

On obtient donc de (44) les équations des cycles limites en coordonnées polaires. La figure IV.6 montre bien qu'on obtient le même cycle limite qu'en coordonnées cartésiennes. Pour mieux comprendre le choix des bornes d'intégrale, la figure IV.7 montre les positions des angles α_i et $\alpha_f = \pi/2$ pour une configuration initiale du robot E_i et une configuration finale E_f .

Fig .IV.6- Comparaison des équations du cycle limite en coordonnées cartésiennes et polaires ($R_c=1$; $Mu=1$; $R_0 = -5$; $\alpha_0 = 2$)

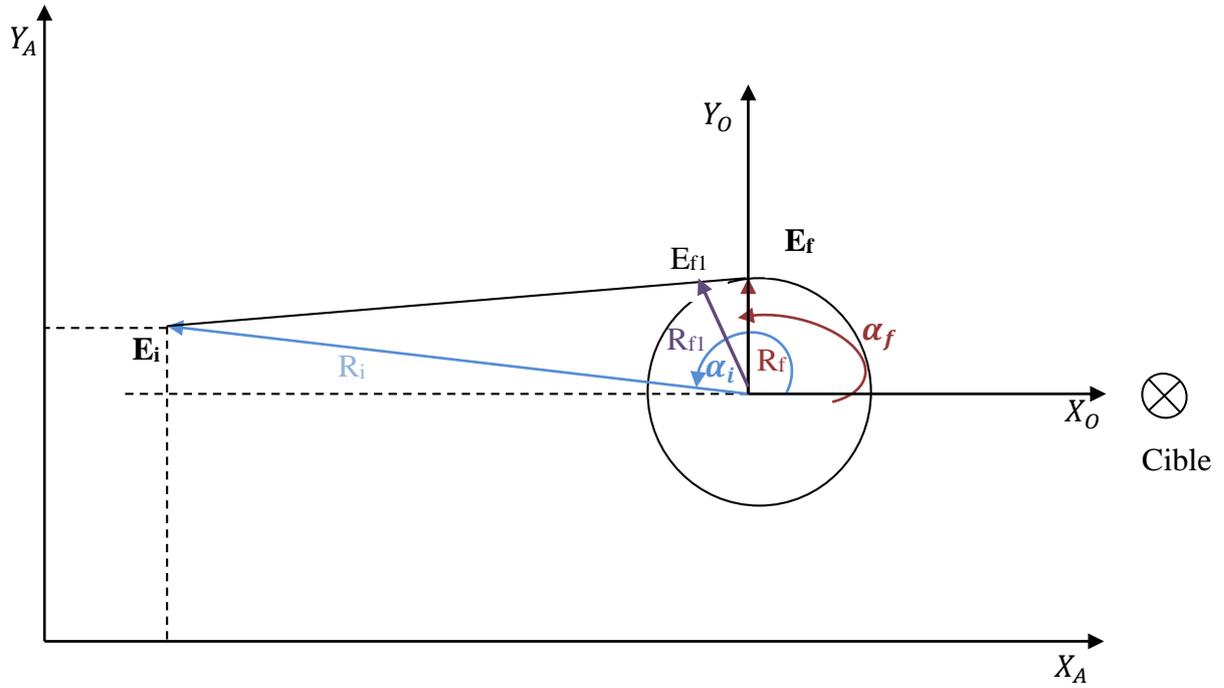


Fig.IV.7 Schéma illustrant l'angle initial et l'angle final

Pour calculer la longueur de la courbe, on utilise la relation (35)

