



HAL
open science

Accrochage immatériel sûr et précis de véhicules automatiques

Jano Yazbeck

► **To cite this version:**

Jano Yazbeck. Accrochage immatériel sûr et précis de véhicules automatiques. Intelligence artificielle [cs.AI]. Université de Lorraine, 2014. Français. NNT : 2014LORR0070 . tel-01750793v2

HAL Id: tel-01750793

<https://inria.hal.science/tel-01750793v2>

Submitted on 15 Oct 2014

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

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



Accrochage immatériel sûr et précis de véhicules automatiques

THÈSE

présentée et soutenue publiquement le 10 juin 2014

pour l'obtention du

Doctorat de l'université de Lorraine

(spécialité informatique)

par

Jano Yazbeck

Composition du jury

<i>Président :</i>	Rochdi Merzouki	Professeur des Universités, Lille 1
<i>Rapporteurs :</i>	Rachid Alami Jean-Pierre Derutin	Directeur de recherche CNRS, LAAS - Toulouse Professeur, Université Blaise Pascal de Clermont Ferrant
<i>Examineurs :</i>	Franck Gechter Oussama Khatib Françoise Simonot-Lion	Maître de Conférences HDR, Université de Technologie de Belfort-Montbéliard Professor of Computer Science, Stanford University Professeur émérite, Université de Lorraine
<i>Directeur de thèse :</i>	François Charpillet	Directeur de Recherche, Loria - Inria
<i>Co-Directeur de thèse :</i>	Alexis Scheuer	Maître de conférences, Université de Lorraine

Mis en page avec la classe thloria.

Remerciements

Table des matières

Table des figures xi

Notations et définitions	1
--------------------------	---

Chapitre 1

Introduction

- 1.1 Le projet Européen InTraDE 5
- 1.2 Objectifs, contributions et plan de la thèse 6

Chapitre 2

État de l'art

- 2.1 Projets de recherche autour du suivi de convoi 11
 - 2.1.1 Le projet *PATH* 12
 - 2.1.2 Le projet *Cristal* 13
 - 2.1.3 Le projet *Sartre* 14
 - 2.1.4 *VIAC* : Le défi intercontinental pour véhicules autonomes de VisLab 15
- 2.2 Architectures de convoi 16
 - 2.2.1 Approches globales 17
 - 2.2.2 Approches locales 18
- 2.3 Approches de contrôle analytiques 19
 - 2.3.1 Suivi de trajectoires 20
 - 2.3.1.1 Les approches basées sur les méthodes physiques 20
 - 2.3.1.2 Les approches représentant la formation comme une structure virtuelle 23
 - 2.3.1.3 Les approches basées sur la théorie des graphes 24
 - 2.3.1.4 Les approches fondées sur le comportement (behavior-based approaches) 25

2.3.1.5	Les approches utilisant des modèles cinématiques	26
2.3.2	Décomposition chemin-vitesse	26
2.3.2.1	Contrôle longitudinal	27
2.3.2.2	Contrôle latéral	35
2.4	Méthodes d'estimation de chemins ou de trajectoires	38
2.4.1	Chemin de Dubins	39
2.4.2	Chemin à courbure continue	40
2.4.3	Approximation de la liste des points mémorisés avec des courbes de Bézier	41
2.4.4	Approximation de la liste de points mémorisés avec des B-splines	43
2.4.5	Synthèse des méthodes d'estimation de chemins et trajectoires	44
2.5	Bilan et discussion	45

Chapitre 3

Algorithme général de *platooning*

3.1	Algorithme général de platooning	48
3.2	Modélisation du système	49
3.2.1	Modèle à variation instantanée	51
3.2.2	Modèle à variation progressive	52
3.2.2.1	Résolution approchée du déplacement des robots	52
3.2.2.2	Saturation des vitesses des roues du robot	55
3.3	Conception du simulateur	56
3.4	Considérations techniques pour l'implémentation des contrôleurs	57
3.5	Conclusion	57

Chapitre 4

Memo-LAT : un algorithme de platooning basé sur la mémorisation du chemin

4.1	Acquisition et mémorisation de positions	60
4.2	Algorithme	61
4.2.1	Choix de la position cible	61
4.2.2	Calcul de la vitesse angulaire	62
4.3	Expérimentations	63
4.3.1	Validation par simulation	63

4.3.1.1	Influence de la courbure sur la distance d'établissement . . .	64
4.3.1.2	Étude de la stabilité du platooning pour des longs convois	66
4.3.2	Validation sur les robots Khépéras	68
4.3.2.1	Considérations techniques	68
4.3.2.2	Formation du convoi	69
4.3.2.3	Influence de la vitesse du leader	71
4.3.2.4	Influence de la distance d'établissement	73
4.4	Conclusion	75

Chapitre 5

NOC et *Abstract NOC* : deux algorithmes de platooning qui s'appuient sur la mémorisation du chemin et la recherche heuristique

5.1	Choix de la position cible et estimation de la courbure	79
5.2	Calcul de la commande latérale	81
5.2.1	Condition Γ_i	82
5.2.2	Condition Γ_i^+	83
5.2.3	La fonction d'erreur	84
5.2.4	Raffinement de la commande latérale	86
5.3	Combinaison des contrôleurs longitudinal et latéral	87
5.4	Étude de la stabilité de <i>NOC</i>	88
5.4.1	Cas d'une droite	90
5.4.2	Cas d'un cercle	92
5.5	Validation expérimentale de l'algorithme <i>NOC</i>	95
5.5.1	Comparaison des deux algorithmes de platooning <i>Memo-LAT</i> et <i>NOC</i>	95
5.5.2	Influence du modèle de déplacement des robots sur <i>NOC</i>	97
5.6	<i>Abstract NOC</i> : Abstraction de l'algorithme <i>NOC</i> par rapport au modèle de mouvement	99
5.6.1	Estimation de la courbure	99
5.6.2	Calcul de la commande latérale	100
5.6.2.1	Condition Γ_i	101
5.6.2.2	Condition Γ_i^+	101
5.6.2.3	Conception de la fonction d'optimisation	102
5.7	Validation expérimentale de <i>Abstract NOC</i>	104

5.7.1	<i>Abstract NOC</i> appliqué sur un modèle de déplacement des robots à variation progressive	104
5.7.2	Validation expérimentale de la preuve de stabilité	105
5.7.3	Vitesse longitudinale du prédécesseur et comportement de son suiveur	106
5.7.4	Vitesse angulaire du prédécesseur et comportement de son suiveur .	108
5.8	Conclusion	108

Chapitre 6

Conclusion

6.1	Synthèse des contributions	111
6.2	Perspectives	114

Appendices

Annexe A

Calcul de l'évolution d'un robot ayant un modèle de mouvement à variation instantanée

A.1	Calcul de la nouvelle orientation $\theta(t)$	119
A.2	Calcul de la nouvelle vitesse longitudinale $v(t)$	119
A.3	Calcul de la nouvelle position $x(t)$	120
A.3.1	Calcul de $\int at \cdot \cos(\omega t + \theta_0)$	120
A.3.2	Calcul de $\int v_0 \cdot \cos(\omega t + \theta_0)$	121
A.3.3	Calcul de $x(t)$	121
A.4	Calcul de la nouvelle position $y(t)$	121
A.4.1	Calcul de $\int at \cdot \sin(\omega t + \theta_0)$	121
A.4.2	Calcul de $\int v_0 \cdot \sin(\omega t + \theta_0)$	122
A.4.3	Calcul de $y(t)$	122

Annexe B

Calcul de l'évolution d'un robot ayant un modèle de mouvement à variation progressive

B.1	Calcul de la nouvelle vitesse longitudinale $v(t)$	123
B.2	Calcul de la nouvelle vitesse angulaire $\omega(t)$	124
B.3	Calcul de la nouvelle orientation $\theta(t)$	124
B.4	Calcul de la nouvelle position $x(t)$	124

B.4.1	Calcul de $\int a.t. \cos(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0).dt$	125
B.4.2	Calcul de $x(t)$	125
B.5	Calcul de la nouvelle position $y(t)$	125
B.5.1	Calcul de $\int a.t. \sin(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0).dt$	126
B.5.2	Calcul de $y(t)$	126

Bibliographie	127
----------------------	------------



Table des figures

1.1	à droite : Platooning de quatre camions, à gauche platooning de trois véhicules électriques.	4
1.2	Robu-tainer.	5
1.3	Le système Robu-tainer - Cassette.	6
2.1	Projet <i>PATH</i> : platooning de huit véhicules sur une autoroute.	12
2.2	Projet <i>Cristal</i> : platooning de trois véhicules.	13
2.3	Représentation du véhicule du projet <i>Cristal</i>	14
2.4	Projet <i>Sartre</i> : convoi de quatre véhicules.	14
2.5	<i>VIAC</i> : navigation autonome du véhicule en absence d'un conducteur.	15
2.6	Principe de l'obstacle fantôme pour éviter les minimums locaux (figure tirée de [Vedova et al., 2012]).	21
2.7	Modélisation de l'interaction entre un couple de véhicules par un système de ressort-amortisseur (figure tirée de [Yi and Chong, 2005]).	22
2.8	Modélisation de l'interaction entre un couple de véhicules par un système de deux ressorts et un amortisseur en parallèle (figure tirée de [El-Zaher et al., 2011]).	22
2.9	Adaptation de la formation carrée de robots selon la trajectoire de référence (figure tirée de [Barfoot and Clark, 2004]).	23
2.10	Schéma bloc d'un régulateur <i>PID</i>	28
2.11	Modélisation d'un véhicule selon le modèle d'Ackermann.	31
2.12	Commande longitudinale hybride basée sur les erreurs longitudinales entre un suiveur et le leader du convoi d'une part et un suiveur et son prédécesseur d'autre part (figure tirée de [Bom, 2006]).	31
2.13	Profil de la vitesse du véhicule de tête (figure tirée de [Scheuer et al., 2009]).	33
2.14	Représentation d'un convoi de robots en mode platooning le long d'une ligne droite (figure tirée de [Scheuer et al., 2009]).	33
2.15	Chemins de Dubins reliant un point de départ (en bleu) et un point d'arrivée (en vert).	39
2.16	Chemin à courbure continue reliant une configuration de départ q_a à une configuration d'arrivée q_b (figure tirée de [Scheuer and Fraichard, 1997]).	40
2.17	Variation de la courbure dans le cas d'un chemin à courbure continue formé d'une rotation à gauche suivie d'un segment de droite et enfin d'une rotation à droite (figure tirée de [Scheuer and Fraichard, 1997]).	41
2.18	Modification en ligne du chemin du robot pour éviter l'obstacle (figure tirée de [Hwang et al., 2009]).	42

2.19	Génération d'une courbe de Bézier par morceaux passant par les coins d'un chemin donné (figure tirée de [Hwang et al., 2009]).	42
2.20	Commande longitudinale hybride basée sur les erreurs longitudinales entre un suiveur et le leader du convoi d'une part et un suiveur et son prédécesseur d'autre part (figure tirée de [Avanzini et al., 2009]).	44
3.1	Représentation d'un robot dans un repère global.	50
3.2	Profil trapézoïdal de variation de la vitesse de chacune des roues droite (v_r) et gauche (v_l) : $v_{r/l,0}$ valeur initiale et $v_{r/l,d}$ valeur désirée.	54
4.1	Acquisition et mémorisation d'une position du robot prédécesseur.	60
4.2	Choix de la position à viser.	62
4.3	Erreur angulaire entre le suiveur et la position cible.	63
4.4	Courbes d'erreur latérales entre les chemins des suiveurs et du leader lors d'un platooning le long d'un chemin à faible courbure.	64
4.5	Platooning de 3 robots le long d'un chemin à forte courbure, $d_l = 20$ mm.	65
4.6	Platooning de 3 robots le long d'un chemin à forte courbure, $d_l = 40$ mm.	65
4.7	Perte de stabilité avec l'augmentation du nombre de robots, $d_l = 20$ mm.	66
4.8	Stabilité améliorée avec l'augmentation de d_l , $d_l = 20$ mm (2 figures en haut) et $d_l = 40$ mm (2 figures en bas).	67
4.9	Le robot Khépéra 3 conçu par <i>K-Team</i>	68
4.10	Erreur due à la dérive de l'odométrie : faible pour le suivi mais importante pour la mémorisation.	69
4.11	Configurations initiales des robots du convoi.	70
4.12	trajectoires suivies par les robots durant un platooning à $v = 0,1$ m/s.	70
4.13	Déviations latérales entre les chemins des suiveurs et leur leader (vert : leader-suiveur 1, rouge : leader-suiveur 2).	70
4.14	Chemins des robots durant un platooning avec $v_{leader} = 0,1$ m/s (leader en rouge, premier suiveur en vert et second suiveur en bleu).	71
4.15	Chemins des robots durant un platooning avec $v_{leader} = 0,15$ m/s (leader en rouge, premier suiveur en vert et second suiveur en bleu).	71
4.16	Mouvement oscillatoire du robot autour du chemin de son prédécesseur.	72
4.17	Inter-distances entre les robots avec $v_{leader} = 0,1$ m/s.	73
4.18	Inter-distances entre les robots avec $v_{leader} = 0,15$ m/s.	73
4.19	Platooning le long d'un cercle avec $d_l = 15$ mm (leader en rouge, premier suiveur en vert et second suiveur en bleu).	74
4.20	Platooning le long d'un cercle où chaque robot vise son prédécesseur (leader en rouge, premier suiveur en vert et second suiveur en bleu).	74
4.21	Platooning sur un cercle : déviation latérale moyenne en fonction de d_l	74
5.1	Approximation du chemin du leader autour de la position cible.	79
5.2	Choix de la nouvelle position cible.	80
5.3	Test de Γ_i dans le cas d'une droite et d'un cercle.	83
5.4	Test de Γ_i^+ dans le cas d'une droite et d'un cercle.	83
5.5	Représentation géométrique de la fonction d'optimisation.	84

5.6	Calcul de la fonction d'optimisation \mathcal{E} .	86
5.7	Raffinement de la vitesse angulaire pour plus minimiser la fonction d'erreur.	87
5.8	Limitation du déplacement du suiveur par deux cercles.	88
5.9	Cas 2 : condition d'initialisation non vérifiée (cas d'une droite et d'un cercle).	89
5.10	Cas 1 : convergence vers un chemin rectiligne.	90
5.11	Cas 2 : première possibilité de convergence vers un chemin rectiligne.	92
5.12	Cas 2 : deuxième possibilité de convergence vers un chemin rectiligne.	92
5.13	Cas 1 : convergence vers un chemin circulaire.	93
5.14	Cas 2 : première possibilité de convergence vers un chemin circulaire.	94
5.15	Cas 2 : deuxième possibilité de convergence vers un chemin circulaire.	94
5.16	À gauche : platooning de 9 robots sous <i>Memo-LAT</i> le long d'un chemin spiral, à droite : courbes d'erreur latérale entre les chemins du leader et chacun des suiveurs.	96
5.17	À gauche : platooning sous <i>NOC</i> de 9 robots le long d'un chemin spiral, à droite : courbes d'erreur latérale entre les chemins du leader et chacun des suiveurs.	96
5.18	À gauche : platooning sous <i>NOC</i> de 9 robots se déplaçant selon un modèle à variation instantanée le long d'un chemin spiral, à droite : courbes d'erreur latérale entre les chemins du leader et chacun des suiveurs.	97
5.19	Choix de la nouvelle cible selon <i>NOC</i> évolué.	100
5.20	Test de Γ_i dans le cas d'une droite et d'un cercle.	101
5.21	Test de Γ_i^+ dans le cas d'une droite et d'un cercle.	102
5.22	Test de Γ_i^+ dans le cas d'une droite et d'un cercle.	103
5.23	Erreur angulaire nulle en un point de l'arc de clothoïde.	103
5.24	Erreur angulaire nulle en un point de l'arc de cercle.	104
5.25	Comparaison des performances de <i>NOC</i> (2 figures en haut) et <i>Abstract NOC</i> (2 figures en bas) implémentés sur un modèle de mouvement à variation progressive.	105
5.26	Convergence d'un robot vers le chemin rectiligne de son prédécesseur ($y=0$). À droite : chemins des 2 robots, à gauche : courbe d'erreur latérale entre les 2 robots.	106
5.27	Convergence d'un robot vers le chemin circulaire de son prédécesseur.	106
5.28	à gauche : suivi en convoi de 2 robots, à droite : courbe d'erreur latérale entre les chemins des 2 robots.	107
5.29	à gauche : suivi en convoi de 2 robots, à droite : courbe d'erreur latérale entre les chemins des 2 robots.	108

Notations et définitions

1. Un convoi est une suite de robots qui se déplacent ensemble vers la même destination selon une forme précise. Dans cette thèse, nous considérons des convois en file.
2. Le leader d'un convoi est le robot de tête qui est suivi par le reste des robots du convoi. Le leader peut être contrôlé par un opérateur humain ou être doté de capacités de navigation autonome.
3. Dans cette thèse, nous désignons par *chemin* la suite de points par lesquels un robot est passé.
4. Le déplacement en convoi (ou *platooning*) est la technique qui permet à un ensemble de robots de se déplacer en file sans accrochage matériel. Le véhicule de tête peut être autonome ou conduit par un pilote.
5. L'écart latéral e_y d'un robot par rapport à un chemin, désigné aussi par le terme "erreur latérale", est la distance du centre de masse du robot à ce chemin. En d'autres termes, c'est la longueur du segment reliant le centre de masse du robot au point le plus proche du chemin.
6. L'erreur angulaire e_θ est la différence d'angle entre l'orientation du robot et la direction de la tangente au chemin au point le plus proche du robot.
7. Le contrôle longitudinal consiste à calculer une commande longitudinale (accélération a ou vitesse v) de façon à réduire l'inter-distance d séparant deux robots consécutifs. d doit toujours rester supérieure ou égale à une distance de sécurité d_s .
8. Le contrôle latéral consiste à calculer une commande latérale (angle de braquage δ , vitesse angulaire ω ou accélération angulaire) de façon à réduire l'écart latéral e_y entre un robot du convoi et le chemin à suivre. Dans le but d'éviter les oscillations, la commande latérale calculée vise à réduire l'erreur angulaire e_θ en même temps que e_y de sorte que le robot converge tangentiellement vers le chemin désiré.
9. Dans cette thèse, nous considérons qu'un suivi en convoi (ou *platooning*) est stable quand les robots se déplacent le long du chemin généré par le leader sans oscillation.
10. Nous considérons également qu'un suivi en convoi (ou *platooning*) est d'autant plus précis que l'erreur latérale entre les chemins du robot suiveur et du leader est faible.
11. Le modèle cinématique d'un robot permet d'étudier son mouvement sans considérer les causes à l'origine de ce mouvement (les forces appliquées au robot).

12. Le modèle dynamique d'un robot étudie son mouvement en considérant toutes les forces agissant sur ce dernier.

Chapitre 1

Introduction

Au cours des dernières années, la robotique mobile s'est intéressée aux systèmes multi-robots pour un nombre croissant d'applications. Plusieurs domaines, dont le domaine du transport, pourraient bénéficier de ces recherches. Le transport de biens et de personnes pourrait être amélioré par des systèmes multi-robots. Des convois de véhicules pourraient déplacer ces biens plus rapidement tout en réduisant le coût de main d'œuvre. Par exemple, dans le cadre du fret routier, un chauffeur pourrait diriger un convoi de camions, démultipliant ainsi la quantité de marchandises transportées par chauffeur en un temps donné. On pourrait imaginer qu'avec cette méthode, trois chauffeurs puissent conduire une douzaine de camions simultanément.

Le déplacement de robots en convoi est appelé *platooning*. Le *platooning* est une technique d'accrochage immatériel visant à diriger, en pleine sécurité, un convoi de robots le long d'un chemin prédéfini ou non. Le convoi est guidé par un robot leader, qui est en général le véhicule de tête et qui peut soit se déplacer de façon autonome soit être dirigé par un opérateur humain.

Pour réaliser un *platooning*, le convoi peut avoir deux architectures : soit une architecture globale, soit une architecture locale (aussi dite de *proche en proche*). Quand le convoi repose sur une architecture globale, le calcul du contrôle de chaque véhicule nécessite la connaissance d'informations relatives à la totalité ou à une partie (en général, les véhicules voisins et le leader) des véhicules du convoi, ces informations pouvant être par exemple leurs positions, orientations et vitesses. La communication est alors indispensable pour l'échange d'informations entre les véhicules. En revanche, dans une architecture locale, chaque véhicule calcule ses propres commandes en utilisant des informations relatives aux véhicules voisins. Dans ce cas, la communication est optionnelle : le véhicule peut soit communiquer avec ses voisins pour acquérir les informations utiles, soit les acquérir par lui-même à l'aide de ses capteurs embarqués.

Le *platooning* peut être appliqué dans différents domaines (figure 1.1) : le transport de marchandises, le déplacement du matériel au sein des industries, les visites touristiques guidées, les applications militaires, etc... Pour des raisons de fiabilité et de coût (communications coûteuses, capteurs précis très chers. Certaines sources estimeraient l'équipement des Google cars à environ 220 k\$/véhicule), le *platooning* n'est pas encore

utilisé dans le cadre d'applications réelles et reste confiné à des tests en environnement contrôlé. Plusieurs projets industriels et travaux de recherche autour du *platooning* ont été réalisés dans le but d'améliorer les approches existantes et de déployer cette technique. Les travaux des chercheurs touchent à différentes problématiques dans le domaine du réseau, de la fusion multi-capteurs, de la localisation et du contrôle. Par exemple, certains chercheurs [Kavathekar and Chen, 2011] s'intéressent à la sûreté des communications, les pertes de communication, le retard dans la transmission et/ou le traitement des paquets, la perte de paquets, etc... Si un de ces problèmes se produit, la sécurité du suivi serait mise en question et le convoi s'arrêterait. L'utilisation de la communication pour l'échange des données entre les véhicules peut également être très coûteuse en bande passante, surtout pour les convois à architecture globale, ce qui limite la taille de ces convois. De plus, les systèmes de perception et de localisation les plus précis sont généralement très coûteux et peuvent souffrir d'une dégradation importante de leurs performances dans certaines conditions climatiques particulières. À titre d'exemple, les systèmes de perception utilisant des caméras sont aveuglés par les flash lumineux (lumière directe du soleil, phares de voitures la nuit) et rendus inefficaces par le brouillard.



FIGURE 1.1 – à droite : Platooning de quatre camions, à gauche platooning de trois véhicules électriques.

Malgré ces difficultés, l'application du *platooning* dans les systèmes de transport permettrait d'offrir beaucoup d'avantages sur les plans écologique, économique et humain. En effet, différentes études ont montré que la formation de convois compacts (dans lesquels l'inter-distance entre chaque couple de véhicules successifs est faible) mènerait à une réduction de la consommation de carburant grâce au phénomène d'aspiration (réduction de la force de traînée subie par un véhicule en suivant un autre à faible distance). Cette réduction de la consommation de carburant permettrait de réduire l'impact du transport de marchandises sur l'environnement. De plus, le contrôle automatique de ces convois renforcerait la sécurité routière (moins d'accidents causés par le manque d'attention et les erreurs des conducteurs), augmenterait la capacité des routes (réduction des distances de sécurité) et fluidifierait la circulation.

1.1 Le projet Européen InTraDE

Cette thèse est financée par le projet européen **InTraDE** "INtelligent TRAsportation in Dynamic Environments". Ce projet, qui a débuté en septembre 2010 pour une période de 3 ans et demi, regroupe les partenaires européens suivants : l'Institut National de Recherche en Informatique et en Automatique INRIA (Fr), le laboratoire d'automatique, génie informatique et signal LAGIS (Fr), le centre régional d'innovation et de transfert de technologie dans les transports et la logistique CRITT Transport et Logistique (Fr), l'agence de développement du sud-est de l'Angleterre SEEDA (UK), le LOOM de l'université de Liverpool John Moores (UK), l'institut de technologie de Dublin DIT (Ie), le port d'Oostende (Be) et les sous-traitants RoboSoft et Oktal. InTraDE a pour but de :

- améliorer la productivité des ports maritimes de petite et moyenne taille pour augmenter leur niveau de compétitivité,
- augmenter la sécurité opérationnelle et réduire l'impact environnemental des ports régionaux,
- réduire les écarts entre les régions les moins développées économiquement et les régions les plus développées.

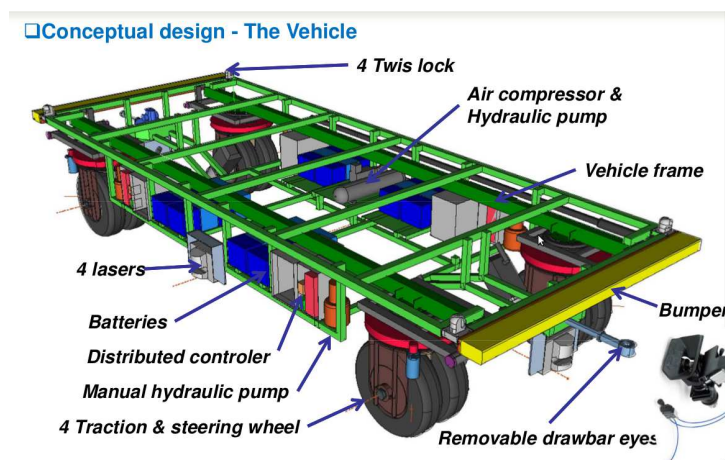


FIGURE 1.2 – Robu-tainer.

Pour cela, le projet InTraDE vise d'une part à automatiser les ports maritimes en introduisant des véhicules électriques autonomes et d'autre part à optimiser leurs espaces de stockage et d'opération afin de faciliter l'activité portuaire.

Dans le cadre du projet, RoboSoft a conçu un véhicule électrique intelligent appelé robu-tainer (Figure 1.2), pour le transport des conteneurs au sein des ports maritimes. Muni de 4 lasers placés chacun d'un côté et d'un GPS pour déterminer sa position, le robu-tainer charge des conteneurs posés sur des cassettes en les fixant à l'aide de son système de verrouillage (Figure 1.3). Pour déplacer le robu-tainer, un contrôleur embarqué se charge d'envoyer les commandes aux roues correspondantes.

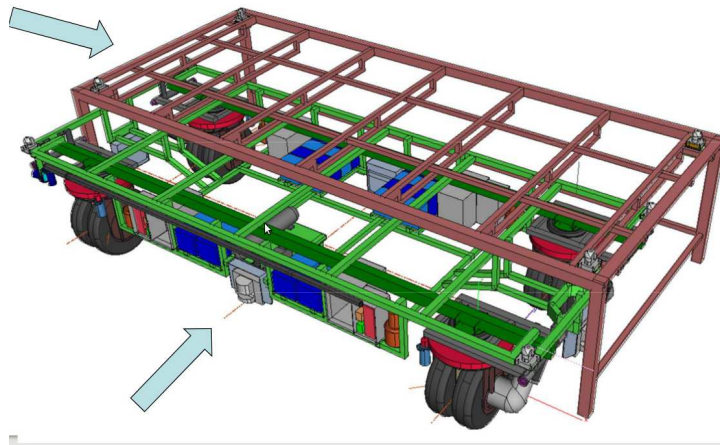


FIGURE 1.3 – Le système Robu-tainer - Cassette.

Un des objectifs du projet est de permettre un déplacement en convoi de plusieurs de ces véhicules, dont le premier est éventuellement piloté par un opérateur humain. Pour cela, nous avons développé des algorithmes permettant à un véhicule de suivre le chemin du véhicule qui le précède, que ce chemin soit défini ou non. Les robu-tainers doivent ainsi respecter le chemin en consigne sans déviation latérale et assurer une sécurité totale en évitant les collisions avec l'environnement comme au sein du convoi. Pour des raisons de robustesse et d'autonomie, nous souhaitons développer une approche décentralisée locale où chaque véhicule calcule ses commandes selon sa perception locale de l'environnement.

1.2 Objectifs, contributions et plan de la thèse

Dans cette thèse, nous souhaitons réaliser un suivi de robots en convoi le long d'un chemin généré par le leader et inconnu des suiveurs. Ce type de suivi sans accrochage matériel est désigné en anglais par le terme "platooning".

Un *platooning* sûr et précis est obtenu en contrôlant le comportement longitudinal et le comportement latéral de chaque véhicule du convoi. D'une part, le contrôle longitudinal consiste à calculer une commande longitudinale qui permet aux véhicules de se déplacer en maintenant une distance minimale par rapport à leur prédécesseur tout en évitant les collisions et en réduisant les phénomènes d'accordéon. D'autre part, le contrôle latéral cherche à calculer une commande latérale qui permette aux véhicules de se déplacer le long du chemin du leader sans déviation ni oscillation. D'après Samson [Samson, 1995], le contrôle longitudinal et le contrôle latéral peuvent être découplés. De même, Daviet et Parent [Daviet and Parent, 1996a] suggèrent que ces deux contrôles peuvent être traités séparément durant le déplacement à faible vitesse du convoi. Nous considérons donc dans cette thèse que le contrôle longitudinal et le contrôle latéral sont indépendants.

Nous avons choisi de contrôler la vitesse longitudinale des robots en utilisant un contrôleur longitudinal développé dans l'équipe *MAIA* qui présente des propriétés intéressantes. Ce contrôleur, que nous appelons *SSC*, a été présenté en détails dans l'article de

conférence [Scheuer et al., 2009]. Celui-ci garantit en effet l'absence de collisions, même dans le cas d'un arrêt d'urgence du leader, tout en maintenant une inter-distance minimale entre les véhicules lors du déplacement.

Nous nous intéressons donc en particulier dans cette thèse au contrôle latéral. Nous souhaitons dès lors concevoir des algorithmes permettant aux robots du convoi de suivre avec précision le chemin emprunté par le leader.

Comme déjà mentionné plus tôt, une approche globale nécessite un échange important d'informations entre les véhicules du convoi pour calculer les commandes de chacun, ces commandes donnant lieu à un suivi stable et précis. Cependant, cet échange est assez coûteux et est sujet aux différents problèmes liés à l'utilisation de la communication (retard dans la transmission et le traitement des paquets, perte de paquets, perte de la communication). Si un de ces problèmes se produit, le *platooning* est altéré. Pour éviter l'arrêt du convoi, nous imaginons la solution suivante : dans le cas où le contrôle du convoi à architecture globale échoue, nous supposons qu'une approche de contrôle décentralisée locale qui n'utilise pas la communication peut prendre le relais et mener le suivi correctement jusqu'à rétablissement de l'architecture globale.

Les approches décentralisées locales sont cependant sujettes à une accumulation de l'erreur latérale qui se propage le long du convoi. Ce phénomène est dû à la perception locale des véhicules et rend ce type d'approches moins précis et moins stable que les approches globales. L'objectif de cette thèse est de montrer qu'une approche décentralisée de proche en proche permet de réaliser un convoi de plusieurs véhicules stable et précis, c'est-à-dire un convoi dans lequel les véhicules suiveurs oscillent très faiblement autour du chemin emprunté par leur leader.

Nous souhaitons aussi concevoir une méthode de contrôle qui ne présente pas de contraintes sur la configuration initiale des robots (dans le cas de [Micaelli and Samson, 1993], le robot suiveur doit vérifier certaines conditions sur son état initial), et qui ne nécessite pas de réglages de paramètres ou de gains, de "bonnes valeurs" étant généralement trouvées par des méthodes heuristiques ou par exploration de paramètres [Levine and Athans, 1966] et nécessitent d'être recalculées pour chaque type de robot [Khatir and Davison, 2004]. En outre, ces valeurs de gains peuvent ne produire de bons résultats que dans un intervalle donné de la courbure de la trajectoire : des gains faibles empêcheront les oscillations.

Deux algorithmes de contrôle, basés sur la mémorisation du chemin du prédécesseur, représentent les contributions principales de cette thèse. La première méthode choisit un point du chemin sauvegardé qui soit plus proche du suiveur que la position courante du prédécesseur. Dans le but d'atteindre ce point, cette méthode utilise une loi de contrôle analytique [Daviet and Parent, 1996a] pour calculer une commande qui asservit l'erreur angulaire autour de zéro. La deuxième méthode reconstruit le chemin mémorisé et calcule une commande qui permet de le suivre avec précision et avec le moins d'oscillations possibles. Cette commande est discrète et est obtenue à partir d'une approche heuristique combinée à une approche géométrique.

Ce manuscrit présente les différents travaux réalisés au cours de cette thèse. Il est organisé en plusieurs chapitres comme suit :

Le chapitre 2 présente les travaux les plus significatifs de la littérature autour du *platooning*. Nous réalisons une synthèse des différentes architectures (globale et locale) qu'un convoi peut prendre et des différentes approches de contrôle (longitudinal et latéral) afin de pouvoir situer nos travaux par rapport à ceux existants.

Le chapitre 3 introduit les travaux de cette thèse en posant les hypothèses de travail et en modélisant le système. Dans ce chapitre, nous précisons la nature de l'approche développée dans cette thèse, à savoir une approche décentralisée locale, en donnant les arguments pour ce choix. Ensuite, nous présentons l'algorithme général de *platooning* considéré dans ce travail et nous introduisons rapidement les deux algorithmes de *platooning* développés dans cette thèse et qui constituent ses principales contributions.

Le chapitre 4 présente le premier algorithme *Memo-LAT* (Memorization and Look-Ahead Target) qui se base principalement sur la mémorisation du chemin du robot prédécesseur, la perception et la mémorisation de ce chemin étant expliquées en détails. Pour suivre le robot prédécesseur, *Memo-LAT* choisit une position cible (du chemin mémorisé) qui est située à une distance d'établissement constante du robot suiveur. Ensuite, il calcule une commande qui permet au robot suiveur d'asservir son orientation dans la direction de la position cible. Cette commande est obtenue en utilisant une loi de contrôle analytique, qui ne garantit malheureusement pas la stabilité du suivi. Pour valider cet algorithme, nous le testons à la fois en simulation et sur des robots khépéras. Enfin, nous concluons ce chapitre par une analyse des points forts et des points faibles de *Memo-LAT* et mettons en avant les motivations qui nous ont poussés à concevoir l'algorithme présenté dans le chapitre suivant.

Le chapitre 5 présente *NOC* (Non Oscillatory Convergence), le deuxième algorithme développé dans cette thèse. Cet algorithme choisit une position cible (du chemin mémorisé) qui permet au robot de suivre son prédécesseur sans osciller autour de son chemin. Pour choisir la prochaine position à viser, *NOC* teste chaque position du chemin mémorisé à partir de la dernière position visée, et ceci jusqu'à trouver une position cible convenable. La position cible est la plus proche position du robot suiveur tel que, en approchant le chemin autour de cette position par une droite ou un cercle, le robot suiveur peut tourner à fond sans couper ce chemin approximé. Ensuite, il estime, au cours du déplacement du robot suiveur, la courbure du chemin du prédécesseur autour du point visé et calcule, en prenant en compte cette courbure, une commande qui permet au suiveur de converger et de suivre ce chemin approché avec précision et sans oscillation. La commande est calculée en combinant une approche géométrique qui évite au suiveur de couper le chemin estimé avec une approche heuristique qui réduit une fonction d'erreur. L'algorithme est ainsi stable par construction et le contrôle réalisé est discret. Pour valider *NOC*, nous le testons sur le simulateur développé dans cette thèse et nous le comparons avec l'algorithme *Memo-LAT*.

Dans le but d'appliquer *NOC* sur des modèles différents de robots, nous proposons une version générale de *NOC* que nous appelons *Abstract NOC*. *Abstract NOC* est obtenu en abstrayant *NOC* par rapport au modèle de mouvement des robots. En d'autres termes, *Abstract NOC* fera appel à une fonction externe qui simule le modèle de déplacement des robots et utilise cette simulation dans le calcul de la commande latérale. *Abstract NOC* a aussi été testé en simulation et les résultats expérimentaux ont montré de bonnes

performances.

Pour conclure cette thèse, le dernier chapitre résume les différents travaux présentés dans les chapitres précédents. Il mettra l'accent sur les améliorations apportées par les algorithmes développés dans cette thèse et sur leurs limites. Enfin, nous discutons des perspectives qui s'ouvrent suite à ce travail.

Chapitre 2

État de l'art

Ces dernières années, la conception des véhicules autonomes a été témoin d'une importante évolution. Plusieurs travaux de recherche ont été menés autour de différentes thématiques allant de la perception et la localisation à la synthèse d'algorithmes de contrôle pour la navigation autonome, l'évitement d'obstacles, etc... Ces travaux sont répartis entre des projets regroupant des partenaires académiques et industriels et des recherches universitaires.

Dans ce chapitre, nous présentons en premier lieu différents programmes de recherche réalisés autour du déplacement en formation (*platooning*) de véhicules autonomes (section 2.1). Nous verrons que pour maintenir une formation, deux approches ont été proposées dans la littérature selon que l'on aborde le problème de manière globale, centralisée ou décentralisée, ou de manière locale. Ces architectures sont décrites dans la section 2.2. La section 2.3 présente différentes méthodes de suivi appliquées sur les robots. La plupart de ces approches reposent sur une connaissance de l'équation analytique de la trajectoire générée par le leader du convoi. Cette trajectoire n'étant pas toujours prédéfinie, nous présentons différentes méthodes d'estimation de trajectoires dans la section 2.4. Dans la section 2.5, nous faisons un bilan de l'apport des travaux présentés dans ce chapitre au problème considéré.

2.1 Projets de recherche autour du suivi de convoi

Industriels et académiques se sont associés autour du concept de l'accrochage immatériel et ont monté ensemble différents projets dans le but de réaliser des navigations en convoi. Dans ce cadre, les véhicules se déplacent le long d'un chemin guidé par un véhicule de tête, désigné aussi par le terme *leader*. Le but de ces projets est d'améliorer la circulation des véhicules sur le réseau routier en réduisant l'inter-distance entre chaque paire de véhicules successifs, augmentant ainsi la capacité des routes et fluidifiant la circulation routière.

2.1.1 Le projet *PATH*



FIGURE 2.1 – Projet *PATH* : platooning de huit véhicules sur une autoroute.

Dans les années 1990, l'université de Californie à Berkley a lancé le projet *PATH* (California Partners for Advanced Transit and Highways) visant à augmenter la capacité des autoroutes pour atténuer la congestion en réalisant des convois de véhicules qui se suivent à faible inter-distance et sans accrochage matériel. En août 1997, une démonstration (figure 2.1)¹ sur une autoroute à San Diego a montré un platooning de 8 voitures Buick distantes de 6,5 m. Les voitures communiquaient entre elles afin d'échanger leurs données et de gérer les manœuvres de changement de voies, d'insertion et/ou d'abandon de convoi.

L'approche utilisée dans *PATH* est une approche globale où chaque suiveur récupère les données relatives au leader et au véhicule prédécesseur. Pour calculer sa nouvelle accélération [Shladover et al., 1991], le suiveur utilise les données suivantes : ses propres vitesse et accélération estimées par ses capteurs embarqués, l'inter-distance le séparant de son prédécesseur qui est mesurée à l'aide de radars. Les vitesses et accélérations relatives au leader du convoi et au véhicule prédécesseur sont transmises au suiveur à l'aide d'une communication radio.

De plus, pour contrôler son comportement latéral, chaque véhicule du convoi détecte à l'aide d'un magnétomètre le champ magnétique d'aimants placés au milieu des voies de l'autoroute. Ces aimants contiennent des informations sur la forme de la prochaine partie du chemin à suivre et aident le véhicule à estimer sa position [Wei-bin Zhang, 1990]. Le suiveur, visant ces aimants successifs, se déplace dans la voie de l'autoroute. En se basant sur la théorie du contrôle appliquée au modèle dynamique des véhicules, une commande de direction est calculée de façon à réduire l'écart latéral entre le véhicule et la succession des aimants [Shladover et al., 1991].

Le problème majeur de ce projet était l'utilisation d'une infrastructure particulière au sol (les aimants). D'une part, installer des aimants sur toutes les autoroutes est coûteux et compliqué à réaliser. D'autre part, les voitures existantes ne peuvent pas utiliser

1. Toutes les figures de l'état de l'art appartiennent à leurs auteurs respectifs. Elles ont été tirées des articles cités dans ce chapitre.

cette technique pour réaliser des convois automatiques. Les industries automobiles ne fabriquent pas des voitures équipées de magnétomètres vu que l'infrastructure nécessaire à leur utilisation n'existe pas.

2.1.2 Le projet *Cristal*



FIGURE 2.2 – Projet *Cristal* : platooning de trois véhicules.

Le projet national *Cristal* (Cellule de Recherche Industrielle en Systèmes de Transports Automatisés Légers), financé par le pôle de compétitivité Alsace Franche Comté "véhicule du futur", a rassemblé plusieurs laboratoires de recherche français (*Inria*, *UTBM*, *LASMEA*) et entreprises (*Vulog*, *Transitec*, *GEA* et *LOHR Industrie*) autour d'un budget de 8 millions d'euros afin d'améliorer la mobilité routière. Le projet *Cristal* constitue un projet innovant qui ne met pas en avance de grandes avancées technologiques mais des innovations techniques. Le but du projet revient à mettre en service des véhicules à double mode :

- un mode "libre-service" pour un usage individuel où une personne pourrait récupérer un véhicule d'une station donnée, s'en servir pour effectuer son déplacement et rendre le véhicule à une station destinataire à la fin du voyage. Plusieurs stations, seront dispersées dans une zone définie selon une étude territoriale et équipées d'un certain nombre de véhicules disponibles à tout moment sans réservation,
- et un mode "convoi" pour un usage semi-collectif où un déplacement d'un ensemble de véhicules en convoi peut être planifié pour une date et un trajet spécifiques afin de transporter des personnes ou des biens (figure 2.2). Le conducteur pourrait choisir le nombre de véhicules formant son convoi ; les tableaux de bord de ces véhicules autre que le véhicule de tête seront inaccessibles. Le mode "convoi" pourrait aussi être utilisé pour rééquilibrer les stations. Pour ce faire, les véhicules se déplaceront en convois dans le but d'être redistribués d'une façon équitable sur les différentes stations.

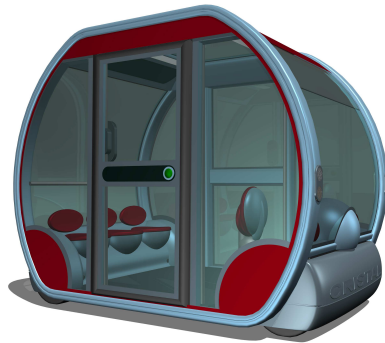


FIGURE 2.3 – Représentation du véhicule du projet *Cristal*.

Les recherches et études menées par les différents acteurs de ce projet ont permis de concevoir un véhicule électrique d'une capacité de 6 personnes, d'une longueur de 3,4 m et d'une largeur de 1,7 m. Le véhicule, présenté dans la figure 2.3, est équipé d'un GPS lui permettant d'estimer sa position absolue, d'un capteur Lidar (qui est un capteur laser) et d'une caméra linéaire pour détecter le véhicule prédécesseur.

2.1.3 Le projet *Sartre*



FIGURE 2.4 – Projet *Sartre* : convoi de quatre véhicules.

Le projet *Sartre* (Safe Road Trains for Environment) a été lancé en 2009 par la communauté européenne. D'une durée de 3 ans, son but était de réaliser des convois automatiques de véhicules [Tom Robinson, 2010] en utilisant l'infrastructure existante (figure 2.4). Ce suivi de convoi (ou platooning) permet de réduire la consommation du carburant et accroît la sécurité routière, comme le montre [A. Davila, 2010]. Le projet s'intéressait à la conduite de ces convois sur autoroute uniquement, en supposant les véhicules initialement alignés du plus lourd en tête du convoi au plus léger pour réduire les risques de collision (un véhicule lourd nécessite plus de temps pour freiner qu'un véhicule plus léger) [Eric Coelingh, 2012].

Pour contrôler la vitesse ou bien maintenir une voie donnée, le projet se base sur le régulateur de vitesse adaptatif et le système de contrôle d'orientation développés pour les voitures Volvo (véhicules utilisés dans le projet). Chaque voiture Volvo était équipée de différents capteurs afin de percevoir son voisinage :

- un système de 3 faisceaux laser d'une portée de 8 m à l'avant de la voiture ;
- un radar à 76 GHz d'une portée de 200 m à l'avant de la voiture ;
- une caméra prospective pour détecter les piétons et les panneaux de signalisation ;
- deux radars à 76 GHz pour aider à gérer la manœuvre d'insertion d'une voiture dans le convoi et pour observer les mouvements des véhicules sur les autres voies : le premier est orienté vers l'arrière et le deuxième est dirigé sur le côté.

L'approche développée dans ce projet dépend fortement de la communication entre les véhicules et associe au véhicule de tête (ou leader) un rôle très important dans la formation et le guidage du convoi. En effet, il gère la sécurité du convoi en particulier quand un nouveau véhicule souhaite s'insérer ou se détacher du convoi et fixe l'inter-distance minimale permise entre un couple de véhicules successifs. Pour éviter les mouvements d'oscillations dans les virages et les phénomènes d'accordéon à l'intérieur du convoi, le leader communique à ses suiveurs son accélération longitudinale et la courbure du chemin qu'il est en train de générer (et non pas la courbure de la route). De plus, durant les phases d'accélération (ou de décélération) et de changement d'orientation, le véhicule de tête envoie au reste du convoi des commandes d'accélération et de direction. Dans le cas où un problème se produit sur le leader, le suivi est perturbé et le convoi s'arrête.

2.1.4 VIAC : Le défi intercontinental pour véhicules autonomes de VisLab



FIGURE 2.5 – VIAC : navigation autonome du véhicule en absence d'un conducteur.