En remplaçant par (40) dans (35) :

$$\text{Longueur} = \int_{\alpha_i}^{\alpha_f} \sqrt{(\dot{R} * \cos \alpha + R * \sin \alpha)^2 + (\dot{R} * \sin \alpha - R * \cos \alpha)^2} d\alpha$$

$$\text{Longueur} = \int_{\alpha_i}^{\alpha_f} \sqrt{(\dot{R} * \cos \alpha)^2 + (R * \sin \alpha)^2 + (\dot{R} * \sin \alpha)^2 + (R * \cos \alpha)^2} d\alpha$$

$$\text{Longueur} = \int_{\alpha_i}^{\alpha_f} \sqrt{\dot{R}^2 * (\sin^2 \alpha + \cos^2 \alpha) + R^2 * (\sin^2 \alpha + \cos^2 \alpha)} d\alpha$$

$$\text{Longueur} = \int_{\alpha_i}^{\alpha_f} \sqrt{\dot{R}^2 + R^2} d\alpha \quad (45)$$

En remplaçant \dot{R} en (45) par sa valeur obtenue en (44) on a :

$$\text{Longueur} = \int_{\alpha_i}^{\alpha_f} \sqrt{(\mu * R * (Rc^2 - R^2)^2 + R^2)} d\alpha \quad (46)$$

Mais pour pouvoir calculer cette intégrale curviligne, il faudrait connaître la relation entre α et R , ceci n'est pas évident dans le cas de la trajectoire d'un cycle limite. Pour résoudre ce souci, on doit recourir à un changement de variable.

On a de (44) :

$$\begin{cases} \dot{\alpha} = -1 \\ \dot{R} = -\dot{\alpha} * \mu * R * (Rc^2 - R^2) \end{cases} \quad (47)$$

On peut déduire pour $R_c \neq R$:

$$d\alpha = \frac{-dR}{\mu * R * (Rc^2 - R^2)} \quad (48)$$

$$Longueur = \int_{R_f}^{R_0} \sqrt{1 + \frac{R^2}{(\mu * R * (Rc^2 - R^2))^2}} dR \quad (49)$$

Comme on a vu dans la figure V.5, plus μ est grand plus notre cycle limite tend vers le cercle limite, on aura alors $R_c = R$. Pour pouvoir calculer notre longueur dans ce cas, l'idée est de partager notre intégrale en deux parties :

- Première partie pour $R_{f1} = R_c + \epsilon$ (avec ϵ une constante qui tend vers 0)

$$Longueur1 = \int_{R_{f1}}^{R_0} \sqrt{1 + \frac{R^2}{(\mu * R * (Rc^2 - R^2))^2}} dR \quad (50)$$

- Deuxième partie pour $R_f = R_c$

De (47) on a Si $R = R_c$ Alors $\dot{R} = 0$

Et de (45) on a :

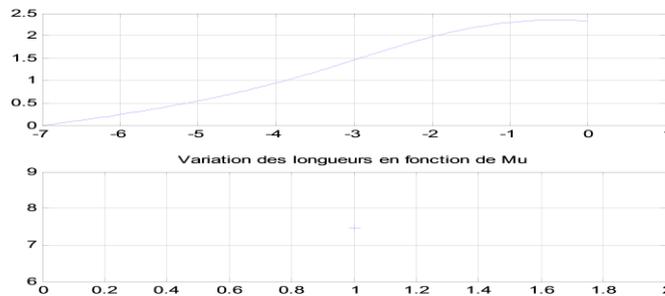
$$Longueur2 = \int_{\alpha_{f1}}^{\alpha_f} R d\alpha \quad (51)$$

Et dans ce cas le problème de la relation entre R et α ne se pose plus, on est dans le cas d'un cercle. La longueur dans ce cas est la somme de *Longueur1* et *Longueur2*.

IV-7 Résultats de simulation

Le calcul de la longueur s'est fait sous matlab de façon numérique. Dans la simulation nous avons utilisé différentes configurations de l'état initial et du rayon du cercle limite. Pour confirmé nos résultats de calcul de longueur nous avons fait une approximation de cette longueur. La figure IV.8 montre l'évolution des longueurs en fonctions de μ pour les conditions initiales : $R_0 = 7$ et $\alpha_0 = \pi$ et un obstacle de rayon $R_c = 2$. La figure IV.9 montre l'évolution des longueurs en fonction des différents Mu pour les mêmes conditions que dans IV.8. Dans ce cas on a $\text{Mu optimale} = 0.2$.

Pour $\text{Mu} = 0.1$ on a Longueur = 7.4580



Pour $\text{Mu} = 1$ on a Longueur = 7.6870

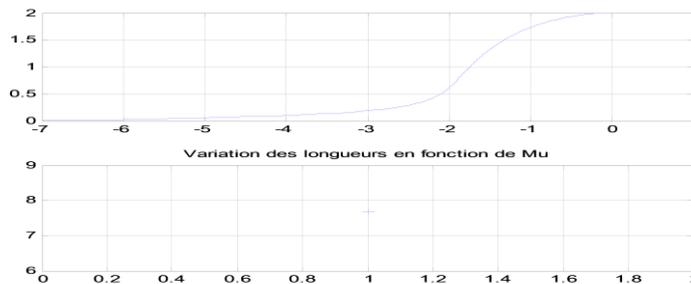


Fig.IV.8- Calcul de longueur pour $\text{Mu} = 0.1$ et $\text{Mu} = 1$

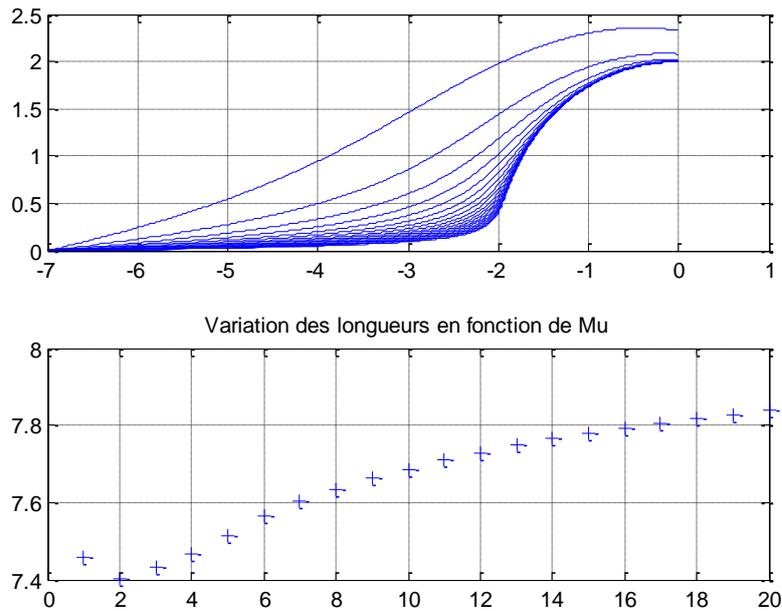


Fig.IV.9 Evolution de la longueur en fonction de $\text{Mu}=0.1*i, i \in \{1,2,\dots,20\}$

On a remarqué lors des différentes configurations que la valeur de Mu optimale dépend des conditions initiales et du rayon R_c du cercle limite vers lequel converge notre cycle limite. La figure IV.10 montre l'évolution de la longueur en fonction de Mu pour les mêmes conditions initiales mais pour $R_c = 5$. On remarque que plus R_c augmente plus Mu optimal tend à être de plus en plus petit.