L'université de Parme, en particulier le laboratoire de vision artificielle et de systèmes intelligents **VisLab**, a lancé le défi *VIAC (Vislab Intercontinental Autonomous Challenge)* [Bertozzi et al., 2011] dans le but de réaliser un suivi en convoi partant d'Italie à destination de la Chine (environ 10000 Km à traverser en conduite autonome). Leur convoi était formé de 4 vans électriques (deux vans roulaient pendant que deux autres chargeaient leur batteries) équipés d'ordinateurs, de caméras et de capteurs, deux camionnettes pour transporter du matériel, un transporteur de voitures avec un générateur électrique à bord et quatre caravanes.

Chacun des véhicules est équipé de 7 caméras, 4 lasers, un GPS et d'un capteur inertiel pour se positionner et pour détecter les voitures et d'autres obstacles (figure 2.5). Deux caméras accrochées à l'avant du véhicule fournissent une vision stéréo pour détecter les voies. Trois autres caméras synchronisées permettent l'obtention d'une vue panoramique frontale à 180°. Les lasers détectent des obstacles, des piétons et d'autres véhicules. Trois ordinateurs se trouvent à bord de chaque véhicule. Un premier ordinateur est dédié aux traitements d'image et des données relatives à ce qui se trouve à l'avant du véhicule. Un deuxième ordinateur traite les données relatives aux deux côtés du véhicule. Le troisième ordinateur récupère toutes ces données, planifie un chemin, calcule les commandes d'accélération et de direction et les envoie aux actionneurs.

L'approche suivie dans ce projet est une approche leader-suiveur où le véhicule de tête communique ses coordonnées GPS à son suiveur. Pour suivre le chemin généré par le leader, le suiveur combine les coordonnées GPS communiquées avec les données acquises par ses capteurs embarqués afin d'avoir une estimation plus précise (que les coordonnées GPS seules) de la position du leader. Ensuite, il planifie en temps réel son chemin et calcule ses commandes de vitesse et de direction afin de suivre le chemin à l'aide de son système de navigation.

2.2 Architectures de convoi

Pour réaliser un suivi de robots sans accrochage matériel, le convoi peut avoir deux architectures différentes : soit une architecture globale où les données de tous les robots (ou une partie) du convoi sont utilisées pour calculer les commandes de chacun, soit une architecture locale où chaque robot du convoi perçoit des informations concernant ses robots voisins pour calculer ses propres commandes. Nous distinguons aussi des convois ayant une architecture mixte qui est une combinaison d'une architecture locale et d'une architecture globale. J. Bom propose dans sa thèse [Bom, 2006] un contrôleur longitudinal qui utilise ce type d'architecture mixte : pour garder un convoi proche, le robot calcule une commande longitudinale qui lui permet de se rapprocher de son leader (architecture globale). Cependant, quand il devient très proche de son prédécesseur, il calcule une commande longitudinale qui lui permet d'éviter leur collision (architecture locale).

Les systèmes multi-agents peuvent être considérés comme un outil pour la représentation de l'architecture d'un convoi. En effet, chaque agent représente un véhicule du convoi et est doté d'une autonomie qui lui permet, à travers les interactions avec son environ-

nement et les autres véhicules, de réaliser l'accroche immatérielle. Selon la nature de ces interactions, cette architecture est alors considérée comme locale ou globale. Simon Hallé a développé dans sa thèse de master [Hallé, 2005] une approche basée sur un système multi-agents, où chaque véhicule, considéré comme un agent autonome, est soumis au contrôle conçu dans [Daviet and Parent, 1996a]. Les agents peuvent communiquer entre eux sous deux scénarios différents : soit le leader communique avec ses suiveurs (architecture globale), soit chaque véhicule communique avec ses voisins (architecture locale). C'est dans ce cadre que l'on trouve aussi les approches physiques dans lesquelles on simule les interactions entre véhicules par des systèmes type ressort ou ressort-capacité. Les auteurs de [Contet et al., 2006] et [Contet et al., 2007] représentent l'interaction entre deux véhicules successifs par un ressort. De plus, les auteurs de [El-Zaher et al., 2011] modélisent l'interaction entre deux véhicules par un amortisseur en parallèle avec deux ressorts.

2.2.1 Approches globales

Dans une approche globale, le calcul de la commande de chaque robot nécessite la connaissance de l'état complet du convoi (ou bien d'une partie du convoi, en général l'état du leader et des robots voisins). Sachant que les capteurs extéroceptifs et proprioceptifs ne suffisent pas pour acquérir ces données, une communication fiable s'avère donc indispensable dans ce type d'architectures. Ce partage d'information vise à construire des lois de commande qui permettent à un véhicule de bénéficier d'une certaine stabilité face aux perturbations.

Le calcul du contrôle peut être centralisé ou décentralisé. Dans une architecture centralisée, une unité de calcul récupère les données relatives à tous les robots du convoi et calcule la commande de chacun. Plusieurs travaux ont adopté cette architecture pour réaliser un suivi en convoi. Dans [Kaku et al., 2012], les auteurs ont développé une approche automatique utilisant un contrôleur linéaire quadratique, tandis que l'article [Levine and Athans, 1966] a présenté une méthode basée sur la théorie du contrôle optimal. Dans [Das et al., 2002], les auteurs ont réalisé un suivi en formation d'un leader avec évitement d'obstacles à l'aide d'une caméra omnidirectionnelle. Étant donné que la localisation par vision n'est pas suffisamment précise, le contrôle (basé sur des données locales) utilise une architecture centralisée pour avoir une localisation précise.

Une architecture centralisée nécessite un nombre important d'échange de données et une unité de calcul puissante. Ceci rend cette approche peu pratique pour les longs convois à cause de la limitation de la portée de la communication et des capacités de l'ordinateur central à traiter les données. Afin de réduire le coût de la communication (en réduisant le nombre de données échangées), le convoi peut adopter une architecture décentralisée globale où chaque robot récupère lui-même les données d'une partie du convoi (en général, le leader et les robots voisins) pour calculer sa commande. Plusieurs travaux [Barfoot and Clark, 2004, Gil-Pinto et al., 2007] ont réalisé un suivi en se contentant des données des robots leader et prédécesseur. Bom a développé dans sa thèse [Bom, 2006] une approche globale de platooning basé sur la théorie du contrôle non linéaire où chaque

robot utilise les données du leader et du véhicule prédécesseur pour calculer ses commandes. L'approche développée dans [Shladover et al., 1991] communique la position et la vitesse des robots leader et prédécesseur pour le calcul d'un comportement longitudinal qui assure la stabilité du convoi.

Dans les approches globales, le suivi réalisé est précis. Les commandes calculées sont optimales et garantissent la stabilité du convoi et une convergence rapide surtout après une perturbation inattendue. Ceci explique les très bonnes performances que ce type d'architectures apporte. Cependant, une architecture globale est assez coûteuse en termes de communication et peu robuste : un retard dans la transmission ou le traitement des données, une panne sur un robot (véhicule) du convoi, ou dans le pire cas une perte de données ou de la communication peut entraîner l'arrêt complet du convoi. Dans les architectures locales, l'utilisation de la communication est optionnelle : dans le cas où le convoi à architecture locale ne repose pas sur les communications, le *platooning* n'est pas affecté par ce type de perturbations. Les architectures locales sont alors une alternative préférable quand ces facteurs perturbants (cités ci-dessus) entrent en jeu.

2.2.2 Approches locales

Comme nous avons vu précédemment dans la section 2.2.1, les architectures globales sont plus coûteuses et moins robustes que les architectures locales. Si on désire améliorer la robustesse du suivi ou réduire le coût des communications, une architecture locale s'avère être une bonne solution. De plus, une architecture locale peut être adoptée dans le cas où une approche globale ne peut pas être utilisée (par exemple, à cause de communications limitées qui entraînent une impossibilité d'acquérir les données de tous les robots). Dans une approche locale, chaque robot calcule sa propre commande en fonction des données qu'il a acquises sur ses voisins. Cette acquisition de données se fait en général en utilisant des capteurs embarqués.

Plusieurs travaux se basent sur un système de vision pour réaliser un suivi en formation. Dans [Mariottini et al., 2007], les robots sont équipés de caméras panoramiques pour se localiser et calculent leur contrôle en utilisant une loi de commande rétroactive basée sur l'extension de leur dynamique. [Benhimane et al., 2005] propose un algorithme de platooning basé sur une approche de suivi visuel où le suiveur estime la position de son prédécesseur à l'aide d'un système de vision et calcule sa commande à l'aide d'un contrôleur LQR (Linear Quadratic Regulator). Un suivi en formation est réalisé dans [Das et al., 2002] en utilisant des caméras catadioptriques embarqués sur les robots. Le contrôle de la formation se base sur l'interaction de plusieurs blocs simples qui représentent des algorithmes de perception, d'estimation de contrôle et de coordination. D'autres travaux reposent sur des approches physiques (l'interaction entre deux robots successifs peut être modélisée par un système à ressort-amortisseur [Contet et al., 2006, Yi and Chong, 2005]) ou des approches d'automatique [Daviet and Parent, 1996b, Khatir and Davison, 2004].

Dans une approche locale où l'acquisition des données se fait à l'aide de capteurs embarqués, les robots du convoi sont autonomes : chaque robot peut rejoindre ou quitter un convoi sans avoir à avertir un contrôleur central. Ainsi, le convoi n'a plus besoin d'être

re-configuré, ce qui rend l'approche plus robuste. De plus, il peut calculer sa commande sans avoir besoin d'attendre de recevoir les données des autres robots. Ainsi, en comparant avec une architecture globale, une architecture locale est moins coûteuse et peut être appliquée sur de longs convois. En revanche, la stabilité du convoi n'est pas toujours garantie et la convergence après une perturbation est moins rapide. De plus, du fait des perceptions locales, l'approche est moins précise à cause de l'accumulation d'une erreur latérale qui se propage d'un robot à un autre.

2.3 Approches de contrôle analytiques

Les approches de contrôle analytiques permettent de synthétiser une loi de commande analytique à partir d'un modèle continu du mouvement du système. Ces approches considèrent généralement qu'à chaque instant t , l'équation de la trajectoire à suivre est connue, ainsi que l'état du robot par rapport à cette trajectoire. Le système peut ainsi être étudié en utilisant les outils de la théorie du contrôle optimal.

La conception des algorithmes de platooning permettant à un convoi de se déplacer en sécurité et avec précision revient à contrôler le comportement longitudinal et le comportement latéral de chaque robot du convoi. Le suivi en convoi ou en formation est alors obtenu soit par des méthodes de suivi de trajectoires (section 2.3.1) qui contrôlent les deux comportements longitudinal et latéral simultanément soit par des méthodes de suivi de chemins (section 2.3.2) qui découplent les deux contrôles longitudinal et latéral et calculent chacun indépendamment de l'autre. Le découplage de ces deux contrôles est possible quand le convoi se déplace à faible vitesse [Daviet and Parent, 1996a]. Le contrôle longitudinal vise à déplacer les robots d'un convoi le plus proche possible les uns des autres tout en évitant les collisions. Le contrôle latéral tend à asservir les robots du convoi autour du chemin à suivre. Les robots doivent dans ce cas suivre avec précision le chemin tracé par leur leader sans osciller autour.

En général, le calcul du contrôle longitudinal et latéral se base sur le modèle de mouvement du robot. Un robot est un système physique dont le mouvement est modélisé par des équations mathématiques donnant la variation du système en réponse à une commande. Cette modélisation est dans la plupart des cas non linéaire, ce qui rend la synthèse des commandes difficile.

J. Bom a présenté dans sa thèse [Bom, 2006] une démarche qui permet le découplage des deux contrôles longitudinal et latéral pour de nombreux modèles dynamiques. D'une façon générale, un système non linéaire est représenté par l'équation suivante :

$$\dot{X} = f(X) + g(X, U) \quad (2.1)$$

où X est le vecteur d'état du système, U est le vecteur de commande, f et g sont des fonctions non linéaires. Si nous considérons le modèle cinématique d'un robot, le vecteur commande U est décomposé en deux vecteurs : le vecteur commande longitudinale u_v et le vecteur commande latérale u_δ . La fonction g se décompose ainsi en fonction des deux

vecteurs commandes u_v et u_δ , ramenant l'équation 2.1 à :

$$\dot{X} = f(X) + g_v(X, u_v) + g_\delta(X, u_v, u_\delta) \quad (2.2)$$

Dans le cas où nous nous intéressons uniquement à la commande latérale, u_v devient un paramètre de contrôle qui peut être constant ou variable au cours du temps. L'équation 2.2 devient :

$$\begin{aligned} \dot{X} &= f(X) + g_{v,u_v}(X) + g_{\delta,u_v}(X, u_\delta), \\ \dot{X} &= (f + g_{v,u_v})(X) + g_{\delta,u_v}(X, u_\delta) \end{aligned} \quad (2.3)$$

Le comportement latéral du robot peut ainsi être représenté par les deux fonctions non linéaires $f + g_{v,u_v}$ et g_{δ,u_v} . Le même raisonnement peut être suivi pour obtenir une modélisation du système dans le cas où nous nous intéressons à contrôler le comportement longitudinal du robot. Nous pouvons ainsi traiter la commande latérale indépendamment de la commande longitudinale.

Dans ce qui suit, nous présentons différentes méthodes de suivi de trajectoires en convoi dans la section 2.3.1. La section 2.3.2 s'intéresse aux méthodes de suivi en convoi qui découplent le contrôle longitudinal et latéral et les calculent séparément. Les approches de calcul du contrôle longitudinal sont présentées dans la section 2.3.2.1 et celles relatives au contrôle latéral sont présentées dans la section 2.3.2.2.

2.3.1 Suivi de trajectoires

Dans ce type d'approche, la commande calculée agit simultanément sur les aspects longitudinal et latéral du robot. Nous distinguons différentes approches : les approches physiques (modélisation par impédances, modélisation par potentiel artificiel), les approches fondées sur une structure virtuelle du groupe de robots, les approches basées sur la théorie des graphes, les approches basées sur le comportement (désigné en anglais pas le terme *behavior-based approach*), les approches utilisant un modèle cinématique.

2.3.1.1 Les approches basées sur les méthodes physiques

Pour modéliser l'interaction entre les robots du convoi, certains travaux se sont inspirés des principes de la physique. Chaque robot est soumis à des forces issues des interactions entre les robots, dont la résultante détermine son nouvel état. Plusieurs approches entrent dans ce cadre.

En 1986, Khatib a proposé dans [Khatib, 1986] une méthode d'évitement d'obstacle basée sur le concept de champ de potentiel artificiel. D'une part, un obstacle génère une force de répulsion, et est ainsi représenté par une fonction de potentiel répulsive. D'autre part, une cible génère une force d'attraction et est représentée par une fonction de potentiel attractive. La combinaison de ces deux fonctions de potentiel définit le comportement du robot. Ce dernier se déplace tout en évitant les obstacles. Plusieurs travaux ont émergé de ce principe de potentiel artificiel. [Olfati-Saber and Murray, 2002] utilise un ensemble

de fonctions de potentiel pour réaliser un convoi en formation. Une approche de contrôle décentralisée a été développée dans [Sabattini et al., 2009] pour réaliser des convois en formation. La stratégie de contrôle est basée sur l'interaction entre champs de potentiel artificiel. Le convoi se déplace ainsi en évitant les obstacles et peut changer de forme en appliquant une transformation bijective de coordonnées. Malheureusement, **l'approche développée n'est pas applicable aux systèmes non holonomes** et ne permet pas de contrôler l'orientation de chaque robot à l'intérieur de la formation. Étant donné que cette approche ne nécessite pas beaucoup de calculs, elle peut être utilisée pour des applications en temps réel. En revanche, le problème rencontré dans ce type d'approches est la possibilité que **le système tombe dans un minimum local**. Comme le montre [Vedova et al., 2012], un robot peut se trouver coincé sur une intersection entre deux obstacles. Pour résoudre ce problème, *Vedova et al* propose dans [Vedova et al., 2012] la notion d'obstacle fantôme où il regroupe les deux obstacles en un seul pour former une enveloppe convexe (voir Figure 2.6). Cette solution permet au système d'éviter de tomber dans un minimum local **mais réduit l'espace de déplacement du robot**. D'autres travaux [Kim and Khosla, 1992, Boumaza and Scherrer, 2007] proposent d'utiliser les fonctions harmoniques pour planifier des trajectoires qui évitent les obstacles. *Connolly et al* [Connolly et al., 1990] ont proposé en 1990 l'utilisation de ces fonctions harmoniques qui permettent de générer un gradient sans minimums locaux. Ces approches nécessitent de connaître l'environnement pour pouvoir calculer une trajectoire permettant au robot d'atteindre son but sans être bloqué par des obstacles. Cependant, ces approches ne permettent pas de considérer des environnements dynamiques puisqu'il serait alors nécessaire de recalculer les trajectoires sur la totalité de l'environnement.

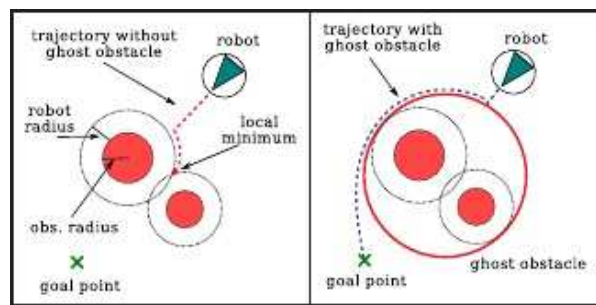


FIGURE 2.6 – Principe de l'obstacle fantôme pour éviter les minimums locaux (figure tirée de [Vedova et al., 2012]).

Nous distinguons aussi d'autres approches qui s'inspirent des principes de la physique. Les auteurs de l'article [Gehrig and Stein, 2001] utilisent des ressorts qui sont des impédances² non amorties pour modéliser l'interaction entre les robots. Cette modélisation est bien adaptée à des convois qui évoluent sur autoroutes, celles-ci ayant une faible courbure. Cependant, cette modélisation est peu adaptée pour des zones urbaines.

Instable par son principe même, cette modélisation physique est améliorée en remplaçant les ressorts par des impédances amorties. [Yi and Chong, 2005] modélise l'in-

2. Une impédance est une mesure physique de la résistance d'un système à une force qui lui est appliquée.

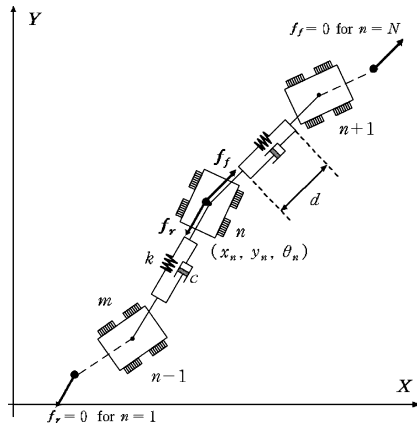


FIGURE 2.7 – Modélisation de l'interaction entre un couple de véhicules par un système de ressort-amortisseur (figure tirée de [Yi and Chong, 2005]).

interaction entre deux véhicules par un système de ressort-amortisseur (voir Figure 2.7). La connexion entre deux robots est modélisée selon un principe similaire dans l'article [Contet et al., 2007] par des forces d'attraction de Newton, d'amortissement et de friction. [El-Zaher et al., 2011] modélise cette interaction par deux ressorts et une capacité en parallèle (voir Figure 2.8). Ainsi, en utilisant le principe fondamental de la dynamique, l'accélération de chaque véhicule est calculée en faisant le bilan des forces agissant dessus ; et le nouvel état (position et vitesse) est déduit par la suite. Les approches par impédance ont comme avantage de pouvoir faire des preuves de stabilité du système simulé. **Ces preuves de stabilité à l'infini** s'appuient sur le principe de l'énergie dissipée. Ainsi, les vitesses relatives entre les suiveurs tendent vers zéro, et l'interdistance séparant chaque couple de véhicules consécutifs tend vers la distance de sécurité représentée par la longueur du ressort au repos. Cependant, ces approches sont connues pour présenter des **problèmes d'accordéon** dus à des retards dans la propagation de l'information le long du train, ces problèmes pouvant causer des collisions entre les robots du convoi.

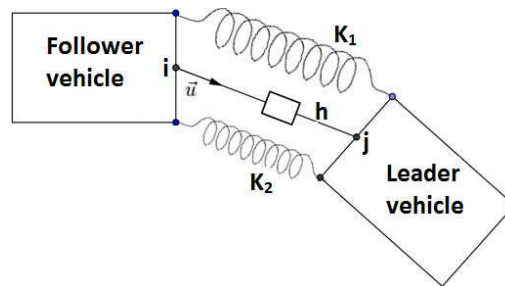


FIGURE 2.8 – Modélisation de l'interaction entre un couple de véhicules par un système de deux ressorts et un amortisseur en parallèle (figure tirée de [El-Zaher et al., 2011]).

2.3.1.2 Les approches représentant la formation comme une structure virtuelle

Plusieurs travaux considèrent que la formation de robots forme une structure qui est exploitée pour contrôler le convoi de robots en formation. Cette approche a été proposée par Tan et Lewis en 1997 [Lewis and Tan, 1997]. Elle considère le groupe de robots comme un seul robot rigide, qui se déplace en conservant les propriétés d'une structure virtuelle donnée. La loi de contrôle de chaque robot est ensuite déduite de la dynamique de la structure virtuelle.

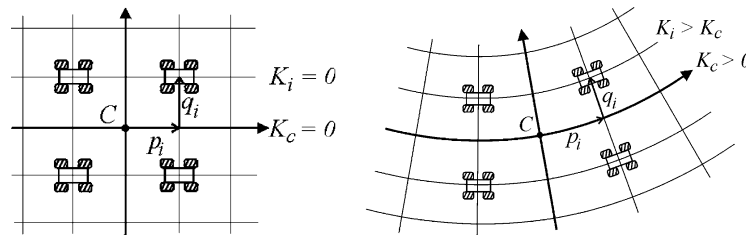


FIGURE 2.9 – Adaptation de la formation carrée de robots selon la trajectoire de référence (figure tirée de [Barfoot and Clark, 2004]).

Puisque la structure est rigide, **un problème peut intervenir si le chemin n'est pas rectiligne** : les robots en formation peuvent ne pas réussir à se déplacer le long d'un chemin curviligne si la voie est relativement étroite (comme lorsque qu'on essaie de faire passer un grand canapé dans un couloir étroit avec un virage à angle droit). Pour résoudre ce problème, Barfoot et al [Barfoot and Clark, 2004] propose de considérer un système de coordonnées curvilignes à la place du système de coordonnées rectilignes, habituellement utilisé. Ce nouveau système utilise la courbure de la trajectoire de référence (qui est celle du leader) pour déformer la formation afin qu'elle s'adapte à la silhouette de la trajectoire à suivre (voir Fig. 2.9). Le superviseur du système communique à chaque robot de la structure une trajectoire de référence et une consigne de décalage par rapport à cette trajectoire. En estimant sa position dans un repère global (grâce à un système de positionnement à l'aide d'une caméra aérienne fixe), chaque robot génère sa propre trajectoire et calcule les commandes (de vitesse linéaire v_i et de courbure K_i) pour la suivre au lieu d'essayer de garder une position relative fixe par rapport à son leader. Le calcul des commandes v_i et K_i se fait à partir des données communiquées au robot concernant le point de référence C de la trajectoire de référence :

$$[x_c(t) \ y_c(t) \ \theta_c(t)]^T \quad [v_c(t) \ K_c(t)]^T \quad (2.4)$$

Cette méthode permet d'obtenir une formation statique où la forme de la structure varie dynamiquement en fonction de la courbure de la trajectoire de référence. Son plus grand avantage consiste à ne pas avoir à programmer chaque robot à la main. Par contre, en comparant cette méthode statique à d'autres méthodes existantes et d'après les auteurs eux-mêmes, **la complexité du problème n'est pas réduite** et la méthode inspirée d'une structure virtuelle **n'est pas considérée avantageuse par rapport aux autres méthodes**.

L'approche proposée dans [Barfoot and Clark, 2004] peut aussi permettre d'obtenir des formations dynamiques dont la topologie de la structure virtuelle n'est pas fixe en modifiant ou en échangeant les consignes de décalage des robots. Le convoi pourra par exemple passer d'une formation en carré à une formation en colonne pour passer dans un couloir étroit.

Toutefois, le point de référence C ainsi que les commandes calculées dans les deux types de formation statique et dynamique doivent vérifier des contraintes de non-holonomie correspondant à la saturation des vitesses linéaires et angulaires :

$$|K_i| \leq K_{i,max} \quad |v_i| \leq v_{i,max} \quad (2.5)$$

Malheureusement, ce travail ne considère ni l'évitement d'obstacles, ni la phase de fusion des robots pour construire la structure. La partie expérimentation ne couvrant que la formation statique, les auteurs doivent encore tester leur approche dans le cas d'une formation dynamique et dans le cas où les capacités de perception sont limitées.

2.3.1.3 Les approches basées sur la théorie des graphes

Pour réaliser des convois en formation, plusieurs travaux se sont basés sur la théorie des graphes [Eren et al., 2004, Hendrickx et al., 2008] qui permet de modéliser les interactions et l'échange des données entre les différents robots. Dans un graphe, un robot est représenté par un nœud et une interaction entre deux robots par un arc. L'interaction peut s'effectuer soit par communication, soit par une acquisition des données à l'aide des capteurs. En observant les interconnexions au sein de la formation, on peut déterminer si l'approche est, d'une part, globale ou locale, et d'autre part, centralisée ou décentralisée. En revanche, **les limites sur la configuration des robots est difficile à présenter dans un graphe**. La théorie des graphes, appliquée à la réalisation des formations, représente un outil important pour étudier leur stabilité et leur commandabilité ([Lafferriere et al., 2004], [Tanner, 2004]).

[Lafferriere et al., 2004] étudie la stabilisation d'une formation de véhicules en utilisant la théorie des graphes algébrique. Les auteurs ont adopté une approche décentralisée locale où les véhicules échangent avec leurs voisins leurs informations. Cet échange s'effectue à l'aide de la communication selon un graphe prédéfini. Les auteurs considèrent un modèle général de véhicules ayant une dynamique identique et calculent l'erreur de chacun par rapport à la position désirée dans la formation. Cette erreur, étant la moyenne des positions relatives du véhicule par rapport à ses véhicules voisins, est utilisée dans le contrôle par retour d'état pour calculer la commande du véhicule. Les auteurs montrent que la formation est stable si le graphe est connecté³. De plus, **la convergence des véhicules vers la formation désirée peut être améliorée en choisissant une autre alternative de graphes de communication**.

Un autre article [Tanner, 2004] étudie la contrôlabilité d'un système de véhicules en formation. Sachant que les interactions entre les différents robots déterminent le graphe

3. Un graphe est dit connecté s'il existe un chemin reliant chaque paire de véhicules.

de connectivité du système, un choix judicieux du leader est nécessaire afin de pouvoir contrôler ce système. Un modèle cinématique du premier ordre détermine le mouvement de chaque véhicule :

$$\dot{x}_i = u_i \quad (2.6)$$

Celui-ci communique avec un certain nombre de véhicules voisins $|\mathcal{N}_i|$. Le contrôle appliqué sur chaque véhicule de la formation est représenté sous la forme suivante :

$$u_i = -\frac{1}{|\mathcal{N}_i|} \cdot \sum (x_i - x_j) \quad (2.7)$$

Cette commande tend à ramener le véhicule à la position du barycentre de ses voisins. Les auteurs montrent qu'**une connectivité accrue du graphe rend le système moins contrôlable** : un graphe complet est incontrôlable, alors qu'un chemin de ce graphe est contrôlable. Ainsi, la contrôlabilité d'un système est déterminée par la topologie du graphe correspondant.

2.3.1.4 Les approches fondées sur le comportement (behavior-based approaches)

Désignée en anglais par *Behavior-based approach*, cette approche a été introduite par Brooks en 1985. Elle consiste à combiner plusieurs comportements élémentaires ; chaque comportement étant pondéré par son poids relatif, afin d'obtenir un comportement global particulier. Étant donné un ensemble de comportements élémentaires comme éviter un obstacle, suivre une cible, éviter une collision et garder une formation définie, un comportement global peut être défini par un ensemble de robots se déplaçant en formation tout en évitant les obstacles et les collisions.

Plusieurs travaux ont découlé de cette approche. [Antonelli et al., 2006] propose un contrôleur *NSB* (Null-Space-Based Behavioral control) pour réaliser un suivi en convoi. [Monteiro and Bicho, 2002] réalise un suivi en formation avec évitement d'obstacles de robots mobiles à l'aide d'un contrôleur à architecture distribuée utilisant la théorie des systèmes dynamiques. Le comportement global est décomposé en plusieurs comportements élémentaires, où chacun est modélisé par un système dynamique non-linéaire (un attracteur si l'état est asymptotiquement stable et un répulseur si l'état est instable). Ensuite, un vecteur combine ces attracteurs et répulseurs pour déduire le comportement global de chaque robot.

Les approches basées sur le comportement sont utilisées dans les environnements inconnus et dynamiques et ne nécessitent pas de communiquer un nombre important de données. De plus, étant donné que ces approches bénéficient d'un calcul parallèle et distribué, elles peuvent être appliquées en temps réel pendant le déplacement du convoi. En revanche, **la stabilité** d'un convoi se déplaçant selon cette approche **n'est pas garantie**. Sachant que la **dynamique du convoi est difficile à décrire**, une **analyse mathématique des performances de cette méthode n'est pas évidente à réaliser**.

2.3.1.5 Les approches utilisant des modèles cinématiques

Certains travaux s'appliquent dans un milieu urbain où les véhicules du convoi se déplacent à faible vitesse. Sous ces contraintes (véhicule léger, vitesse faible donc pas de glissement), un modèle cinématique suffit pour décrire le comportement d'un véhicule. Le modèle cinématique d'un véhicule est une représentation simplifiée qui ne tient pas compte de la dynamique de son comportement. Ces approches s'appuient sur une modélisation bidimensionnelle du mouvement du véhicule qui impliquent l'usage des deux contrôles longitudinal et latéral.

[Xiang and Bräunl, 2010] considèrent le problème de suivi en convoi comme un jeu de poursuite, où le prédateur ne connaît que la position de sa proie. Il développe alors une stratégie de poursuite où chaque véhicule du convoi perçoit son prédécesseur comme une proie à chasser. Dans ce travail, la connaissance de l'équation analytique du chemin à suivre n'est pas nécessaire puisque le suiveur vise directement la position courante de son prédécesseur et ne cherche pas à s'asservir autour du chemin de ce dernier. Les commandes de vitesse longitudinale v et angulaire ω d'un véhicule visent à contrôler l'inter-distance ρ et la différence d'orientation γ avec son prédécesseur. Elles sont ainsi proportionnelles à ρ et γ , dont leurs positions d'équilibre sont obtenues en fonction des gains k_ρ et k_γ :

$$v = k_\rho \cdot \rho \quad \text{et} \quad \omega = k_\gamma \cdot \gamma$$

Sachant que les gains doivent vérifier certaines inéquations, le choix de leurs valeurs n'est pas évident et peut affecter le suivi. En observant les expérimentations réalisées dans [Xiang and Bräunl, 2010], nous constatons cependant que les gains sont dépendant du type de trajectoire à suivre. Les auteurs ont étudié deux modes de déplacement dans [Xiang and Bräunl, 2010] : le déplacement le long d'une droite et le déplacement le long d'un cercle. Dans les deux cas, une stabilité locale exponentielle a été prouvée. La stabilité globale d'un convoi se déplaçant à vitesse constante a été seulement considérée le long d'un chemin linéaire. Malheureusement, **la stabilité du système n'est pas étudiée pour un chemin à courbure variable**, mais seulement pour des chemins à courbure constante.

2.3.2 Décomposition chemin-vitesse

Contrairement au problème de suivi de trajectoire (section 2.3.1), nous découplons ici les deux contrôles longitudinal et latéral dans le but de traiter chaque contrôle séparément. Nous parlons dans ce cas d'une décomposition chemin-vitesse. D'une part, un contrôle de vitesse vise à déplacer un convoi de robots à faible inter-distance en toute sécurité. D'autre part, suivre un chemin consiste à contrôler le comportement latéral du convoi de façon à ce que l'écart latéral entre les chemins des différents robots soit très faible.

Plusieurs approches ont été suivies pour concevoir des contrôleurs longitudinaux et latéraux. Certains se basent sur un modèle linéaire du système, et d'autres utilisent le modèle non linéaire pour développer les lois de commandes.

Comme nous avons déjà vu dans l'introduction de la section 2.3, le déplacement d'un robot à roues est non linéaire. Pour faciliter la conception des lois de commande, le système peut être linéarisé autour d'un point de fonctionnement. Cependant, ce modèle linéarisé n'est valable que dans un voisinage de ce point de fonctionnement, et la commande conçue ne garantit ni le bon suivi ni la stabilité du système en dehors de ce voisinage.

Nous allons par la suite présenter différentes approches de contrôle basées sur des systèmes linéarisés (PID, commandes optimales linéaires quadratiques, ...) ou non linéaires (approches utilisant des fonctions de Lyapunov, approches basées sur une linéarisation exacte, ...), et ceci pour le contrôle du comportement longitudinal et latéral.

2.3.2.1 Contrôle longitudinal

Le contrôle longitudinal calcule des commandes de vitesse ou d'accélération longitudinale visant à avoir un convoi dont les véhicules sont proches tout en gardant une distance de sécurité. La conception de la commande longitudinale doit garantir soit la stabilité du suivi, soit sa sécurité. Dans ce qui suit, nous présentons différentes méthodes de synthèse du contrôle longitudinal, se basant sur la théorie de contrôle optimal, sur l'automatique linéaire ou bien sur des méthodes de linéarisation exacte.

Approches utilisant un régulateur *PID*

Plusieurs travaux se sont basés sur un régulateur *PID* (Proportionnel - Intégral - Dérivé) pour concevoir un contrôleur longitudinal. Le régulateur *PID* calcule l'erreur du système, qui est la différence entre son état actuel et la consigne désirée, et calcule une commande qui minimise cette erreur. Le calcul de la commande se fait en ajustant les trois paramètres du régulateur : *P* relatif à l'erreur courante, *I* relatif à l'accumulation des erreurs passées et *D* relatif à la prédiction des futures erreurs. Ces trois paramètres doivent être choisis de façon à réduire l'erreur statique par rapport à la consigne, les oscillations et le dépassement de l'erreur.

Quand le système sous-jacent n'est pas connu, le régulateur *PID* constitue une bonne approche pour construire un contrôleur efficace [Bouglès and Guillard, 2012]. Cependant, **la commande calculée n'est pas optimale et ne garantit pas la stabilité du système.**

Dans le cadre du projet Praxitéle [Daviet and Parent, 1995, Daviet and Parent, 1996a], la régulation de l'inter-distance entre deux véhicules consécutifs est réalisée à l'aide d'une loi de commande linéaire proportionnelle dérivée. La commande appliquée sur le suiveur est une accélération longitudinale a_f exprimée sous la forme suivante :

$$a_f = \frac{1}{h} \cdot (\Delta v + k_p \cdot (\Delta x - (d_0 + h \cdot v_f))) \quad (2.8)$$

où Δx et Δv représentent respectivement l'inter-distance et la vitesse relative entre le suiveur et son prédécesseur. k_p représente le coefficient du régulateur *PD* et $(d_0 + h \cdot v_f)$

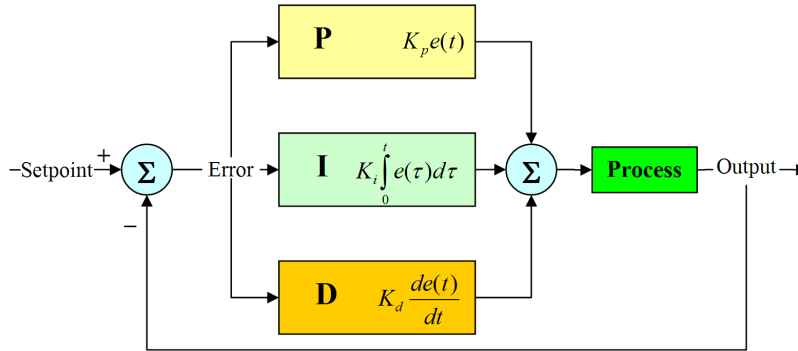


FIGURE 2.10 – Schéma bloc d'un régulateur *PID*.

la consigne d'inter-distance désirée entre les deux véhicules. Cette consigne est la somme d'un écart en distance d_0 et d'un écart temporel $h.v_f$ correspondant à la distance que le suiveur traverse durant une durée h .

Les expérimentations ont été réalisées sur des véhicules électriques équipés d'un système caméra linéaire/flash infrarouge permettant de détecter le véhicule prédécesseur. Les auteurs ont fixé la consigne d'inter-distance à 4,5 m ($d_0 = 1$ m et $h = 0,35$) pour des véhicules qui se déplacent à 10 m/s. **L'erreur longitudinale non négligeable** est inférieure à 30 cm pour des accélérations de ± 2 m/s², et à 50 cm pour des arrêts d'urgence à une décélération de 5 m/s². Les paramètres d_0 et h affectent ainsi le comportement du convoi, et **leur choix doit se faire en fonction des capteurs et du temps de réponse des actionneurs** dans le but d'avoir une circulation fluide (faible inter-distance) et sûre (absence de collision).

[Khatir and Davison, 2004] présente une approche décentralisée pour le contrôle longitudinal d'un convoi de robots. Les auteurs définissent l'erreur longitudinale en fonction des positions d'un couple de véhicules comme suit :

$$e_i = x_{i-1} - x_i - d^* ; \quad (2.9)$$

où d^* représente l'inter-distance minimale admissible.

Pour réduire l'erreur longitudinale entre le leader et le suiveur, un contrôleur de type PID est alors proposé, avec **des gains qui dépendent de chaque véhicule suiveur**. Les contrôleurs PID ne sont donc pas identiques pour tous les véhicules. Les gains du PID du véhicule i sont exprimés en fonction de ceux du véhicule $i - 1$ comme suit :

$$\begin{cases} KI_i \geq KI_{i-1} \\ KP_i = \frac{KI_i}{KI_{i-1}} KP_{i-1} + \frac{m}{KD_{i-1}} KI_{i-1} \\ KD_i = \frac{KI_i}{KI_{i-1}} KD_{i-1} + \frac{m}{KD_{i-1}} KP_{i-1} - b \end{cases} \quad (2.10)$$

En observant ces équations, nous constatons que les gains augmentent avec l'indice du véhicule dans le convoi. Sachant que des gains plus importants conduisent à des commandes plus brusques, l'approche proposée dans [Khatir and Davison, 2004] pourrait poser problème sur de longs convois.

Approche basée sur la théorie du contrôle optimal La théorie du contrôle optimal consiste à calculer la commande d'un système soumis à des contraintes dans le but d'optimiser un critère de performance [Bertsekas, 2007]. Ce critère de performance est une fonction de coût qui dépend de l'état du système et des variables de contrôle.

Dans [Levine and Athans, 1966], La commande d'une flotte de véhicules est abordée sous forme d'un problème d'optimisation. Sachant que le convoi se déplace à grande vitesse, les auteurs désirent concevoir un contrôleur longitudinal optimal afin de contrôler la position et la vitesse de chaque véhicule du convoi. Pour cela, ils expriment les équations dynamiques de mouvement en fonction des forces agissant sur le robot et définissent l'expression des erreurs sur la position, la vitesse et l'accélération (ou la force appliquée). **En supposant que l'erreur sur la vitesse est faible**, les auteurs linéarisent le système et définissent les équations de variation des erreurs sur la position et la vitesse. Ensuite, ils définissent une fonction coût quadratique J basée sur les erreurs de position, de vitesse et d'accélération (ou de force étant donné que $force = masse \cdot accélération$). J est alors la somme pondérée de ces trois erreurs, **les coefficients de pondération sont choisis plus ou moins arbitrairement (de l'aveu même des auteurs) de façon à donner plus de poids à une erreur qu'à une autre**. Étant donné le système de variation des erreurs et la fonction d'optimisation J , les auteurs calculent un vecteur commande du convoi qui est alors l'ensemble des forces à appliquer sur les différents véhicules de façon à minimiser J .

Des simulations réalisées sur un convoi de trois véhicules ont montré que l'erreur sur la position de chaque robot a été réduite à zéro rapidement mais avec un faible dépassement⁴ (un passager n'apprécie pas un dépassement important de l'erreur vu que la dynamique calculée induit un mouvement brutal du véhicule). Étant donné que l'approche adoptée est centralisée, la force appliquée sur chaque véhicule est fonction des positions et vitesses de tous les autres véhicules du convoi. Ainsi, **plus la taille du convoi est importante, plus l'échange de données au sein du convoi est coûteux**. Les auteurs soulignent 4950 ($99 * 50$) communications pour un convoi de 50 véhicules. En considérant que l'influence du premier véhicule sur le dernier n'est pas remarquable, les auteurs proposent de réduire ce coût de communication en ne considérant que les véhicules voisins (le véhicule devant et celui derrière) pour le calcul de la force à appliquer sur un véhicule donné du convoi.

Contrôle basé sur des techniques de linéarisation exacte La technique de linéarisation exacte utilise une transformation exacte dans le but d'écrire le système non linéaire

4. Un dépassement d'erreur se produit quand la sortie du système est plus importante que la réponse désirée (consigne).

sous forme d'un système linéaire de façon à pouvoir concevoir des lois de commande en exploitant les méthodes de l'automatique linéaire. Isidori présente dans [Isidori, 1995] plusieurs techniques de linéarisation exacte dont la plus célèbre est une approche géométrique par retour d'état.

[Sheikholeslam and Desoer, 1993] formule le problème du suivi de convoi le long d'une ligne droite comme un problème d'optimisation sous contraintes. Les auteurs proposent une approche de contrôle longitudinal sans communication utilisant une méthode de linéarisation exacte. Étant sur un chemin linéaire, la configuration d'un véhicule i est représenté par son abscisse x_i , et son erreur longitudinale est représentée par $\Delta_i = x_{i-1} - x_i - L$, L étant la consigne d'inter-distance entre deux véhicules. Le modèle dynamique du mouvement longitudinal du véhicule est exprimé sous forme de deux équations :

- une première équation représente la dynamique du moteur du véhicule i ;
- une deuxième équation représente le mouvement du véhicule i : $\Sigma F = m.a$.

Ensuite, les auteurs linéarisent le comportement dynamique du robot en appliquant des méthodes de linéarisation exacte [Isidori, 1995, Sastry and Isidori, 1989] et formalisent le problème de suivi sous forme d'un problème d'optimisation. Ils proposent par la suite un contrôle dont les constantes sont calculées de façon à avoir les fonctions de transfert⁵ des véhicules stables. En testant leur approche de contrôle, les auteurs ont constaté que **l'inter-distance entre deux véhicules consécutifs augmente le long du convoi**. En comparaison avec les performances de la méthode présentée ici, les performances des approches qui utilisent la communication pour calculer un contrôle basé sur les données de tous les véhicules du convoi se dégradent mais ne sont pas catastrophiques. Ainsi, ce contrôle peut être utilisé dans le cas où une perte de communication surgit sur le système.

Les approches de linéarisation exacte présentent quelques inconvénients : d'une part, **certaines problématiques comme la robustesse et l'atténuation de la perturbation ne peuvent pas être abordées** ; d'autre part, **tous les systèmes ne peuvent pas être exactement linéarisés (en suivant cette approche) tout en maintenant leur stabilité**. Dans le cadre de la robotique mobile, Samson a montré dans [Samson, 1995] que le modèle cinématique d'un robot mobile ne peut pas être linéarisé d'une manière exacte. Il propose alors de le convertir en systèmes chaînés où les contrôles longitudinal et latéral sont découplés et conçus chacun à part.

Bom a présenté dans sa thèse [Bom, 2006] une approche de platooning basée sur la transformation chaînée. Se basant sur un RTK-GPS pour localiser les robots, le travail considère un convoi à faible dimension, se déplaçant à faible vitesse longitudinale sans glissement. Chaque véhicule est représenté par un modèle d'Ackermann (voir Fig. 2.11), où les roues arrières sont motrices et les roues avant sont directrices. Pour modéliser l'évolution d'un véhicule le long d'une trajectoire de référence, J.Bom utilise un repère de Frenet. L'état du véhicule est alors représenté par sa distance curviligne parcourue, son écart latéral à la trajectoire et son erreur angulaire par rapport à la tangente à la trajectoire au point le plus proche ([Bom, 2006]). Pour découpler les deux contrôles longitudinal

5. Une fonction de transfert d'un système est une expression mathématique qui relie son entrée (qui est la commande) à sa sortie (son nouvel état).

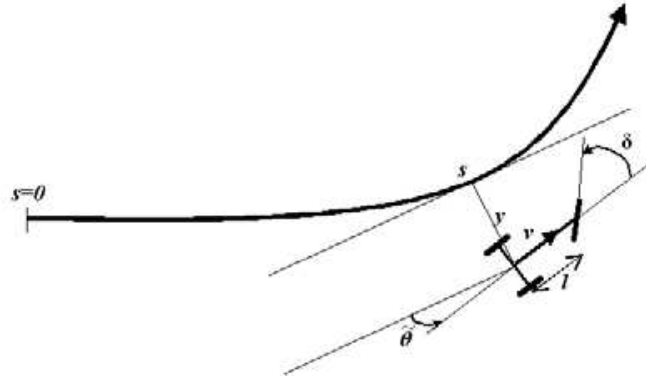


FIGURE 2.11 – Modélisation d'un véhicule selon le modèle d'Ackermann.

et latéral, une linéarisation exacte du système est alors réalisée en utilisant la transformation chaînée présentée dans [Samson, 1995]. Ceci permet de contrôler l'orientation du véhicule indépendamment de sa vitesse longitudinale. Dans ce paragraphe, nous présentons le contrôle longitudinal développé par Bom, et dans la section relative au contrôle latéral, nous présentons le contrôle latéral.