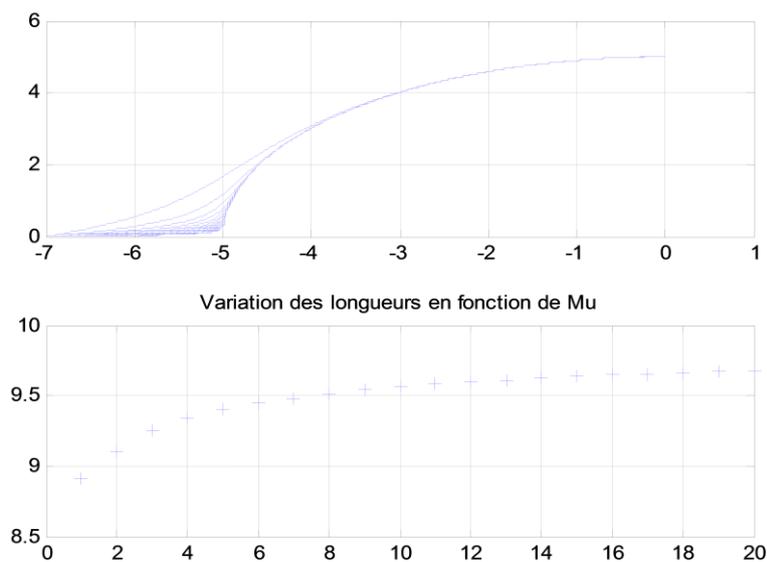


Fig.IV.10 Evolution de la longueur en fonction de $\text{Mu}=0.1*i, i \in \{1,\dots,20\}$ et $R_c = 5$

IV-7-1 Optimisation de Mu

Afin d'obtenir Mu optimal, c'est-à-dire celui qui minimise la longueur de la trajectoire à suivre, on fait une interpolation polynômiale au sens des moindres carrés des Mu en fonction des longueurs déjà calculé, puis on fait l'optimisation en utilisant les fonctions que nous offre le Toolbox d'optimisation de Matlab. Dans le figure IV.11, on voit l'évolution de la longueur en fonction des les conditions initiales $R_0 = 7$, $\alpha_0 = \pi$ et $R_c = 2$. Comme on a 5 points on doit faire l'interpolation par un polynôme d'ordre 4, après l'optimisation on obtient $Mu = 0.99$ donc l'erreur n'est pas conséquente si on compare au résultat obtenu en trouvant le minimum du vecteur de Mu mais cette méthode montre rapidement ses limitations dès qu'on commence à augmenté le degré du polynôme d'interpolation et à faire un pas d'échantillonnage plus petit pour avoir une meilleur précision quand à la valeur de Mu optimale en fonction de la longueur.

Fig.IV.11 Variation de la valeur "Longueur" en fonction de Mu

La figure IV.12, montre l'évolution de la longueur en fonction de la valeur du rayon du cercle limite R_c et de μ . On constate que plus R_c est grand plus la longueur optimale est pour μ petit.

IV.12- Evolution de la longueur en fonction de R_c et μ

IV-7-2 Exemple de simulation

Afin de montrer l'intérêt du choix optimal de μ lors de l'évitement d'obstacles, on se propose de simuler le mouvement d'un robot mobile pour différentes valeurs de μ et calculer à chaque fois le temps que prend le robot pour atteindre la cible.

Les figures IV.12 jusqu'à IV.17 montre l'évolution du mouvement du robot mobile en fonction de μ . La première figure montre la trajectoire du robot en fonction de μ optimal qui est calculé à chaque fois selon l'état du système, c'est-à-dire la position initiale du robot et le rayon du cercle limite que doit suivre le robot, et qui sont donnés par le fonction $f((X_{robot}, Y_{robot}), R_c)$. Le reste des figures montre l'évolution du robot en fonction de μ constant. Le tableau ci-dessous résume les valeurs de t qui est le temps que prend le robot pour atteindre sa cible.

| μ | $f((X_{robot}, Y_{robot}), R_c)$ | 0.2 | 0.5 | 0.8 | 1.5 | 3 |
|-------------------------|----------------------------------|-------|-------|-------|-------|-------|
| t (seconde) | 121.0 | 122.6 | 122.4 | 123.8 | 125.3 | 127.5 |

Ces résultats confirment bien que le choix de μ optimale réduit le temps du parcours du robot.