Le calcul du contrôle longitudinal, **nécessitant une communication** entre les véhicules, est une combinaison de deux approches ([Bom et al., 2005]) : une approche locale et une approche globale. Dans une approche locale, chaque robot tend à garder une inter-distance curviligne constante avec son prédécesseur. **La propagation de l'erreur longitudinale dans le convoi crée un mouvement oscillatoire et par la suite une instabilité sur les longs convois.** Ce mouvement oscillatoire, traduit par un effet ressort, induit un manque de confort chez les passagers. Pour éliminer les inconvénients de la méthode locale, une méthode globale consiste à calculer une vitesse réduisant l'erreur longitudinale entre le véhicule et le leader du convoi. Un risque de collision avec son prédécesseur peut apparaître durant le déplacement du véhicule en convoi.

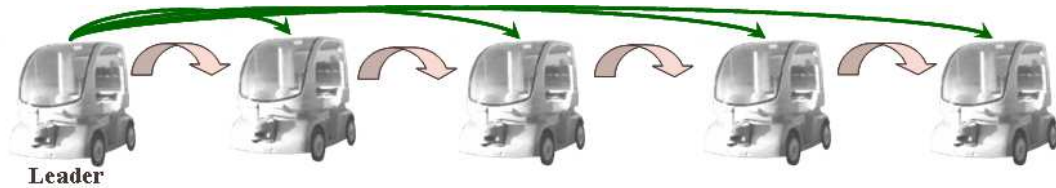


FIGURE 2.12 – Commande longitudinale hybride basée sur les erreurs longitudinales entre un suiveur et le leader du convoi d'une part et un suiveur et son prédécesseur d'autre part (figure tirée de [Bom, 2006]).

Pour décider quel contrôle adopté durant le déplacement du convoi, une commande longitudinale hybride (voir Fig. 2.12) est exprimée en fonction des deux erreurs longitudinales (mentionnées précédemment) comme suit :

$$c_{i+1} = \sigma_{i+1}.e_{i+1}^1 + (1 - \sigma_{i+1}).e_{i+1}^i \quad (2.11)$$

où σ définit la prédominance d'une approche par rapport à l'autre, c_{i+1} est la commande

longitudinale du robot $i+1$, e_{i+1}^1 est l'erreur longitudinale entre le robot ($i+1$) et le leader du convoi et e_{i+1}^i est l'erreur longitudinale entre le robot ($i+1$) et son robot prédécesseur (i). À titre d'exemple, si le véhicule est à une distance très proche (voir dangereuse) de son prédécesseur, la priorité du contrôle longitudinal est donné à l'approche locale.

Approche analytique exacte : contrôleur longitudinal SSC Tous les contrôleurs longitudinaux présentés ci-dessus ont été conçus pour le suivi mais, à notre connaissance, aucun ne garantit l'absence de collision à l'exception du contrôleur *SSC* (Scheuer-Simonin-Charpillet) [Scheuer et al., 2009] détaillé dans ce paragraphe. Ce contrôleur décentralisé local n'a pas besoin d'une infrastructure particulière (pas besoin de GPS ni de communication). De plus, le contrôleur *SSC* est conçu de façon à gérer les erreurs de perception.

Pour concevoir ce contrôleur, le critère principal considéré est l'absence de collision, contrairement aux autres contrôleurs longitudinaux où le critère principal est la stabilité. La conception de ce contrôleur découle de l'étude d'un cas dangereux de suivi de deux véhicules où le suiveur avance avec l'accélération maximale admise pendant que son prédécesseur freine à fond. Sachant que, d'une façon générale, la propriété de non collision est obtenue quand l'inter-distance entre deux robots est strictement positive, ce travail considère que l'inter-distance doit être supérieure à une certaine distance de sécurité d_{crit} (prise aussi petite que l'on désire) pour vérifier cette propriété de non collision.

Le travail présenté dans [Scheuer et al., 2009] s'intéresse au contrôle longitudinal le long d'une ligne droite ; les résultats pouvant être étendus au problème général du platooning. Dans les chapitres relatifs aux deux algorithmes de contrôle *Memo-LAT* et *NOC*, le contrôleur *SSC* est appliqué sur un chemin curviligne quelconque. Le contrôleur conçu n'est pas contraint à un type particulier de robots, au contraire, il considère un modèle générique de robots. [Scheuer et al., 2009] suppose qu'un robot a un contrôleur bas niveau lui permettant d'atteindre l'accélération consigne. De plus, tous les robots du convoi se déplacent à vitesse v et accélération a bornées :

$$\begin{aligned} 0 \leq v_{min} \leq v \leq v_{max} \\ a_{min} \leq a \leq a_{max} \quad \text{avec : } a_{min} < 0 < a_{max} \end{aligned}$$

Un conducteur humain dirige le convoi en appliquant à chaque pas de temps t_i une nouvelle consigne de vitesse v_i . Le profil global de la vitesse du véhicule de tête est alors une suite de couples (t_i, v_i) . Pour passer d'une vitesse v_i à une autre v_{i+1} , le leader applique soit une accélération maximale a_{max} soit une décélération maximale a_{min} selon que v_i est supérieure ou bien inférieure à v_{i+1} . Une fois v_{i+1} atteinte, l'accélération appliquée devient nulle et la vitesse v_{i+1} est maintenue jusqu'au prochain pas de temps. Le profil de la vitesse du leader prend alors entre deux pas de temps le profil semblable à celui présenté dans la figure 2.13.

Pour calculer son comportement, chaque suiveur répète successivement les trois étapes suivantes :

- percevoir l'environnement ;

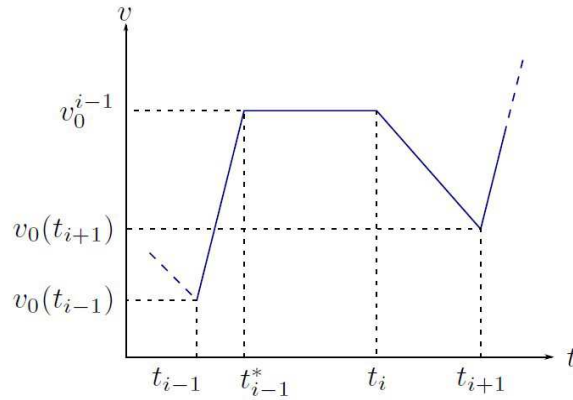


FIGURE 2.13 – Profil de la vitesse du véhicule de tête (figure tirée de [Scheuer et al., 2009]).

- calculer la commande d'accélération longitudinale ;
- envoyer la commande calculée au contrôleur bas niveau.

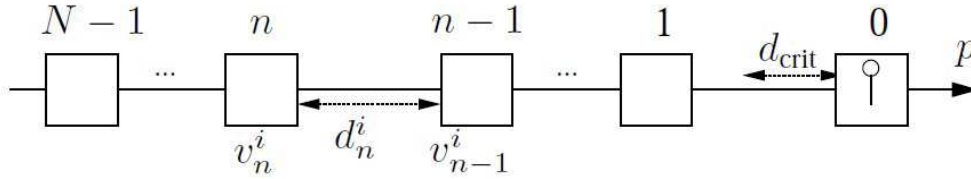


FIGURE 2.14 – Représentation d'un convoi de robots en mode platooning le long d'une ligne droite (figure tirée de [Scheuer et al., 2009]).

L'approche de contrôle développée ici distingue deux durées de temps :

- τ représente le temps nécessaire pour la perception, le calcul de la commande et sa transmission au contrôleur bas niveau ;
- δt représente la durée d'un cycle de contrôle qui comprend la durée τ plus le temps d'application de la commande.

À chaque pas de temps $i\delta t$, le robot suiveur estime sa propre vitesse longitudinale v_n^i , celle de son prédécesseur v_{n-1}^i , et l'inter-distance les séparant d_n^i (la position des robots dans un repère global n'est pas utilisée dans le calcul du contrôle longitudinal).

Un déplacement sans collision existe si la capacité maximale du robot suiveur à freiner permet d'éviter cette collision. En d'autres termes, l'inter-distance doit vérifier l'inégalité suivante :

$$d_n^i \geq d_{crit} + \max \left(0, \frac{v_n^{i2} - v_{n-1}^{i2}}{-2a_{min}} \right) \quad (2.12)$$

Ayant un convoi de robots initialement au repos et assez éloignés les uns des autres, l'inégalité 2.12 est vérifiée si l'accélération longitudinale du suiveur a_n^i reste inférieure à une accélération limite a_{lim} donnée par la formule 2.13 :

$$a_{lim}(d_n^i, v_n^i, v_{n-1}^i) = \min\{a_1, a_2, a_3\} \quad (2.13)$$

avec :

$$\begin{aligned} a_1 &= a_{min} + 2 \frac{\tilde{d}_n^i - d_{crit} + (\tilde{v}_{n-1,n}^i - \tilde{v}_n^i)\delta t}{3\delta t^2}, \\ a_2 &= \frac{\sqrt{(\tilde{v}_n^i - a_{min}\frac{\delta t}{2})^2 - (2a_{min}\tilde{\delta}d_n^i) - (\tilde{v}_n^i - \frac{3}{2}a_{min}\delta t)}}{\delta t}, \\ a_3 &= \frac{\sqrt{(\tilde{v}_n^i + (a_{max} - \frac{a_{min}}{2})\delta t)^2 - 2a_{min}\tilde{D}_n^i}}{\delta t} - \frac{\tilde{v}_n^i + (a_{max} - \frac{3}{2}a_{min})\delta t}{\delta t} \end{aligned}$$

et où :

$$\begin{aligned} \delta d_n^i &= d_n^i - d_{crit} + \min\left\{0, \frac{v_n^{i2} - v_{n-1}^{i2}}{2a_{min}}\right\} \\ \tilde{d}_n^i &= d_n^i + (v_{n-1}^i - v_n^i)\delta t + (a_{min} - a_{max})\frac{\delta^2}{2} \\ \tilde{v}_n^i &= v_n^i + a_{max}\delta t \\ \tilde{v}_{n-1,n}^i &= v_{n-1}^i + a_{min}\delta t \\ \tilde{\delta}d_n^i &= \tilde{d}_n^i - d_{crit} + \frac{v_n^{i2} - v_{n-1}^{i2}}{2a_{min}} \\ \tilde{D}_n^i &= \max\{0, \tilde{\delta}d_n^i - (a_{max} - a_{min})(\tilde{v}_n^i + a_{max}\delta t/2)\delta t/(-a_{min})\} + (a_{max} - a_{min})\delta t^2 \end{aligned}$$

a_{lim} représente l'accélération maximale qu'on peut appliquer sur un robot du convoi tout en garantissant l'absence de collisions. Le robot suiveur a ainsi une accélération longitudinale appartenant à l'intervalle $[a_{min}, a_{lim}]$. En appliquant a_{lim} , les robots du convoi se déplacent tout en restant les uns proches des autres. Pour avoir un autre comportement longitudinal du convoi, le contrôleur *SSC* peut aussi être combiné à d'autres contrôleurs longitudinaux qui calculent une accélération γ . La commande appliquée sur le robot est dans ce cas $\min\{a_{lim}, \gamma\}$.

Après avoir calculé la commande longitudinale, son application sur le robot se fait comme suit : au pas de temps $i\delta t$ et durant un cycle de contrôle δt , le robot passe une durée τ à percevoir son environnement, calculer sa commande a_i et la transmettre au contrôleur bas niveau. Avant d'obtenir cette commande d'accélération, le robot continue à appliquer l'accélération a_{i-1} précédemment calculée au pas de temps précédent $(i-1)\delta t$. Ensuite, à $i\delta t + \tau$, le robot applique la commande a_i jusqu'au calcul de la nouvelle accélération. Le calcul de la position du robot ayant une position p et une vitesse v se fait alors à deux instants : à l'instant $i\delta t + \tau$ en utilisant l'ancienne accélération a_{i-1} et à l'instant $(i+1)\delta t$ en appliquant a_i . Sachant que les résultats expérimentaux présentés

dans [Scheuer et al., 2009] sont obtenus par simulation, le calcul de la position du robot est obtenu selon la fonction $move : (p, v, a, t) \rightarrow (p, v)$ suivante :

$$move(p, v, a, t) = \begin{cases} (v_{min}, p + v_{min}t - (v - v_{min})^2/a) & \text{si } (v + at < v_{min}) \\ (v_{max}, p + v_{max}t - (v_{max} - v)^2/a) & \text{si } (v + at > v_{max}) \\ (v + at, p + vt + at^2/2) & \text{sinon} \end{cases} \quad (2.14)$$

2.3.2.2 Contrôle latéral

Le contrôle latéral calcule une suite de vitesses angulaires tout au long du déplacement du convoi visant à mener les robots à suivre avec précision le chemin de leur leader sans osciller autour. La précision et la stabilité sont ainsi les critères de conception des lois de commandes latérales. Dans ce qui suit, nous présentons différentes méthodes de synthèse du contrôle latéral.

Approches utilisant un régulateur *PID* Plusieurs travaux ont utilisé des régulateurs proportionnel P , proportionnel dérivé PD , ou bien proportionnel dérivé intégral PID afin d'asservir l'erreur du système autour de zéro. Le fonctionnement des régulateurs PID est expliqué dans le paragraphe 2.3.2.1.

[Micaelli and Samson, 1993] présentent un contrôle basé sur la linéarisation du système et l'application d'un régulateur proportionnel dérivé. La commande issue de cette approche de contrôle peut être appliquée sur des robots uni-cycles ou bien sur des robots à deux roues (avant et arrière) directionnelles dans le but de suivre un chemin curviligne quelconque (C). Un robot uni-cycle est contrôlé par sa vitesse angulaire ω , alors qu'un robot à deux roues directionnelles est contrôlé par deux variables : une première commande agit sur l'angle de braquage de la roue avant α et une deuxième commande agit sur la distance du centre de la roue avant au centre de rotation du robot $\frac{1}{\sigma}$.

Un robot mobile M est représenté dans un repère orthonormé par ses coordonnées (X, Y) ou leurs équivalents (s, y) où s représente l'abscisse curviligne du robot le long de (C) et y l'écart latéral de M par rapport à (C). Après avoir modélisé le mouvement au cours du temps de chaque type de robots, les auteurs définissent une variable η comme étant la distance curviligne parcourue par le robot le long du chemin (C) et représentent l'évolution de l'état de M (s', y', θ') en fonction de η (θ étant l'erreur angulaire de M par rapport à (C)).

Dans le cas d'un robot uni-cycle, en dérivant encore une fois y' par rapport à η , la commande ω apparaît dans y'' . L'équation y'' est linéarisée en introduisant un contrôle auxiliaire u de façon à obtenir $y'' = u$. Ce contrôle auxiliaire est alors calculé de sorte que le système soit stable. Les auteurs choisissent alors un régulateur $PD : u = -k_{p_y}y - k_{v_y}y'$, où k_{p_y} et k_{v_y} sont deux coefficients positifs. Ensuite, ils expriment le contrôle ω en fonction de u , de l'état du robot $\{v, \theta, y\}$ et du chemin à suivre. La même démarche est suivie pour établir les lois de commande d'un robot à deux roues directionnelles. Les auteurs

dérivent y' et $\tilde{\theta}'$ par rapport à η et obtiennent les deux contrôles $\hat{\alpha}$ et $\hat{\sigma}$ en utilisant aussi des régulateurs *PID*.

Ce contrôle suppose que la courbure et la variation de la courbure en fonction de l'abscisse curviligne, étant toutes les deux bornées, sont connues. De plus, **la convergence n'est garantie que si certaines conditions initiales sont satisfaites.** Parmi ces conditions, **une contrainte s'impose sur l'erreur angulaire initiale** dans le cas d'un robot uni-cycle. En revanche, si $\frac{\pi}{2} < \theta(0) < \frac{3\pi}{2}$, l'écart latéral tend vers zéro tandis que l'erreur angulaire tend vers π . De même, si $\theta(0) = \pm\frac{\pi}{2}$, le robot ne converge pas vers le chemin désiré. Pour que les erreurs latérale et angulaire tendent vers zéro, $|\theta(0)|$ doit être strictement inférieur à $\frac{\pi}{2}$. Dans le cas d'un robot à deux roues directionnelles, ces mêmes contraintes s'appliquent non pas sur l'angle θ mais plutôt sur $\theta + \alpha$ où α est l'angle de braquage des roues avant. De plus, les erreurs latérale et angulaire sont découplées de façon à ce qu'une petite perturbation sur l'une n'affecte pas l'autre.

Malgré la convergence exponentielle du robot vers le chemin (C), **le domaine de stabilité de la commande construite dans cette approche reste très petit** à cause des singularités citées auparavant.

De même, [Daviet and Parent, 1996a] propose aussi un régulateur *PID* visant à minimiser l'erreur angulaire entre un couple de robots successifs du convoi. Après avoir calculé l'orientation désirée du robot suiveur, les auteurs déduisent l'erreur angulaire qui n'est autre que la différence entre l'orientation courante du robot et son orientation désirée et appliquent un régulateur *PID* qui asservit cette erreur autour de zéro.

Le contrôleur développé dans [Daviet and Parent, 1996a] **ne garantit pas la stabilité** du système étant donné que des oscillations surgissent durant le suivi. De plus, **le suivi n'est pas précis** surtout au niveau des virages : en visant l'arrière du robot prédécesseur, **le robot coupe les virages et dévie du chemin emprunté par son prédécesseur.** Pour améliorer la précision du suivi, [Daviet and Parent, 1996a] proposent d'autres lois de contrôle plus complexes synthétisées à partir de polynômes d'ordre 3 et 4. Ils proposent ensuite une autre méthode qui consiste à mémoriser le chemin du prédécesseur pour réduire l'écart latéral entre les chemins du leader et des suiveurs. En revanche, aucun résultat expérimental concernant cette méthode n'a été présenté.

Approche utilisant des fonctions de Lyapunov Ce type d'approche s'appuie sur la théorie de stabilité de Lyapunov pour synthétiser la commande. Étant donné un système non linéaire $\dot{X} = f(X)$, l'étude de sa stabilité autour d'un point d'équilibre $X_e = 0$ peut être réalisée à l'aide des fonctions de Lyapunov $V(X)$. Ces fonctions sont définies dans un voisinage D du point d'équilibre X_e comme suit :

- $V(X)$ est une fonction continue et définie positive tel que $V(X_e) = V(0) = 0$ et $V(X) > 0 \forall X \in D - \{0\}$.
- La dérivée de $V(X)$ est négative : $\dot{V}(X) < 0$. Ceci signifie que le système est asymptotiquement stable (le système est stable et tend vers la configuration d'équilibre

X_e).

Plusieurs travaux exploitent les fonctions de Lyapunov dans la conception des lois de commande U de systèmes non linéaires $\dot{X} = f(X) + g(X, U)$. Pour qu'un système converge vers un point d'équilibre X_e (qui représente la consigne), la commande est synthétisée en suivant la procédure suivante :

- choisir une fonction $V(X)$ définie positive dans un voisinage de X_e .
- exprimer la dérivée de $V(X)$ en fonction de l'état X et de la commande U .
- vérifier s'il est possible de construire une fonction $U(X)$ tel que $\dot{V}(X, U(X)) < 0$.
- si $U(X)$ existe de façon à ce que $V(X)$ soit asymptotiquement stable, le système tend vers X_e . De plus, vu que la commande $U(X)$ est fonction de son état X , le système est autonome. $V(X)$ est alors une fonction de Lyapunov. **Si ce n'est pas possible de construire une fonction $U(X)$, il faut choisir une autre fonction $V(X)$ et recommencer cette procédure.**

Selon J. Bom [Bom, 2006], *"La synthèse de lois de commande à partir de fonctions de Lyapunov s'appuie sur des outils mathématiques relativement simples, réclame en général peu de calcul, mais en contre-partie requiert de l'intuition et de l'expérience pour choisir une fonction de Lyapunov qui peut conduire à une loi de commande pertinente."*

Dans [Micaelli and Samson, 1993], une loi de commande est synthétisée en s'appuyant sur des fonctions de Lyapunov dans le but de faire converger un robot vers un chemin (C). Deux types de robots sont considérés pour la synthèse du contrôle : un robot uni-cycle contrôlé par sa vitesse angulaire ω , et un robot à deux roues (avant et arrière) directionnelles contrôlé par l'angle de braquage de sa roue avant α et la distance $\frac{1}{\sigma}$ du centre de la roue avant au centre de rotation du robot. Pour contrôler un robot uni-cycle, les auteurs choisissent une fonction de Lyapunov V basée sur l'écart latéral et l'erreur angulaire et obtiennent l'expression de la commande ω en dérivant V . Dans le cas d'un robot à deux roues directionnelles, deux fonctions de Lyapunov sont choisies et dérivées afin d'obtenir les commandes $\dot{\alpha}$ et $\dot{\sigma}$ permettant de réduire les erreurs latérale et angulaire. **La convergence des erreurs latérale et angulaire n'est garantie que si certaines contraintes sur l'écart latéral initial du robot sont satisfaites.**

Contrôle basé sur une technique de linéarisation exacte Comme déjà mentionné dans 2.3.2.1, les deux contrôles longitudinal et latéral sont découplés. Après avoir présenté la synthèse du contrôle longitudinal, nous nous intéressons ici au contrôle latéral. En effet, le calcul de la commande latérale consiste à asservir à zéro les erreurs latérale et angulaire par rapport au chemin du leader du convoi. Ainsi, en se basant sur le système linéarisé exact, [Bom, 2006] utilise un régulateur proportionnel dérivé dont les coefficients déterminent la vitesse de convergence du système. **Un bon réglage de ces coefficients est**

alors indispensable afin d'éviter les comportements oscillatoires des véhicules autour de la trajectoire de référence. Ensuite, l'angle de braquage à appliquer sur le suiveur est obtenu en inversant les transformations chaînées **à condition que certaines conditions soient vérifiées** : le robot se déplace à une vitesse non nulle, avec une erreur angulaire strictement inférieure en valeur absolue à $\frac{\pi}{2}$, **sans jamais se retrouver au centre de courbure du chemin de son leader**.

Ce travail suppose que chaque véhicule du convoi **connaît l'expression analytique de la trajectoire à suivre afin de récupérer, en tout point, la courbure et la dérivée de la courbure**. En plus la localisation de chaque véhicule est basée sur un GPS. Or, **l'utilisation du signal GPS seul ne garantit pas toujours une localisation continue et fiable** des véhicules du fait que dans les environnements urbains, la communication avec les satellites peut être perdue. Pour résoudre ce problème, [Bom, 2006] propose de combiner le signal GPS avec des capteurs inertiels, ou bien d'utiliser la vision.

Ces deux problèmes (localisation à l'aide d'un GPS et nécessité de connaître l'expression analytique de la trajectoire du leader) sont traités dans [Avanzini, 2010]. La localisation est réalisée par vision où le GPS est remplacé par des caméras moins chères. Sachant que l'endroit de déplacement du convoi est riche en objets de référence, plusieurs amers sont repérés par traitement d'images après une première navigation du véhicule. Ensuite, pendant son déplacement en mode autonome, le véhicule récupère sa position par appariement d'amers [Avanzini et al., 2008]. Dans le cas où la trajectoire du leader n'est pas connue mais seulement ses positions successives, [Avanzini et al., 2009] présente une méthode permettant de générer en ligne l'expression analytique de la trajectoire du leader (voir section 2.4).

2.4 Méthodes d'estimation de chemins ou de trajectoires

Les méthodes de suivi de trajectoires ou de chemins considèrent généralement que l'équation analytique de la courbe générée par le leader est connue. En d'autres termes, chaque robot du convoi connaît à l'avance l'équation de la courbe à suivre et applique un contrôle pour s'asservir autour de cette courbe sans rentrer en collision avec les autres robots du convoi. Or, ce n'est pas toujours le cas. À moins que l'équation analytique de la courbe à suivre ne soit donnée directement à chaque robot avant le début du suivi, ce dernier n'aura qu'une suite de points acquis ou communiqués durant son déplacement. Du coup, plusieurs chercheurs se sont intéressés à élaborer des méthodes permettant de reconstruire la trajectoire du leader. Leur défi est d'avoir une approximation aussi fidèle que possible de la trajectoire réelle.

Chaque robot suiveur acquiert, à chaque pas de temps, un nouveau point de la courbe à suivre. Ce point peut être représenté de différentes façons : un point dans un référentiel cartésien ayant les coordonnées x et y , un point daté (x, y, t) , ou bien un point orienté (x, y, θ) .

Dans certains travaux, le robot vise à rejoindre successivement les différents points mémorisés, ce qui rend la connaissance de la courbe à suivre inutile. Dans ce cas, le robot planifie un chemin lui permettant de passer de sa configuration courante à une configuration désirée (la configuration désirée est le point visé de la courbe du leader). Pour déplacer un robot d'un point de départ vers un point d'arrivée, [Albagul and Wahyudi, 2004] propose de planifier un chemin formé de deux segments de droite reliés par un arc de cercle. Les deux articles [Gil-Pinto et al., 2005] et [Gil-Pinto et al., 2007] proposent d'utiliser des courbes B-splines pour planifier une trajectoire permettant à un robot de passer de sa configuration courante à une configuration désirée. Les coefficients des B-splines sont déterminées à l'aide d'un solveur SQP (sequential quadratic programming) de façon à satisfaire certaines contraintes sur le système (commande et angle de braquage bornés) et un critère d'optimisation consistant à calculer une trajectoire permettant au robot d'atteindre son but le plus vite possible.

2.4.1 Chemin de Dubins

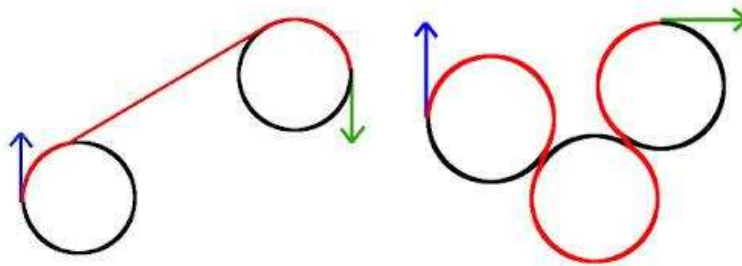


FIGURE 2.15 – Chemins de Dubins reliant un point de départ (en bleu) et un point d'arrivée (en vert).

Pour relier deux points orientés, Dubins a proposé un chemin optimal composé d'une succession d'arcs de cercle et/ou de segments de droite. Il a montré que le chemin reliant deux configurations est formé, au plus, de trois morceaux de chemins : arc de cercle - segment de droite - arc de cercle, ou bien arc de cercle - arc de cercle - arc de cercle. Ainsi, le premier et le dernier arc de cercle, à courbure positive ou négative, sont reliés tangentiuellement par un segment de droite ou bien par un autre arc de cercle (Fig. 2.15). Si le robot a des contraintes de mouvement, les arcs de cercle du chemin de Dubins ont une courbure maximale correspondant à la capacité maximale du robot à tourner. Ce chemin est alors le plus court chemin joignant deux points orientés en respectant des contraintes sur sa courbure. Malgré le fait que le chemin de Dubins est optimal, il représente certains inconvénients. En effet, la courbure n'est pas continue aux points reliant les arcs de cercle avec les segments de droite ou les autres arcs de cercle. Dans les applications réelles de robotique, aucun robot ne peut changer instantanément de courbure sans passer par une vitesse longitudinale nulle. Ce comportement non acceptable dans de nombreuses applications force les chercheurs à trouver d'autres solutions de planification de trajectoires.

2.4.2 Chemin à courbure continue

Pour éliminer la limite sur la discontinuité de la courbure, [Scheuer and Fraichard, 1997] propose de relier deux points orientés par un chemin à courbure continue bornée par κ_{max} et dont la variation est limitée par σ_{max} . Ce chemin est représenté par une succession d'arcs de cercle et de clothoïde, et de segments de droite. Les clothoïdes sont des courbes dont la courbure varie proportionnellement à l'abscisse curviligne. [Scheuer and Fraichard, 1997] propose un chemin reliant deux configurations données et composé au plus de huit morceaux, chaque morceau étant un segment de droite, un arc de cercle ou un arc de clothoïde.

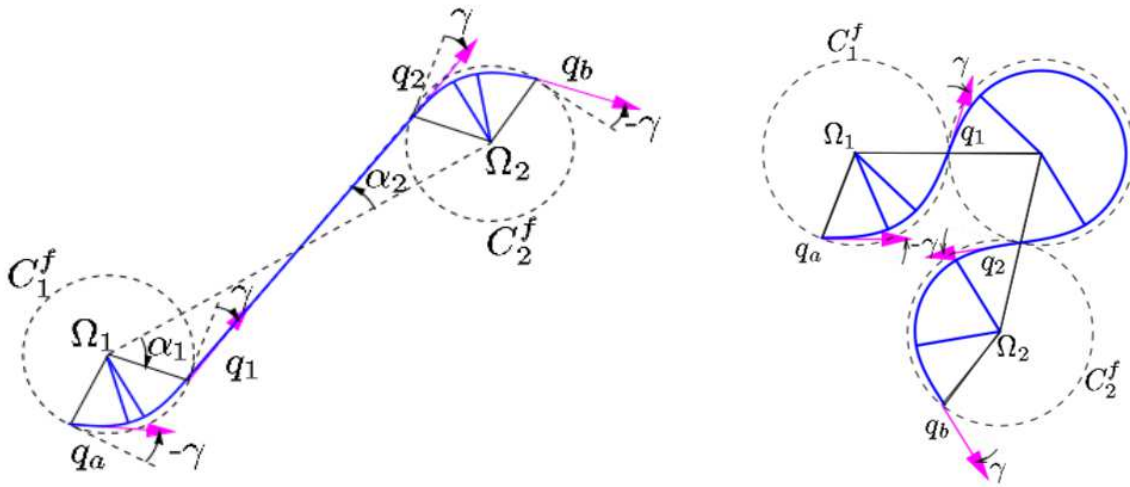


FIGURE 2.16 – Chemin à courbure continue reliant une configuration de départ q_a à une configuration d'arrivée q_b (figure tirée de [Scheuer and Fraichard, 1997]).

Au lieu de relier deux configurations par une succession d'arcs de cercles et de segments de droites (comme c'est le cas dans la figure 2.15), la figure 2.16 montre que ces deux configurations peuvent être reliées par des chemins à courbure continue formés d'arcs de cercles et de clothoïdes et/ou de segments de droites. L'arc de cercle de Dubins est alors remplacé par un arc ayant un point de symétrie et formé de trois morceaux : un premier arc de clothoïde dont la courbure croît de 0 jusqu'à la valeur maximale κ_{max} , un arc de cercle de rayon $\frac{1}{\kappa_{max}}$ et un arc de clothoïde dont la courbure décroît de κ_{max} jusqu'à 0. Le premier arc de clothoïde a une variation de courbure égale à σ_{max} alors que le deuxième a une variation de courbure égale à $-\sigma_{max}$; les deux arcs ayant la même longueur $l = \frac{\kappa_{max}}{\sigma_{max}}$. En suivant ce raisonnement, la figure 2.17 montre la variation de la courbure dans le cas du chemin à gauche représenté dans la figure 2.16. Dans certains cas, l'arc à courbure continue remplaçant l'arc de Dubins n'est pas toujours formé de trois morceaux, mais seulement de deux morceaux correspondant à deux arcs de clothoïdes de courbure $\pm\kappa$, où $\kappa \leq \kappa_{max}$ et de variation de courbure σ_{max} . Ainsi, l'arc à courbure continue garde toujours sa symétrie étant donné que les deux arcs de clothoïdes ont la même longueur.

Ces chemins à courbure continue peuvent être utilisés dans des planificateurs de chemins locaux évitant les collisions. Pour ce faire, tous les chemins possibles reliant deux config-

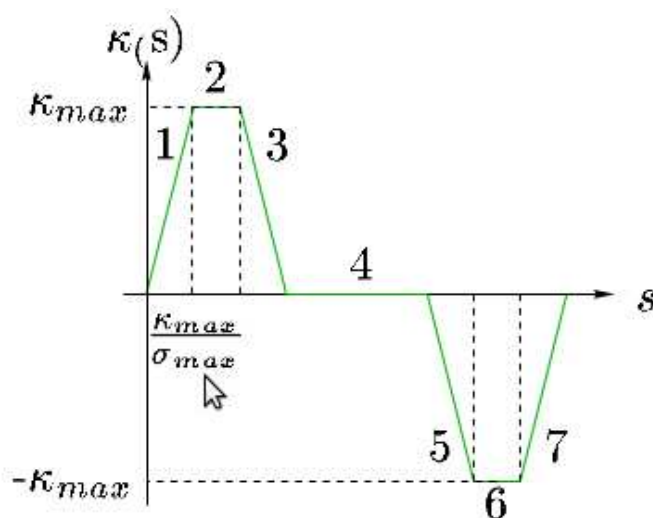


FIGURE 2.17 – Variation de la courbure dans le cas d'un chemin à courbure continue formé d'une rotation à gauche suivie d'un segment de droite et enfin d'une rotation à droite (figure tirée de [Scheuer and Fraichard, 1997]).

urations sont calculés. Ensuite, les chemins qui évitent les collisions sont gardés. Enfin, le chemin retenu est le plus court chemin parmi les chemins retenus.

Plusieurs travaux nécessitent la connaissance d'une expression analytique de la trajectoire à suivre pour pouvoir calculer le contrôle de chaque robot du convoi. Quand elle est inconnue, le robot vise à l'approcher en faisant passer par les points acquis une courbe dont l'expression analytique est connue (comme courbes de Bézier, les splines, ...).

2.4.3 Approximation de la liste des points mémorisés avec des courbes de Bézier

Les courbes de Bézier sont des courbes polynomiales paramétriques ayant plusieurs points de contrôle. La disposition de ces points détermine la forme de la courbe, cette dernière passe par le premier et le dernier point de contrôle.

Dans [MATKO et al., 2008], les auteurs proposent une stratégie de contrôle d'un convoi à deux robots où le suiveur acquiert, à l'aide de ses capteurs, les positions successives de son leader et les mémorise en fonction de leur temps d'acquisition. Ensuite, il approche le chemin de son leader à l'aide d'une courbe exprimée sous forme paramétrique polynomiale d'ordre 2, dont les coefficients sont calculées en utilisant la méthode des moindres carrés.

[Hwang et al., 2009] propose un algorithme permettant d'approcher un chemin dont l'expression analytique est inconnue par une courbe de Bézier. L'algorithme proposé extrait alors des points significatifs de ce chemin (ces points correspondent principalement à

un changement de courbure important) et fait passer par ces points une courbe de Bézier continue et lisse. Si un obstacle apparaît devant le robot le long de la courbe de Bézier à suivre, le robot la modifie en ligne pour éviter de rentrer en collision avec l'obstacle. Comme la figure 2.18 le montre, quand un obstacle apparaît sur la courbe de Bézier générée par morceaux, le robot mobile modifie le chemin à suivre pour éviter la collision avec l'obstacle. Ensuite, il re-converge vers et suit son chemin planifié par avance.

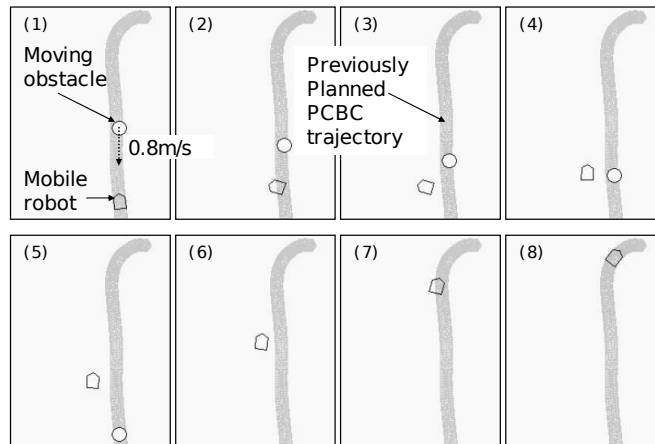


FIGURE 2.18 – Modification en ligne du chemin du robot pour éviter l'obstacle (figure tirée de [Hwang et al., 2009]).

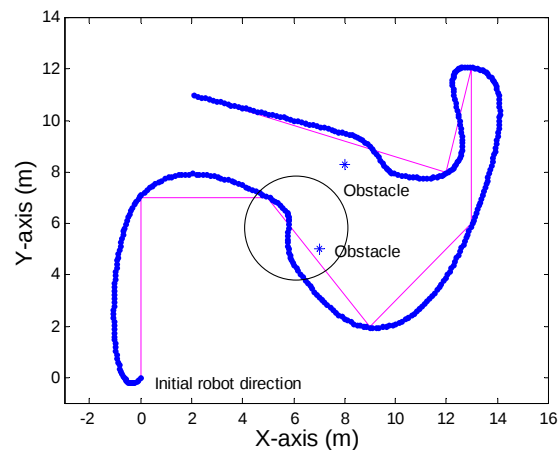


FIGURE 2.19 – Génération d'une courbe de Bézier par morceaux passant par les coins d'un chemin donné (figure tirée de [Hwang et al., 2009]).

La courbe générée pour relier les points significatifs du chemin est construite par morceaux à partir de courbes de Bézier cubiques (d'ordre 3). Chaque courbe de Bézier nécessite 4 points de contrôle, dont le premier et le dernier correspondent à deux points du chemin : le point de départ du robot et son point d'arrivée. De cette façon, l'algorithme proposé garantit le passage du robot par les points extraits du chemin à suivre.

Les deux autres points sont fixés de façon à avoir une connexion homogène entre les différents morceaux de Bézier : la connexion entre deux courbes de Bézier doit être tangente à l'orientation du robot.

Les expérimentations montrent que l'algorithme proposé génère une courbe de Bézier par morceaux passant par les points extraits du chemin original (dont l'équation analytique est inconnue). **En revanche, la courbe générée n'est pas fidèle au chemin original : un écart latéral important entre ces deux courbes est visible sur la figure 2.19.**

2.4.4 Approximation de la liste de points mémorisés avec des B-splines

Plusieurs travaux font passer par une succession de points donnés une courbe B-spline, qui est une généralisation d'une courbe de Bézier. [Komoriya and Tanie, 1989] utilise des B-splines d'ordre 3 qui représentent des courbes lisses dont la dérivée seconde est continue. Le chemin reliant les points est alors à courbure continue.

[Avanzini et al., 2009] considère un convoi en déplacement où le leader, conduit manuellement, communique à ses suiveurs sa nouvelle position à chaque pas de temps. Chaque suiveur essaie alors de reconstruire durant son déplacement la trajectoire du leader en faisant passer des B-splines par les points du leader stockés dans sa mémoire. La trajectoire globale du leader est alors obtenue en concaténant plusieurs courbes B-splines, dont chacune est une combinaison linéaire de polynômes de degré d admettant plusieurs points de contrôle qui déterminent sa forme. Ces points de contrôle représentent les coefficients des polynômes et sont choisis de façon à minimiser le critère d'optimisation suivant :

- la trajectoire globale doit passer par les points de leader et être le plus possible fidèle à sa trajectoire ;
- la trajectoire globale doit vérifier des contraintes de continuité aux points connectant les différentes B-splines concaténées.

Après chaque acquisition d'une nouvelle position du leader, le robot suiveur étend la trajectoire déjà construite de façon à y intégrer cette nouvelle position (voir Fig. 2.20). Pour étendre la trajectoire courante, le suiveur garde actifs n_{ac} polynômes et n_{ap} points de contrôle de cette trajectoire pour pouvoir les ajuster pendant la concaténation de la nouvelle B-spline avec la trajectoire courante. Les coordonnées des n_{ap} points de contrôle sont alors à modifier pour minimiser le critère d'optimisation.

Le choix du degré d des polynômes et du nombre de polynômes n_{ac} et de points de contrôle n_{ap} actifs est obtenu suite à une étude expérimentale où **beaucoup de simulations ont été réalisées tout en modifiant ces 3 paramètres pour trouver la meilleure combinaison**. À notre avis, cette combinaison n'est pas forcément optimale et son calcul est assez lourd et coûteux.

Les auteurs ont testé cette méthode dans un suivi en convoi de 4 véhicules où l'écart latéral entre les trajectoires des suiveurs et celle de leur leader était aux alentours de 15

cm. Malgré ses bonnes performances, nous constatons que la génération en-ligne de la trajectoire du leader selon cette méthode semble être très coûteuse en termes de calculs : à chaque pas de temps, certains polynômes et points de contrôle doivent être ajustés pour pouvoir concaténer une nouvelle B-spline intégrant le nouveau point acquis du leader.

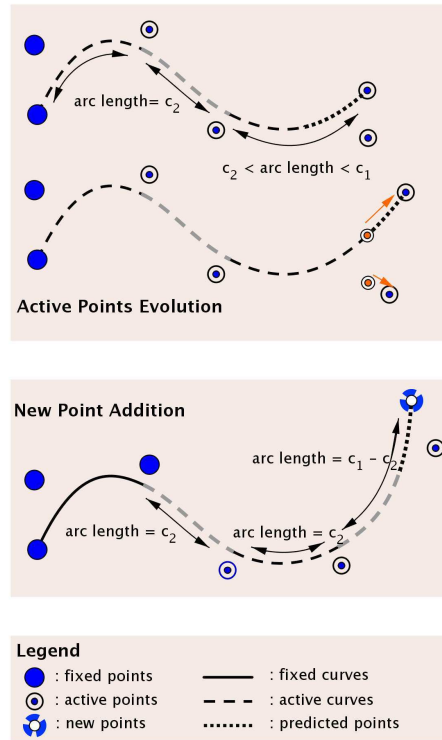


FIGURE 2.20 – Commande longitudinale hybride basée sur les erreurs longitudinales entre un suiveur et le leader du convoi d'une part et un suiveur et son prédécesseur d'autre part (figure tirée de [Avanzini et al., 2009]).

2.4.5 Synthèse des méthodes d'estimation de chemins et trajectoires

Nous avons présenté ci-dessus plusieurs méthodes d'estimation de chemins ou de trajectoires pour deux problèmes différents. Le premier problème était d'estimer une trajectoire permettant à un robot de passer d'une configuration à une autre. Dans le cas du suivi en formation ou en convoi, la configuration désirée correspond à des positions par lesquelles le leader est déjà passé. Estimer le chemin (ou la trajectoire) qui relie deux points à l'aide de courbes à expression analytique connue (comme des droites, des cercles, des clothoïdes ou des B-splines) peut induire un écart latéral non négligeable entre le chemin du leader et celui du suiveur, ce qui rend le suivi moins précis. D'un côté, les chemins de Dubins dont la courbure est discontinue ne présentent pas une solution efficace vu que le robot doit s'arrêter à chaque point de discontinuité de la courbure pour se réorienter. D'un autre côté, les clothoïdes sont des courbes à courbure continue, dont le calcul est difficile puisque

leur forme analytique est exprimée par des intégrales de Fresnel (qui sont des intégrales non analytiques dont l'évaluation est fastidieuse).

Le deuxième problème était de reconstruire le chemin du leader en faisant passer par ses positions successives une courbe à équation analytique connue. Les courbes de Bézier et les B-splines étaient le plus utilisées dans ce cas, malgré la complexité de leurs calculs : l'obtention de ces courbes ainsi que leurs coefficients nécessitent beaucoup de calculs.

2.5 Bilan et discussion

Dans ce chapitre, nous avons présenté plusieurs approches du maintien en formation d'un ensemble de véhicules autonomes. Ces approches, basées sur l'automatique linéaire ou non linéaire, conçoivent une loi de contrôle continue exprimée sous une forme analytique.

Pour faciliter la conception des lois de commande, certains travaux linéarisent le système et exploitent les méthodes de l'automatique linéaire. Ainsi, nous avons distingué l'utilisation des régulateurs *PID* dont les gains ne sont pas déterminés selon une méthode générique mais ajustés au cas par cas, de sorte que le suivi soit stable et précis. D'autres travaux considèrent le suivi en convoi comme étant un problème d'optimisation et cherchent la commande qui minimise une fonction d'erreur.

Ces méthodes présentent certains inconvénients et ne garantissent pas toujours un bon fonctionnement du système. En effet, la linéarisation du système est réalisée autour d'un point de fonctionnement et la commande synthétisée ne garantit le contrôle sûr du système que tant que celui-ci évolue dans le voisinage de ce point de fonctionnement. En d'autres termes, la sécurité du suivi est réduite à ce voisinage.

Pour remédier à ce problème, d'autres travaux se sont appuyés sur le modèle non linéaire du système pour concevoir des lois de commande applicables sur une plus large plage de fonctionnement. Nous distinguons plusieurs méthodes. Une première famille qui repose sur des techniques de linéarisation exacte comme la transformation chaînée offre de bonnes performances mais à l'image des approches fondées sur des PID, elles nécessitent un réglage de paramètres. D'autres méthodes s'appuient quant à elles sur le théorème de stabilité de Lyapunov pour la synthèse de la commande. La difficulté dans ces approches est de trouver une fonction de Lyapunov qui puisse générer une loi de contrôle stable (c'est-à-dire que si cette fonction ne donne pas de loi contrôle stable, il faut choisir une nouvelle fonction et recommencer la synthèse d'une nouvelle loi). De plus, la convergence du système n'est garantie que si certaines contraintes sur son état initial sont satisfaites.

Que le modèle du système soit linéarisé ou non linéaire, deux modèles de mouvement du robot peuvent être utilisés pour la synthèse de la loi de commande. Dans le cas où les robots se déplacent à faible vitesse, certains travaux se contentent du modèle cinématique de mouvement, et ceci pour simplifier la conception des lois de commande. Pour un contrôle plus précis et plus réaliste, d'autres travaux se basent sur une modélisation dynamique du mouvement du robot. Dans ce deuxième cas, la conception des lois de commande est une tâche ardue à cause de la non-linéarité des équations dynamiques.

Toutes les méthodes citées ci-dessus supposent que la trajectoire à suivre est connue par tout le convoi. De plus, à tout instant, chaque robot connaît la courbure du chemin (et sa dérivée) et peut calculer son écart latéral et son erreur angulaire par rapport à ce chemin. Cela n'étant pas toujours le cas, plusieurs travaux se concentrent sur l'estimation de la trajectoire consigne à partir d'une suite de points qui représente l'historique de déplacement du leader. Cette reconstruction est réalisée la plupart du temps avec des fonctions B-splines d'autant plus complexes que la précision désirée est élevée. Comme nous l'avons vu dans 2.4, le calcul d'une expression analytique de la trajectoire à suivre peut alors être coûteux quand on désire avoir une trajectoire très fidèle à celle générée par le leader.

Pour conclure sur les approches de contrôle analytiques, le système est régi par un modèle non linéaire d'autant plus complexe que la précision de contrôle souhaitée est importante. Cette complexité, provenant de tous les facteurs cités ci-dessus, rend la synthèse des lois de commande assez compliquée et difficile (cette synthèse est obtenue en manipulant les équations analytiques du système).

Nous désirons dans cette thèse concevoir une méthode qui facilite le calcul du contrôle, tout en conservant l'objectif d'aboutir à un suivi stable et précis. Nous proposons pour cela de découpler le modèle de mouvement des robots (qui peut devenir assez rapidement complexe) de la méthode de calcul du contrôle. Le contrôleur conçu pourrait alors être appliqué à n'importe quel type de robots. Pour ce faire, il suffirait de changer le modèle de mouvement considéré et d'appliquer la même méthode de contrôle pour calculer les nouvelles commandes du robot. Nous considérons ainsi que ce découplage représente un avantage par rapport aux méthodes analytiques et mériterait d'être exploité dans la conception des lois de contrôle.

Chapitre 3

Algorithme général de *platooning*

Comme nous l'avons déjà vu dans la section 2.2, les approches reposant sur des architectures globales sont généralement plus précises que celles reposant sur des architectures locales mais sont aussi plus coûteuses et moins robustes aux perturbations. Une perte de communication ou un retard dans la transmission et le traitement des données peut entraîner l'arrêt du convoi. Dans le cas où un problème se produit durant le platooning, les architectures locales pourraient prendre le relais et permettre de poursuivre le suivi dans des conditions de sécurité satisfaisantes.

Les architectures locales proposées dans la littérature (section 2.2.2) sont quant à elles généralement moins précises (puisque l'erreur latérale tend à se cumuler le long du convoi) et ne garantissent pas nécessairement la stabilité. Ainsi, nous désirons concevoir dans cette thèse des algorithmes de platooning selon une architecture décentralisée locale qui améliorent ces deux points. Nous nous intéressons en particulier à l'aspect contrôle latéral du platooning, c'est-à-dire au contrôle permettant à un véhicule de suivre avec précision un chemin de référence en minimisant à la fois l'écart latéral et les oscillations autour de ce chemin.

Ce chapitre se concentre sur la définition de nos hypothèses de travail et est organisé comme suit :

la section 3.1 décrit l'algorithme général de platooning, c'est-à-dire la méthode que nous suivons dans le but de lier entre eux les contrôles longitudinal et latéral.

La section 3.2 présente la modélisation du mouvement des robots. Deux modèles sont considérés dans ce travail : le premier, appelé *modèle à variation instantanée*, permet la résolution exacte des équations de mouvement du robot mais considère que le robot peut atteindre instantanément n'importe quelle commande en vitesse angulaire (ce qui correspond à considérer une accélération angulaire infinie) ; le deuxième modèle, appelé *modèle à variation progressive* offre une résolution approchée des équations de mouvement du robot mais permet de considérer une accélération angulaire finie.

La section 3.3 présente le simulateur réalisé durant cette thèse pour tester et valider nos algorithmes de *platooning*. Pour réaliser des expérimentations, le simulateur utilise le modèle de mouvement à variation progressive puisqu'il est plus fidèle au déplacement réel

des robots que le modèle à variation instantanée.

La section 3.4 présente les contraintes techniques auxquelles doivent répondre les robots afin de pouvoir implémenter *Memo-Lat* et *NOC*.

Enfin, nous concluons ce chapitre dans la section 3.5 en résumant les objectifs de ce travail et ce que nous allons présenter dans les chapitres suivants.

3.1 Algorithme général de platooning

Nous considérons ici un convoi de robots se déplaçant à faible vitesse. Comme suggéré dans [Daviet and Parent, 1996a], nous pouvons alors **découpler les deux contrôles longitudinal et latéral**.

L'algorithme général de platooning que nous considérons dans ce travail repose sur ce découplage et est divisé en quatre étapes décrites ci-après. À chaque pas de temps τ , chaque robot suiveur du convoi :

- acquiert et mémorise la position courante du robot qui le précède (section 4.1) ;
- calcule l'accélération longitudinale avec le contrôleur *SSC* en utilisant la position courante du prédécesseur (section 2.3.2.1) ;
- choisit la meilleure position cible à viser (parmi les positions mémorisées) et calcule la vitesse angulaire en utilisant un des deux contrôleurs latéraux détaillés dans les chapitres 4 et 5 ;
- applique les deux commandes calculées.

L'équipe *MAIA* a développé un contrôleur longitudinal fondé sur une approche décentralisée locale permettant de garantir l'absence de collisions entre les robots tout en minimisant leur inter-distance. Ce contrôleur longitudinal, que nous appelons *SSC* (Scheuer - Simonin - Charpillat), sera utilisé dans ce travail pour contrôler la vitesse longitudinale des robots formant le convoi. Le calcul de l'accélération longitudinale par le contrôleur *SSC* est détaillé dans la section 2.3.2.1.

Nous nous intéressons donc dans cette thèse au comportement latéral du convoi et souhaitons synthétiser un contrôleur latéral qui permet aux robots de suivre le chemin tracé par le leader avec précision en garantissant la stabilité du convoi. Plus précisément, nous souhaitons concevoir des algorithmes de contrôle permettant à un robot du convoi de calculer, tout au long du suivi, une suite de vitesses angulaires ω qui le mène à **converger le plus rapidement possible** vers le chemin de son prédécesseur (qui est dans ce cas une succession de points) et à le suivre **en minimisant l'écart latéral e_y et l'erreur angulaire e_θ** (définitions 5 et 6). Nous étudions également la stabilité des contrôleurs latéraux que nous développons.

Nous présentons dans les chapitres 4 et 5 deux contrôleurs latéraux *Memo-LAT* et *NOC*,

qui s'appuient sur la mémorisation du chemin du robot prédécesseur sous forme d'une liste de positions et que nous décrivons succinctement ci-dessous.

Algorithme de contrôle latéral *Memo-LAT* Le premier contrôleur *Memo-LAT* (**M**emorization and **L**ook-**A**head **T**arget) choisit une position cible de la liste des positions mémorisées du robot prédécesseur et calcule une vitesse angulaire selon le principe de Daviet et Parent (*DP*) présenté dans [Daviet and Parent, 1996a]. En comparant les performances des deux algorithmes de contrôle (*Memo-LAT* et *DP*), nous observons l'amélioration apportée par la mémorisation à la précision du suivi. En revanche, la stabilité du suivi n'est pas garantie et des mouvements oscillatoires apparaissent au niveau des virages.

Algorithme de contrôle latéral *NOC* Nous proposons dès-lors un second contrôleur dont le but est d'éliminer ces oscillations et d'obtenir un suivi stable. *NOC* (**N**on **O**scillatory **C**onvergence), approxime le chemin du prédécesseur autour de la cible qu'il cherche à atteindre dans le but d'estimer localement sa courbure. Il utilise ensuite cette estimation dans le calcul de la consigne de vitesse angulaire. *NOC* combine une approche géométrique, qui interdit (ou évite autant que possible) par sa conception, l'apparition d'oscillations avec une approche heuristique qui cherche à réduire une fonction d'erreur fondée sur l'écart latéral et l'erreur angulaire. *NOC* raisonne sur plusieurs pas de temps dans le but de garantir l'absence d'oscillations : il prévoit le déplacement du robot dans le futur afin de choisir une commande qui évite les oscillations. En gardant ce même principe, *NOC* a ensuite été généralisé en *Abstract NOC* dans le but d'être appliqué sur différents types de robots. Ces deux principaux contrôleurs (*Memo-LAT* et *NOC*) seront présentés et expliqués en détails dans la suite de cette thèse.

3.2 Modélisation du système

Nous avons choisi de tester nos algorithmes sur des robots à deux roues différentielles non holonomes qui se déplacent à faible vitesse. Nous pouvons ainsi négliger les glissements. Sachant qu'un robot est dit holonome quand il peut se déplacer librement (dans n'importe quelle direction et quelle que soit son orientation) sans aucune contrainte sur son mouvement, un robot non holonome est soumis à des contraintes cinématiques et/ou dynamiques et ne peut pas suivre n'importe quel chemin.

Nous considérons ici un robot R qui évolue dans son référentiel fixe. Il est représenté par sa position (x, y) , son orientation θ (Fig. 3.1), sa vitesse longitudinale v et sa vitesse angulaire ω . Ses deux roues droite et gauche sont distantes d'une distance l . Étant donné les contraintes mécaniques du robot, les vitesses et les accélérations longitudinales des deux roues droite (right) $\{v_r, a_r\}$ et gauche (left) $\{v_l, a_l\}$ sont bornées. Par conséquent, les vitesses longitudinales v et angulaires ω ainsi que les accélérations longitudinales a et angulaires η du centre de masse du robot sont aussi bornées :

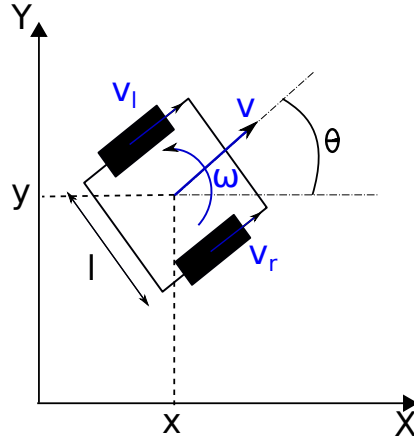


FIGURE 3.1 – Représentation d'un robot dans un repère global.

$$\begin{cases} v_{\min} \leq v \leq v_{\max}, & \omega_{\min} \leq \omega \leq \omega_{\max} \\ a_{\min} \leq a \leq a_{\max}, & \eta_{\min} \leq \eta \leq \eta_{\max} \end{cases} \quad (3.1)$$

Nous considérons dans cette thèse deux modèles de mouvement des robots :

- *un modèle à variation instantanée* (section 3.2.1) où la vitesse angulaire du robot varie instantanément d'une valeur à une autre à chaque pas de temps τ . L'évolution des robots selon ce modèle est représentée d'une façon continue par des équations analytiques qui ne considèrent pas l'accélération angulaire.
- *un modèle à variation progressive* (section 3.2.2) qui est plus fidèle au déplacement réel des robots. En effet, ce modèle, qui prend en compte l'accélération angulaire du robot, cherche à simuler son déplacement réel en considérant que sa vitesse angulaire ne peut pas changer instantanément d'une valeur à une valeur.

Dans cette thèse, nous avons réalisé un simulateur pour tester nos algorithmes de *platooning*. Celui-ci considère que les robots se déplacent selon le modèle de mouvement à variation progressive, puisqu'il est plus fidèle au déplacement réel des robots que le modèle à variation instantanée.

Par ailleurs, *NOC*, de part sa conception, est fondé sur le modèle de déplacement à variation instantanée. En effet, comme nous l'avons déjà mentionné, *NOC* anticipe la trajectoire de chaque robot du convoi dans le but de choisir une vitesse angulaire qui évite l'oscillation autour du chemin suivi. *NOC* calcule ainsi le déplacement du robot en faisant l'hypothèse que celui-ci évolue à vitesse angulaire constante entre deux prises de décision et vérifie s'il y a intersection avec le chemin de son prédécesseur. En considérant une vitesse angulaire constante, le modèle de mouvement des robots à variation progressive est équivalent au modèle à variation instantanée. Nous pouvons alors utiliser ce modèle plus simple pour prévoir le déplacement de chaque robot du convoi dans le futur et calculer

sa commande latérale. L'avantage du modèle à variation instantanée est qu'il permet d'utiliser les équations analytiques du mouvement du robot pour vérifier mathématiquement si le robot coupe le chemin de son prédécesseur.

3.2.1 Modèle à variation instantanée

Le modèle de mouvement à variation instantanée ne prend pas en compte l'accélération angulaire du robot, mais seulement son accélération longitudinale a . Il considère que la vitesse angulaire du robot peut passer instantanément d'une valeur à une autre au début de chaque pas de temps. Le modèle à variation instantanée est utilisé dans la conception et le test de l'algorithme *NOC* qui suppose que les robots se déplacent à vitesse constante. Ce modèle considère que le passage d'une commande à une autre est instantané ; c'est-à-dire que l'évolution de la vitesse angulaire du robot n'est pas continue.

Nous expliquerons ce point plus en détail dans le chapitre 5. Ce modèle simplifié est représenté par les équations de mouvement suivants :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{v} = a \\ \dot{\theta} = \omega \end{cases} \quad (3.2)$$

Dans un modèle à variation instantanée, a et ω représentent les commandes appliquées sur le robot à chaque pas de temps τ . En supposant que ces commandes restent constantes durant τ , l'évolution du centre de masse du robot, dont l'état initial est représenté par la position (x_0, y_0) , l'orientation θ_0 et la vitesse v_0 , à tout instant $t \in [0, \tau]$ est décrite par les équations du mouvement suivantes (Annexe A) si ω est non nulle :

$$\begin{cases} \theta(t) = \omega.t + \theta_0 \\ x(t) = x_0 + \frac{at + v_0}{\omega} \cdot \sin(\omega t + \theta_0) + \frac{a}{\omega^2} \cdot \cos(\omega t + \theta_0) - \frac{a}{\omega^2} \cdot \cos \theta_0 - \frac{v_0}{\omega} \cdot \sin \theta_0 \\ y(t) = y_0 - \frac{at + v_0}{\omega} \cdot \cos(\omega t + \theta_0) + \frac{a}{\omega^2} \cdot \sin(\omega t + \theta_0) - \frac{a}{\omega^2} \cdot \sin \theta_0 + \frac{v_0}{\omega} \cdot \cos \theta_0 \end{cases} \quad (3.3)$$

ou bien par les équations suivantes si ω est nulle :

$$\begin{cases} \theta(t) = \theta_0 \\ x(t) = x_0 + \left(v_0.t + \frac{1}{2}.a.t^2 \right) \cdot \cos \theta_0 \\ y(t) = y_0 + \left(v_0.t + \frac{1}{2}.a.t^2 \right) \cdot \sin \theta_0 \end{cases} \quad (3.4)$$

Cette modélisation simplifiée ne correspond pas au mouvement réel du robot. En effet, le robot ne peut pas changer instantanément de vitesse angulaire. Dans le but d'obtenir une modélisation plus fidèle au déplacement réel d'un robot, nous proposons le modèle à variation progressive détaillé dans 3.2.2 qui prend en compte son accélération angulaire.

3.2.2 Modèle à variation progressive

Comme déjà mentionné dans l'introduction de la section 3.2, le modèle à variation progressive cherche à reproduire un déplacement plus proche du déplacement réel des robots. En effet, ce modèle considère que le robot ne peut pas changer instantanément de vitesse angulaire, celle-ci évoluant linéairement au cours du pas de temps. En incluant l'accélération angulaire η dans les équations de mouvement, nous obtenons :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{v} = a \\ \dot{\theta} = \omega \\ \dot{\omega} = \eta \end{cases} \quad (3.5)$$

Les accélérations a et η étant supposées constantes sur un pas de temps τ , le centre de masse du robot évolue selon les équations suivantes (voir Annexe B) :

$$\begin{cases} v(t) = a.t + v_0 \\ \omega(t) = \eta.t + \omega_0 \\ \theta(t) = \frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0 \\ x(t) = x_0 + \frac{a}{\eta}.\sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) - \int \left(\frac{a.\omega_0}{\eta} - v_0\right).\cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt - \frac{a}{\eta}.\sin \theta_0 \\ y(t) = y_0 - \frac{a}{\eta}.\cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) - \int \left(\frac{a.\omega_0}{\eta} - v_0\right).\sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt + \frac{a}{\eta}.\cos \theta_0 \end{cases} \quad (3.6)$$

Les équations de mouvement 3.6 contiennent des intégrales de Fresnel qui sont des intégrales non analytiques. Nous ne pouvons donc pas obtenir des expressions analytiques exactes de x et y comme dans le modèle à variation instantanée. Pour cela, nous proposons d'approcher le mouvement du robot représenté par les équations 3.5 comme expliqué ci-dessous.

3.2.2.1 Résolution approchée du déplacement des robots

Paramètres de contrôle du modèle — Dans ce modèle, les robots sont commandés en vitesse par deux paramètres. Ces commandes sont la vitesse longitudinale v_d et la vitesse angulaire ω_d et sont déterminées comme suit.

- Sachant que nous utilisons dans ce travail un contrôleur longitudinal qui calcule une accélération longitudinale a , nous traduisons cette commande en accélération en une commande en vitesse. La vitesse longitudinale désirée sera égale à $v_d = v_0 + a\tau$.

- La vitesse angulaire ω_d est quant à elle donnée par l'un de nos algorithmes de contrôle latéral *Memo-LAT*, *NOC* ou *Abstract NOC*.

Calcul des commandes bas niveau — Comme nous l'avons déjà mentionné, nous considérons tout au long de cette thèse des robots à deux roues différentielles. Il nous faut donc passer des commandes de *haut niveau* (vitesses longitudinale et angulaire) à des commandes de *bas niveau* (vitesses longitudinales de chaque roue).

En utilisant les relations qui relient les vitesses des roues avec la vitesse linéaire et la vitesse angulaire du robot ($v = (v_r + v_l)/2$ et $\omega = (v_r - v_l)/l$), nous représentons son modèle dynamique sous la forme suivante (Equations 3.7) :

$$\begin{cases} \dot{x} = \frac{1}{2}(v_r + v_l) \cos \theta \\ \dot{y} = \frac{1}{2}(v_r + v_l) \sin \theta \\ \dot{\theta} = \frac{1}{l}(v_r - v_l) \\ \dot{v}_r = a_r \\ \dot{v}_l = a_l \end{cases} \quad (3.7)$$

Connaissant v_d et ω_d , nous déduisons les vitesses désirées des deux roues selon les équations suivantes :

$$\begin{cases} v_{r,d} = v_d + \frac{l}{2}\omega_d \\ v_{l,d} = v_d - \frac{l}{2}\omega_d \end{cases} \quad (3.8)$$

Modélisation de l'asservissement — Théoriquement, les moteurs à courant continu des deux roues sont commandés en vitesse selon un profil trapézoïdale (Fig. 3.2). En pratique, pour atteindre la consigne $v_{r/l,d}$, la plupart des travaux en robotique visent à asservir la vitesse des moteurs à l'aide d'un régulateur *PID* (paragraphe 2.3.2.1), dont les coefficients sont réglés de façon à atteindre la consigne le plus vite possible en réduisant le dépassement de la consigne, les oscillations et l'erreur statique.

Pour modéliser l'asservissement, nous divisons le pas de temps τ (qui correspond au temps écoulé entre deux modifications de la consigne) en N pas d'intégration, chacun noté par dt . La précision de la simulation du mouvement dépend directement de la durée d'intégration dt . Plus dt est faible, plus la simulation sera précise⁶.

Pour chaque dt , nous calculons les accélérations longitudinales à appliquer sur chacune des deux roues dans le but d'atteindre leurs vitesses désirées le plus rapidement possible. Pour cela, à chaque $n.dt$ (où $n \in \{1, \dots, (N - 1)\}$) :

6. Dans le reste de cette thèse, nous considérons que $N = 100$, ce qui correspond, avec $\tau = 0.1$ seconde, à des pas de temps dt de 1 milliseconde.

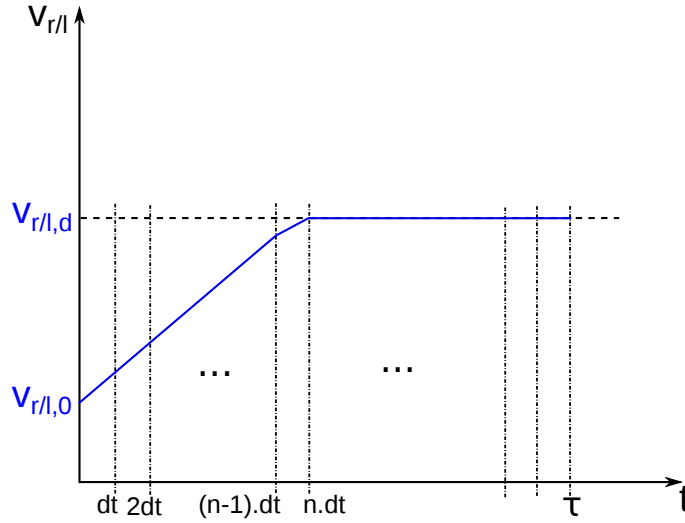


FIGURE 3.2 – Profil trapézoïdal de variation de la vitesse de chacune des roues droite (v_r) et gauche (v_l) : $v_{r/l,0}$ valeur initiale et $v_{r/l,d}$ valeur désirée.

- nous estimons la configuration courante du robot (x_{n-1} , y_{n-1} , θ_{n-1} , v_{n-1} , ω_{n-1}) et calculons les vitesses longitudinales des deux roues selon les équations suivantes :

$$\begin{cases} v_{r,n-1} = v_{n-1} + \frac{l}{2}\omega_{n-1} \\ v_{l,n-1} = v_{n-1} - \frac{l}{2}\omega_{n-1} \end{cases} \quad (3.9)$$

- nous calculons l'accélération linéaire $a_{r/l,n}$ à appliquer sur la roue (droite ou gauche) comme suit :

- si ($v_{r/l,d} > v_{r/l,n-1}$) et ($v_{r/l,n-1} + a_{r/l,\max}.dt \leq v_{r/l,d}$) $\Rightarrow a_{r/l,n} = a_{r/l,\max}$.
- si ($v_{r/l,d} < v_{r/l,n-1}$) et ($v_{r/l,n-1} - a_{r/l,\max}.dt \geq v_{r/l,d}$) $\Rightarrow a_{r/l,n} = -a_{r/l,\max}$.
- sinon : $a_{r/l,n} = (v_{r/l,d} - v_{r/l,n-1})/dt$.

- une fois que les accélérations $\{a_{r,n}, a_{l,n}\}$ à appliquer sur chacune des deux roues sont calculées, nous déterminons la nouvelle configuration du robot (position, vitesse et orientation) à la fin de $n.dt$ en utilisant une méthode d'intégration numérique de Runge-Kutta de second ordre désignée en anglais par le terme *Midpoint method* [Süli and Mayers, 2003]. Étant donné que $a_{r,n}$ et $a_{l,n}$ sont constantes sur dt , nous calculons la nouvelle configuration du robot en utilisant les vitesses moyennes sur dt $v_{r,m}$ et $v_{l,m}$ des deux roues et la configuration du robot (x_{n-1} , y_{n-1} , θ_{n-1} , $v_{r,n-1}$, $v_{l,n-1}$) correspondant au début de $n.dt$. Les vitesses moyennes $v_{r,m}$ et $v_{l,m}$ sont calculées selon les équations suivantes :

$$\begin{cases} v_{r,n} = v_{r,n-1} + a_r \cdot dt \\ v_{l,n} = v_{l,n-1} + a_l \cdot dt \\ v_{r,m} = (v_{r,n-1} + v_{r,n})/2 \\ v_{l,m} = (v_{l,n-1} + v_{l,n})/2 \end{cases} \quad (3.10)$$

La nouvelle configuration du robot à la fin de $n \cdot dt$ est calculée selon les équations suivantes :

$$\begin{cases} v_n = (v_{r,n} + v_{l,n})/2 \\ v_{n,m} = (v_{r,m} + v_{l,m})/2 \\ \omega_n = (v_{r,n} - v_{l,n})/l \\ \theta_n = \theta_{n-1} + \omega_n \cdot dt \\ \theta_{n,m} = (\theta_{n-1} + \theta_n)/2 \\ x_n = x_{n-1} + v_{n,m} \cdot \cos \theta_{n,m} \cdot dt \\ y_n = y_{n-1} + v_{n,m} \cdot \sin \theta_{n,m} \cdot dt \end{cases} \quad (3.11)$$

Nous répétons ces étapes jusqu'à la fin du pas de temps τ où nous obtenons la nouvelle configuration du robot qui correspond à l'application des commandes données durant τ . Nous obtenons également l'ensemble des points $\mathcal{P} : \{p_1, \dots, p_n, \dots, p_N\}$ (avec $p_n = (x_n, y_n)$) par lesquels le robot est passé durant cette période.

À la fin du pas de temps τ et si celui-ci est suffisamment long⁷, les vitesses v et ω du robot sont passées de leurs vitesses de départ v_0 et ω_0 aux valeurs désirées v_d et ω_d .

3.2.2.2 Saturation des vitesses des roues du robot

Dans le cas où la vitesse désirée d'au moins une des roues du robot est supérieure à sa vitesse maximale, la vitesse de la roue sature et le robot ne pourra pas exécuter le mouvement prévu par ses contrôleurs.

Dans ce cas, nous proposons de résoudre ce problème de saturation des vitesses des roues en réduisant à la fois la vitesse longitudinale désirée v_d à v_d^* et la vitesse angulaire désirée ω_d à ω_d^* de façon à ce que le robot puisse toujours se déplacer selon la courbure désirée $\kappa_d = \frac{\omega_d}{v_d}$ (ou le rayon de courbure $r_d = \frac{v_d}{\omega_d}$), mais avec une vitesse plus faible. Cette solution est intégrée dans le modèle de mouvement des robots implémenté dans notre simulateur.

Pour calculer v_d^* et ω_d^* à partir de v_d et ω_d , nous procédons comme suit :

7. Si le pas de temps n'est pas suffisamment grand pour atteindre les vitesses désirées des roues, le robot aura une vitesse longitudinale v et une vitesse angulaire ω inférieures aux valeurs désirées. Nous tenons en compte de ce cas puisque nous considérons toujours la configuration courante du robot dans le calcul du contrôle, et non sa configuration désirée.

1. calculer les vitesses désirées des deux roues selon les équations suivantes :

$$\begin{cases} v_{r,d} = v_d + \frac{l}{2}\omega_d \\ v_{l,d} = v_d - \frac{l}{2}\omega_d \end{cases} \quad (3.12)$$

2. déterminer laquelle des deux vitesses $v_{r,d}$, $v_{l,d}$ est la plus grande et la poser égale à la vitesse maximale admissible de la roue $v_{r,\max}$ ou $v_{l,\max}$.
3. Si $v_{r,d}$ est la vitesse la plus élevée entre $v_{r,d}$ et $v_{l,d}$, nous calculons ω_d^* et v_d^* suivant la démarche suivante :

$$\begin{cases} v_{r,d} = v_d + \frac{l}{2}\omega_d \\ v_{r,\max} = r_d \cdot \omega_d^* + \frac{l}{2}\omega_d^* \\ \omega_d^* = \frac{v_{r,\max}}{r_d + \frac{l}{2}} \\ v_d^* = \frac{r_d \cdot v_{r,\max}}{r_d + \frac{l}{2}} \end{cases} \quad (3.13)$$

4. Si $v_{l,d}$ est la vitesse la plus élevée entre $v_{r,d}$, $v_{l,d}$, nous calculons ω_d^* et v_d^* suivant la démarche suivante :

$$\begin{cases} v_{l,d} = v_d - \frac{l}{2}\omega_d \\ v_{l,\max} = r_d \cdot \omega_d^* - \frac{l}{2}\omega_d^* \\ \omega_d^* = \frac{v_{l,\max}}{r_d - \frac{l}{2}} \\ v_d^* = \frac{r_d \cdot v_{l,\max}}{r_d - \frac{l}{2}} \end{cases} \quad (3.14)$$

3.3 Conception du simulateur

Comme nous l'avons déjà mentionné, nous avons réalisé dans cette thèse un simulateur dans le but de tester nos algorithmes de *platooning*. Pour calculer le déplacement des robots, nous implémentons dans ce simulateur le modèle de mouvement à variation progressive. Comme nous avons déjà expliqué dans la section 3.2.2, ce modèle est plus fidèle au déplacement réel des robots que le modèle à variation instantanée (section 3.2.1) puisqu'il considère des accélérations longitudinale et angulaire finies.

Les commandes longitudinale et latérale du robot sont calculées avec une fréquence égale à 10 Hz. Le déplacement des robots est calculé avec une fréquence égale à 1000 Hz afin de garder les imprécisions dues à l'intégration temporelle du mouvement les plus faibles possibles.

Le simulateur considère que les robots se déplacent dans un environnement parfait. Nous considérons que les informations relatives aux différents robots (positions, orientations et vitesses) sont traitées avant d'être utilisées dans le calcul des commandes. Les robots ne sont donc pas soumis aux bruits de perception et de localisation.

3.4 Considérations techniques pour l'implémentation des contrôleurs

L'implémentation de notre algorithme de platooning sur une plate-forme expérimentale nécessite que chaque robot soit équipé de capteurs permettant d'estimer son propre état (représenté par ses position, orientation et vitesse, à l'aide par exemple d'un odomètre) et celui du robot qui le précède (en utilisant une caméra ou un télémètre). Tout au long de ce travail, nous considérons que les données acquises par les capteurs sont déjà traitées et que chaque robot du convoi connaît sa position, son orientation et sa vitesse longitudinale et angulaire dans le référentiel global relatif à sa configuration de départ. Chaque robot connaît également l'inter-distance le séparant de son prédécesseur et l'orientation relative de ce dernier. Nous pouvons ainsi estimer la position du robot prédécesseur et sa vitesse longitudinale. Sauvegarder les positions estimées du prédécesseur dans la mémoire de chaque robot permet alors de mémoriser le chemin qu'il a emprunté.

3.5 Conclusion

Dans ce chapitre, nous avons défini le cadre du travail où l'approche de contrôle adoptée est décentralisée locale. Ensuite, nous avons présenté l'algorithme général de platooning pour un couple de robots successifs et nous avons introduit les deux contrôleurs latéraux *Memo-LAT* et *NOC*. Étant donné que ces deux contrôleurs s'appuient sur la mémorisation du chemin, nous allons expliquer dans la section 4.1 comment un robot acquiert et mémorise les positions successives de son prédécesseur. Pour contrôler la vitesse longitudinale des robots, nous utilisons le contrôleur *SSC* expliqué dans la section 2.3.2.1. Ce contrôleur permet d'avoir un suivi proche en garantissant l'absence de collisions. Pour valider les algorithmes développés dans cette thèse, nous avons modélisé le mouvement d'un robot à deux roues différentielles. Deux modèles ont été ainsi présentés dans la section 3.2 : un modèle à variation instantanée qui considère que la vitesse angulaire peut passer instantanément d'une valeur à une autre et un modèle à variation progressive qui prend en compte le temps nécessaire au robot pour passer de sa vitesse angulaire courante à celle désirée.

Dans les chapitres suivants, nous détaillons les deux algorithmes de contrôle latéral *Memo-LAT* et *NOC*. Le chapitre 4 explique comment calculer la commande latérale et montre des résultats expérimentaux obtenus par implémentation sur des robots khépéras

et par simulation en utilisant le modèle de déplacement de robots à variation progressive. De plus, ce chapitre identifie les limites de l'algorithme *Memo-LAT* qui nous ont poussé à concevoir l'algorithme *NOC*.

Le chapitre 5 introduit l'algorithme *NOC*. Nous proposons dans un premier temps une comparaison des performances des deux algorithmes *Memo-LAT* et *NOC*. Nous réalisons ensuite des expérimentations où nous simulons des robots à deux roues différentielles en utilisant le modèle de mouvement à variation progressive présenté dans la section 3.2.2. L'observation des résultats du platooning selon le modèle à variation progressive nous permet d'abstraire l'algorithme *NOC* par rapport au modèle de mouvement des robots. Nous obtenons ainsi *NOC évolué* qui utilise, dans le calcul de la commande latérale, une fonction externe qui simule le déplacement du robot.

Chapitre 4

Memo-LAT : un algorithme de platooning basé sur la mémorisation du chemin

Comme expliqué précédemment dans l'état de l'art (chapitre 2), les approches locales ont l'inconvénient de ne pas toujours permettre un suivi stable et précis. En effet, les robots suiveurs ne connaissent pas l'équation analytique du chemin tracé par leur leader puisqu'ils ne communiquent pas avec lui. Chaque robot perçoit alors son prédécesseur et calcule son contrôle en se basant sur les données acquises localement. Ainsi, une erreur latérale due aux perceptions locales se propage le long du convoi en s'accumulant de robot en robot.

Dans ce chapitre, nous souhaitons réduire cette erreur latérale dans le but d'améliorer la précision du suivi. Pour cela, nous proposons l'algorithme de platooning *Memo-LAT* (**M**emorization and **L**ook-**A**head **T**arget). Celui-ci consiste à mémoriser le chemin du robot prédécesseur sous forme d'une liste de points. Ensuite, au lieu de viser l'arrière du robot prédécesseur, il choisit comme position cible un point du chemin mémorisé et calcule la commande latérale qui permet de l'atteindre. Cette commande, qui représente l'orientation désirée du robot suiveur, est proportionnelle à l'erreur angulaire qui existe entre l'orientation du robot et la direction avec son point visé. Cette loi de contrôle a été proposée par Daviet et Parent dans [Daviet and Parent, 1996a]. Le robot suiveur cherche à garder son orientation dirigée vers l'arrière de son prédécesseur. Si ce contrôleur fonctionne bien en ligne droite, son principal défaut est que, lorsque le prédécesseur effectue des mouvements à fortes variations de courbure, le suiveur coupe les virages. Ce phénomène est d'autant plus marqué que l'écart longitudinal entre les robots est important. L'intérêt du travail présenté dans la suite de ce chapitre est de montrer que la mémorisation du chemin améliore la précision du suivi puisque le robot suiveur cherche à suivre le chemin emprunté par son prédécesseur au lieu d'essayer de rejoindre ce dernier.

L'article [Yazbeck et al., 2011], publié dans la conférence internationale IROS (International Conference on Intelligent Robots and Systems), résume le travail réalisé autour de l'algorithme *Memo-LAT*.

Ce chapitre est organisé comme suit : tout d'abord, nous expliquons la méthode de mémorisation du chemin du robot prédécesseur ainsi que les différentes étapes de l'algorithme *Memo-LAT* qui permettent de calculer la commande latérale. Ensuite, pour valider l'algorithme, nous menons plusieurs expériences à la fois en simulation et sur des robots khépéras. Finalement, nous concluons ce travail en résumant les améliorations apportées par *Memo-LAT* et les points faibles qui limitent son utilisation.

4.1 Acquisition et mémorisation de positions

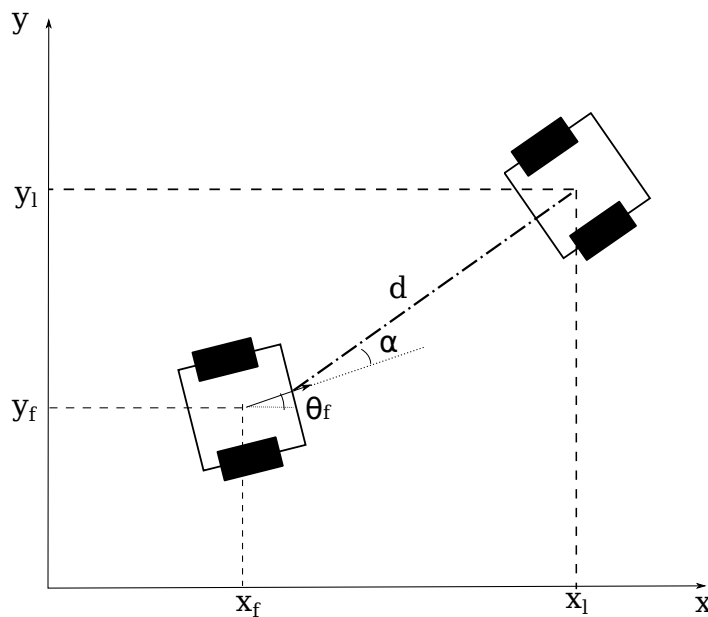


FIGURE 4.1 – Acquisition et mémorisation d'une position du robot prédécesseur.

L'algorithme de contrôle latéral *Memo-LAT* s'appuie sur la mémorisation du chemin du robot prédécesseur. L'approche adoptée dans cette thèse étant décentralisée locale, nous menons le travail sur un couple de robots successifs que nous appelons leader - suiveur.

Pour sauvegarder le chemin de son prédécesseur, le robot suiveur a deux choix de repères possibles : soit il considère le référentiel mobile correspondant à sa position courante et re-projette dans ce référentiel toutes les positions du prédécesseur sauvegardées dans sa mémoire ; soit il considère un référentiel fixe dont l'origine correspond à sa position initiale et projette dans ce référentiel la nouvelle position du prédécesseur acquise dans son référentiel mobile. Pour simplifier les calculs, nous choisissons d'attribuer au robot suiveur un référentiel fixe (\mathcal{R}) où nous estimons les positions et les vitesses des deux robots leader et suiveur.

Les capteurs embarqués sur le robot suiveur permettent d'estimer sa configuration courante (x_f, y_f, θ_f) , sa vitesse et les coordonnées polaires d et α de son prédécesseur

dans son repère mobile (figure 4.1). Pour estimer la position du robot prédécesseur dans (\mathcal{R}), nous faisons un changement de repère. En désignant par L la distance entre le capteur et le centre de masse du robot où les commandes sont appliquées, les coordonnées du robot prédécesseur (x_l, y_l) sont :

$$\begin{aligned} x_l &= x_f + L \cdot \cos \theta_f + d \cdot \cos(\theta_f + \alpha) \\ y_l &= y_f + L \cdot \sin \theta_f + d \cdot \sin(\theta_f + \alpha) \end{aligned} \quad (4.1)$$

En résumé, pour mémoriser une position du robot prédécesseur, le suiveur :

- estime sa nouvelle configuration (x_f, y_f, θ_f) dans son référentiel fixe ;
- estime les coordonnées polaires d et α de son prédécesseur dans son référentiel mobile ;
- estime la position du robot prédécesseur (x_l, y_l) dans son référentiel fixe.

Pour estimer la vitesse de son prédécesseur, le robot suiveur calcule la variation Δd de l'inter-distance séparant les deux robots sur un pas de temps ; la vitesse du prédécesseur v_l est alors calculée en fonction de la vitesse du robot suiveur v comme suit :

$$v_l = v + \Delta d / \tau$$

4.2 Algorithme

Comme expliqué dans la section 4.1, pour sauvegarder le chemin à suivre, le suiveur acquiert les positions successives de son prédécesseur en utilisant ses capteurs embarqués et les mémorise sous forme d'une suite de points dans un référentiel fixe dont l'origine correspond à sa position initiale.

Le suivi des robots en convoi est obtenu en appliquant l'algorithme général présenté dans le chapitre 3. Pour calculer le contrôle latéral du convoi le long du platooning, l'algorithme *Memo-LAT* effectue les étapes suivantes à chaque pas de temps τ :

- choisir parmi les points mémorisés une position cible à viser (voir paragraphe 4.2.1) ;
- calculer, en adaptant le contrôle présenté dans [Daviet and Parent, 1996a] à notre modèle, la vitesse angulaire à appliquer sur le suiveur pour atteindre la position cible (voir paragraphe 4.2.2).

4.2.1 Choix de la position cible

Pour augmenter la précision du suivi, le robot suiveur vise un point de la trajectoire mémorisée de son prédécesseur. Ce point étant plus proche de sa position actuelle que le

prédécesseur lui même, l'écart latéral est plus faible que dans le cas où le robot suiveur vise la position courante de son prédécesseur. La position cible choisie est alors la première position, parmi celles en mémoire, dont l'inter-distance avec le robot suiveur est supérieure ou égale à une distance d'établissement d_l (voir figure 4.2). Étant donné que l'inter-distance entre deux robots successifs du convoi doit être supérieure ou égale à d_{crit} (inter-distance de sécurité séparant deux robots consécutifs), nous fixons une valeur constante à d_l , cette valeur étant choisie inférieure à d_{crit} .

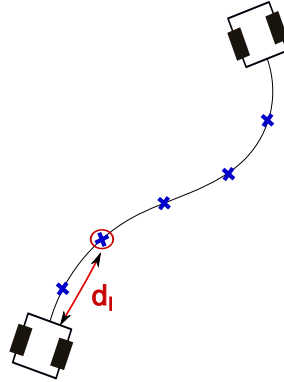


FIGURE 4.2 – Choix de la position à viser.

4.2.2 Calcul de la vitesse angulaire

Pour atteindre la position cible (x_{cible}, y_{cible}) choisie, le robot calcule une vitesse angulaire qui permet d'asservir autour de zéro son erreur angulaire par rapport à sa cible. Ce but est atteint en adaptant le contrôle développé dans [Daviet and Parent, 1996a] à notre modèle.

Daviet et Parent ont proposé dans [Daviet and Parent, 1996a] un contrôleur latéral pour un modèle de mouvement tricycle qui mène chaque robot du convoi à rejoindre la position courante du robot qui le précède. En d'autres termes, chaque robot cherche à maintenir son orientation dans la direction de son prédécesseur. Ce contrôle consiste tout d'abord à calculer l'erreur angulaire e_θ (voir figure 4.3) entre l'orientation courante du suiveur et l'orientation désirée pour atteindre la position cible. Nous utilisons ensuite cette erreur pour calculer une vitesse angulaire proportionnelle qui permet de la réduire. Comme nous avons signalé dans le chapitre 3, la vitesse angulaire calculée est toujours bornée entre ω_{min} et ω_{max} .

Pour calculer l'erreur angulaire e_θ , la position cible, initialement mémorisée dans le référentiel fixe du robot⁸, est projetée dans le référentiel mobile du suiveur.

Le calcul de la vitesse angulaire est ensuite donné par la formule suivante :

$$\omega = \frac{e_\theta}{\tau} \quad (4.2)$$

8. pour rappel, il s'agit du repère d'origine relatif à la configuration de départ du robot

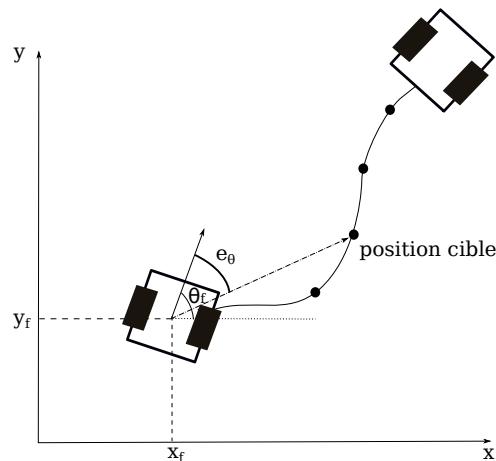


FIGURE 4.3 – Erreur angulaire entre le suiveur et la position cible.

4.3 Expérimentations

Nous avons choisi de mener des expériences à la fois en simulation (en simulant des robots à deux roues différentielles utilisant un modèle cinématique de type unicycle) et sur des robots Khépéras. Dans un premier temps, nous présentons les résultats des simulations de suivi sur un convoi de robots. Ces simulations ont pour but de déterminer les limites de l'algorithme. Nous avons choisi de déterminer les limites de *Memo-LAT* en simulation parce que nous aurons un environnement parfait non perturbé par le bruit qui nous permet d'étudier et de comprendre plus facilement le comportement des robots.

Ensuite, nous présentons la plate-forme expérimentale et les résultats des expériences réalisées qui montrent l'efficacité de l'approche proposée et les améliorations apportées par la mémorisation au platooning en comparaison avec le contrôle proposé par Daviet et Parent dans [Daviet and Parent, 1996a] (ce dernier n'utilisant pas la mémorisation du chemin du prédécesseur).

4.3.1 Validation par simulation

Dans un premier temps, nous cherchons à tester l'algorithme *Memo-LAT* sur notre simulateur afin d'étudier son comportement dans un environnement parfait. Comme nous l'avons déjà mentionné, le simulateur utilise le modèle de mouvement de robots à variation progressive expliqué dans la section 3.2.2 pour calculer le mouvement des robots. Nous cherchons plus particulièrement à déterminer les limites de l'algorithme en réalisant plusieurs simulations de suivis en convoi. En premier lieu, nous étudions l'influence de la distance d'établissement sur la précision du suivi en faisant varier la courbure du chemin à suivre. Ensuite, nous étudions la stabilité de l'algorithme quand la taille du convoi varie.

Dans toutes les simulations, nous considérons que les robots ont les caractéristiques suivantes :

- $v_{\max} = 0,65$ m/s, $v_{\min} = 0$ m/s ;
- $\omega_{\max} = -\omega_{\min} = 0,65$ rad/s.