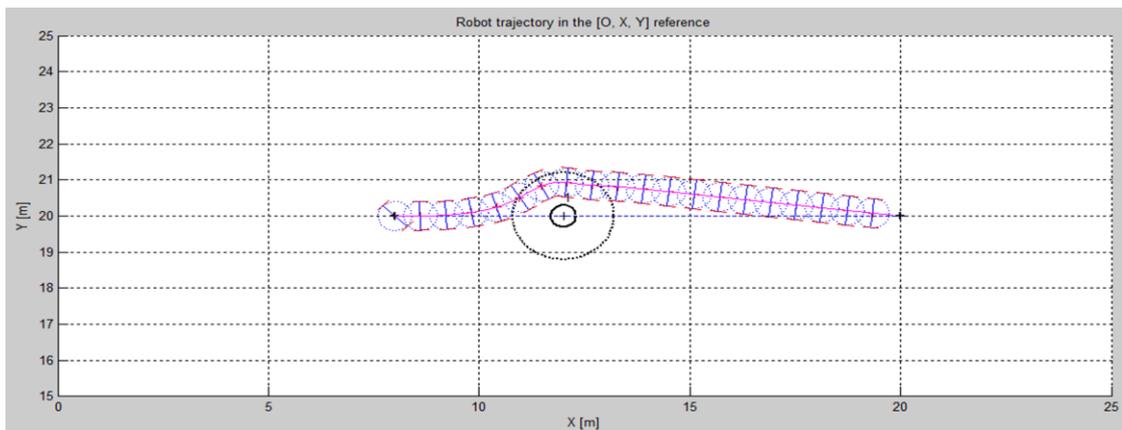


Fig.IV.12 Evolution du robot pour $\mu = f((X_{robot}, Y_{robot}), R_c)$

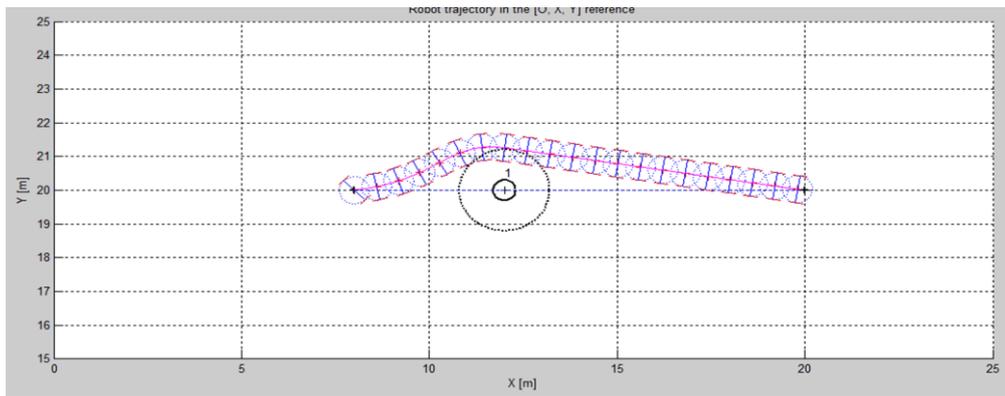


Fig.IV.13 Evolution du robot pour $\mu = 0.2$

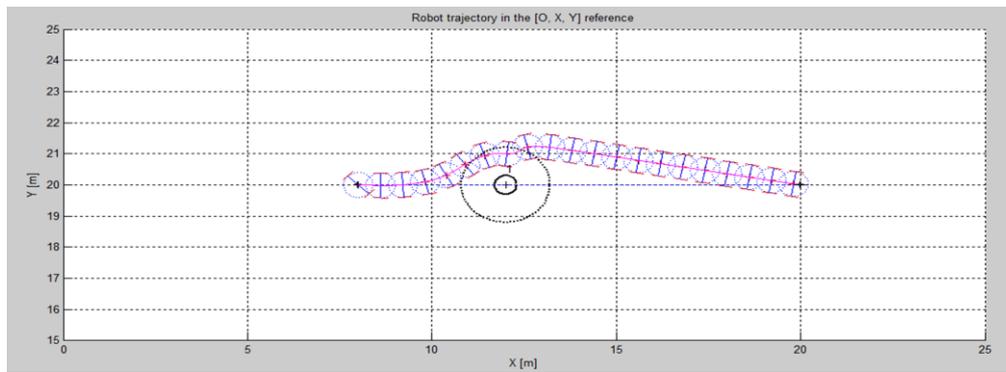


Fig.IV.14 Evolution du robot pour $\mu = 0.5$

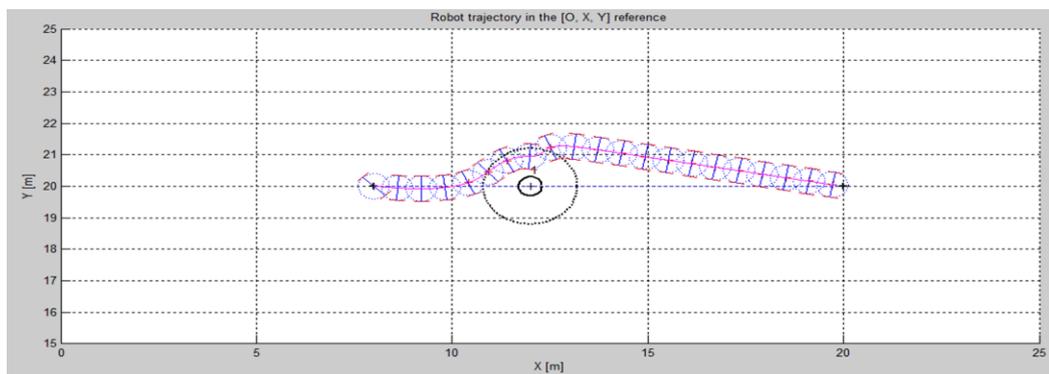


Fig.IV.15 Evolution du robot pour $\mu = 0.8$

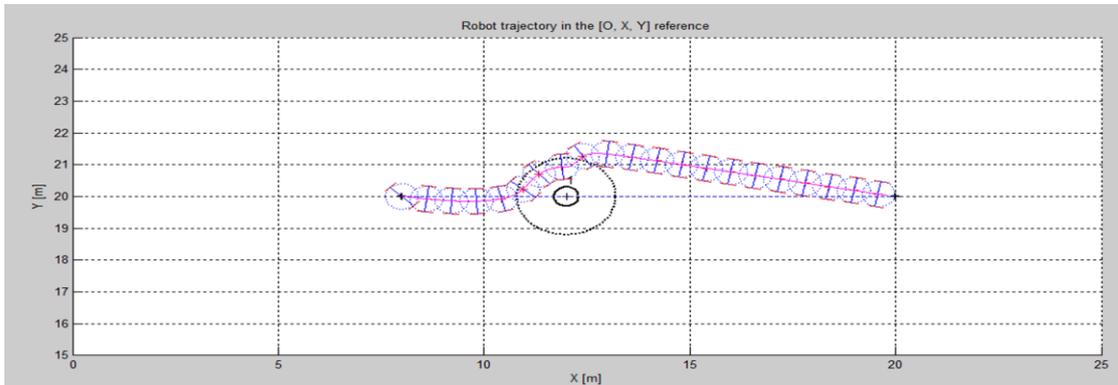


Fig.IV.16 Evolution du robot pour $\mu = 1.5$

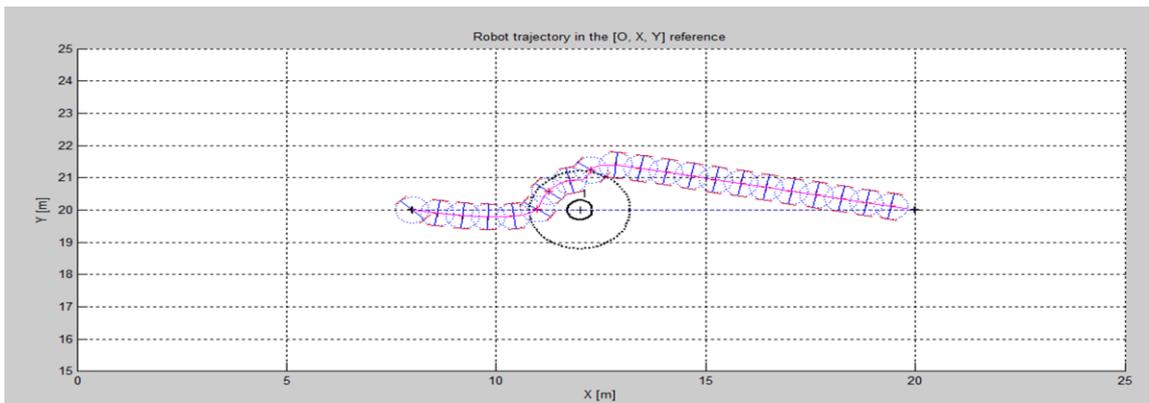