4.3.1.1 Influence de la courbure sur la distance d'établissement

Comme expliqué dans la section 4.2.1, la distance d'établissement d_l constitue un paramètre fondamental de l'algorithme *Memo-LAT*. Elle représente la distance à partir de laquelle le robot cherche sa prochaine cible (c'est à dire que $distance_{robot-cible} > d_l$). Dans cette expérimentation, nous cherchons à étudier la relation entre la courbure du chemin à suivre et la distance d'établissement. La distance d'établissement étant constante durant le suivi, le choix de la valeur de d_l influe sur l'écart latéral (et par la suite, sur la précision du suivi), en particulier quand la courbure du chemin du leader augmente : plus la distance d'établissement augmente, plus la déviation latérale devient importante.

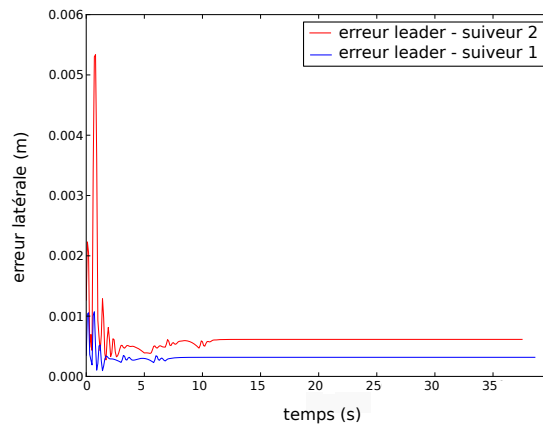
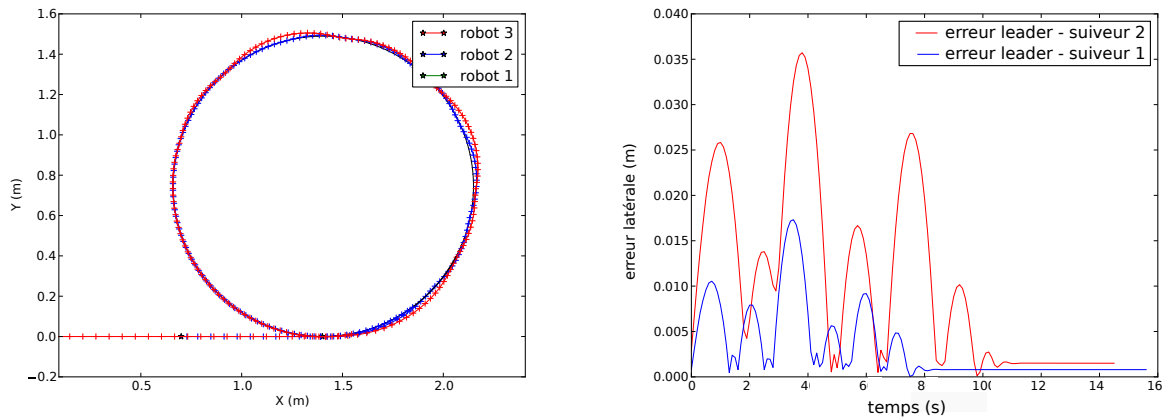


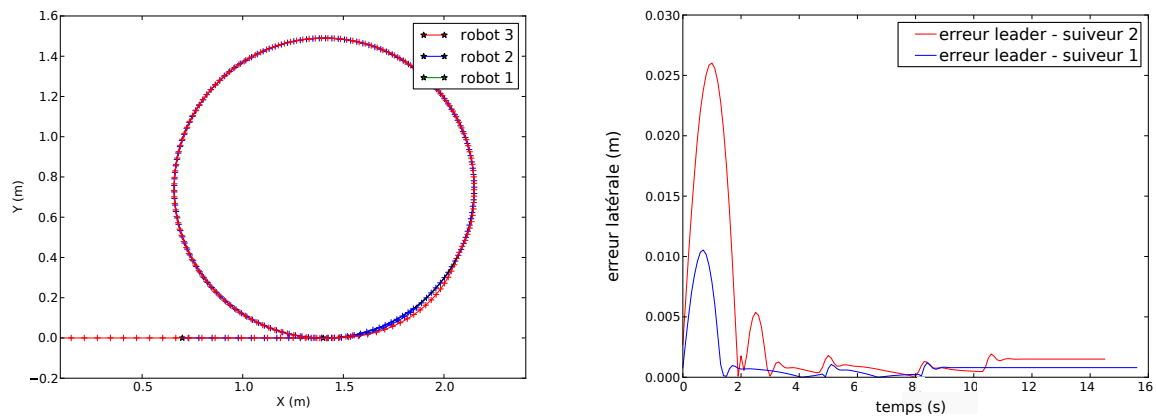
FIGURE 4.4 – Courbes d'erreur latérales entre les chemins des suiveurs et du leader lors d'un platooning le long d'un chemin à faible courbure.

Pour étudier l'impact de la courbure sur la stabilité du suivi, nous réalisons un platooning d'un convoi de 3 robots initialement alignés, de même orientation et distants de 0,7 m. L'inter-distance minimale admise pour le contrôle longitudinal est $d_{crit} = 0,1$ m et la distance d'établissement à laquelle les robots visent leurs cibles est égale à $d_l = 20$ mm. Le convoi évolue le long d'un chemin circulaire de rayon égal à 1,86 m à vitesse longitudinale constante égale à $v_{\max}/2$. La figure 4.4 montre les écarts latéraux entre les chemins des suiveurs et celui du leader.

Ensuite, en gardant les mêmes paramètres du système, nous réalisons un suivi le long d'un cercle de plus grande courbure (rayon égal à 0,74 m). Comme le montre la figure 4.5, un mouvement oscillatoire apparaît lorsque les robots rejoignent le chemin de leur leader. En effet, vu l'inter-distance importante de 0,7 m séparant les robots au début du suivi par rapport à $d_{crit} = 0,1$ m, le contrôleur longitudinal essaie de réduire cette erreur en calculant des fortes accélérations : les robots s'approchent alors du chemin de leur leader

FIGURE 4.5 – Platooning de 3 robots le long d'un chemin à forte courbure, $d_l = 20$ mm.

avec une vitesse longitudinale relativement élevée. Faisant face à un changement important de courbure et visant des points assez proches, les robots suivent le chemin circulaire du leader en oscillant autour. Ces oscillations, qui apparaissent aussi sur les courbes d'erreur latérale, s'amortissent graduellement jusqu'à leur disparition totale : l'écart latéral entre le chemin du leader et chacun des chemins des suiveurs diminue et tend vers une valeur constante très faible qui présente un état stable.

FIGURE 4.6 – Platooning de 3 robots le long d'un chemin à forte courbure, $d_l = 40$ mm.

Pour réduire considérablement ce mouvement oscillatoire, nous proposons d'augmenter la distance d'établissement de $d_l = 20$ mm à $d_l = 40$ mm. En comparant les courbes d'erreur dans les figures 4.5 et 4.6, nous constatons qu'en visant des positions plus éloignées ($d_l = 40$ mm), les suiveurs réduisent plus vite leurs oscillations autour du chemin de leur leader et convergent plus rapidement vers la consigne.

En conclusion de cette simulation, la précision du suivi dépend de la valeur de la distance d'établissement d_l . Cette distance ne pouvant pas être la même pour n'importe quel suivi, la précision du platooning peut être bien améliorée en adaptant la valeur de d_l à la courbure

du chemin à suivre. Nous pouvons ainsi déduire que **la distance d'établissement doit également être choisie en fonction de la courbure du chemin à suivre.**

4.3.1.2 Étude de la stabilité du platooning pour des longs convois

Les approches locales souffrent d'un inconvénient majeur : la stabilité du convoi n'est pas toujours garantie durant le suivi. La question que nous nous posons dans ce paragraphe est la suivante : est-ce que la stabilité du platooning est maintenue pour de longs convois contrôlés par l'algorithme *Memo-LAT* ?

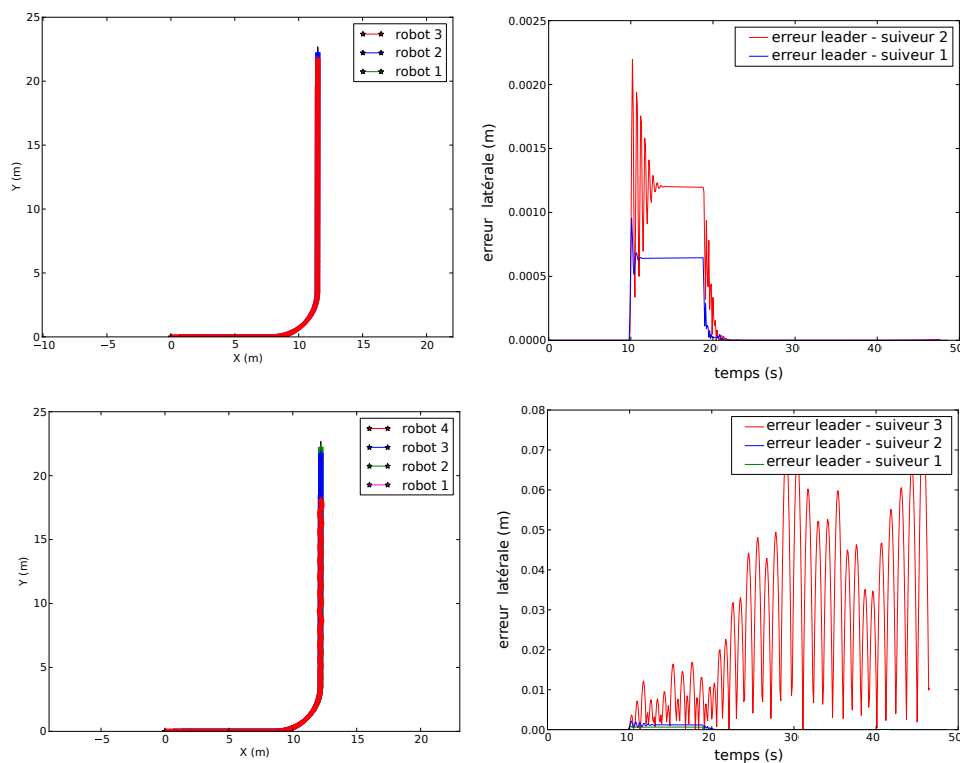


FIGURE 4.7 – Perte de stabilité avec l'augmentation du nombre de robots, $d_l = 20$ mm.

Pour répondre à cette question, nous réalisons plusieurs simulations de suivis, tous réalisés le long du même chemin qui est un virage serré à 90° , et où les robots sont initialement à l'arrêt et visent leurs cibles à une distance d'établissement $d_l = 20$ mm. Dans chaque simulation, nous faisons varier le nombre de robots dans le convoi.

La figure 4.7 compare deux platoonnings : le premier réalisé avec un convoi de 3 robots, et le second avec un convoi de 4 robots. Chaque suivi est représenté par les chemins tracés par les différents robots (figures à gauche) et par les courbes d'erreur latérale entre chacun des chemins des suiveurs et le chemin de leur leader (figures à droite).

En observant ces deux suivis, nous constatons que la stabilité du système diminue avec l'augmentation du nombre de robots dans le convoi : le mouvement oscillatoire des robots

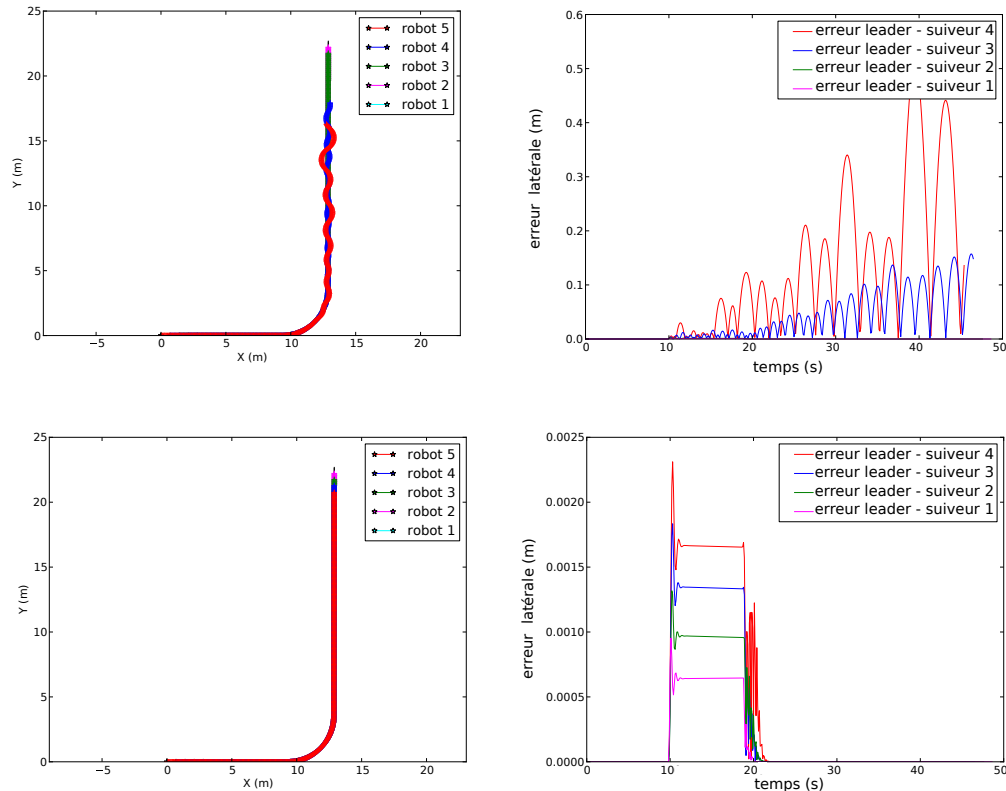


FIGURE 4.8 – Stabilité améliorée avec l’augmentation de d_l , $d_l = 20$ mm (2 figures en haut) et $d_l = 40$ mm (2 figures en bas).

autour du chemin du leader s’amplifie d’un robot à un autre. En effet, nous pouvons observer sur les courbes d’erreur latérale de la figure 4.7 (en haut à droite) que le comportement oscillatoire du premier suiveur affecte le comportement du deuxième suiveur qui oscille plus que son prédécesseur. En ajoutant un troisième suiveur au convoi, ce dernier cherche à viser des points du chemin oscillatoire du robot prédécesseur. Ces points, proches de lui ($d_l = 20$ mm) et proches les uns des autres, représentent fidèlement le mouvement oscillatoire du robot prédécesseur et entraînent son amplification, comme nous pouvons l’observer dans le suivi du dernier robot du convoi (figure 4.7 en bas à droite).

Dans le but de rétablir la stabilité du suivi, nous proposons d’augmenter la distance d’établissement avec le nombre de robots dans le convoi. Une distance d’établissement plus importante permet de viser des points, non seulement plus loin du robot suiveur, mais aussi plus éloignés les uns des autres. Ceci représente un lissage du chemin oscillatoire du robot prédécesseur. La figure 4.8 montre deux suivis réalisés avec deux valeurs de d_l différentes. Chaque suivi est représenté par les chemins tracés par les différents robots et par les courbes d’erreur latérale entre chacun des chemins des suiveurs et celui du leader du convoi. Les deux figures en haut correspondent à $d_l = 20$ mm, et les deux figures en bas à $d_l = 40$ mm. Nous remarquons que, pour ce suivi, viser plus loin permet de réduire le mouvement oscillatoire des robots suiveurs. Malgré l’accumulation des écarts latéraux

au niveau du virage, les robots arrivent à converger vite vers un état stable.

En conclusion, pour assurer la stabilité du suivi, d_l doit varier avec la taille des convois.

4.3.2 Validation sur les robots Khépéras



FIGURE 4.9 – Le robot Khépéra 3 conçu par *K-Team*.

Nous avons choisi de tester l'algorithme *Memo-LAT* sur des robots Khépéras puisque nous n'avons pas accès au robot *InTraDE* (le véhicule dédié au projet *InTraDE*). Les robots Khépéras (figure 4.9) sont des robots mobiles à deux roues différentielles d'une longueur de 130 mm et de vitesse maximale d'environ 0,17 m/s. Le diamètre des roues est égal à 41 mm.

Dans cette section, nous montrons des expérimentations de suivis en convoi obtenus en implémentant l'algorithme *Memo-LAT* sur les robots Khépéras décrites ci-dessus. Pour ce faire, nous présentons dans la section 4.3.2.1 les capteurs utilisés pour acquérir les informations nécessaires au calcul du contrôle. Une première expérience montre la phase de formation du convoi et son suivi le long d'un chemin spiral. Ensuite, nous observons l'influence de la vitesse du leader sur le comportement des suiveurs. Enfin, nous étudions l'influence de la distance d'établissement sur la précision du suivi.

4.3.2.1 Considérations techniques

Pour estimer les coordonnées polaires (inter-distance et inter-angle) d'un robot dans le repère courant de son suiveur, nous utilisons une table interactive [Simonin et al., 2011] qui simule un télémètre laser. En effet, la table interactive est capable de détecter, avec une fréquence de 45 Hz, la position et l'orientation de chaque robot avec une précision de 2 mm. Elle renvoie à chaque robot suiveur la distance d et la direction α de son prédécesseur. La table interactive présente un intérêt particulier puisqu'elle affiche et sauvegarde les chemins suivis par les différents robots du convoi. Nous pouvons ainsi récupérer tous ces chemins et tracer les courbes d'erreur latérale entre eux.

Pour estimer son état, le robot est muni d'un odomètre qui récupère le déplacement effectué par chaque roue et calcule la nouvelle configuration du robot (position, orientation, vitesse) par rapport à un référentiel fixe dont l'origine correspond à l'état initial du robot. Cependant, il est connu que plus le robot avance, plus l'estimation de sa position par odométrie est bruitée. Cette dérive affecte également l'estimation de la position du

prédécesseur dans le référentiel fixe du robot suiveur puisque cette dernière est obtenue en utilisant la position du robot suiveur et les coordonnées polaires du prédécesseur obtenues à l'aide de la table interactive (comme nous l'avons expliqué dans 4.1).

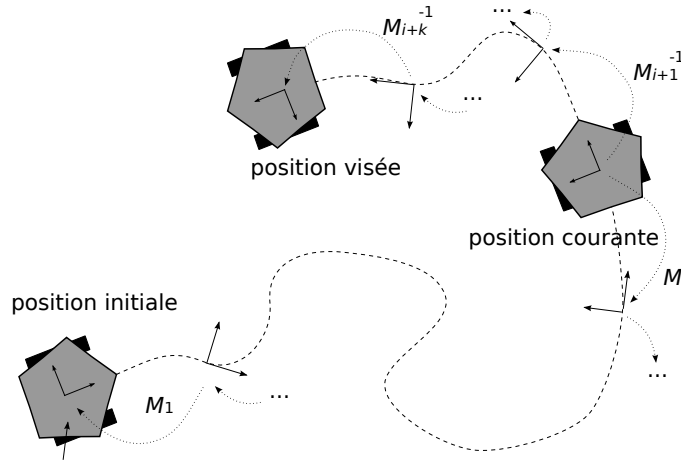


FIGURE 4.10 – Erreur due à la dérive de l’odométrie : faible pour le suivi mais importante pour la mémorisation.

Après avoir acquis une position du prédécesseur, sa projection Π_i dans le référentiel fixe du suiveur n’est qu’une composition de toutes les projections M_j ($j \in \{1, \dots, i\}$) effectuées pour passer d’un référentiel à un autre, comme le montre la figure 4.10. Si l’erreur pour passer d’un référentiel à un autre est faible, l’erreur obtenue en passant du référentiel mobile du robot à son référentiel fixe, qui n’est que l’accumulation de ces erreurs, devient plus importante. Or, après avoir choisi une position cible (x_{cible}, y_{cible}) , le calcul du contrôle n’utilise pas les coordonnées de cette position dans le référentiel fixe mais plutôt sa position relative au suiveur, obtenue en faisant une transformation inverse. Ainsi, l’erreur due à la dérive de l’odométrie est réduite à l’erreur obtenue en passant du référentiel courant du robot au référentiel dans lequel la position cible a été acquise.

4.3.2.2 Formation du convoi

Pour valider notre approche, nous réalisons un platooning de trois robots le long d’une spirale ouvrente (c’est-à-dire, parcourue dans le sens qui nous éloigne de son centre) où le leader se déplace à $v = 0,1$ m/s. L’intérêt du chemin en spirale réside dans le fait que la variation continue de sa courbure rend le travail du contrôleur latéral difficile. Initialement, les robots ne sont pas alignés sur la table interactive (Fig. 4.11), mais décalés arbitrairement dans le but d’observer la phase de convergence et de formation du convoi.

La figure 4.12 montre les trajectoires suivies par les différents robots au cours du suivi. Nous remarquons qu’au début du suivi, les robots suiveurs visent à atteindre les positions initiales de leurs prédécesseurs en générant des chemins rectilignes. Ils visent et suivent ensuite les chemins décrits par leurs prédécesseurs.



FIGURE 4.11 – Configurations initiales des robots du convoi.

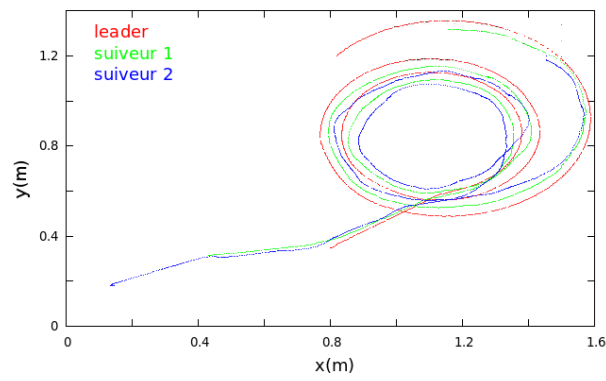


FIGURE 4.12 – trajectoires suivies par les robots durant un platooning à $v = 0,1$ m/s.

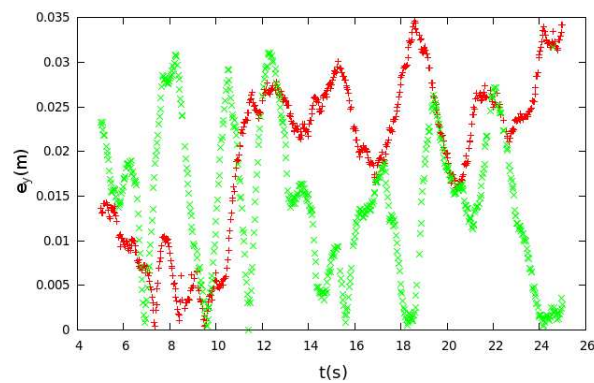


FIGURE 4.13 – Déviations latérales entre les chemins des suiveurs et leur leader (vert : leader-suiveur 1, rouge : leader-suiveur 2).

Pour évaluer la précision du suivi, nous calculons la déviation latérale⁹ entre le chemin de chaque suiveur et le chemin du leader. La figure 4.13 illustre la variation de la déviation latérale (axe des ordonnées, en m) en fonction du temps (axe des abscisses, en s). Nous remarquons que l'écart latéral maximal est de 35 mm, l'écart moyen est de 19 mm et l'écart latéral maximal entre deux robots consécutifs est de 30 mm. De plus, nous constatons que la déviation latérale se cumule d'un robot au suivant tout au long du convoi. Il s'agit d'un inconvénient typique des approches locales.

4.3.2.3 Influence de la vitesse du leader

Dans le but d'étudier l'influence de la vitesse longitudinale sur la précision du suivi, nous avons réalisé plusieurs expériences en faisant varier la vitesse du leader.

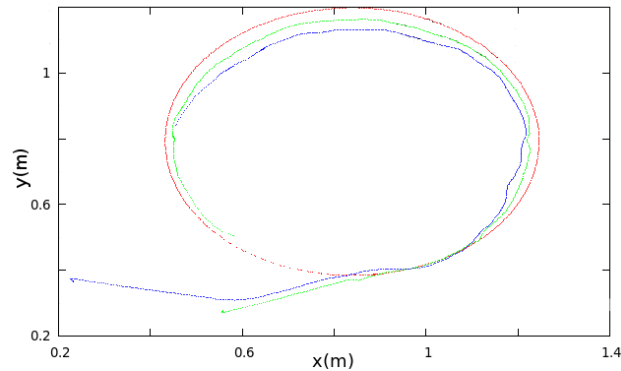


FIGURE 4.14 – Chemins des robots durant un platooning avec $v_{leader} = 0,1$ m/s (leader en rouge, premier suiveur en vert et second suiveur en bleu).

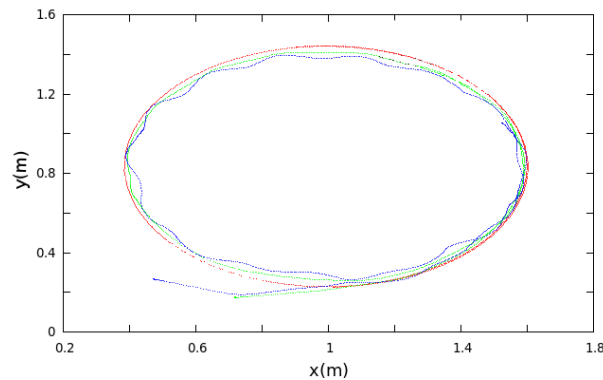


FIGURE 4.15 – Chemins des robots durant un platooning avec $v_{leader} = 0,15$ m/s (leader en rouge, premier suiveur en vert et second suiveur en bleu).

Nous disposons d'un convoi de 3 robots initialement décalés et orientés arbitrairement. Le leader se déplace à vitesse longitudinale constante le long d'un chemin circulaire et les

9. La déviation latérale est la distance de chaque point de la trajectoire du leader à la droite formée par les deux points les plus proches de la trajectoire du suiveur.

suiveurs sont soumis à la fois au contrôle longitudinal *SSC* et au contrôle latéral *Memo-LAT*. La distance d'établissement d_l considérée au cours de ces expériences est égale à 15 mm. Les figures 4.14 et 4.15 montrent les chemins empruntés par les robots lors de deux platoonings où le leader se déplace respectivement à $v_l = 0,1$ m/s et $v_l = 0,15$ m/s. Quand le leader se déplace à une vitesse longitudinale plus élevée, le contrôleur longitudinal de chaque suiveur calcule une accélération plus forte dans le but de garder une faible inter-distance avec le prédécesseur. Tout le convoi se déplace ainsi à une vitesse plus élevée. Nous remarquons que plus les robots avancent vite, plus l'écart latéral entre les chemins des robots est faible (une vitesse plus élevée permet de réduire plus rapidement l'écart latéral). En effet, la déviation latérale moyenne entre le chemin du leader et le chemin du dernier suiveur diminue de 40 mm quand $v_l = 0,1$ m/s à 27 mm quand $v_l = 0,15$ m/s.

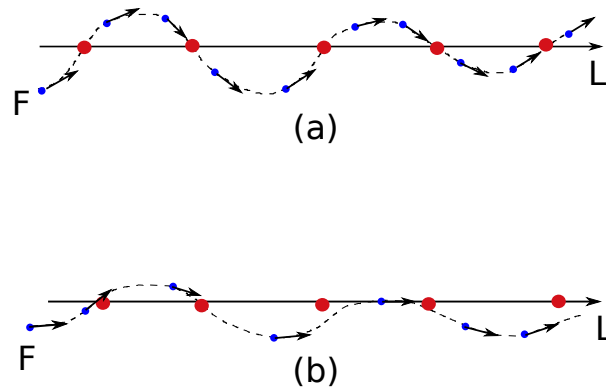
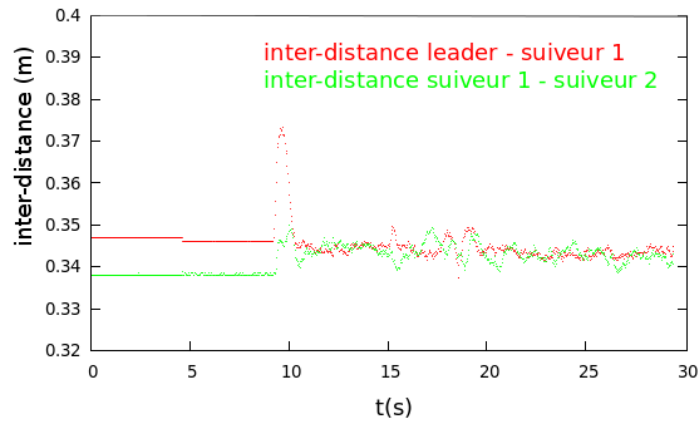
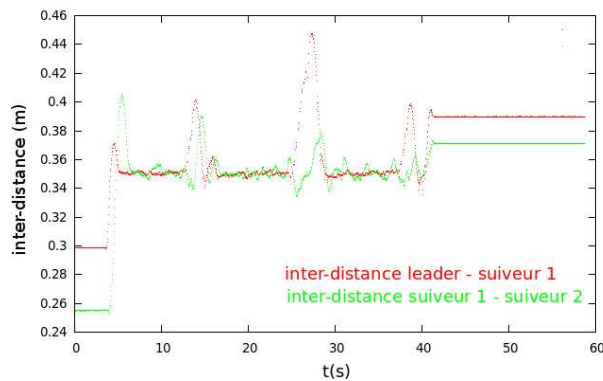


FIGURE 4.16 – Mouvement oscillatoire du robot autour du chemin de son prédécesseur.

En revanche, en comparant le suivi présenté par les figures 4.14 et 4.15, nous observons qu'un mouvement oscillatoire apparaît dans le déplacement des robots lorsque la vitesse du convoi devient plus importante. Nous essayons d'expliquer ce mouvement oscillatoire comme suit : quand les robots avancent plus rapidement, ils atteignent leurs positions visées plus vite. Viser une position cible à une faible distance d'établissement n'est pas une bonne idée parce qu'elle génère des oscillations. À cause de sa vitesse relativement importante, le robot parcourt une distance qui peut être plus grande que la distance à la position cible et peut la dépasser (figure 4.16-(a)) ou se rapprocher d'elle avec une erreur angulaire par rapport au chemin du leader d'autant plus importante que d_l est faible (figure 4.16-(b)). Le robot change ensuite de cible et le même processus se répète. Pour réduire les oscillations, les robots suiveurs doivent viser plus loin quand ils se déplacent plus vite. Nous déduisons ainsi que **la distance d'établissement doit être choisie en fonction de la vitesse longitudinale** de déplacement du robot : plus le robot avance vite, plus il doit regarder loin (plus la distance d'établissement est importante).

En observant les figures 4.17 et 4.18, nous remarquons que, quelle que soit la vitesse du leader et dans le cas où la courbure du chemin du leader est constante, le convoi atteint un état d'équilibre. C'est à dire un état où l'inter-distance entre chaque couple de robots successifs tend vers une valeur constante.

FIGURE 4.17 – Inter-distances entre les robots avec $v_{leader} = 0,1$ m/s.FIGURE 4.18 – Inter-distances entre les robots avec $v_{leader} = 0,15$ m/s.

4.3.2.4 Influence de la distance d'établissement

Un convoi formé de trois robots initialement alignés et de même orientation se déplace le long d'un cercle à vitesse constante égale à $0,1$ m/s. Pour étudier l'impact de la distance d'établissement sur les performances du suivi, nous avons réalisé plusieurs expériences de suivi sur un chemin circulaire en faisant varier à chaque fois la valeur de d_l .

La figure 4.19 montre un suivi réalisé avec une distance d'établissement égale à 15 mm. Sur la figure 4.20, chaque robot suiveur vise la position courante de son prédécesseur. Comme prévu, la déviation latérale est beaucoup plus importante dans la figure 4.20. Cette déviation latérale mène le dernier robot du convoi à tourner sur lui-même.

La figure 4.21 montre la variation de la déviation latérale moyenne en fonction de d_l : la courbe rouge correspond à l'écart entre le leader et le premier suiveur, et celle en vert correspond à l'écart entre le leader et le second suiveur. Par le terme *inf*, nous entendons l'inter-distance courante séparant deux robots consécutifs. En d'autres termes, chaque robot du convoi vise la position courante de son prédécesseur. En observant ces deux courbes, nous pouvons constater l'accumulation de l'erreur latérale d'un robot à un autre (vu que l'écart latéral est plus important pour le dernier suiveur que pour le premier).

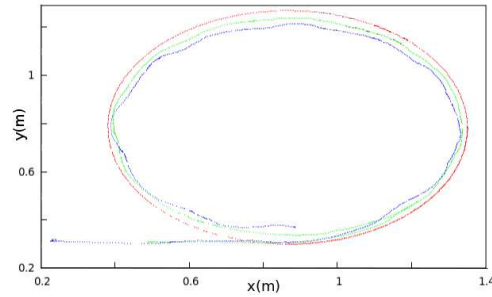


FIGURE 4.19 – Platooning le long d’un cercle avec $d_l = 15$ mm (leader en rouge, premier suiveur en vert et second suiveur en bleu).

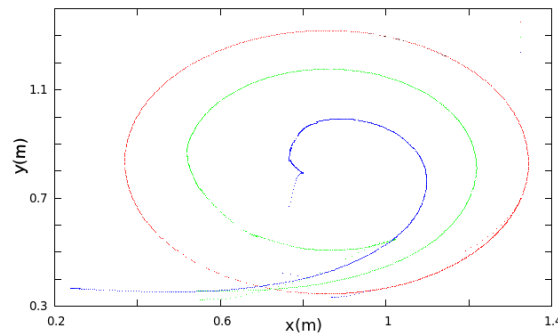


FIGURE 4.20 – Platooning le long d’un cercle où chaque robot vise son prédécesseur (leader en rouge, premier suiveur en vert et second suiveur en bleu).

De plus, comme prévu, plus d_l augmente, plus l’écart latéral entre le chemin d’un suiveur et celui du leader devient important. Nous remarquons aussi que l’erreur latérale augmente linéairement avec d_l en notant une rupture de continuité de l’augmentation à $d_l = 40$ mm.

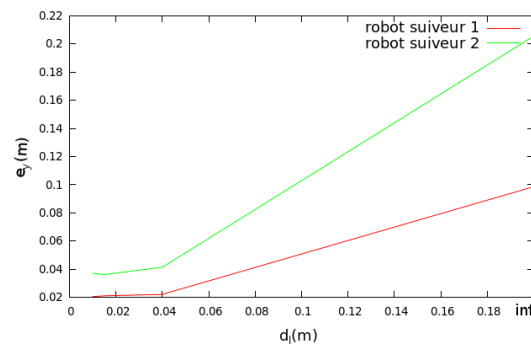


FIGURE 4.21 – Platooning sur un cercle : déviation latérale moyenne en fonction de d_l .

L’ensemble des 3 figures 4.21, 4.19 et 4.20 confirment l’efficacité de notre approche due à la mémorisation du chemin du prédécesseur. En mémorisant et suivant ce chemin plutôt qu’en essayant d’atteindre la position courante du prédécesseur, nous réduisons l’écart latéral entre les 2 chemins du robot et de son prédécesseur. Cependant, en visant

des positions très proches du suiveur, des mouvements oscillatoires apparaissent dans le comportement du suiveur. Le choix de la distance d'établissement d_l doit être alors un compromis entre vouloir réduire l'écart latéral et vouloir éliminer les oscillations.

4.4 Conclusion

Ce chapitre a présenté l'algorithme de platooning *Memo-LAT* : **Memorization and Look-Ahead Target**. Implémenté sur un robot du convoi, cet algorithme consiste à mémoriser le chemin de son prédécesseur et à suivre ce chemin au lieu de viser la dernière position détectée du prédécesseur. Pour suivre ce chemin, le robot suiveur choisit parmi les points mémorisés, des cibles situées à une distance d'établissement d_l (look-ahead distance) plus faible que l'inter-distance minimale admise entre deux robots consécutifs. Ensuite, il calcule une vitesse angulaire proportionnelle à l'erreur angulaire entre l'orientation courante du robot et son orientation désirée pour atteindre le point.

Les expériences réalisées sur des robots Khépéras montrent l'amélioration de la précision apportée au suivi grâce à la mémorisation du chemin. En revanche, cette précision, pour une valeur constante donnée de la distance d'établissement, varie selon la courbure du chemin à suivre et la vitesse du convoi. De plus, la stabilité du suivi est détériorée quand la taille du convoi devient importante. Pour rétablir la stabilité du convoi et améliorer la précision du suivi, les expériences nous ont montré qu'il faut ajuster la valeur de la distance d'établissement en fonction de la courbure du chemin et la vitesse du convoi. Ne connaissant pas à l'avance le chemin que le leader va générer durant le platooning, nous ne serons pas capables de choisir une valeur adéquate de d_l . De plus, vu que la courbure varie durant le déplacement, avoir d_l constante n'est pas un choix judicieux : nous souhaitons que d_l change dynamiquement et s'adapte à la courbure.

Plusieurs simulations ont été réalisées dans le but d'établir une expression analytique reliant la distance d'établissement d_l aux différents paramètres du système. Mais, nous n'avons pas réussi à trouver cette expression.

La courbure du chemin et la vitesse longitudinale constituent des paramètres essentiels dans le choix de la valeur de d_l . Sachant que la loi de contrôle utilisée dans *Memo-LAT* vise à suivre des points et non pas un chemin, ces paramètres ne sont pas alors pris en compte dans le calcul de la commande. Nous désirons ainsi améliorer *Memo-LAT* en proposant une loi de contrôle qui prend en compte la courbure et la vitesse du robot dans le calcul du comportement latéral, sans avoir besoin de choisir durant le déplacement du convoi des valeurs convenables de la distance d'établissement relatives à chaque robot. Nous proposons ainsi l'algorithme de contrôle *NOC* présenté dans le chapitre 5.

Chapitre 5

NOC et *Abstract NOC* : deux algorithmes de platooning qui s'appuient sur la mémorisation du chemin et la recherche heuristique

Comme déjà vu au chapitre 4, l'algorithme de platooning *Memo-LAT* ne garantit pas toujours la stabilité du suivi. En effet, il permet de réduire la déviation latérale entre les différents chemins des robots mais crée dans certains cas des oscillations, surtout au niveau des virages serrés. De plus, les performances du suivi dépendent fortement de la valeur de la distance d'établissement d_l (distance à partir de laquelle le robot choisit sa cible). Or, comme nous avons pu observer dans la partie expérimentations 4.3 du chapitre *Memo-LAT*, d_l dépend de la vitesse de déplacement du robot et de la courbure du chemin à suivre. Pour une meilleure précision, d_l doit être réglée en fonction de ces paramètres. Cependant, nous n'avons pas pu établir une expression analytique qui permet de calculer une valeur convenable de la distance d'établissement.

Dans ce chapitre nous souhaitons concevoir un algorithme permettant d'obtenir de meilleures précision et stabilité que *Memo-LAT* en prenant en compte la courbure du chemin à suivre et ne nécessitant pas de réglage de paramètres de contrôle. Cet algorithme doit minimiser le comportement oscillatoire des robots dans le but d'améliorer la stabilité du suivi tout en ayant un suivi précis permettant aux robots de converger au plus vite vers le chemin de leur leader. Le critère principal que nous cherchons à maximiser reste la stabilité même si cela se fait au prix de la vitesse de convergence. Étant donné que le modèle de déplacement des robots est non linéaire, il semble difficile d'établir une expression analytique de notre commande (qui est la vitesse angulaire) qui permette d'atteindre nos objectifs. Nous avons donc eu recours à une méthode géométrique (qui permet d'améliorer la stabilité) combinée à de la recherche heuristique (qui permet d'améliorer la précision). Nous proposons ainsi l'algorithme *NOC* (Non Oscillatory Convergence) suivant : chaque suiveur, ne disposant pas de l'équation analytique du chemin du leader ou de son prédécesseur mais seulement des positions successives de son prédécesseur, choisit des

positions cibles au fur et à mesure de son déplacement et reconstruit, autour de celles-ci, le chemin du prédécesseur par morceaux. Il calcule ensuite une commande lui permettant de converger vers ce chemin estimé et de le suivre, avec précision et sans oscillations.

Le contrôle proposé dans *NOC* est discret : après avoir discrétisé le domaine de variation de la vitesse angulaire, nous retenons l'ensemble des vitesses angulaires qui permettent au robot d'éviter d'osciller autour du chemin du prédécesseur. Ensuite, nous appliquons sur cet ensemble une recherche heuristique pour choisir la commande qui minimise une fonction d'optimisation \mathcal{E} que nous définissons plus tard dans ce chapitre. Si aucune vitesse angulaire ne permet au robot d'éviter de couper le chemin de son prédécesseur, *NOC* choisit comme commande la vitesse angulaire qui optimise \mathcal{E} .

L'algorithme *NOC* se divise en deux étapes réalisées à chaque pas de temps τ dans l'ordre suivant :

- choisir une cible, approcher le chemin autour de la cible par un cercle (\mathcal{C}) ou une droite (Δ) et déduire sa courbure (section 5.1) ;
- calculer la commande ω en prenant en compte la courbure estimée (section 5.2).

Dans la suite de ce chapitre, nous utilisons d'abord un modèle de déplacement simplifié de robots pour expliquer les différentes étapes de l'algorithme *NOC*. Ce modèle, appelé modèle à variation instantanée et présenté dans la section 3.2.1, suppose que la vitesse angulaire varie instantanément d'une valeur à une autre au début de chaque pas de temps. Après avoir validé cet algorithme dans la section 5.5, nous le testons sur un modèle plus proche du comportement réel de robots. Ce modèle, appelé modèle à variation progressive et présenté dans la section 3.2.2, prend en compte le temps nécessaire à la vitesse angulaire du robot pour passer d'une valeur à une autre. En observant les simulations réalisées sur ce modèle, nous remarquons que les performances de *NOC* deviennent moins bonnes. Pour améliorer ces performances, nous proposons d'abstraire l'algorithme *NOC* par rapport au modèle de mouvement. Cette abstraction, présentée dans la section 5.6, nous mène à l'algorithme de platooning *Abstract NOC* qui sera validé par simulations dans la section 5.7. À la fin du chapitre, la section 5.8 résume le travail et conclut.

L'article [Yazbeck et al., 2013], publié dans la conférence nationale CAR (Control Architecture of Robots), présente les deux algorithmes *Memo-LAT* et *NOC* et compare leurs performances en observant leurs résultats expérimentaux.

De plus, l'article [Yazbeck et al., 2014], à paraître dans les actes de la conférence internationale ICRA (International Conference on Robotics and Automation), explique les différentes étapes de l'algorithme *NOC* et discute des résultats expérimentaux.

5.1 Choix de la position cible et estimation de la courbure

Comme dans le cas de l'algorithme *Memo-LAT*, *NOC* mémorise aussi le chemin du robot prédécesseur. L'acquisition et la mémorisation de ce chemin a déjà été expliquée dans la section 4.1. Sachant qu'un de nos critères de conception du contrôleur *NOC* est l'évitement des oscillations (tant que possible), nous souhaitons choisir une cible qui permet au robot de suivre le chemin de son prédécesseur sans le couper. Pour choisir sa cible, le suiveur parcourt, dans sa mémoire, les positions non dépassées de son prédécesseur, et ceci à partir de la dernière position visée au pas de temps précédent. Ensuite, pour chaque position p à tester, le suiveur effectue les deux étapes suivantes jusqu'à ce qu'il trouve sa position cible :

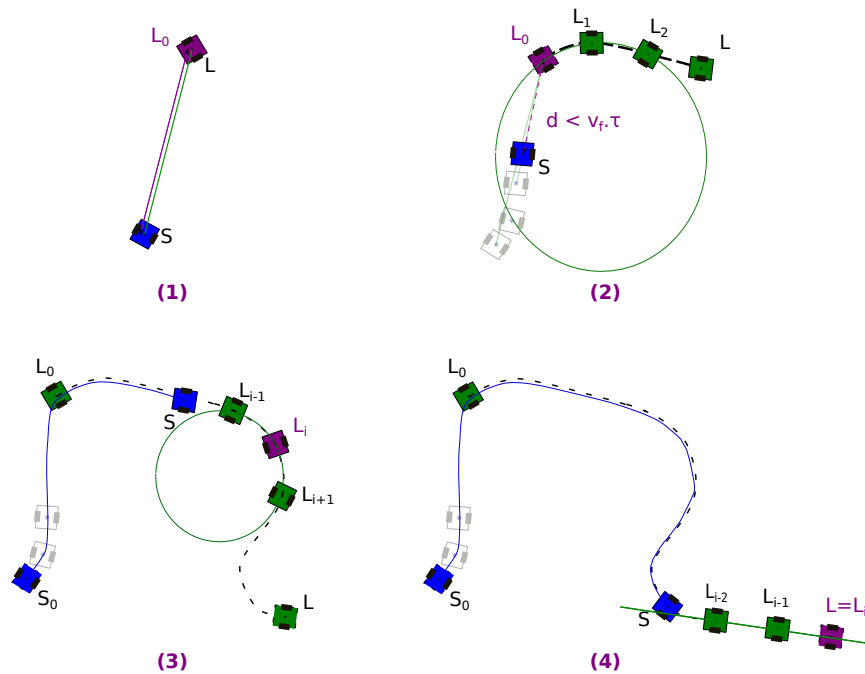


FIGURE 5.1 – Approximation du chemin du leader autour de la position cible.

étape 1 : estimer la courbure du chemin du leader autour de la position p en considérant les deux positions voisines de p et en faisant passer par ces 3 points une droite (Δ) ou un cercle (\mathcal{C}). La figure 5.1 représente plusieurs cas possibles quant au choix de ces trois points :

1. Au début du suivi, le suiveur ne connaît que la position du robot prédécesseur L (ou L_0) et la sienne S . Le chemin approché est la droite obtenue en joignant ces deux positions (figure 5.1-(1)).
2. À un instant quelconque, si le suiveur vise la position initiale L_0 du prédécesseur et si la distance séparant sa position courante de sa position cible est supérieure

à la distance qu'il parcourra durant τ ($< v_f \cdot \tau$), le chemin approché est la droite obtenue en joignant L_0 avec la position initiale du robot suiveur.

Quand la distance séparant sa position courante de sa position cible devient inférieure à $< v_f \cdot \tau$, le chemin approché est la courbe (droite ou cercle) qui passe par les trois premières positions L_0 , L_1 et L_2 du prédécesseur. Si le suiveur n'a que deux positions mémorisées du prédécesseur, le chemin approché est la courbe qui passe par la position initiale du suiveur S_0 et les deux positions sauvegardées du prédécesseur L_0 et L_1 (figure 5.1-(2)).

3. Si le robot vise un point intermédiaire L_i du chemin du prédécesseur, le chemin approché est la courbe (droite ou cercle) qui joint la cible avec le point précédent L_{i-1} et le point suivant L_{i+1} (figure 5.1-(3)).
4. Si la position cible L_i est la position courante L du prédécesseur, le suiveur considère les 2 points précédant la cible L_{i-1} et L_{i-2} et fait passer une droite ou un cercle par ces trois points (figure 5.1-(4)).

Si le chemin construit est un segment de droite, la courbure estimée est nulle. Dans le cas où ce chemin est un arc de cercle, la courbure estimée est l'inverse de son rayon.

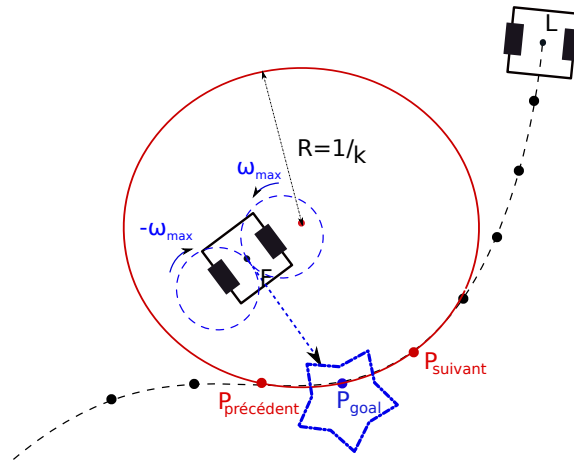


FIGURE 5.2 – Choix de la nouvelle position cible.

étape 2 : vérifier que le robot suiveur peut éviter de couper (Δ) ou (\mathcal{C}) en tournant à vitesse angulaire maximale, tout en gardant une vitesse longitudinale constante. Pour vérifier cela, nous traçons les deux cercles de rayon $R = \frac{v_0}{\omega_{max}}$, un en tournant à $+\omega_{max}$ et un autre en tournant à $-\omega_{max}$. Si au moins un de ces 2 cercles ne coupe pas le chemin approché, le suiveur a trouvé sa position cible et ne teste pas les points suivants.

Un exemple de choix d'une position cible est montré dans la Figure 5.2 où le suiveur choisit une position P_{goal} et approche le chemin autour de P_{goal} par un cercle (en rouge). Ensuite, il trace les deux cercles obtenus en tournant à $\pm\omega_{max}$ et constate qu'aucun des deux ne coupe le cercle en rouge. Dans ce cas, la nouvelle position cible du suiveur est P_{goal} .

Si le robot teste toutes positions possibles de son prédécesseur et ne trouve pas sa position cible, il vise la position courante de son prédécesseur. Ceci nous ramène au dernier point énuméré dans la première étape.

5.2 Calcul de la commande latérale

Dans un premier lieu, nous supposons que le convoi de robots se déplacent à vitesse longitudinale constante v_0 . Nous combinerons le contrôleur longitudinal avec le contrôleur latéral dans la section 5.3.

Le calcul de la commande, qui est ici discrète, repose sur la combinaison d'une approche géométrique avec la recherche heuristique. D'une part, l'approche géométrique améliore la stabilité du suivi en interdisant le mouvement oscillatoire des robots autour du chemin du leader. D'autre part, la recherche heuristique améliore la précision du suivi en choisissant une vitesse angulaire qui réduit au mieux la déviation latérale et l'erreur angulaire. En premier lieu, nous décomposons le domaine de variation de la vitesse angulaire $[\omega_{min}, \omega_{max}]$ en n pas. Ensuite, nous éliminons, selon une méthode géométrique, toutes les vitesses angulaires de l'ensemble $\{\omega_{min}, \dots, \omega_{max}\}$ qui mènent le suiveur à couper ou à osciller autour du chemin construit. De l'ensemble des vitesses angulaires retenues, nous choisissons celle qui minimise une fonction d'optimisation \mathcal{E} (que nous détaillerons plus tard) basée sur les erreurs latérale et angulaire du suiveur par rapport au chemin construit (Δ) ou (\mathcal{C}). La commande choisie parmi l'ensemble des commandes possibles est la commande optimale par rapport au chemin (Δ) ou (\mathcal{C}).

Pour éliminer les vitesses angulaires ne respectant pas les contraintes d'intersection avec (Δ) ou (\mathcal{C}), nous testons pour chaque ω_i appartenant à l'ensemble $\{\omega_{min}, \dots, \omega_{max}\}$ deux conditions Γ_i et Γ_i^+ . Ces deux conditions vérifient qu'un véhicule ayant appliqué la commande ω_i pendant un temps τ , ne coupera pas dans le futur (au delà de $2.\tau$) le chemin estimé du véhicule qui le précède en appliquant sa vitesse angulaire maximale. Le calcul de la commande latérale se résume alors aux étapes suivantes :

Pour chaque ω_i de l'ensemble $\{\omega_{min}, \dots, \omega_{max}\}$, $i : 1 \rightarrow n$:

- calculer le chemin ((Δ_i) ou (\mathcal{C}_i)) tracé par le suiveur en tournant avec une vitesse angulaire ω_i .
- tester la condition Γ_i décrite dans 5.2.1 : si la condition est satisfaite, continuer. Sinon, tester la vitesse angulaire suivante ω_{i+1} ;
- tester la condition Γ_i^+ décrite dans 5.2.2 : si la condition est vérifiée, retenir ω_i .

L'expression analytique de la commande latérale ω^* est alors :

$$\omega^* = \underset{\omega \in \mathcal{W}^*}{\operatorname{argmin}}(\mathcal{E}(\omega)) \quad (5.1)$$

où étant donné l'ensemble $\mathcal{W} = \{\omega_{min}, \dots, \omega_{max}\}$,

$$\mathcal{W}^* = \{\omega \in \mathcal{W}, / \Gamma_i \wedge \Gamma_i^+ = 1\} \quad (5.2)$$

Si l'ensemble \mathcal{W}^* est vide, nous formons l'ensemble \mathcal{W}_\cap des vitesses angulaires appliquées pendant τ qui permettent au robot d'éviter de couper le chemin de son prédécesseur. Mais il y aura alors forcément une amorce d'oscillation puisque le chemin du véhicule prédécesseur sera forcément coupé.

$$\mathcal{W}_\cap = \{\omega \in \mathcal{W}, / \Gamma_i = 1 \text{ et } \Gamma_i^+ = 0\} \quad (5.3)$$

Si l'ensemble \mathcal{W}_\cap n'est pas vide, la commande ω^* devient :

$$\boxed{\omega^* = \underset{\omega \in \mathcal{W}_\cap}{\operatorname{argmin}}(\mathcal{E})} \quad (5.4)$$

Dans le cas contraire (\mathcal{W}_\cap est vide), ω^* est la vitesse angulaire de l'ensemble \mathcal{W} qui minimise la fonction d'erreur \mathcal{E} :

$$\boxed{\omega^* = \underset{\omega \in \mathcal{W}}{\operatorname{argmin}}(\mathcal{E})} \quad (5.5)$$

Dans la suite, nous décrirons les deux conditions Γ_i et Γ_i^+ dans le cas où le chemin estimé du prédécesseur est un cercle ou bien une droite. Avant cela, nous rappelons les expressions des différents chemins utilisés plus loin. Le chemin du prédécesseur peut être approximé par un segment de droite (Δ) ou un arc de cercle (\mathcal{C}), alors que le suiveur peut générer un segment de droite (Δ_i) (si sa vitesse angulaire ω_i est nulle) ou bien un arc de cercle (\mathcal{C}_i) de centre $\Omega_i(x_i, y_i)$ et de rayon r_i (si $\omega_i \neq 0$) :

- (Δ) : $ax + by + c = 0$;
- (\mathcal{C}) : $(x - x_c)^2 + (y - y_c)^2 = r_c^2$;
- (Δ_i) : $\begin{cases} x \cdot \tan \theta_0 - y + y_0 - x_0 \cdot \tan \theta_0 = 0, & \text{si } \theta_0 \neq \pm \frac{\pi}{2}. \\ x = x_0, & \text{si } \theta_0 = \pm \frac{\pi}{2}. \end{cases}$
- (\mathcal{C}_i) : $(x - x_i)^2 + (y - y_i)^2 = r_i^2$,
avec $x_i = (x_0 - \frac{v_0}{\omega_i} \cdot \sin \theta_0)$, $y_i = (y_0 + \frac{v_0}{\omega_i} \cdot \cos \theta_0)$ et $r_i = |\frac{v_0}{\omega_i}|$

5.2.1 Condition Γ_i

Cette première condition permet de vérifier que le robot suiveur, soumis à la commande ω_i , peut se déplacer durant un pas de temps τ sans couper le chemin estimé (Δ) ou (\mathcal{C})

de son prédécesseur. En appliquant ω_i pendant τ , le suiveur passe de la configuration $q_0 = q(\omega_0) = (x_0, y_0, \theta_0, v_0, \omega_0)$ à la configuration $q_i = q(\omega_i) = (x_i, y_i, \theta_i, v_0, \omega_i)$ ¹⁰. Pour que la condition Γ_i soit vérifiée, l'arc $\widehat{q_0 q_i}$ et (Δ) ou (\mathcal{C}) ne doivent pas se couper.

Dans le cas d'une droite comme dans le cas d'un cercle, la Figure 5.3 montre que la seule vitesse angulaire retenue est ω_l . En tournant avec ω_j ou ω_k , le suiveur coupe (Δ) et (\mathcal{C}) .

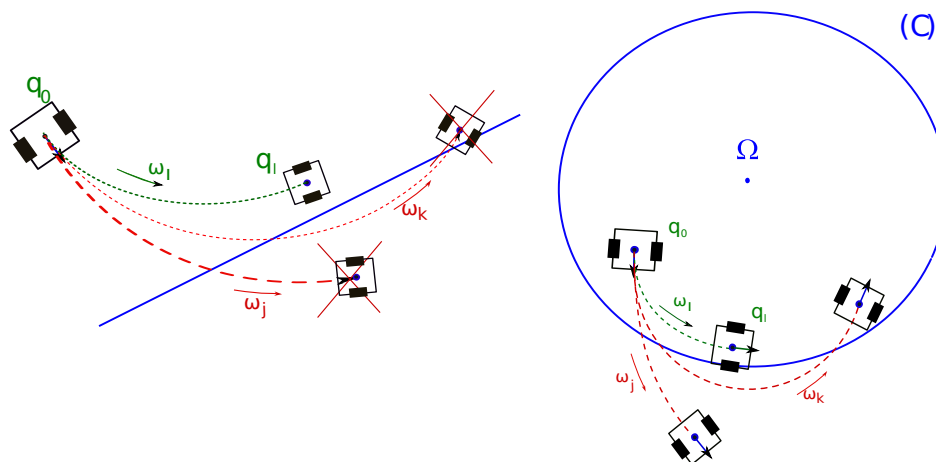


FIGURE 5.3 – Test de Γ_i dans le cas d'une droite et d'un cercle.

5.2.2 Condition Γ_i^+

Afin de garantir l'absence d'oscillation autour du chemin estimé (Δ) ou (\mathcal{C}) pour

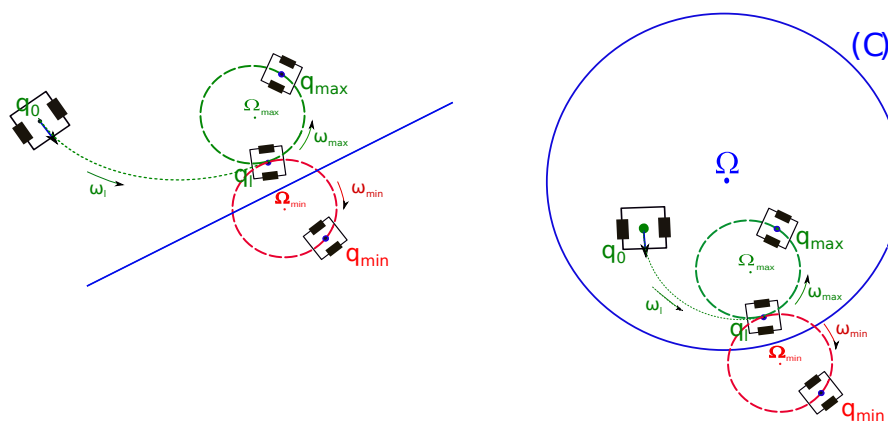


FIGURE 5.4 – Test de Γ_i^+ dans le cas d'une droite et d'un cercle.

au moins le prochain pas de temps, le suiveur vérifie qu'à partir de sa configuration q_l , il

10. Pour simplifier les notations dans les sections relatives aux conditions Γ_i et Γ_i^+ , $q(\omega_i)$ sera noté q_i . Dans le reste de la thèse, q_i est relative à l'instant t_i .

pourra éviter toute intersection avec (Δ) ou (\mathcal{C}) en tournant à vitesse angulaire maximale. Ainsi, le robot calcule sa nouvelle configuration q_i obtenue s'il se déplace à v_0 et ω_i pendant τ . Ensuite, en se déplaçant à vitesse longitudinale constante v_0 , le suiveur génère deux cercles de centres Ω_{min} et Ω_{max} en tournant à des vitesses angulaires $-\omega_{max}$ et $+\omega_{max}$ respectivement. La condition Γ_i^+ est vraie quand au moins un de ces deux cercles ne coupe pas (Δ) ou (\mathcal{C}) . En d'autres termes, si la distance d'un des deux centres Ω_{min} et Ω_{max} au chemin estimé est supérieure au rayon du cercle $\frac{v_0}{\omega_{max}}$, la condition Γ_i^+ est satisfaite.

Comme le montre la figure 5.4, ω_l peut être une éventuelle commande latérale puisqu'elle vérifie les deux conditions Γ_i et Γ_i^+ : d'une part, l'arc $\widehat{q_0 q_l}$ ne coupe ni (Δ) ni (\mathcal{C}) . D'autre part, en partant de la configuration q_l , le suiveur peut éviter de couper le chemin estimé du prédécesseur en tournant à $+\omega_{max}$.

5.2.3 La fonction d'erreur

La fonction d'erreur a un rôle déterminant dans la qualité du suivi. Cette fonction

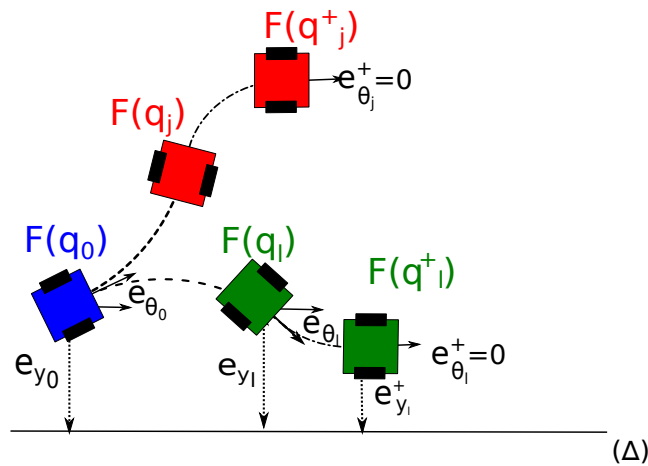


FIGURE 5.5 – Représentation géométrique de la fonction d'optimisation.

est utilisée comme un critère que doit optimiser la recherche heuristique afin de déterminer parmi l'ensemble des vitesses angulaires retenues celle permettant au suiveur de réduire le plus vite possible son écart latéral par rapport au chemin de son prédécesseur. Cette convergence rapide doit permettre au suiveur d'atteindre le chemin visé avec une orientation tangentielle pour éviter de couper le chemin visé et ainsi éviter une oscillation autour de celui-ci. Pour cela, l'erreur angulaire apporte une information précieuse sur le comportement latéral futur du suiveur. En partant d'une configuration initiale q_0 du robot suiveur, et pour chaque ω_i de l'ensemble des vitesses angulaires possibles :

- nous calculons la nouvelle configuration q_i obtenue en appliquant ω_i sur q_0 ;
- ensuite, nous calculons l'écart latéral et l'erreur angulaire du suiveur de configuration q_i par rapport au chemin approché de son prédécesseur ;

- le robot suiveur étant en q_i , nous appliquons une vitesse angulaire maximale sur ce dernier pendant un certain temps jusqu'à atteindre une configuration q_i^+ où son erreur angulaire est nulle. $F(q_i^+)$ est la position du robot suiveur ayant une configuration q_i^+ ;
- la fonction d'erreur est alors égale à l'écart latéral du suiveur de configuration q_i^+ par rapport au chemin approché.

La Figure 5.5 montre un exemple d'un robot suiveur, initialement à $F(q_0)$, et tel que le chemin de son prédécesseur est approché par une droite (Δ). Si le suiveur applique ω_l , il atteint la configuration q_l où il va tourner à ω_{max} jusqu'à arriver à q_l^+ où son erreur angulaire est nulle. Le même comportement est obtenu si le suiveur applique ω_j . Dans ce cas, il va atteindre q_j^+ avec une erreur angulaire nulle. Nous remarquons que le sens de rotation adopté par le robot pour passer de q_l à q_l^+ n'est pas le même que pour passer de q_j à q_j^+ . En effet, le robot tourne avec sa vitesse angulaire maximale $+\omega_{max}$ quand son erreur angulaire est positive et avec $-\omega_{max}$ quand celle-ci est négative. Ainsi, la vitesse de rotation du robot est égale à $\text{signe}(e_\theta) \cdot \omega_{max}$. En procédant ainsi, le robot atteint plus rapidement une position où son erreur angulaire est nulle. En comparant les deux écarts latéraux aux configurations q_l^+ et q_j^+ , nous constatons que e_{yl}^+ est la valeur optimale de la fonction d'erreur car elle représente l'erreur la plus faible. Ainsi, le suiveur choisit, entre les deux commandes possibles ω_j et ω_l , d'appliquer ω_l sur le robot suiveur.

Le même raisonnement peut être suivi dans le cas où le chemin du prédécesseur est approché par un cercle. Dans les deux cas (droite ou cercle), l'équation analytique de la fonction d'optimisation \mathcal{E} s'exprime en fonction de la configuration q_0 du robot, de la vitesse angulaire ω et de la vitesse longitudinale constante v_0 .

Pour passer de q_0 à q_i , le robot applique ω_i . Sa nouvelle configuration est dans ce cas :

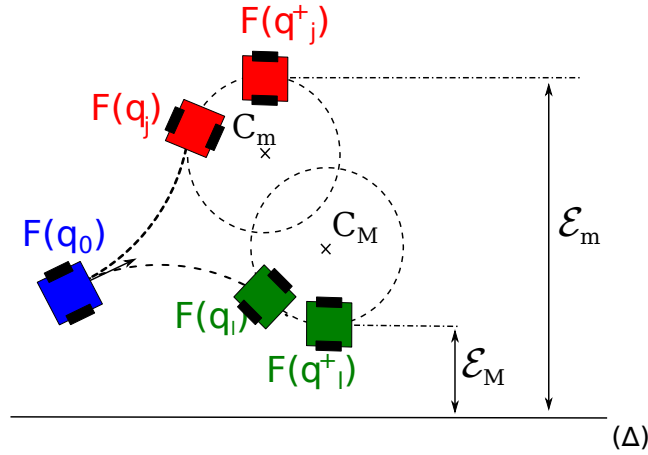
$$\begin{cases} x_i = x_0 + \frac{v_0}{\omega_i} (\sin(\theta_0 + \omega_i \tau) - \sin \theta_0) \\ y_i = y_0 - \frac{v_0}{\omega_i} (\cos(\theta_0 + \omega_i \tau) - \cos \theta_0) \\ \theta_i = \theta_0 + \omega_i \tau \end{cases} \quad (5.6)$$

Ensuite, étant à $F(q_i)$, le robot tourne avec une vitesse angulaire égale à $\text{signe}(e_{\theta_i}) \cdot \omega_{max}$ jusqu'à atteindre $F(q_i)$ où l'erreur angulaire est nulle. Il trace donc un arc d'un cercle centré à C_{max} de coordonnées (x_c, y_c) :

$$\begin{cases} x_c = x_i - \text{signe}(e_{\theta_i}) \cdot \frac{v_0}{\omega_{max}} \cdot \sin \theta_i \\ y_c = y_i + \text{signe}(e_{\theta_i}) \cdot \frac{v_0}{\omega_{max}} \cdot \cos \theta_i \end{cases} \quad (5.7)$$

En posant $\text{signe}(e_{\theta_i}) = \frac{e_{\theta_i}}{|e_{\theta_i}|}$ et en remplaçant x_i et y_i par leurs expressions 5.6, nous obtenons :

$$\begin{cases} x_c = x_0 + \frac{v_0}{\omega_i} (\sin(\theta_0 + \omega_i \tau) - \sin \theta_0) - \frac{e_{\theta_i}}{|e_{\theta_i}|} \cdot \frac{v_0}{\omega_{max}} \cdot \sin(\theta_0 + \omega_i \tau) \\ y_c = y_0 - \frac{v_0}{\omega_i} (\cos(\theta_0 + \omega_i \tau) - \cos \theta_0) + \frac{e_{\theta_i}}{|e_{\theta_i}|} \cdot \frac{v_0}{\omega_{max}} \cdot \cos(\theta_0 + \omega_i \tau) \end{cases} \quad (5.8)$$


 FIGURE 5.6 – Calcul de la fonction d'optimisation \mathcal{E} .

La fonction d'optimisation \mathcal{E} est égale à la distance du robot suiveur au chemin visé. En observant la figure 5.6, nous considérons ainsi deux cas selon que le robot tourne à $+\omega_{max}$ (passage de q_l à q_l^+) ou $-\omega_{max}$ (passage de q_j à q_j^+) :

- si le robot tourne à $+\omega_{max}$ ($e_{\theta l} > 0$), l'écart latéral \mathcal{E}_M à $F(q_l^+)$ est égal à la distance du centre C_M au chemin visé (Δ) moins le rayon $\frac{v_0}{\omega_{max}}$;
- si le robot tourne à $-\omega_{max}$ ($e_{\theta j} < 0$), l'écart latéral \mathcal{E}_m à $F(q_j^+)$ est égal à la distance du centre C_m au chemin visé (Δ) plus le rayon $\frac{v_0}{\omega_{max}}$.

Nous remarquons que l'ajout ou la suppression du rayon $\frac{v_0}{\omega_{max}}$ à la distance du centre C_M ou m au chemin estimé du robot prédécesseur dépend du sens de rotation du robot adopté pour passer de sa configuration q à q^+ ; en d'autres termes, ça dépend du signe de l'erreur angulaire e_θ . Ainsi, nous pouvons généraliser l'expression de la fonction d'optimisation comme suit :

$$\mathcal{E} = \left| \text{distance}(C_M \text{ ou } m, \text{ chemin visé}) - \frac{e_{\theta i}}{|e_{\theta i}|} \cdot \frac{v_0}{\omega_{max}} \right|$$

Nous considérons ensuite deux cas selon que le chemin du prédécesseur est approché par une droite (Δ) d'équation $ax + by + c = 0$ ou un cercle (\mathcal{C}) de centre C et de rayon R :

$$\mathcal{E} = \left| \frac{|ax_c + by_c + c|}{\sqrt{a^2 + b^2}} - \frac{e_\theta}{|e_\theta|} \cdot \frac{v_0}{\omega_{max}} \right| \quad (5.9)$$

$$\mathcal{E} = \left| R - CC_{max} - \frac{e_\theta}{|e_\theta|} \cdot \frac{v_0}{\omega_{max}} \right|$$

5.2.4 Raffinement de la commande latérale

Après avoir choisi la vitesse angulaire ω^* de l'ensemble discrétisé $\{\omega_{min}, \dots, \omega_{max}\}$ comme commande latérale selon l'équation 5.1, ω^* peut ne pas être la commande optimale.

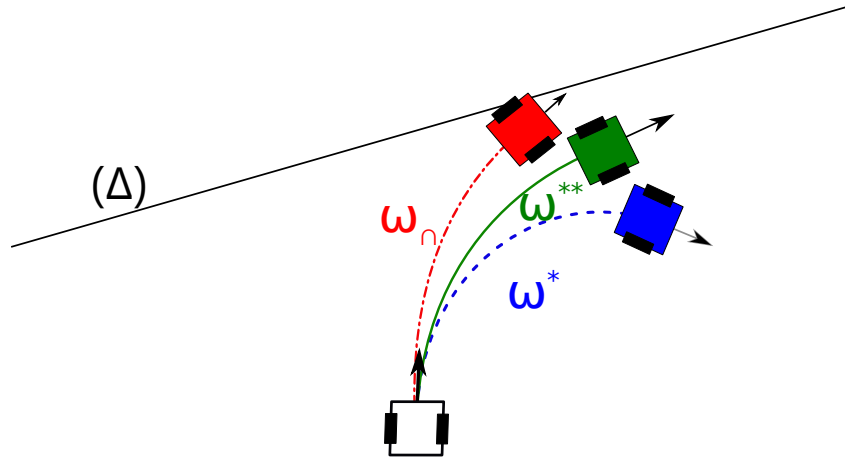


FIGURE 5.7 – Raffinement de la vitesse angulaire pour plus minimiser la fonction d’erreur.

En effet, le domaine de variation de la vitesse angulaire $[\omega_{min}, \omega_{max}]$ peut contenir une meilleure commande latérale ω^{**} , qui vérifie à la fois Γ_i et Γ_i^+ et qui réduit plus que ω^* la fonction d’erreur \mathcal{E} , mais qui n’apparaît pas dans l’ensemble discrétisé $\{\omega_{min}, \dots, \omega_{max}\}$ à cause d’une discrétisation trop grossière. En effet, pendant le calcul de la commande latérale ω^* , et en testant les ω_i de l’ensemble $\{\omega_{min}, \dots, \omega_{max}\}$, le contrôleur latéral peut trouver une ou plusieurs vitesses angulaires qui satisfont uniquement Γ_i (Γ_i^+ est fausse). En désignant par ω_η la vitesse angulaire qui minimise \mathcal{E} parmi les vitesses angulaires vérifiant uniquement Γ_i , une meilleure commande latérale appartient à l’intervalle $[\omega_\eta, \omega^*]$. ω^{**} est alors obtenue en décomposant $[\omega_\eta, \omega^*]$ en n_{raf} pas et en appliquant les étapes suivies pour obtenir ω^* . Par analogie, l’expression analytique de la nouvelle vitesse angulaire raffinée est obtenue comme suit :

étant donné l’ensemble $\mathcal{W} = \{\omega_{min}, \dots, \omega_{max}\}$, nous déterminons l’ensemble \mathcal{W}_η et la vitesse angulaire ω_η selon les expressions suivantes :

$$\begin{aligned} \mathcal{W}_\eta &= \{\omega \in \mathcal{W}, / \Gamma_i = 1 \text{ et } \Gamma_i^+ = 0\} \\ \omega_\eta &= \underset{\omega \in \mathcal{W}_\eta}{\operatorname{argmin}}(\mathcal{E}) \end{aligned}$$

Ensuite, en utilisant l’ensemble $\mathcal{W}^{**} = \{\omega_\eta, \dots, \omega^*\}$, nous calculons la nouvelle commande selon l’expression suivante :

$$\omega^{**} = \underset{\omega \in \mathcal{W}^{**}}{\operatorname{argmin}}(\mathcal{E})$$

5.3 Combinaison des contrôleurs longitudinal et latéral

La conception de notre algorithme *NOC* a été réalisée dans le cas d’un convoi qui se déplace à vitesse longitudinale constante. Or, pour avoir un suivi sécurisé, un contrôle

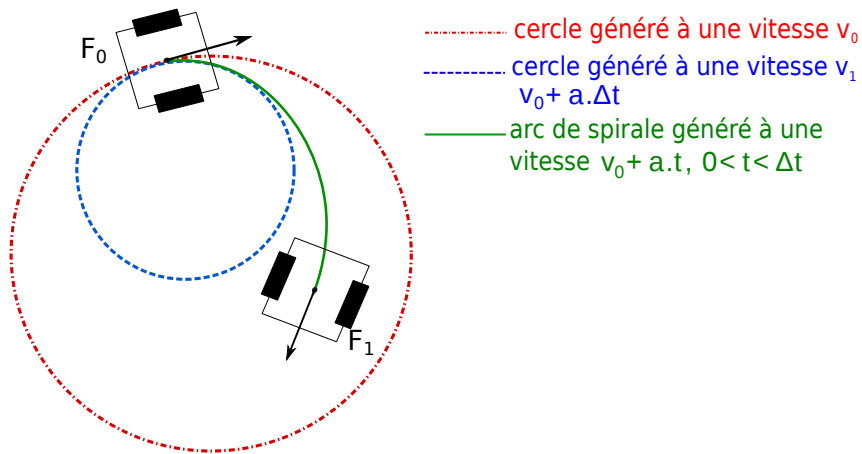


FIGURE 5.8 – Limitation du déplacement du suiveur par deux cercles.

longitudinal doit être pris en compte afin d'éviter les collisions. Ainsi, les deux contrôles longitudinal et latéral sont exécutés à chaque pas de temps τ . Un robot, initialement localisé à F_0 , ayant une vitesse de départ v_0 et soumis aux commandes d'accélération longitudinale a et de vitesse angulaire ω constantes pendant τ , se déplace selon un arc de spirale $\widehat{F_0 F_1}$ (arc en vert sur la Figure 5.8). Cet arc évolue dans l'espace borné par deux cercles générés avec la même vitesse angulaire ω :

- le premier cercle (en rouge), de rayon $r_0 = |v_0/\omega|$, est obtenu si le robot avance en gardant sa vitesse longitudinale initiale v_0 ;
- le deuxième cercle (en bleu), de rayon $r_1 = |v_1/\omega|$, est obtenu si le robot avance avec sa nouvelle vitesse $v_1 = v_0 + a.\tau$ (c'est la vitesse obtenue à la fin du pas de temps). Dans cet exemple, nous supposons que l'accélération a calculée par le contrôleur longitudinal est négative. Ainsi, le cercle bleu a un rayon r_1 plus petit que le rayon r_0 du cercle rouge.

Les tests des deux conditions Γ_i et Γ_i^+ utilisent l'arc de cercle généré par le suiveur en se déplaçant à vitesses longitudinale et angulaire constantes. Pour garantir l'absence d'intersection avec le chemin du prédécesseur, le robot suiveur considère le cercle (un des deux cercles bleu ou rouge) qui a le plus grand rayon. Ainsi, après avoir calculé la nouvelle commande longitudinale du suiveur, le contrôle latéral utilise la vitesse v dans le calcul de la nouvelle vitesse angulaire. v est obtenue selon l'expression suivante :

$$v = \max(v_0, v_0 + a.\tau) \quad (5.10)$$

5.4 Étude de la stabilité de *NOC*

Nous nous intéressons dans cette section à l'étude de la stabilité de l'algorithme *NOC*. En d'autres termes, nous souhaitons vérifier si un robot convergera toujours vers le chemin

de son prédécesseur, même dans des cas extrêmes. Nous constatons deux cas critiques où le robot peut diverger du chemin de son prédécesseur :

cas 1 : le robot n'est pas orienté dans la direction de son chemin visé ;

cas 2 : le robot est orienté vers le chemin à suivre mais son écart latéral par rapport à ce chemin n'est pas suffisant pour éviter de couper celui-ci.

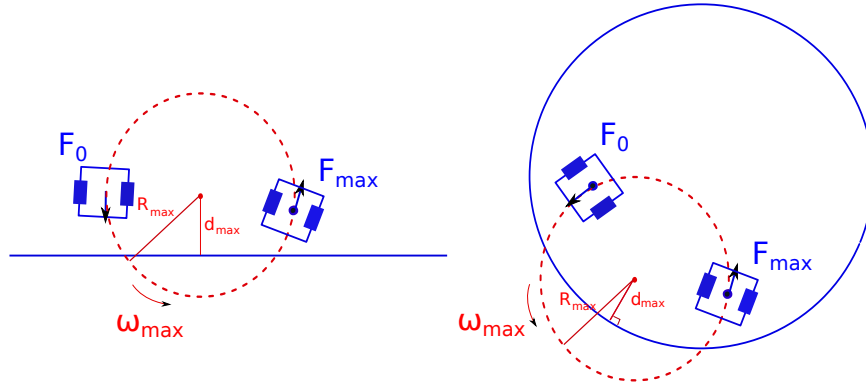


FIGURE 5.9 – Cas 2 : condition d'initialisation non vérifiée (cas d'une droite et d'un cercle).

Le premier cas peut se produire quand le robot suiveur change de chemin visé et ne se trouve pas orienté vers celui-ci. Si sa vitesse de rotation maximale ne lui permet pas de se réorienter correctement dans la direction du chemin de son prédécesseur, nous calculerons le nombre de pas de temps nécessaires pour corriger son orientation.

Le second cas peut se produire quand le robot change de chemin à suivre et se retrouve bien orienté vers le nouveau chemin à suivre mais pas suffisamment éloigné de ce dernier pour éviter de le couper. Ainsi, même en tournant à vitesse angulaire maximale, il n'arrive pas à éviter de couper celui-ci. Comme le montre la figure 5.9, si le robot n'est pas initialement bien positionné par rapport au chemin approché de son prédécesseur, la distance du centre du cercle formé en tournant à vitesse angulaire maximale à ce chemin est inférieure au rayon de ce cercle. Pour que la condition d'initialisation soit vérifiée, il faut que :

$$d_{\max} > R_{\max} \quad (5.11)$$

Dans ce cas, nous souhaitons étudier quand le robot, même en coupant une fois le chemin visé, arrivera à corriger son orientation et à suivre sans oscillations le chemin de son prédécesseur.

Étant donné que le chemin du prédécesseur peut être approché par une suite de segments de droites et d'arcs de cercles, l'étude de la stabilité revient donc à étudier la convergence du suiveur vers une droite et sa convergence vers un cercle. La convergence est atteinte quand l'erreur latérale du suiveur par rapport au chemin visé décroît : $|\frac{e_{y_{i+N}}}{e_{y_i}}| < 1$. Cette convergence, obtenue en un ou plusieurs pas de temps ($N \geq 1$), n'est garantie que si le chemin visé par le suiveur reste le même durant les N pas de temps. De plus, comme dans

la section 5.3, nous supposons que la vitesse longitudinale d'un robot suiveur à l'instant i est $v = \max(v_i, v_i + a.\tau)$ et reste constante pendant le pas de temps τ .

Dans la suite, nous étudions la convergence du robot vers le chemin estimé de son prédécesseur. Pour chaque type de chemin (droite ou cercle), nous calculons le nombre de pas de temps nécessaires au robot pour converger vers le chemin à suivre dans le cas où il n'est pas orienté vers celui-ci. Ensuite, dans le cas où le robot est orienté dans la direction du chemin à suivre, nous étudions son comportement lors de son suivi de ce chemin. Le robot ne vérifiant pas la condition d'initialisation 5.11, nous distinguons différents cas où le robot coupera une première fois le chemin estimé et ensuite, soit il arrivera à suivre sans oscillation ce chemin, soit il continuera à osciller autour.

5.4.1 Cas d'une droite

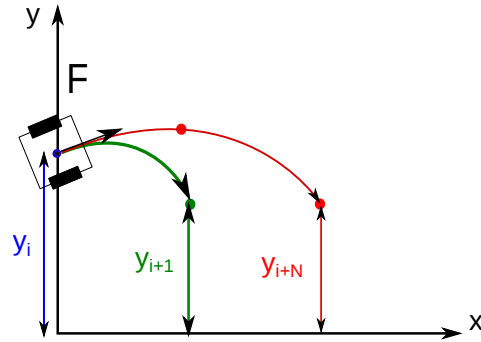


FIGURE 5.10 – Cas 1 : convergence vers un chemin rectiligne.

Cas 1 : Pour simplifier les calculs, nous considérons un repère orthogonal où le chemin linéaire du prédécesseur est confondu avec l'axe des x , et où le robot suiveur est situé sur les axes des y . Sa configuration à un instant i est alors $(0, y_i, \theta_i)$ (avec $0 \leq \theta_i \leq \frac{\pi}{2}$) représenté par la Figure 5.10. L'erreur latérale du suiveur par rapport au chemin visé est $e_y = y_i$. Le robot converge vers le chemin du prédécesseur si e_y diminue au cours du temps et tend vers 0 : $e_{y_{i+1}} < e_{y_i} \Leftrightarrow y_{i+1} < y_i$.

Selon les équations cinématiques de mouvement 3.3 d'un robot, l'évolution de l'erreur latérale après N pas de temps où on applique ω s'exprime sous la forme suivante :

$$y_{i+N} = y_i - \frac{v}{\omega_i} (\cos(N.\omega.\tau + \theta_i) - \cos \theta_i), \quad N > 0$$

En posant $\theta_{i+N} = N.\omega.\tau + \theta_i$ et supposant que $-\frac{\pi}{2} \leq \theta_{i+N} \leq \frac{\pi}{2}$, pour avoir $y_{i+1} < y_i$, il faut que :

$$\frac{v}{\omega_i} (\cos(N.\omega.\tau + \theta_i) - \cos \theta_i) \geq 0 \Rightarrow A = \frac{\cos(N.\omega.\tau + \theta_i) - \cos \theta_i}{\omega_i} \geq 0$$

Nous allons donc étudier le signe de A .

En considérant l'exemple illustré par la figure 5.10, le contrôleur latéral calcule une vitesse angulaire négative ω_i afin de réduire l'écart latéral. La nouvelle orientation du suiveur est dans ce cas $\theta_{i+1} \leq \theta_i$. Ceci contribue à corriger l'orientation du robot. Nous pouvons ainsi distinguer les cas suivants :

- si $|\theta_{i+1}| > |\theta_i|$, $\cos \theta_{i+1} < \cos \theta_i \Rightarrow A > 0$; ceci revient à dire que si la commande ω_i est suffisante pour ramener l'orientation du suiveur à $-\frac{\pi}{2} \leq \theta_{i+1} \leq -\theta_i$, l'erreur latérale par rapport au chemin du prédécesseur ($y = 0$) décroît ;
- si $|\theta_{i+1}| < |\theta_i|$, $\cos \theta_{i+1} > \cos \theta_i \Rightarrow A < 0$; dans ce cas, l'erreur latérale n'est pas réduite après un seul pas de temps ; le suiveur nécessite plusieurs pas de temps N pour corriger son orientation et converger vers le chemin du prédécesseur. Pendant ces N pas de temps, le suiveur tourne à vitesse angulaire maximale (ω_{min} quand il est au dessus de son chemin visé (Figure 5.10) et ω_{max} quand il en est en dessous) jusqu'à atteindre une orientation $\theta_{i+N} \leq -\theta_i$. Nous pouvons ainsi calculer le nombre N de pas de temps τ nécessaires pour bien se réorienter :

$$\begin{aligned}\theta_{i+N} &= \theta_i - N \cdot \omega_{max} \cdot \tau \leq -\theta_i \\ \text{cas limite : } \theta_i - N \cdot \omega_{max} \cdot \tau &= -\theta_i \\ \Rightarrow N \cdot \tau &= \frac{2\theta_i}{\omega_{max}}\end{aligned}$$

Cas 2 : Le robot, orienté dans la direction du chemin approché de son prédécesseur et dont la position initiale ne vérifie pas la condition d'initialisation (formule 5.11), ne trouve pas une commande lui permettant d'éviter de couper le chemin estimé de son prédécesseur. La commande choisie est dans ce cas la vitesse angulaire qui minimise la fonction d'erreur présentée dans la section 5.2.3. Quand le robot applique la nouvelle commande latérale calculée et passe de l'autre côté du chemin rectiligne de son prédécesseur, deux cas sont possibles selon sa nouvelle orientation et sa nouvelle position par rapport à ce chemin :

- soit le robot ne se trouve pas orienté dans la direction du chemin visé. Nous retrouvons donc le premier cas présenté dans le paragraphe précédent (voir figure 5.11);
- soit le robot se trouve orienté dans la direction du chemin visé et ne vérifie pas la condition d'initialisation (formule 5.11). Dans ce cas, comme le montre la figure 5.12-a, il va continuer à couper le chemin de son prédécesseur jusqu'à trouver une commande convenable. Pour éviter un mouvement oscillatoire, nous pouvons exiger à l'algorithme de choisir une vitesse angulaire qui mène le robot à s'éloigner du chemin de son prédécesseur dans un premier temps pour ensuite re-converger vers celui-ci sans oscillation, comme le montre la figure 5.12-b. Dans ce travail, nous avons choisi de ne pas adopter cette solution puisqu'elle crée une erreur latérale importante. Il faut noter que nous avons réalisé plusieurs expériences et nous n'avons pas été confronté à la situation présentée dans la figure 5.12-a.

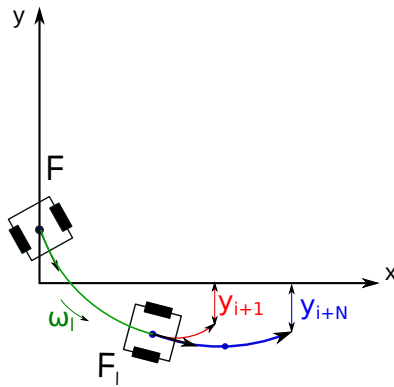


FIGURE 5.11 – Cas 2 : première possibilité de convergence vers un chemin rectiligne.

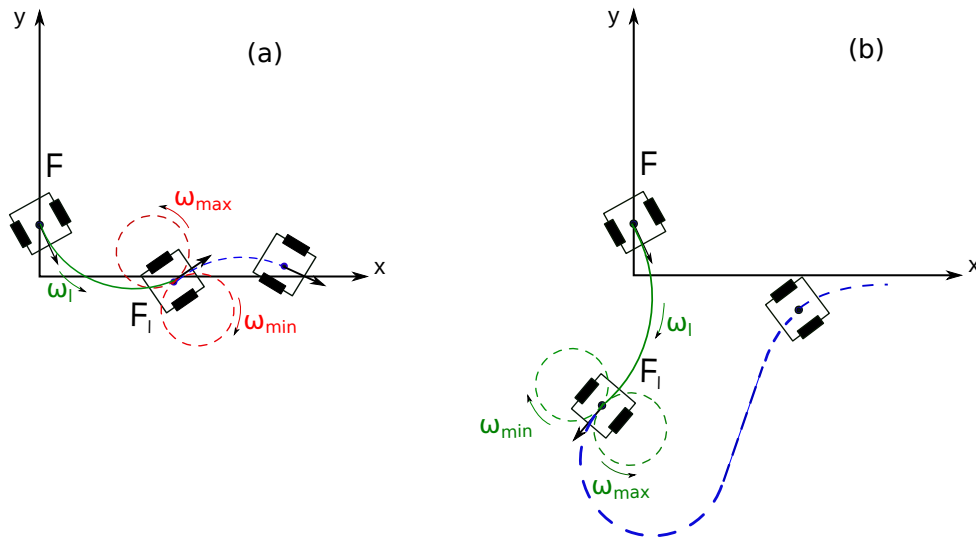


FIGURE 5.12 – Cas 2 : deuxième possibilité de convergence vers un chemin rectiligne.

5.4.2 Cas d'un cercle

Cas 1 : Dans ce cas, nous considérons que le chemin circulaire (\mathcal{C}) du robot prédécesseur est centré sur O (le centre du repère orthogonal), et que le robot suiveur est positionné sur l'axe des y avec une orientation comprise entre $\pm\frac{\pi}{2}$. La configuration du robot est noté $(0, y_i, \theta_i)$. L'erreur latérale e_y du suiveur par rapport à (\mathcal{C}) est le rayon R de (\mathcal{C}) moins la distance (en valeur absolue) du suiveur au centre O du cercle que nous noterons d : $e_y = |R - d|$. À l'instant i , l'erreur latérale e_y notée e_i est égale à $|R - |y_i||$, comme le montre la Figure 5.13.