Fig.IV.17 Evolution du robot pour $\mu = 3$

La figure IV.19 montre l'évolution de μ optimale en fonction du temps. On voit que de plus en plus le robot s'approche de l'obstacle μ optimale augmente, ce qui est logique car l'angle d'incidence Φ (figure IV.18) vers le cercle limite doit être plus important. Dans la deuxième phase d'évitement d'obstacle le μ optimale diminue ce qui s'explique par le fait que le robot s'éloigne de l'obstacle, donc l'angle d'incidence vers le cercle limite diminue. Enfin le $\mu = 0$ c'est lorsque le robot est dans la phase d'attraction vers la cible.

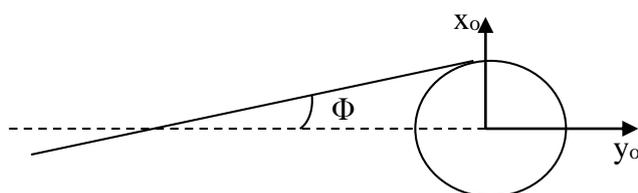


Fig.IV.18 Schéma illustrant l'angle d'incidence vers le cercle limite.

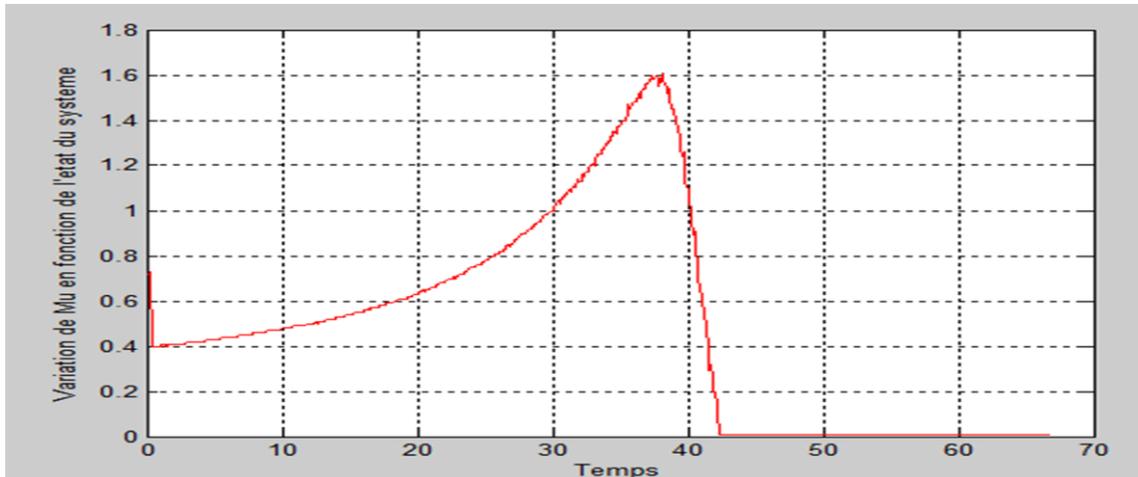


Fig.IV.18 Variation de Mu optimale en fonction du temps.

IV-8 Conclusion

Dans ce chapitre nous avons développé, dans la cadre de l'évitement d'obstacles, la caractérisation et le calcul du plus court chemin. Nous avons illustré par ailleurs l'efficacité de cette stratégie de commande par des résultats de simulation.