En raisonnant comme dans le cas d'une droite, la convergence est obtenue quand l'erreur latérale par rapport au cercle (\mathcal{C}) diminue au cours du temps :

$$\begin{cases} e_{i+1} < e_i \Rightarrow d_{i+1} > d_i \\ x_{i+1}^2 + y_{i+1}^2 > y_i^2 \end{cases} \quad (5.12)$$

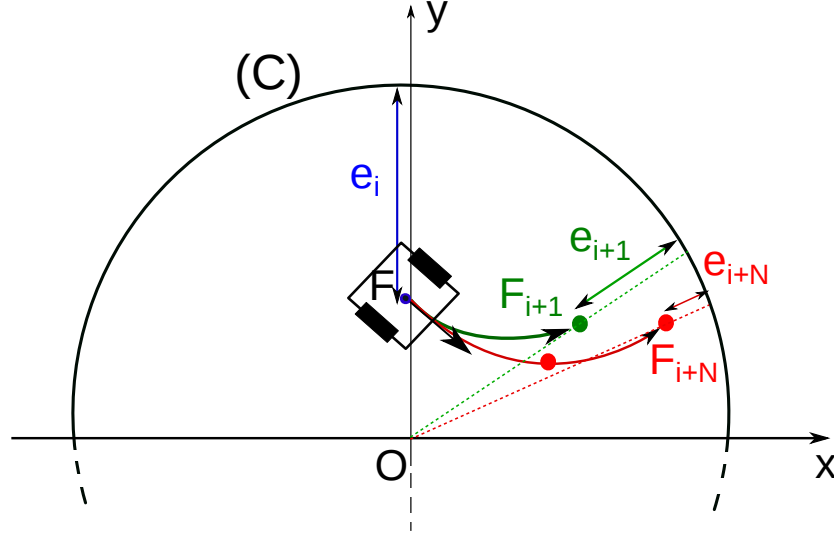


FIGURE 5.13 – Cas 1 : convergence vers un chemin circulaire.

En utilisant les équations cinématiques 3.3, la position du suiveur après N pas de temps est exprimée sous la forme suivante :

$$\begin{cases} x_{i+N} = x_i + \frac{v}{\omega_i} (\sin(N.\omega_i.\tau + \theta_i) - \sin \theta_i), & x_i = 0 \\ y_{i+N} = y_i - \frac{v}{\omega_i} (\cos(N.\omega_i.\tau + \theta_i) - \cos \theta_i), & N \geq 0 \end{cases} \quad (5.13)$$

Dans le cas où $N = 1$ et pour avoir $x_{i+1}^2 + y_{i+1}^2 > y_i^2$, il faut que :

$$2 \cdot \frac{v_0^2}{\omega_i^2} \cdot (1 - \cos(\omega_i \tau)) - 2 \cdot y_i \cdot \frac{v_0}{\omega_i} (\cos \theta_{i+1} - \cos \theta_i) > 0 \quad (5.14)$$

Cette égalité est toujours satisfaite sauf dans deux cas :

- si $\theta_i > 0$, le suiveur, tournant avec une vitesse angulaire négative $\omega_i < 0$, aura une nouvelle orientation $\theta_{i+1} < 0$ tel que $|\theta_{i+1}| > \theta_i$;
- si $\theta_i < 0$, le suiveur tourne avec une vitesse angulaire positive $\omega_i > 0$ mais sa nouvelle orientation $\theta_{i+1} < -\theta_i$;

Dans ces deux cas, un seul pas de temps n'est pas suffisant pour réduire l'écart latéral. Le suiveur tourne ainsi à vitesse angulaire maximale jusqu'à atteindre une configuration où son orientation θ_{i+N} est supérieure ou égale à $-\theta_i$. Le nombre N de pas de temps nécessaire pour réduire l'écart latéral est ainsi calculé selon l'équation suivante :

$$\begin{aligned} \theta_{i+N} &= \theta_i + N\omega^*\tau \geq -\theta_i \\ \text{cas limite : } \theta_i + N\omega^*\tau &= -\theta_i \\ \Rightarrow N\tau &= -\frac{2\theta_i}{\omega^*} \end{aligned}$$

ω^* est égale à $\pm\omega_{max}$ selon l'orientation du suiveur : si $\theta_i > 0$, le robot va tourner avec ω_{min} (premier cas) ; au contraire, si $\theta_i < 0$, le robot va tourner avec ω_{max} (deuxième cas).

Cas 2 : Le même raisonnement suivi dans le cas d'une droite est adopté dans le cas d'un chemin circulaire. La position initiale du robot ne vérifiant pas la condition d'initialisation (formule 5.11), celui-ci choisit une commande latérale qui minimise la fonction d'erreur présentée dans la section 5.2.3 sans lui permettant d'éviter de couper le chemin circulaire de son prédécesseur. Quand le robot applique sa commande latérale, il passe de l'autre côté du cercle. Pour étudier la stabilité du suivi, nous distinguons deux cas selon le signe de l'erreur angulaire du robot par rapport au chemin visé :

- soit le robot ne se trouve pas orienté dans la direction du chemin visé. Nous retrouvons donc le premier cas présenté dans le paragraphe précédent (voir figure 5.14);

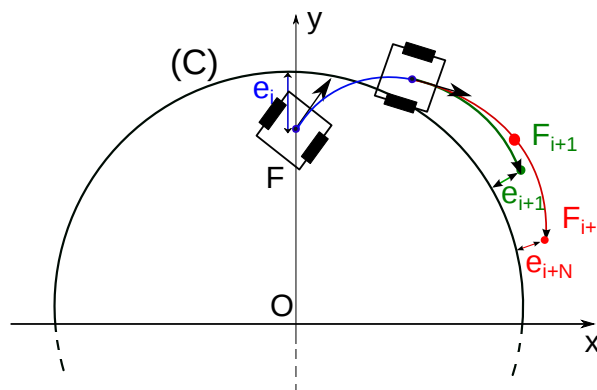


FIGURE 5.14 – Cas 2 : première possibilité de convergence vers un chemin circulaire.

- soit le robot se trouve orienté dans la direction du chemin visé et ne vérifie pas la condition d'initialisation (formule 5.11).

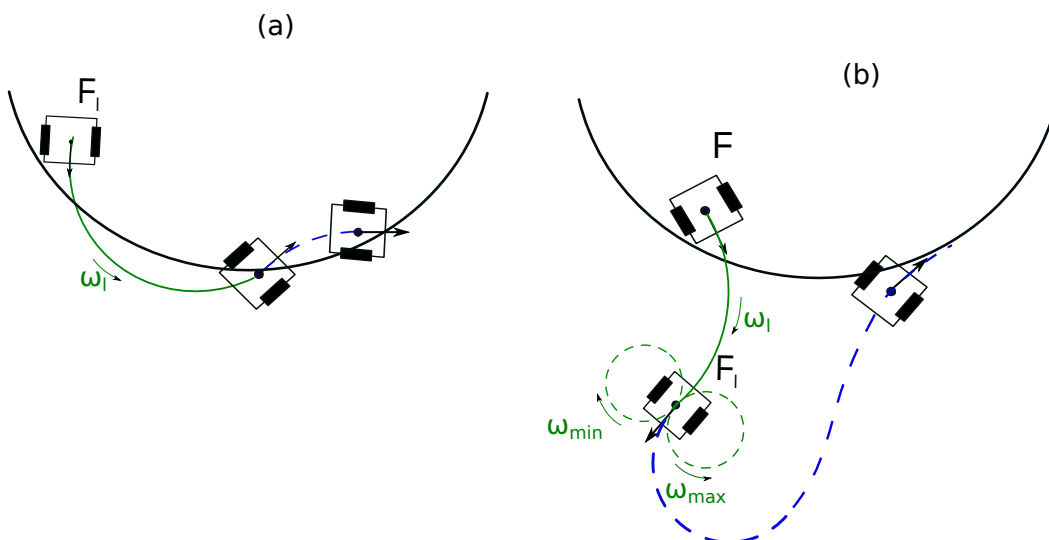


FIGURE 5.15 – Cas 2 : deuxième possibilité de convergence vers un chemin circulaire.

Dans ce cas, comme le montre la figure 5.15-a, il va continuer à couper le chemin de son prédécesseur jusqu'à trouver une commande convenable. Pour éviter un mouvement oscillatoire, nous pouvons exiger à l'algorithme de choisir une vitesse angulaire qui mène le robot à s'éloigner du chemin de son prédécesseur dans un premier temps pour ensuite re-converger vers celui-ci sans oscillation, comme le montre la figure 5.15-b. Comme dans le cas d'une droite, nous avons préféré de garantir un suivi précis.

Ainsi, nous n'adoptons pas dans ce travail la solution proposée dans la figure 5.15-b puisqu'elle génère une erreur latérale importante. En observant les simulations réalisées dans cette thèse, nous avons constaté que le robot n'a jamais subi une situation d'oscillation comme celle présentée dans la figure 5.15-b.

5.5 Validation expérimentale de l'algorithme *NOC*

Après avoir expliqué les différentes étapes de l'algorithme *NOC*, nous souhaitons valider celui-ci en réalisant plusieurs simulations. Ces expérimentations sont réalisées sur un modèle de robots dont la vitesse longitudinale maximale est de $v_{max} = 0,65$ m/s.

Pour cela, nous réalisons une expérimentation dans le but de comparer les performances des deux algorithmes de platooning *Memo-LAT* et *NOC* afin de vérifier l'amélioration apportée par *NOC* à un suivi pour lequel *Memo-LAT* ne permet pas d'avoir de bons résultats. Les deux algorithmes *Memo-LAT* et *NOC* sont implémentés sur un modèle de déplacement de robots à variation progressive qui permet de simuler un mouvement fidèle à leur déplacement réel.

5.5.1 Comparaison des deux algorithmes de platooning *Memo-LAT* et *NOC*

L'objectif de cette section est de valider l'algorithme *NOC* en montrant son efficacité en particulier pour des suivis en convoi où *Memo-LAT* n'affiche pas de bonnes performances.

Dans le chapitre précédent, nous avons vu que *Memo-LAT* nécessite un réglage de la distance d'établissement d_l en fonction de plusieurs facteurs (vitesse longitudinale, courbure du chemin à suivre, nombre de robots dans le convoi). Dans la simulation de la section 4.3.1.2, nous avons montré que le choix de la distance d'établissement d_l influe sur la stabilité du suivi des longs convois. Ainsi, pour rétablir la stabilité d'un suivi, d_l doit être ajustée.

Au début de ce chapitre, nous avons souligné que l'algorithme *NOC* a été conçu dans le but d'améliorer *Memo-LAT*. En plus de chercher à éviter les oscillations de par sa construction, un des avantages de *NOC* sur *Memo-LAT* est l'absence de paramètres à régler explicitement.

Pour réaliser cette comparaison entre les deux approches, nous réalisons un *platooning* d'un convoi de 9 robots le long d'un chemin spiral. Les robots sont initialement alignés

et distant de 0,2 m. La distance minimale permise entre deux robots consécutifs est de 0,1 m. La distance d'établissement pour *Memo-LAT* est de 17 mm. En ce qui concerne *NOC*, le domaine de variation de la vitesse angulaire $[\omega_{min}, \omega_{max}]$ est décomposé en 10 échantillons, et si un raffinement a lieu, le nouveau domaine de variation de la commande est à nouveau discrétisé en 10 échantillons.

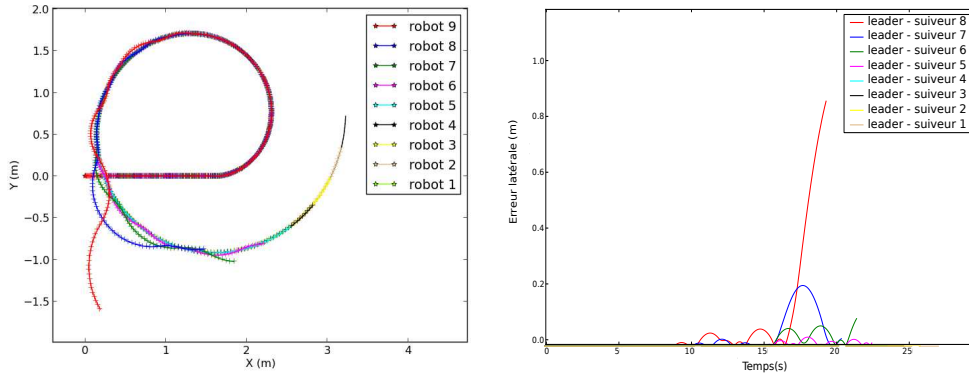


FIGURE 5.16 – À gauche : platooning de 9 robots sous *Memo-LAT* le long d'un chemin spiral, à droite : courbes d'erreur latérale entre les chemins du leader et chacun des suiveurs.

La figure 5.16 montre les chemins tracés par les différents robots soumis à *Memo-LAT* ainsi que les courbes d'erreur latérale entre le chemin du leader et chacun des suiveurs. Comme nous pouvons le remarquer, plus les robots avancent, plus l'erreur latérale s'accumule menant ainsi les robots à osciller autour du chemin du leader. Le suivi n'est alors pas stable. Ce mouvement oscillatoire est observé sur les courbes d'erreur latérale qui montre l'augmentation progressive de l'écart latéral jusqu'à 1,1 m.

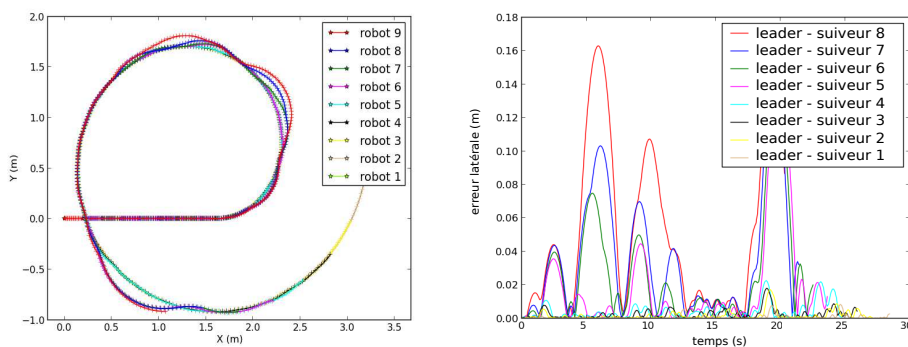


FIGURE 5.17 – À gauche : platooning sous *NOC* de 9 robots le long d'un chemin spiral, à droite : courbes d'erreur latérale entre les chemins du leader et chacun des suiveurs.

Comme nous pouvons l'observer sur la figure 5.17, *NOC* permet d'avoir un suivi plus stable des longs convois de robots, puisque le convoi ne diverge pas, sans avoir besoin de régler des paramètres. L'atténuation du comportement oscillatoire observé sous *Memo-LAT* et la réduction de l'erreur latérale de 1,1 m (figure 5.16) jusqu'à 0,17 m sont obtenues

grâce au principe de *NOC* : au lieu de suivre une suite de points non orientés, comme c'est le cas dans *Memo-LAT*, *NOC* reconstruit le chemin du robot prédécesseur et calcule la commande latérale qui permet au robot suiveur de suivre ce chemin en évitant les oscillations.

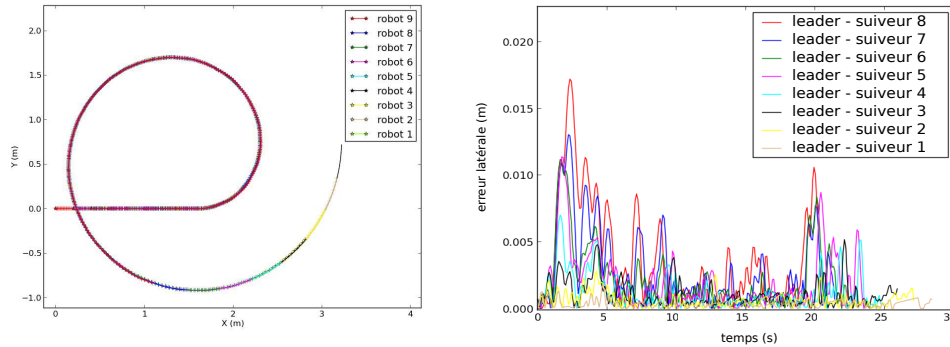


FIGURE 5.18 – À gauche : platooning sous *NOC* de 9 robots se déplaçant selon un modèle à variation instantanée le long d'un chemin spiral, à droite : courbes d'erreur latérale entre les chemins du leader et chacun des suiveurs.

5.5.2 Influence du modèle de déplacement des robots sur *NOC*

Malgré l'amélioration du suivi observée en comparant les figures 5.16 et 5.17, *NOC* montre des performances moins bonnes que celles à quoi nous nous attendons. Ceci est dû au fait que *NOC* utilise un modèle de déplacement à variation instantanée dans le calcul de la commande latérale.

Comme nous pouvons l'observer sur la figure 5.18, le suivi pour le modèle à variation instantanée est stable et précis¹¹, l'écart latéral maximal étant inférieur 17 mm. Une

11. L'observation de la courbe d'erreur du suivi obtenu avec *NOC* sur la figure 5.18 montre plusieurs pics, même si l'erreur latérale maximale reste très faible (< 17 mm). Ces pics d'erreur sont dus à deux raisons principales :

- d'une part, le contrôle calculé est discret et la commande est choisie de l'ensemble des vitesses angulaires $\{\omega_{min}, \dots, \omega_{max}\}$. Cette commande est la meilleure sur l'ensemble des commandes possibles. Cependant, elle ne représente pas forcément la valeur optimale de vitesse angulaire, sur le domaine continu des vitesses angulaires, que le robot pourrait appliquer pour avoir un suivi plus doux (moins de pics), cette vitesse angulaire optimale ne figurant pas nécessairement dans la liste des commandes possibles à cause de la discrétisation de son domaine de variation.
- d'autre part, le robot reconstruit par morceaux le chemin de son prédécesseur et cherche à suivre durant son déplacement ces morceaux chemins reconstruits. En changeant de chemins à suivre, le robot peut se retrouver dans un état (position, orientation) par rapport au nouveau chemin pour lequel la condition d'initialisation (eq. 5.11) n'est pas vérifiée, ce qui génère une intersection entre le robot et ce chemin.

fois implémenté sur un modèle de déplacement à variation progressive (figure 5.17) qui prend en compte les accélérations angulaires, les performances de l'algorithme sont moins bonnes : un mouvement oscillatoire apparaît lors du suivi menant à une erreur latérale maximale de 170 mm. Ce mouvement oscillatoire est dû au retard correspondant au temps nécessaire au robot pour passer de sa vitesse angulaire courante à la vitesse angulaire désirée. Ce retard affecte le calcul de la commande latérale de deux façons :

1. Les conditions Γ_i , Γ_i^+ ainsi que la fonction d'optimisation \mathcal{E} considèrent que le robot, se déplaçant à vitesse longitudinale v_0 constante, génère un arc de cercle pour un ω_i donné. Or, ce n'est pas le cas quand il s'agit du modèle de mouvement à variation progressive. Pour que le robot puisse tourner à ω_i , il faudrait que sa vitesse angulaire passe instantanément de sa valeur courante à la valeur désirée ω_i . Le robot ne trace alors pas un arc de cercle, mais un arc de clothoïde qui peut éventuellement être suivi d'un arc de cercle si le pas de temps τ est suffisamment grand. Ainsi, le déplacement simulé du robot est différent de la prévision de *NOC*. Les conditions Γ_i , Γ_i^+ sont alors potentiellement mal évaluées et l'optimisation de la fonction d'erreur peut être erronée.
2. La combinaison des contrôleurs longitudinal et latéral, comme expliquée dans la section 5.3, ne convient pas au modèle de déplacement des robots à variation progressive. Dans le cas d'un modèle à variation instantanée de la vitesse angulaire, cette combinaison considère que, durant un pas de temps, le robot évolue le long d'un arc de spirale (vitesse angulaire constante et vitesse longitudinale uniformément variée) qui peut être borné entre deux cercles. En revanche, dans le cas d'un modèle à variation progressive, le robot ne génère plus un arc de spirale puisque sa vitesse angulaire n'est pas constante durant ce pas de temps. Puisque le chemin du robot ne peut plus être borné par deux cercles, la combinaison des deux contrôleurs ne garantit plus l'absence d'intersections du robot avec le chemin du prédécesseur.

Comme nous l'avons déjà indiqué, *NOC* considère implicitement un modèle de déplacement des robots dans le calcul des consignes. Lorsque ce modèle de déplacement (ici, modèle à variation instantanée) ne correspond pas au modèle de déplacement effectif des robots (modèle à variation progressive), nous constatons une dégradation notable des performances. Nous cherchons alors à adapter *NOC* aux modèles de déplacement des robots afin de garder de bonnes performances. La généralisation de l'algorithme implique d'abstraire *NOC* par rapport au modèle de déplacement des robots. Pour cela, nous avons choisi de faire appel à une fonction simulant précisément le mouvement réel des robots (plutôt que de le résoudre analytiquement) et utilisons sa sortie dans le calcul de la commande latérale.

5.6 *Abstract NOC* : Abstraction de l'algorithme *NOC* par rapport au modèle de mouvement

Comme nous avons expliqué dans la section 5.2, l'algorithme *NOC*, tel qu'il est conçu, est fondé sur le modèle de déplacement des robots à variation instantanée : les conditions qui permettent d'éliminer les vitesses angulaires de l'ensemble $\{\omega_{min}, \dots, \omega_{max}\}$ induisant des comportements non désirés testent si des intersections d'arcs de cercles et de segments de droites avec le chemin du prédécesseur peuvent apparaître. De plus, la combinaison du contrôle longitudinal avec le contrôle latéral considère que le robot, dont la vitesse longitudinale varie uniformément et la vitesse angulaire reste constante durant un pas de temps, trace des arcs de spirale.

Dans l'expérimentation de la section 5.5.2, nous avons montré que les performances de *NOC* sont moins bonnes quand le modèle de mouvement du robot change : les conditions à tester et la fonction d'optimisation utilisées pour calculer la commande latérale ne sont plus conformes au déplacement effectué par le robot.

Pour cela, nous choisissons d'abstraire *NOC* par rapport au modèle de mouvement du robot et nous obtenons *Abstract NOC*. Au lieu d'intégrer le mouvement du robot dans les différentes étapes de l'algorithme, nous couplerons *Abstract NOC* à une fonction externe qui simule le mouvement. Ainsi, *Abstract NOC* utilisera le déplacement simulé pour calculer la commande latérale. Cette abstraction permettra à l'algorithme *NOC* d'être utilisé avec n'importe quel modèle de mouvement. De plus, contrairement à *Memo-LAT* et la première version de *NOC*, aucun découplage des contrôleurs longitudinal et latéral est nécessaire. En effet, la commande longitudinale a_d sera prise en compte dans la simulation du mouvement, et par la suite, dans le processus de calcul de la vitesse angulaire. Dans la suite de ce chapitre, la fonction externe appelée par *Abstract NOC* simule le modèle de mouvement à variation progressive présenté dans la section 3.2. Nous supposons que les vitesses longitudinale et angulaire courantes du robot sont représentées par v_0 et ω_0 et les vitesses désirées par $v_d = v_0 + a_d \cdot \tau$ et ω_d (le contrôleur longitudinal calcule la commande a_d et le contrôleur latéral calcule la commande ω_d).

Même si le principe de *NOC* reste le même, quelques modifications sont à apporter pour obtenir l'algorithme *Abstract NOC*. Ces modifications, qui touchent principalement aux conditions Γ et Γ^+ et à la fonction d'optimisation, sont présentées par la suite.

5.6.1 Estimation de la courbure

La deuxième étape du choix de la position cible présentée dans la section 5.1 consiste à tester si le robot pourrait éviter de couper le chemin de son prédécesseur en tournant à $\pm\omega_{max}$ et en gardant une vitesse longitudinale constante v_0 , ce qui revenait à évaluer la condition Γ^+ , conçue de façon à être toujours testée à vitesse longitudinale constante. En considérant un modèle de déplacement à variation progressive, ce test n'est plus valide puisque le robot ne génère plus des cercles en tournant à $\pm\omega_{max}$. Sachant que sa vitesse angulaire courante est ω_0 et qu'il ne peut pas atteindre instantanément sa vitesse angulaire

maximale, le robot trace un arc de clothoïde (courbe générée par un robot se déplaçant à vitesse longitudinale constante et une vitesse angulaire qui varie linéairement) pour atteindre ω_{max} ou $-\omega_{max}$. Ensuite, il continue à tourner à ω_{max} (ou $-\omega_{max}$) tout en générant un cercle de rayon v_0/ω_{max} (figure 5.19). Si au moins un des deux chemins générés par le robot ne coupe pas (Δ) ou (\mathcal{C}), le suiveur a trouvé sa cible. Sinon, il teste le point suivant.

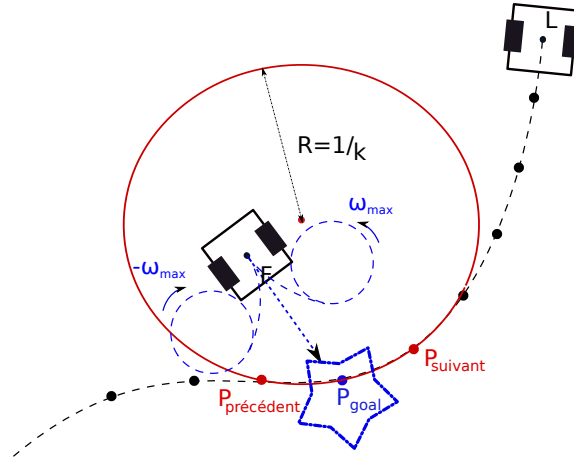


FIGURE 5.19 – Choix de la nouvelle cible selon *NOC* évolué.

5.6.2 Calcul de la commande latérale

Pour calculer le comportement latéral du robot, l'algorithme *NOC* parcourt l'ensemble des vitesses angulaires possibles afin de trouver la meilleure commande. Pour chaque ω_i de l'ensemble $\{\omega_{min}, \dots, \omega_{max}\}$, il teste les conditions Γ_i , Γ_i^+ et évalue la fonction d'erreur. La première étape de cet algorithme présenté dans la section 5.2 consiste à calculer l'équation de la droite ou du cercle que le robot, ayant un modèle de déplacement à variation instantanée, trace en se déplaçant à une vitesse longitudinale constante v_0 et une vitesse angulaire ω_i nulle ou non nulle.

En passant à un modèle à variation progressive (pour rappel, voir section 3.2.2), cette première étape calcule le chemin tracé par le suiveur soumis aux commandes v_d et ω_i . Ce chemin, qui n'est pas obligatoirement une droite ou bien un cercle mais une courbe dont la courbure ($\kappa = \omega/v$) est variable, est généré en passant des vitesses courantes (v_0 et ω_0) du robot aux vitesses désirées (v_d et ω_i). Comme nous avons déjà mentionné dans la section 3.2, la courbe tracée par le robot peut être représentée par une suite de points $\mathcal{P} : \{p_1, p_2, \dots, p_N\}$. Cette suite de points permet de nous donner un déplacement précis du robot durant un pas de temps τ .

Les conditions Γ_i , Γ_i^+ et la fonction d'erreur seront évaluées comme expliqué dans les paragraphes suivants.

5.6.2.1 Condition Γ_i

Cette condition permet de vérifier que le robot suiveur, soumis aux commandes $\{v_d, \omega_i\}$, peut se déplacer durant un pas de temps τ sans couper le chemin estimé de son prédécesseur. En appliquant ces commandes pendant τ , le suiveur passe de sa configuration courante $q_0(x_0, y_0, \theta_0, v_0, \omega_0)$ à la configuration $q_i(x_i, y_i, \theta_i, v_d, \omega_i)$.

Pour que la condition Γ_i soit satisfaite, la courbe (q_0q_i) et le chemin approximé (Δ) ou (\mathcal{C}) représentés dans la figure 5.20 ne doivent pas se couper. Pour évaluer Γ_i , au lieu de tester si l'arc de cercle $\widehat{q_0q_i}$ et le chemin approximé du prédécesseur se coupaient comme dans le cas de *NOC*, nous formons un segment avec chaque couple de points consécutifs de la liste $\mathcal{P} : \{p_1, p_2, \dots, p_N\}$ et nous vérifions si ce segment coupe (Δ) ou (\mathcal{C}) . Dans le cas où aucune intersection a lieu entre les différents segments $[p_i p_{i+1}]$ et le chemin approché du prédécesseur (Δ) ou (\mathcal{C}) , la condition Γ_i est vérifiée.

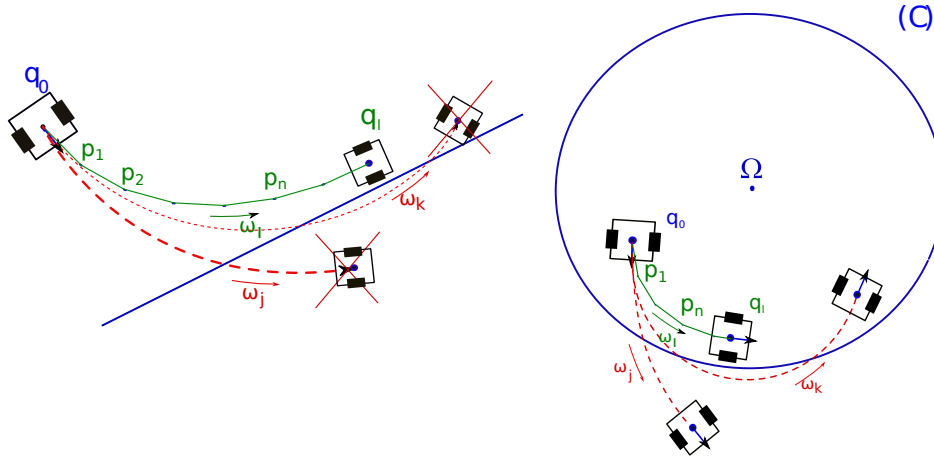
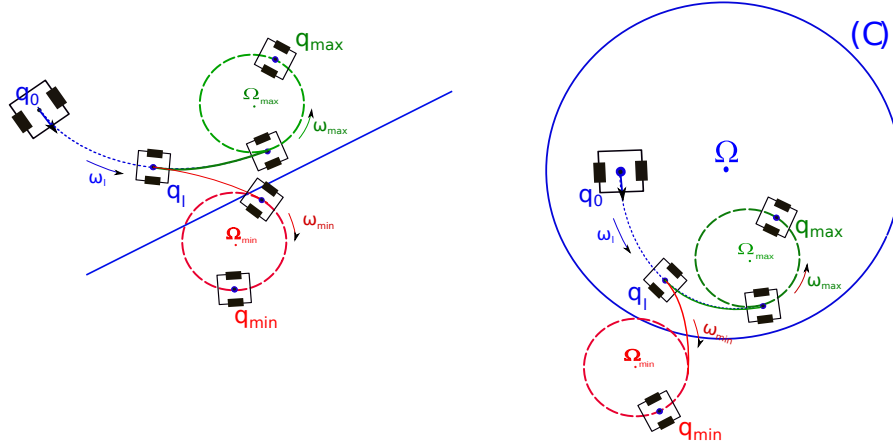


FIGURE 5.20 – Test de Γ_i dans le cas d'une droite et d'un cercle.

5.6.2.2 Condition Γ_i^+

La condition Γ_i^+ , étant satisfaite, permet de garantir l'absence d'oscillation du robot autour du chemin estimé (Δ) ou (\mathcal{C}) de son prédécesseur dans le futur. Cette condition utilise la nouvelle configuration q_i du robot, obtenue en appliquant ω_i sur la configuration q_0 (figure 5.21), et considère que le robot se déplace à une vitesse longitudinale constante. Pour évaluer cette condition, le suiveur vérifie qu'à partir de sa configuration q_i , il pourra éviter toute intersection avec le chemin approché en tournant à vitesse angulaire maximale au moins dans un des deux sens : le sens trigonométrique ou bien le sens horaire. À partir de q_i , nous traçons les deux trajectoires que le robot peut générer en tournant à fond : une première trajectoire est obtenue en tournant à ω_{max} , et une deuxième trajectoire est obtenue en tournant à $\omega_{min}(= -\omega_{max})$.

Pour tourner à fond, le robot ne peut pas changer instantanément de vitesse angulaire. Pendant ce temps de transition (passage de ω_i à ω_{max} (respectivement ω_{min})), il décrit


 FIGURE 5.21 – Test de Γ_i^+ dans le cas d'une droite et d'un cercle.

un arc de clothoïde. Ensuite, une fois qu'il a atteint ω_{max} (ou ω_{min}), le robot continue à tourner et décrit un cercle de centre Ω_{max} (ou Ω_{min}).

La condition Γ_i^+ est vraie quand au moins une de ces deux trajectoires n'intersecte pas (Δ) ou (\mathcal{C}) . Pour évaluer cette condition, nous vérifions tout d'abord si l'arc de clothoïde coupe (Δ) ou (\mathcal{C}) . Pour faire ceci, nous récupérons la liste des points générés par le robot en passant de ω_i à ω_{max} (ou ω_{min}), nous formons les segments de droite en reliant chaque couple de points successifs et nous testons si ces segments coupent (Δ) ou (\mathcal{C}) . Si aucune intersection a lieu, nous testons si le cercle coupe (Δ) ou (\mathcal{C}) : si la distance du centre Ω_{max} (ou Ω_{min}) au chemin approché est supérieure au rayon du cercle $\frac{v^*}{\omega_{max}}$, la condition Γ_i^+ est satisfaite.

Comme le montre la figure 5.21, la vitesse angulaire ω_l peut être une éventuelle commande latérale puisqu'elle vérifie les deux conditions Γ_i et Γ_i^+ : d'une part, l'arc $\widehat{q_0q_l}$ ne coupe ni (Δ) ni (\mathcal{C}) . D'autre part, en partant de la configuration q_l , le suiveur peut éviter le chemin estimé de son prédécesseur en tournant à $+\omega_{max}$.

5.6.2.3 Conception de la fonction d'optimisation

Le principe de calcul de la fonction d'optimisation \mathcal{E} reste le même : ayant une configuration initiale $q_0(x_0, y_0, \theta_0, v_0, \omega_0)$, le robot calcule pour une vitesse angulaire ω_i donnée la configuration $q_i(x_i, y_i, \theta_i, v_i, \omega_i)$ obtenue en appliquant les commandes v_d et ω_i . Ensuite, en partant de q_i , il cherche à annuler son erreur angulaire par rapport au chemin du leader en tournant à vitesse angulaire maximale. Une fois que le robot atteint la configuration q_i^+ (figure 5.22), son erreur angulaire s'annule et la valeur de la fonction d'optimisation \mathcal{E} est égale à l'écart latéral du robot par rapport au chemin approché de son prédécesseur.

Suite au changement du modèle de déplacement du robot, le calcul de q_i^+ n'est plus le même. Pour corriger son orientation, le robot trace un arc de clothoïde $\widehat{q_iq'_i}$ lui permettant de passer de sa vitesse angulaire courante ω_i à la vitesse angulaire maximale. Connaissant

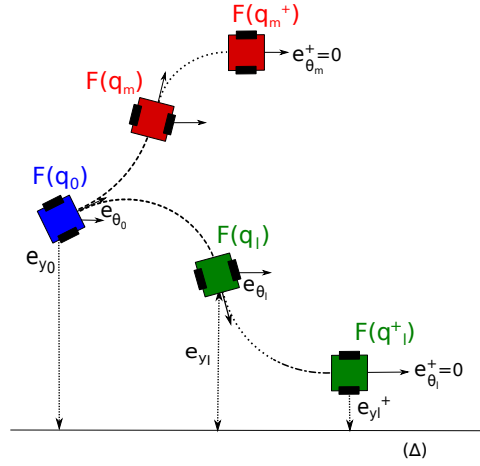
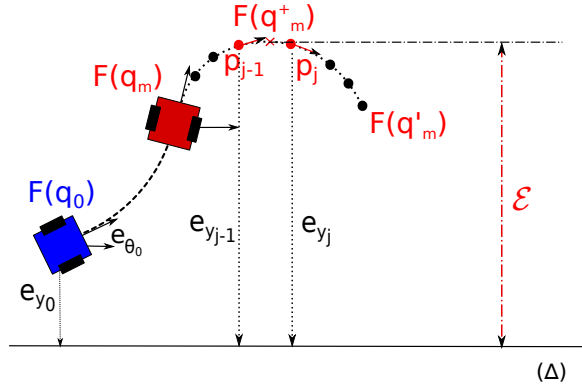

 FIGURE 5.22 – Test de Γ_i^+ dans le cas d'une droite et d'un cercle.


FIGURE 5.23 – Erreur angulaire nulle en un point de l'arc de clothoïde.

le signe de l'erreur angulaire à q_i , et ayant la liste des points orientés $\mathcal{P} : \{p_1, p_2, \dots, p_N\}$ qui forment la clothoïde, nous commençons par tester si l'erreur angulaire e_j correspondant au point p_j ($j = 1, \dots, N$) a changé de signe. Si oui, nous déduisons que le robot a changé d'orientation vers le chemin de son prédécesseur durant son passage de p_{j-1} à p_j (figure 5.23). En d'autres termes, q_i^+ se trouve entre p_{j-1} et p_j . Une approximation de \mathcal{E} est ainsi égale à l'écart latéral moyen des écarts latéraux $e_{y_{j-1}}$ et e_{y_j} du robot par rapport au chemin (Δ) du prédécesseur aux points p_{j-1} et p_j .

Si l'erreur angulaire ne s'annule pas le long de l'arc de clothoïde $\widehat{q_i q'_i}$, le robot continue à tourner avec sa vitesse angulaire maximale jusqu'à atteindre une position q_i^+ où l'erreur angulaire est nulle (figure 5.24). Il génère dans ce cas un arc de cercle. Le calcul de \mathcal{E} se fait ainsi de la même façon que dans 5.2.3 : nous déterminons le centre du cercle en utilisant la configuration q'_i selon les équations suivantes :

$$\begin{cases} x_c = x'_i - \frac{e_{\theta_i}}{|e_{\theta_i}|} \cdot \frac{v_i}{\omega_{max}} \cdot \sin \theta'_i \\ y_c = y'_i + \frac{e_{\theta_i}}{|e_{\theta_i}|} \cdot \frac{v_i}{\omega_{max}} \cdot \cos \theta'_i \end{cases} \quad (5.15)$$

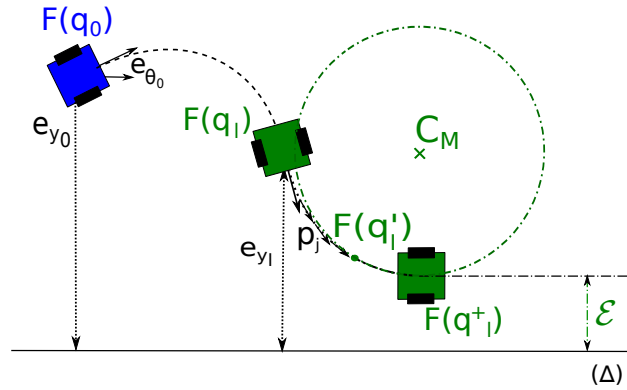


FIGURE 5.24 – Erreur angulaire nulle en un point de l’arc de cercle.

Ensuite, en suivant le même raisonnement que dans 5.2.3, la fonction d’optimisation est égale à :

$$\mathcal{E} = \left| \text{distance}(C_M \text{ ou } m, \text{ chemin visé}) - \frac{e_{\theta_i}}{|e_{\theta_i}|} \cdot \frac{v_i}{\omega_{max}} \right|$$

En observant la figure 5.22, la vitesse angulaire ω_l représente une meilleure commande puisque l’écart latéral en $F(q_l^+)$ est inférieur à celui en $F(q_m^+)$.

5.7 Validation expérimentale de *Abstract NOC*

Après avoir abstrait l’algorithme *NOC* par rapport au modèle de mouvement, nous réalisons des simulations pour valider *Abstract NOC*. Nous commençons par tester *Abstract NOC* sur un modèle de déplacement des robots à variation progressive (présenté dans la section 3.2.2). Ensuite, nous réalisons une expérimentation pour montrer la convergence d’un robot vers le chemin rectiligne ou circulaire de son prédécesseur. Nous discutons de l’influence du contrôle longitudinal sur le comportement latéral du robot suiveur. Pour cela, nous réalisons un suivi de deux robots en faisant varier la vitesse longitudinale du robot leader et nous étudions le comportement du robot suiveur. De plus, nous étudions la stabilité d’un suivi de deux robots qui se déplacent à vitesse longitudinale maximale et où le leader tourne avec une vitesse angulaire qui passe de $+\omega_{max}$ à $-\omega_{max}$ et réciproquement. Nous étudions alors l’influence de la variation de la vitesse angulaire sur le comportement des robots.

5.7.1 *Abstract NOC* appliqué sur un modèle de déplacement des robots à variation progressive

Dans une simulation précédente (section 5.5.1), nous avons testé *NOC* sur un modèle de déplacement des robots à variation progressive et nous avons remarqué que les per-

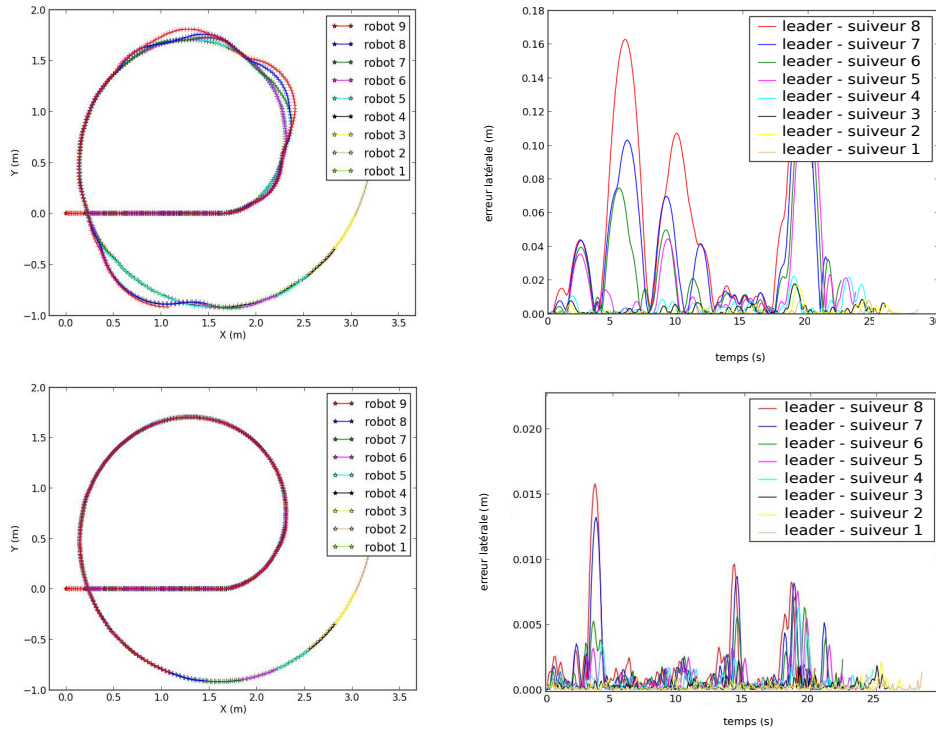


FIGURE 5.25 – Comparaison des performances de *NOC* (2 figures en haut) et *Abstract NOC* (2 figures en bas) implémentés sur un modèle de mouvement à variation progressive.

performances de l’algorithme sont considérablement affectées : le suivi perd énormément en termes de précision et de stabilité.

Nous reprenons cette même simulation et nous réalisons un suivi selon *Abstract NOC* avec des robots dont le modèle de mouvement est à variation progressive. La figure 5.25 montre une validation de *Abstract NOC*. Comme nous pouvons l’observer, le suivi des 9 robots le long de la spirale est précis et stable : l’erreur latérale maximale est de 15 mm et le mouvement oscillatoire des robots a disparu.

5.7.2 Validation expérimentale de la preuve de stabilité

Dans la section 5.4, nous avons fait la preuve de stabilité de l’algorithme *NOC*. Dans le but de vérifier si cette preuve reste valide dans le cas de l’algorithme *Abstract NOC*, nous considérons un convoi de 2 robots initialement au repos. Le leader se déplace avec une accélération longitudinale égale à $0,5 \text{ m/s}^2$ et une vitesse angulaire nulle, il décrit ainsi un chemin rectiligne. Le robot suiveur, n’étant pas initialement sur le chemin de son leader, converge vers ce chemin et le suit avec une erreur latérale nulle comme le montre la figure 5.26.

Pour vérifier la stabilité de l’algorithme *Abstract NOC* dans le cas d’un chemin circulaire, nous considérons un leader se déplaçant à vitesse longitudinale constante $v_{max}/2$ et à

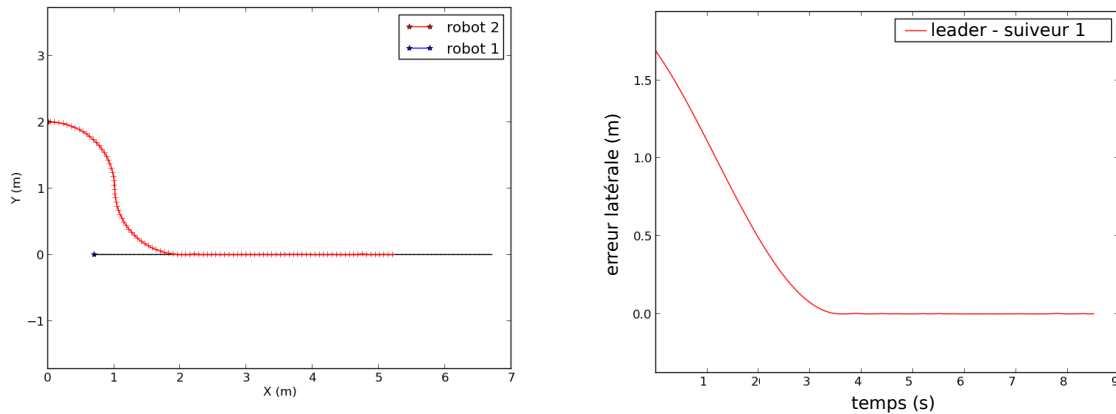


FIGURE 5.26 – Convergence d’un robot vers le chemin rectiligne de son prédécesseur ($y=0$). À droite : chemins des 2 robots, à gauche : courbe d’erreur latérale entre les 2 robots.