Conclusion générale et perspectives

Durant ce stage de master recherche, plusieurs éléments liés à la commande optimale des robots mobiles ont été abordés. L'état de l'art réalisé sur les travaux proposés dans la littérature nous a montré l'engouement de la communauté scientifique pour la planification de trajectoires optimales. Ceci stipule donc la connaissance a priori totale de l'environnement dans lequel évolue le robot. Pour garantir la capacité du robot à réagir de façon réactive, nous nous sommes plutôt orienté vers une architecture de commande hybride (continu-discrète). Plus spécifiquement, on a proposé de trouver la longueur minimale du cycle limite à suivre lors de l'évitement d'obstacles. Ceci en trouvant le meilleur paramètre μ qui modifie l'angle d'incidence vers le cercle limite à suivre.

En perspective, il faut prendre en considération le cap du robot et recalculer à chaque changement de cap le nouveau μ qui minimise la longueur du nouveau cycle limite à suivre. Ensuite, il serait intéressant de faire une étude comparative entre celle-ci et celle déjà faite dans le cas où on détermine le μ optimale qui caractérise la trajectoire à suivre pour conclure sur la meilleure stratégie à prendre. Enfin, il serait également intéressant de mettre en œuvre ces travaux dans un environnement dynamique.

Bibliographie

Adouane L. (a) An adaptive multi-controller architecture for mobile robot navigation. *10th IAS, Intelligent Autonomous Systems, 2008*, 23-25

Adouane L. (b) Orbital obstacle avoidance algorithm for reliable and online mobile robot navigation. - Portugal : 9 th Conference on Autonomous Robot Systems and Competitions, 2009.

Alamir M. La commande prédictive pour systèmes non linéaires à dynamiques rapides. - Strasbourg : JNNR 07, 2007.

Ancenay A. and Frederic,M. A Time and Energy Optimal Controller // Springer-Verlag Berlin Heidelberg. - 2004. - pp. AI 2004, LNAI 3339, pp. 1181–1186.

Arakin R.C. Motor schema-based mobile robot navigation // International Journal of Robotics Research . - 1989. - pp. vol.8,no. 4,pp.92-112.

Boccardo M. Wardi,Y. Egerstedt.M. and Verriest,E. Optimal Control of Switching Surfaces in Hybrid Dynamical Systems// Springer Netherlands, Journal of Discrete Event Dynamic Systems. - 2005. - pp. Volume 15, Number 4, PP.433-448.

Brooks R.A. A robust layered control system for a mobile robot// IEEE Journal of Robotics and Automation RA-2. - 1986. - pp. pp.14-23.

Chevallereau, C. and Adouane, L. On line reference trajectory adaptation for the control of a planar biped, *5th international Conference on Climbing and Walking Robots CLAWAR, 2002*, 427-435.

Defroot M. Contribution à la planification et à la commande pour les robots mobiles coopératifs: thèse de l' Université de l'école centrale de Lille, 2007.

Devinemeux J.P. Les robots dans l'industrie. Hermes, 1989.

Dubins L.E. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents// American Journal of Mathematics. - 1957. - pp. 79:497-516.

Egerstedt M. Hu,X. A hybrid control approach to action coordination for mobile robots// Automatica 38(1). - 2002. - pp. 125-130.

Jie M.S. Baek J.H., Hong Y.S., and K.Lee Woong Real time obstacle avoidance for mobile robot using limit-cycle and vector field method// Knowledge-Based Intelligent Information and Engineering Systems. - 2006. - pp. pp. 866-873.

Kim D. and Kim,J A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer// Robotics and Autonomous Systems,42. - 2003. - pp. 17-30.

Kühne F. Fetter Lages,W. and Gomes da Silva Jr,J.M. MOBILE ROBOT TRAJECTORY TRACKING USING MODEL PREDICTIVE // VII SBAI / II IEEE LARS. - São Luís , 2005.

Pontriaguine L. V. Boltianski, G. Gamkrelidze, and E. Michtchenko Théorie mathématique des processus optimaux . - Moscou : Mir, 1974.

Shepp J.A. Reeds and L.A. Optimal paths for a car that goes both forwards and backwards // Pacific Journal of Mathemathecs. - 1990. - pp. 145(2):367-393.

Souères P. Commande optimale et robots mobiles non-holonomes: thèse HDR de l'Université Paul Sébastien.1554, 1993.