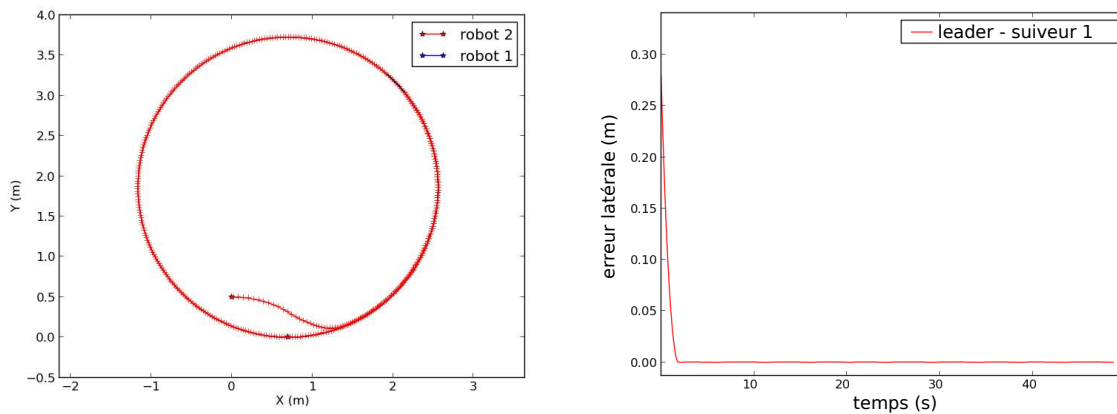


FIGURE 5.27 – Convergence d’un robot vers le chemin circulaire de son prédécesseur.

vitesse angulaire constante égale à $\pi/18$ rad/s. Le suiveur, initialement au repos, converge vers le chemin de son leader en réduisant sa déviation latérale et son erreur angulaire. Une fois que le robot atteint le chemin de son prédécesseur, il le suit avec précision, comme nous pouvons constater sur la figure 5.27.

5.7.3 Vitesse longitudinale du prédécesseur et comportement de son suiveur

Nous souhaitons dans cette expérimentation étudier l’influence du contrôle longitudinal sur le comportement latéral. Pour cela, nous faisons varier la vitesse longitudinale du robot prédécesseur et observons l’impact de cette variation sur le comportement de son suiveur.

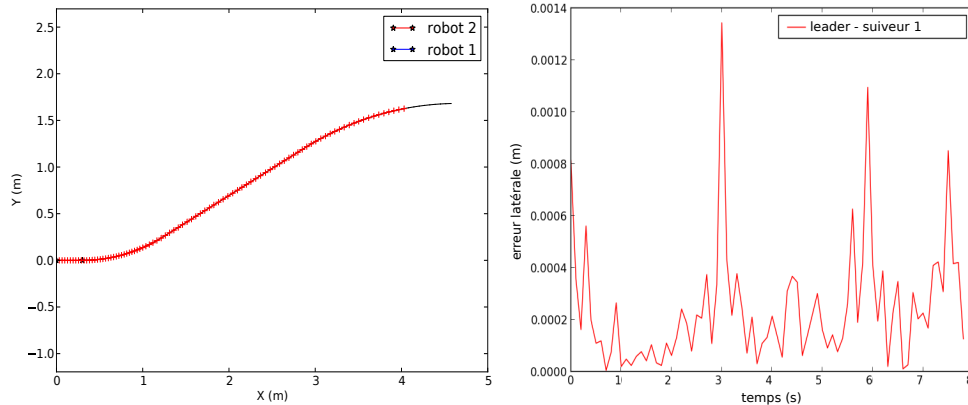


FIGURE 5.28 – à gauche : suivi en convoi de 2 robots, à droite : courbe d’erreur latérale entre les chemins des 2 robots.

Nous considérons deux robots initialement distants de 0,3 m se déplaçant à vitesse longitudinale constante $v_{max}/2 = 0,325$ m/s. Durant le suivi, le leader génère le chemin suivant :

- un arc de cercle en se déplaçant à des vitesses longitudinale et angulaire constantes pendant 3 secondes ($v = v_{max}/2 = 0,325$ m/s, $\omega = \pi/18$ rad/s) ;
- un segment de droite en se déplaçant avec une accélération longitudinale égale à 0,5 m/s² pendant 3 secondes ;
- un arc de cercle en se déplaçant à des vitesses longitudinale et angulaire constantes pendant 3 secondes ($v = v_{max} = 0,65$ m/s, $\omega = -\pi/18$ rad/s).

Au départ, le robot suiveur se déplace sur la droite joignant sa position initiale avec celle de son prédécesseur. Ensuite, il suit le chemin circulaire avec une erreur latérale maximale de 0,8 mm. Une fois le leader commence à accélérer, le robot suiveur tend aussi à augmenter sa vitesse longitudinale dans le but de réduire l’inter-distance avec son prédécesseur.

Quand le suiveur atteint le chemin rectiligne, son erreur angulaire non nulle induit une déviation latérale représentée par le pic sur la courbe d’erreur (figure à droite de 5.28) à 3 secondes. Cette déviation, d’une valeur de 1,4 mm, reste cependant minime. Le robot cherche alors à réduire cette déviation en calculant des valeurs convenables de vitesses angulaires.

Enfin, le robot suiveur s’approche du dernier morceau circulaire du chemin de son prédécesseur avec une vitesse longitudinale plus élevée. Il suit ce chemin avec précision où la déviation latérale maximale atteint 1,1 mm.

Le calcul de la commande latérale prend en compte la vitesse longitudinale du robot. En effet, il utilise une approximation du mouvement du robot suiveur qui se déplace avec

une accélération longitudinale au lieu de borner ce mouvement par un cercle (comme la section 5.3 l'explique). Ceci permet d'élargir l'ensemble des commandes ω_i qui n'induisent pas d'intersections (qui vérifient les conditions Γ_i et Γ_i^+) et qui peuvent mener à un suivi plus précis.

5.7.4 Vitesse angulaire du prédécesseur et comportement de son suiveur

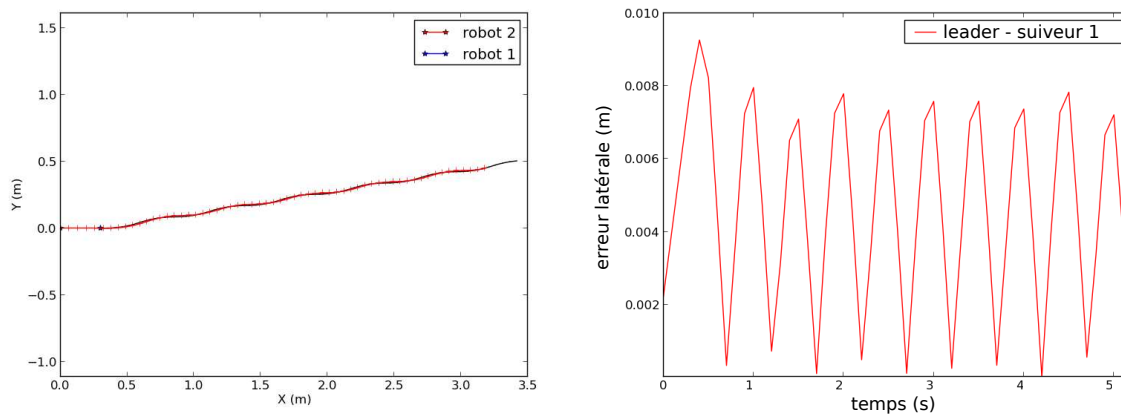


FIGURE 5.29 – à gauche : suivi en convoi de 2 robots, à droite : courbe d'erreur latérale entre les chemins des 2 robots.

Nous considérons un robot qui se déplace avec une vitesse longitudinale constante égale à la vitesse maximale v_{max} . Sa vitesse angulaire passe de ω_{max} à $-\omega_{max}$, puis réciproquement, les changements étant espacés de 0,5 s. Un robot suiveur cherche à suivre le chemin de son prédécesseur en se déplaçant à la même vitesse longitudinale v_{max} . Les deux robots sont initialement distant de 0,3 m. L'inter-distance minimale permise entre deux robots consécutifs est de 0,1 m.

La figure 5.29 montre un suivi précis (erreur latérale inférieure à 9 mm) et stable le long d'un chemin où les robots se déplacent à vitesse longitudinale maximale et où la vitesse angulaire du leader oscille entre $\pm\omega_{max}$. Le retard dû au changement brutal de la courbure quand le suiveur atteint la position initiale de son prédécesseur mène le robot à dévier légèrement le long des arcs tracés par le leader à vitesse angulaire positive et à compenser ce retard et réduire cette déviation latérale le long des arcs tracés par le leader à vitesse angulaire négative. Ceci explique la légère atténuation des pics observés sur la courbe d'erreur (figure à droite de la figure 5.29).

5.8 Conclusion

Dans ce chapitre, nous avons présenté deux algorithmes de platooning *NOC* et *Abstract NOC* pour réaliser des suivis de convois de robots. Comme dans le cas de l'algorithme

Memo-LAT présenté dans le chapitre 4, ces deux algorithmes, utilisant une approche décentralisée locale, s'appuient sur la mémorisation du chemin du prédécesseur. Ce chapitre améliore l'algorithme *Memo-LAT* en termes de précision et de stabilité. Cependant, au lieu d'utiliser une loi de contrôle analytique pour calculer un contrôle continu, les deux algorithmes conçus dans ce chapitre utilisent une méthode d'intelligence artificielle et calculent un contrôle discret. Pour calculer la meilleure commande, *NOC* et *Abstract NOC* choisissent une position cible du chemin mémorisé qui permet au robot de converger vers le chemin du prédécesseur sans osciller autour. Ensuite, ils approchent le chemin du prédécesseur par un cercle ou une droite. Pour trouver la meilleure commande, ces deux algorithmes discrétisent son domaine de variation $[\omega_{min}, \omega_{max}]$ et choisissent la vitesse angulaire qui optimise une fonction d'optimisation et permet au robot de suivre le chemin de son prédécesseur sans le couper.

L'algorithme *NOC*, publié dans [Yazbeck et al., 2014], a été conçu pour des robots ayant un modèle de déplacement à variation instantanée, qui représente un modèle simplifié où la vitesse angulaire peut passer instantanément d'une valeur à une autre au début de chaque pas de temps. Les simulations ont validé l'amélioration apportée à *Memo-LAT* en montrant un suivi plus stable. En revanche, les performances du suivi deviennent moins bonnes quand nous implémentons *NOC* sur un modèle de déplacement des robots différent du modèle à variation instantanée. En effet, puisque *NOC* considère implicitement le modèle à variation instantanée dans le calcul du contrôle latéral, les conditions testées Γ et Γ^+ ainsi que la fonction d'optimisation sont mal évaluées dans le cas d'un autre modèle de déplacement de robots (dans ce chapitre, le modèle à variation progressive).

C'est pourquoi, nous avons proposé d'abstraire *NOC* par rapport au modèle de déplacement des robots. Nous avons obtenu l'algorithme *Abstract NOC* qui peut s'appliquer sur n'importe quel modèle de déplacement. Pour tester *Abstract NOC*, nous avons utilisé une fonction qui simule un modèle de déplacement des robots plus proche de la réalité. *Abstract NOC* était alors implémenté sur le modèle de déplacement à variation progressive qui prend en compte le temps nécessaire à la vitesse angulaire pour passer d'une valeur à une autre. Une première validation expérimentale de l'algorithme *Abstract NOC* a été ainsi montrée dans la section 5.7.

Chapitre 6

Conclusion

Cette thèse a présenté une méthode pour la mise en œuvre d'un convoi de robots autonomes sans accrochage matériel. Le convoi étant guidé par un opérateur humain le long d'un chemin généré en ligne, les robots ne connaissent pas à l'avance l'expression analytique de ce dernier. Le challenge de cette thèse était de concevoir des algorithmes permettant aux robots formant ce convoi de reconstruire et de suivre avec précision, durant leurs déplacements, le chemin de leur leader.

Dans ce travail, nous avons choisi d'utiliser une architecture de convois décentralisée locale étant donné ses différents avantages par rapport à une architecture globale (autonomie des robots, robustesse à la perte des communications, coût de calcul des commandes réduit). Nous avons aussi considéré que le convoi se déplace à faible vitesse, ce qui nous a permis de découpler les contrôles longitudinal et latéral pour les traiter chacun à part. Pour contrôler le comportement longitudinal des robots, nous avons utilisé le contrôleur *SSC* qui garantit l'absence des collisions. Le travail mené dans cette thèse a alors consisté à concevoir des algorithmes pour le contrôle latéral des robots formant le convoi. Des expérimentations réalisées en simulation et sur robots ont permis de valider ces algorithmes.

Par la suite, nous rappelons les différentes contributions de la thèse et nous discutons des différentes perspectives qui s'ouvrent suite à ce travail.

6.1 Synthèse des contributions

Cette thèse a présenté deux algorithmes de *platooning* pour le contrôle latéral : *Memo-LAT* et *NOC* qui s'est ensuite abstrait en *Abstract NOC*. Ces deux algorithmes s'appuient sur la mémorisation du chemin du prédécesseur. D'une part, *Memo-LAT* (Memorization and Look-Ahead Target) calcule un contrôle continu en utilisant une loi de contrôle analytique, d'autre part, *NOC* (Non Oscillatory Convergence) calcule un contrôle discret en utilisant une recherche heuristique.

Memo-LAT

Memo-LAT choisit, parmi la séquence des points du chemin emprunté par le robot prédécesseur, une position cible située à une distance d'établissement d_l constante qui soit inférieure à la distance de sécurité entre deux robots successifs. Pour atteindre cette position, le robot suiveur calcule une vitesse angulaire proportionnelle à l'erreur angulaire entre l'orientation courante du robot et l'orientation désirée. Les résultats expérimentaux ont montré l'amélioration de la précision du suivi apportée par la mémorisation du chemin du prédécesseur en comparaison avec un suivi où chaque robot vise la position courante de son prédécesseur. En revanche, cette approche pose un problème au niveau de la stabilité du suivi, en particulier lors des virages. En observant ces résultats expérimentaux, nous avons constaté que le choix de d_l influe sur la stabilité et la précision du suivi. En effet, le suivi est d'autant plus précis que d_l est faible. Une faible distance d'établissement accentue cependant les oscillations dans les virages serrés.

Suite à ces résultats, nous avons déduit qu'il est préférable d'utiliser une distance d'établissement d_l dynamique qui varie au cours du suivi en fonction de la courbure du chemin à suivre, de la vitesse du convoi et de l'erreur angulaire (écart latéral + erreur angulaire) initiale du robot suiveur. Malheureusement, malgré les différentes simulations réalisées en modifiant les paramètres citées ci-dessus, nous n'avons pas pu établir d'expression analytique de la distance d'établissement. Le réglage de la distance d'établissement selon la courbure de la trajectoire suivie est délicate et dépend des situations rencontrées. Pour pallier à ces problèmes, nous avons développé les algorithmes *NOC* et *Abstract NOC* qui prennent en compte la vitesse longitudinale de déplacement des robots et la courbure du chemin à suivre.

NOC

L'algorithme *NOC* a été conçu dans le but d'améliorer la stabilité et la précision du suivi en calculant une vitesse angulaire qui prend en compte la courbure du chemin à suivre. Pour ce faire, *NOC* choisit une position cible parmi les positions sauvegardées, il reconstruit le chemin autour de la position cible et estime la courbure en ce point. Pour calculer la commande, le contrôleur réalise une recherche heuristique sur l'ensemble des vitesses angulaires possibles $\{\omega_{min}, \dots, \omega_{max}\}$ et choisit celle qui évite au robot de couper (tant que possible) le chemin reconstruit de son prédécesseur et qui optimise une fonction d'erreur basée sur la déviation latérale et l'erreur angulaire. Une preuve mathématique a été menée dans ce travail dans le but de montrer la stabilité de *NOC*. Afin de prendre en compte le contrôle longitudinal au cours du suivi, *NOC* (qui considère que la vitesse longitudinale est constante durant le calcul de la commande latérale) utilise la valeur maximale que la vitesse longitudinale peut atteindre durant le pas de temps ($v = \max(v_0, v_0 + a.\tau)$). Ceci permet de garantir l'absence d'oscillations du robot suiveur autour du chemin de son prédécesseur mais ajoute une contrainte forte sur les conditions d'élimination des vitesses angulaires. En d'autres termes, nous éliminons des vitesses angulaires qui pourraient être des commandes possibles (c'est-à-dire que nous éliminons

potentiellement des faux négatifs) et qui généreraient un suivi plus précis que la vitesse angulaire retenue comme commande latérale.

En comparaison avec les performances de *Memo-LAT*, les expérimentations ont montré l'atténuation des oscillations (et donc l'amélioration de la stabilité) lors d'un suivi réalisé sous *NOC*. L'algorithme *NOC* a aussi été testé en simulation sur un modèle de déplacement simplifié des robots que nous avons appelé *modèle à variation instantanée*. Ce modèle suppose que les commandes de vitesses angulaires changent instantanément de valeur au début de chaque pas de temps. Sous ces conditions, les simulations réalisées ont permis de valider l'algorithme *NOC* en montrant un suivi précis sans oscillation.

Abstract NOC

L'algorithme *NOC*, tel qu'il a été conçu à l'origine, considère implicitement le modèle de déplacement des robots à variation instantanée dans le calcul de la commande latérale. Ainsi, les expérimentations ont montré une réduction des performances de *NOC* une fois implémenté sur un autre modèle de déplacement des robots, le modèle à variation progressive qui prend en compte le temps nécessaire au robot pour passer d'une vitesse angulaire à une autre.

Dans le but d'utiliser *NOC* avec d'autres modèles de déplacement de robots, nous avons conçu l'algorithme *Abstract NOC* qui n'est autre qu'une abstraction de *NOC* par rapport au modèle de mouvement des robots. Ne pouvant plus prévoir le déplacement du robot (comme dans le cas du *modèle à variation instantanée*), *Abstract NOC* fait appel à une fonction externe qui simule le mouvement du robot, selon un modèle de mouvement donné, suite à l'application des vitesses durant un pas de temps et qui lui renvoie une liste de points représentant la simulation de son évolution durant ce pas de temps. *Abstract NOC* utilise ensuite cette liste de points dans le calcul de la commande latérale sur le même principe que *NOC*. Contrairement à *NOC* et à *Memo-LAT*, *Abstract NOC* ne découple pas les deux contrôles longitudinal et latéral, puisqu'il utilise une fonction externe de simulation du mouvement du robot qui prend en entrée l'accélération longitudinale du robot (le contrôle latéral prend donc en compte la consigne du contrôleur longitudinal). En comparaison avec *NOC*, *Abstract NOC* permet d'avoir un suivi plus précis puisqu'il n'élimine pas les vitesses angulaires qui auraient été considérés par *NOC* comme des faux négatifs. Ceci est dû au fait que, à l'image de *NOC* simulé avec le modèle à variation instantanée, la prévision du mouvement correspond précisément au mouvement effectif du robot.

L'algorithme *Abstract NOC* a ensuite été validé en le testant sur le modèle de déplacement à variation progressive. En comparant les performances de *NOC* et d'*Abstract NOC* sur un suivi le long d'un chemin spiral d'un convoi de 9 robots ayant un modèle de mouvement à variation progressive, les simulations ont montré un suivi plus stable et précis ($e_y \leq 15$ mm) dans le cas d'*Abstract NOC* que dans le cas de *NOC* ($e_y \leq 160$ mm). De plus, pour étudier l'influence de la vitesse longitudinale sur le contrôle latéral, nous avons réalisé des simulations dans lesquelles la vitesse longitudinale change brusquement à un instant donné. La vitesse longitudinale du robot étant prise en compte directement dans

le calcul de la vitesse angulaire, l'erreur latérale est très faible ($e_y = 1,5$ mm) quand le robot est contrôlé par *Abstract NOC*.

6.2 Perspectives

Le travail réalisé dans cette thèse ouvre à différentes perspectives que nous discutons par la suite.

Étude de la stabilité de l'algorithme *Abstract NOC*

Dans la section 5.4, nous avons réalisé une preuve mathématique de la stabilité de *NOC* en utilisant les équations de mouvement (équations 3.3) du modèle simplifié d'un robot (modèle à variation instantanée). Cette même étude mathématique semble être très difficile à réaliser dans le cas d'*Abstract NOC*. En effet, *Abstract NOC*, tel que présenté ici utilisant le modèle de mouvement à variation progressive, repose sur des équations plus compliquées (contenant des intégrales de Fresnel) qui prennent en compte à la fois l'accélération angulaire et l'accélération longitudinale du robot (équations 3.6). Bien que les simulations réalisées sur *Abstract NOC* montrent toujours un suivi stable, nous souhaitons dans les travaux futurs mener une étude mathématique dans le but de prouver d'une manière théorique la stabilité de *Abstract NOC* avec le modèle à variation progressive.

Validation de l'algorithme *Abstract NOC* sur des modèles de mouvement dynamiques et réalistes

Après avoir réalisé une première validation de *Abstract NOC*, nous désirons l'implémenter sur un modèle de déplacement des robots plus sophistiqué et observer ses performances. En faisant ceci, nous pourrions étudier le comportement de l'algorithme *Abstract NOC* une fois que la dynamique du robot (inertie, glissements) sera prise en compte. Sachant que *Abstract NOC* utilise une simulation du mouvement du robot pour prévoir son déplacement durant le prochain pas de temps, il faudrait modéliser la dynamique des robots sur lesquels cet algorithme est implémenté. Une solution pour avoir le mouvement simulé des robots est de coupler *Abstract NOC* avec un moteur physique (à titre d'exemple celui de *TORCS*) ou avec les simulateurs de ces robots (comme celui des *turtlebots*). Pour avoir une simulation fidèle au mouvement réel du robot, il faudrait donner au simulateur l'état initial du robot, son état lors de chaque prise de décision (accélération, vitesse, position du centre d'inertie, etc...) et l'état de la route sur laquelle le robot se déplace (caractéristiques, température et humidité du revêtement de la chaussée). Toutes ces informations peuvent être obtenues à partir de différents capteurs pouvant être montés sur le robot. À titre d'exemple, en équipant le véhicule automatique de pneus intelligents ¹²,

12. Plusieurs constructeurs tels que Continental ou Pirelli ont déjà déposé des brevets concernant cette technologie qui devrait équiper les véhicules de série dans quelques années.

les sondes insérées dans leurs bandes de roulement permettent de transmettre l'état de la route, à savoir le taux d'humidité, si le sol est gelé ou enneigé, etc. De plus, nous pouvons aussi récupérer les vitesses de rotation des roues, l'angle de braquage du volant, la vitesse de lacet, l'accélération transversale du robot, ses suspensions, etc. Toutes ses données peuvent alors être utilisées dans la simulation du déplacement du robot afin de prévoir si la vitesse angulaire que nous désirons appliquer peut causer son sous-virement ou son sur-virement et de choisir par la suite la meilleure commande.

Durant cette thèse, nous avons considéré que le convoi se déplaçait à une faible vitesse longitudinale afin de pouvoir découpler le contrôle longitudinal du contrôle latéral. Ce découplage nous a permis de concevoir les deux algorithmes *Memo-LAT* et *NOC*. Ensuite, en synthétisant *Abstract NOC*, nous avons pu prendre en compte directement la vitesse longitudinale du robot dans le calcul de sa vitesse angulaire. En faisant ceci, c'est le contrôleur latéral qui s'adapte au comportement longitudinal du robot. Ceci peut mener à des comportements illogiques dans certains cas. À titre d'exemple, imaginons un convoi de 2 robots au niveau d'un virage et où le robot suiveur est assez loin de son prédécesseur de sorte que son contrôleur longitudinal calcule une accélération longitudinale forte pour réduire l'inter-distance entre les deux robots. En simulant le déplacement du robot suiveur qui est à l'entrée d'un virage serré, nous remarquons qu'il ne pourra pas accélérer fortement sans déraper. Une solution serait de réduire sa vitesse longitudinale de sorte que nous puissions calculer une vitesse angulaire qui permette au robot de suivre précisément le chemin de son prédécesseur sans déraper. Le contrôleur latéral proposerait ensuite cette nouvelle vitesse longitudinale au contrôleur longitudinal pour vérifier la garantie d'un suivi sûr (sans collisions). Une boucle s'établirait ainsi entre les deux contrôleurs latéral et longitudinal jusqu'à trouver les meilleures commandes de vitesse longitudinale et latérale.

Adaptation de l'algorithme *Abstract NOC* pour la réalisation de convois en formation

Enfin, dans cette thèse, nous avons conçu des algorithmes de contrôle pour réaliser un suivi de convoi en file. Dans les travaux futurs, nous souhaitons adapter *Abstract NOC* pour réaliser des convois en formation. Une solution possible serait de choisir, pour chaque robot du convoi, un point cible qui est un point d'ancrage virtuel relatif à la position d'un ou de plusieurs robots voisins. Ensuite, au fur et à mesure du déplacement du point d'ancrage, nous acquérons sa position et nous la sauvegardons. Nous obtenons ainsi le chemin consigne que le robot doit suivre. Enfin, nous appliquons notre contrôleur *Abstract NOC* dans le but d'asservir le robot sur son chemin consigne.

Appendices

Annexe A

Calcul de l'évolution d'un robot ayant un modèle de mouvement à variation instantanée

Dans cet annexe, nous calculons les équations de mouvement (en fonction de temps) d'un robot qui se déplace selon un modèle de mouvement où sa vitesse angulaire passe instantanément d'une valeur à une autre. Nous appelons ce modèle *un modèle à variation instantanée*. L'état initial du robot est représenté par sa position (x_0, y_0) , son orientation θ_0 et sa vitesse longitudinale v_0 . Il évolue selon les équations de mouvement suivantes :

$$\begin{cases} \dot{x}(t) = v(t) \cdot \cos \theta(t) \\ \dot{y}(t) = v(t) \cdot \sin \theta(t) \\ \dot{\theta}(t) = \omega \\ \dot{v}(t) = a \end{cases} \quad (\text{A.1})$$

A.1 Calcul de la nouvelle orientation $\theta(t)$

En intégrant l'équation $\dot{\theta}(t) = \omega$ en fonction du temps, nous obtenons la nouvelle orientation du robot :

$$\theta(t) = \omega \cdot t + \theta_0$$

A.2 Calcul de la nouvelle vitesse longitudinale $v(t)$

L'équation A.2 représente la variation de la vitesse longitudinale du robot :

$$\dot{v}(t) = a \quad (\text{A.2})$$

En supposant que l'accélération longitudinale a reste constante le long du pas de temps τ et en intégrant cette équation, nous obtenons l'évolution de la vitesse longitudinale en fonction du temps :

$$v(t) = a.t + v_0$$

A.3 Calcul de la nouvelle position $x(t)$

L'équation A.3 représente la variation de la position en x du robot :

$$\begin{aligned} \dot{x}(t) &= v(t) \cdot \cos \theta(t) \\ \dot{x}(t) &= (a.t + v_0) \cdot \cos(\omega.t + \theta_0) \\ \dot{x}(t) &= a.t \cdot \cos(\omega.t + \theta_0) + v_0 \cdot \cos(\omega.t + \theta_0) \end{aligned} \quad (\text{A.3})$$

En intégrant cette équation, nous obtenons l'évolution de x en fonction du temps :

$$\begin{aligned} x(t) &= x_0 + \int (at + v_0) \cdot \cos(\omega t + \theta_0) \\ x(t) &= x_0 + \int at \cdot \cos(\omega t + \theta_0) + \int v_0 \cdot \cos(\omega t + \theta_0) \\ x(t) &= x_0 + X1 + X2 \end{aligned} \quad (\text{A.4})$$

$X1$ sera évaluée en intégrant par parties, alors que $X2$ sera une intégrale simple.

A.3.1 Calcul de $\int at \cdot \cos(\omega t + \theta_0)$

En posant :

$$\begin{cases} v = t \\ du \cdot dt = \cos(\omega t + \theta_0) \cdot dt \end{cases} \quad (\text{A.5})$$

Nous obtenons :

$$\begin{cases} dv \cdot dt = 1 \cdot dt \\ u = \frac{1}{\omega} \cdot \sin(\omega t + \theta_0) \end{cases} \quad (\text{A.6})$$

Ainsi, nous résolvons $X1$ comme suit :

$$\begin{aligned}
 X1 &= \int at. \cos(\omega t + \theta_0) \\
 X1 &= at. \frac{1}{\omega}. \sin(\omega t + \theta_0) - a. \int \frac{1}{\omega}. \sin(\omega t + \theta_0). dt \\
 X1 &= \frac{at}{\omega}. \sin(\omega t + \theta_0) + \frac{a}{\omega^2}. \cos(\omega t + \theta_0) - \frac{a}{\omega^2}. \cos \theta_0
 \end{aligned} \tag{A.7}$$

A.3.2 Calcul de $\int v_0. \cos(\omega t + \theta_0)$

$$\begin{aligned}
 X2 &= \int v_0. \cos(\omega t + \theta_0) \\
 X2 &= \frac{v_0}{\omega}. \sin(\omega t + \theta_0) - \frac{v_0}{\omega}. \sin \theta_0
 \end{aligned} \tag{A.8}$$

A.3.3 Calcul de $x(t)$

L'évolution de x en fonction du temps est donc exprimée sous la forme suivante :

$$x(t) = x_0 + \frac{at + v_0}{\omega}. \sin(\omega t + \theta_0) + \frac{a}{\omega^2}. \cos(\omega t + \theta_0) - \frac{a}{\omega^2}. \cos \theta_0 - \frac{v_0}{\omega}. \sin \theta_0$$

A.4 Calcul de la nouvelle position $y(t)$

L'équation A.9 représente la variation de la position en y du robot :

$$\begin{aligned}
 \dot{y}(t) &= v(t). \sin \theta(t) \\
 \dot{y}(t) &= (a.t + v_0). \sin(\omega.t + \theta_0) \\
 \dot{y}(t) &= a.t. \sin(\omega.t + \theta_0) + v_0. \sin(\omega.t + \theta_0)
 \end{aligned} \tag{A.9}$$

En intégrant cette équation, nous obtenons l'évolution de y en fonction du temps :

$$\begin{aligned}
 y(t) &= y_0 + \int at. \sin(\omega t + \theta_0) + \int v_0. \sin(\omega t + \theta_0) \\
 y(t) &= y_0 + Y1 + Y2
 \end{aligned} \tag{A.10}$$

A.4.1 Calcul de $\int at. \sin(\omega t + \theta_0)$

En posant :

$$\begin{cases} v = t \\ du.dt = \sin(\omega t + \theta_0).dt \end{cases} \quad (\text{A.11})$$

Nous obtenons :

$$\begin{cases} dv.dt = 1.dt \\ u = \frac{-1}{\omega} \cdot \cos(\omega t + \theta_0) \end{cases} \quad (\text{A.12})$$

Ainsi, nous résolvons $Y1$ comme suit :

$$\begin{aligned} Y1 &= \int at \cdot \sin(\omega t + \theta_0) \\ Y1 &= at \cdot \frac{-1}{\omega} \cdot \cos(\omega t + \theta_0) + a \int \frac{1}{\omega} \cdot \cos(\omega t + \theta_0).dt \\ Y1 &= \frac{-at}{\omega} \cdot \cos(\omega t + \theta_0) + \frac{a}{\omega^2} \cdot \sin(\omega t + \theta_0) - \frac{a}{\omega^2} \cdot \sin \theta_0 \end{aligned} \quad (\text{A.13})$$

A.4.2 Calcul de $\int v_0 \cdot \sin(\omega t + \theta_0)$

$$\begin{aligned} Y2 &= \int v_0 \cdot \sin(\omega t + \theta_0) \\ Y2 &= -\frac{v_0}{\omega} \cdot \cos(\omega t + \theta_0) + \frac{v_0}{\omega} \cdot \cos \theta_0 \end{aligned} \quad (\text{A.14})$$

A.4.3 Calcul de $y(t)$

L'évolution de y en fonction du temps est donc exprimée sous la forme suivante :

$$y(t) = y_0 - \frac{at + v_0}{\omega} \cdot \cos(\omega t + \theta_0) + \frac{a}{\omega^2} \cdot \sin(\omega t + \theta_0) - \frac{a}{\omega^2} \cdot \sin \theta_0 + \frac{v_0}{\omega} \cdot \cos \theta_0$$

Annexe B

Calcul de l'évolution d'un robot ayant un modèle de mouvement à variation progressive

Dans cet annexe, nous calculons les équations de mouvement (en fonction de temps) d'un robot qui se déplace selon un modèle de mouvement qui prend en compte son accélération angulaire. Le robot ne peut donc pas changer instantanément de vitesse angulaire. Nous appelons ce modèle *un modèle à variation progressive*.

À l'état initial, le robot a une position (x_0, y_0) , une orientation θ_0 , une vitesse longitudinale v_0 et une vitesse angulaire ω_0 . Le robot change de vitesse longitudinale v et de vitesse angulaire ω selon respectivement une accélération longitudinale a et une accélération angulaire η .

Les équations de mouvement du robot sont alors exprimées selon la forme suivante :

$$\begin{cases} \dot{x}(t) = v(t) \cdot \cos \theta(t) \\ \dot{y}(t) = v(t) \cdot \sin \theta(t) \\ \dot{\theta}(t) = \omega(t) \\ \dot{v}(t) = a \\ \dot{\omega}(t) = \eta \end{cases} \quad (\text{B.1})$$

B.1 Calcul de la nouvelle vitesse longitudinale $v(t)$

L'équation B.2 représente la variation de la vitesse longitudinale du robot :

$$\dot{v}(t) = a \quad (\text{B.2})$$

En supposant que l'accélération longitudinale a reste constante le long du pas de temps

τ et en intégrant cette équation, nous obtenons l'évolution de la vitesse longitudinale en fonction du temps :

$$v(t) = a.t + v_0$$

B.2 Calcul de la nouvelle vitesse angulaire $\omega(t)$

L'équation B.3 représente la variation de la vitesse angulaire du robot :

$$\dot{\omega}(t) = \eta \quad (\text{B.3})$$

En supposant que l'accélération angulaire η reste constante le long du pas de temps τ et en intégrant cette équation, nous obtenons l'évolution de la vitesse angulaire en fonction du temps :

$$\omega(t) = \eta.t + \omega_0$$

B.3 Calcul de la nouvelle orientation $\theta(t)$

L'équation B.4 représente la variation de l'orientation du robot :

$$\dot{\theta}(t) = \omega(t) = \eta.t + \omega_0 \quad (\text{B.4})$$

En intégrant cette équation, nous obtenons l'évolution de l'orientation du robot en fonction du temps :

$$\theta(t) = \frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0$$

B.4 Calcul de la nouvelle position $x(t)$

L'équation B.5 représente la variation de la position en x du robot :

$$\begin{aligned} \dot{x}(t) &= v(t). \cos \theta(t) \\ \dot{x}(t) &= (a.t + v_0). \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) \\ \dot{x}(t) &= a.t. \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) + v_0. \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) \end{aligned} \quad (\text{B.5})$$

En intégrant cette équation, nous obtenons l'évolution de x en fonction du temps :

$$x(t) = x_0 + \int a.t. \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt + \int v_0. \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt \quad (\text{B.6})$$

$$x(t) = x_0 + X1 + X2$$

$X2$ est une intégrale de Fresnel que nous ne pouvons pas résoudre analytiquement. Par la suite, nous calculons $X1$.

B.4.1 Calcul de $\int a.t. \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt$

Nous exprimons $X1$ sous la forme $\int u'. \cos u$ dont l'intégral est $\sin u$. Nous détaillons dans ce qui suit les calculs :

$$\begin{aligned} X1 &= \int a.t. \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt \\ X1 &= a. \int \frac{\eta.t + \omega_0}{\eta}. \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt \\ X1 &= \frac{a}{\eta}. \int (\eta.t + \omega_0). \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt - \frac{a.\omega_0}{\eta}. \int \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt \\ X1 &= \frac{a}{\eta}. \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) - \frac{a.\omega_0}{\eta}. \int \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt - \frac{a}{\eta}. \sin \theta_0 \end{aligned} \quad (\text{B.7})$$

B.4.2 Calcul de $x(t)$

L'évolution de x en fonction du temps est donc exprimée sous la forme suivante :

$$x(t) = x_0 + \frac{a}{\eta}. \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) - \int \left(\frac{a.\omega_0}{\eta} - v_0\right). \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt - \frac{a}{\eta}. \sin \theta_0$$

B.5 Calcul de la nouvelle position $y(t)$

L'équation B.8 représente la variation de la position en y du robot :

$$\begin{aligned} \dot{y}(t) &= v(t). \sin \theta(t) \\ \dot{y}(t) &= (a.t + v_0). \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) \\ \dot{y}(t) &= a.t. \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) + v_0. \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) \end{aligned} \quad (\text{B.8})$$

En intégrant cette équation, nous obtenons l'évolution de y en fonction du temps :

$$y(t) = y_0 + \int a.t. \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt + \int v_0. \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt \quad (\text{B.9})$$

$$y(t) = y_0 + Y1 + Y2$$

$Y2$ est une intégrale de Fresnel que nous ne pouvons pas résoudre analytiquement. Par la suite, nous calculons $Y1$.

B.5.1 Calcul de $\int a.t. \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt$

Nous exprimons $Y1$ sous la forme $\int u'. \sin u$ dont l'intégral est $-\cos u$. Nous détaillons dans la suite les calculs :

$$Y1 = \int a.t. \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt$$

$$Y1 = a. \int \frac{\eta.t + \omega_0}{\eta}. \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt$$

$$Y1 = \frac{a}{\eta}. \int (\eta.t + \omega_0). \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt - \frac{a.\omega_0}{\eta}. \int \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt$$

$$Y1 = -\frac{a}{\eta}. \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) - \frac{a.\omega_0}{\eta}. \int \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt + \frac{a}{\eta}. \cos \theta_0 \quad (\text{B.10})$$

B.5.2 Calcul de $y(t)$

L'évolution de y en fonction du temps est donc exprimée sous la forme suivante :

$$y(t) = y_0 - \frac{a}{\eta}. \cos\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right) - \int \left(\frac{a.\omega_0}{\eta} - v_0\right). \sin\left(\frac{1}{2}.\eta.t^2 + \omega_0.t + \theta_0\right).dt + \frac{a}{\eta}. \cos \theta_0$$

Bibliographie

- [A. Davila, 2010] A. Davila, M. N. (September 2010). Sartre : Safe road trains for the environment.
- [Albagul and Wahyudi, 2004] Albagul, A. and Wahyudi (2004). Dynamic modelling and adaptive traction control for mobile robots. *International Journal of Advanced Robotic Systems*, 1 :149 – 154.
- [Antonelli et al., 2006] Antonelli, G., Arrichiello, F., Chiaverini, S., and Rao, K. J. (2006). Preliminary experiments of formation control using the null-space-based behavioral control. In *IFAC*.
- [Avanzini, 2010] Avanzini, P. (2010). *Modélisation et Commande d'un Convoi de Véhicules Urbains par Vision*. thèse de doctorat, Université Blaise Pascal – Clermont II, Clermont-Ferrand (FR).
- [Avanzini et al., 2008] Avanzini, P., Royer, E., Thuilot, B., and Martinet, P. (2008). A global decentralized control strategy for urban vehicle platooning using monocular vision and a laser rangefinder. In *International Conference on Control, Automation, Robotics and Vision*.
- [Avanzini et al., 2009] Avanzini, P., Thuilot, B., Dallej, T., Martinet, P., and Derutin, J. (2009). On-line reference trajectory generation for manually convoying a platoon of automatic urban vehicles. In *IROS*, pages 1867–1872.
- [Barfoot and Clark, 2004] Barfoot, T. and Clark, C. (29 February 2004). Motion planning for formations of mobile robots. In *Robotics and Autonomous Systems*, pages vol. 46, Issue 2, pp. 65–78.
- [Benhimane et al., 2005] Benhimane, S., Malis, E., Rives, P., and Azinheira, J. R. (April 2005). Vision-based control for car platooning using homography decomposition. In *International Conference on Robotics and Automation*, pages 2173–2178.
- [Bertozzi et al., 2011] Bertozzi, M., Broggi, A., Cardarelli, E., Fedriga, R. I., Mazzei, L., and Porta, P. P. (September 2011). Viac expedition toward autonomous mobility.
- [Bertsekas, 2007] Bertsekas, D. (2007). *Dynamic Programming and Optimal Control*.
- [Bom, 2006] Bom, J. (2006). *Etude et mise en œuvre d'un convoi de véhicules urbains avec accrochage immatériel*. thèse de doctorat, Université Blaise Pascal – Clermont II, Clermont-Ferrand (FR).
- [Bom et al., 2005] Bom, J., Thuilot, B., Marmouton, F., and Martinet, P. (2005). A global control strategy for urban vehicles platooning relying on nonlinear decoupling laws. In

- IEEE International Conference on Intelligent Robots and Systems*, pages pp. 2875 – 2880.
- [Boumaza and Scherrer, 2007] Boumaza, A. and Scherrer, B. (2007). Optimal control subsumes harmonic control. In *International Conference on Robotics and Automation (ICRA)*.
- [Bourlès and Guillard, 2012] Bourlès, H. and Guillard, H. (2012). *Commande des systèmes, performance et robustesse*.
- [Connolly et al., 1990] Connolly, C. I., Burns, J. B., and Weiss, R. (1990). Path planning using laplace’s equation. In *International Conference on Robotics and Automation ICRA*.
- [Contet et al., 2006] Contet, J.-M., Gechter, F., Gruer, P., and Koukam, A. (2006). Multiagent system model for vehicle platooning with merge and split capabilities. In *3rd International Conference on Autonomous Robots and Agents (ICARA)*, Palmerston North (NZ).
- [Contet et al., 2007] Contet, J.-M., Gechter, F., Gruer, P., and Koukam, A. (2007). Application of reactive multiagent system to linear vehicle platoon. In *19th IEEE International Conference on Tools with Artificial Intelligence*.
- [Das et al., 2002] Das, A. K., Fierro, R., Kumar, V., Ostrowski, J. P., Spletzer, J., and Taylor, C. J. (2002). A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*.
- [Daviet and Parent, 1995] Daviet, P. and Parent, M. (1995). Platooning techniques for the intelligent highway : How close can we get ? In *3rd IEEE Mediterranean Symposium on New Directions in Control and Automation*, volume 2, pages 237–242, Limassol, Cyprus (GR).
- [Daviet and Parent, 1996a] Daviet, P. and Parent, M. (1996a). Longitudinal and lateral servoing of vehicles in a platoon. In *Proc. of the IEEE Int. Symp. on Intelligent Vehicles*, pages 41–46, Tokyo (JP).
- [Daviet and Parent, 1996b] Daviet, P. and Parent, M. (1996b). Platooning technique for empty vehicles distribution in the praxitele project. In *4th IEEE Mediterranean Symposium on New Directions in Control and Automation*, Malem, Krete (GR).
- [El-Zaher et al., 2011] El-Zaher, M., Gechter, F., Gruer, P., and Hajjar, M. (2011). A new linear platoon model based on reactive multi-agent systems. In *23rd IEEE International Conference on Tools with Artificial Intelligence*.
- [Eren et al., 2004] Eren, T., Anderson, B. D. O., Morse, A. S., Whiteley, W., and Belhumeur, P. N. (2004). Operations on rigid formations of autonomous agents. *COMMUNICATIONS IN INFORMATION AND SYSTEMS*, 3 :223–258.
- [Eric Coelingh, 2012] Eric Coelingh, S. S. (November 2012). All aboard the robotic road train.
- [Gehrig and Stein, 2001] Gehrig, S. K. and Stein, F. J. (2001). Elastic bands to enhance vehicle following. In *IEEE Intelligent Transportation Systems Conference*.

-
- [Gil-Pinto et al., 2005] Gil-Pinto, A., Fraisse, P., and Zapata, R. (2005). A decentralized adaptive trajectory planning approach for a group of mobile robots. In *Towards Autonomous Robotic Systems (TAROS)*.
- [Gil-Pinto et al., 2007] Gil-Pinto, A., Fraisse, P., and Zapata, R. (2007). Decentralized strategy for car-like robot formations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages pp 4176 – 4181.
- [Hallé, 2005] Hallé, S. (2005). Automated highway systems : Platoons of vehicles viewed as a multiagent system. Master’s thesis, Faculté des sciences et de génie, Université Laval, Québec.
- [Hendrickx et al., 2008] Hendrickx, J. M., Fidan, B., Yu, C. B., Anderson, B. D. O., and Blondel, V. D. (2008). Formation reorganization by primitive operations on directed graphs. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, 53.
- [Hwang et al., 2009] Hwang, J.-H., Arkin, R. C., and Kwon, D.-S. (2009). Mobile robots at your fingertip : Bezier curve on-line trajectory generation for supervisory control. In *IROS*.
- [Isidori, 1995] Isidori, A. (1995). *Nonlinear Control Systems*. Springer-Verlag.
- [Kaku et al., 2012] Kaku, A., Mukai, M., and Kawabe, T. (2012). A centralized control system for ecological vehicle platooning using linear quadratic regulator theory. In *17th International Symposium on Artificial Life and Robotics*, pages Volume 17, Issue 1, pp 70 – 74.
- [Kavatthekar and Chen, 2011] Kavathekar, P. and Chen, Y. (2011). Draft : Vehicle platooning : a brief survey and categorization. In *Proceedings of The ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.
- [Khatib, 1986] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5.
- [Khatir and Davison, 2004] Khatir, M. E. and Davison, E. J. (2004). Decentralized control of a large platoon of vehicles using non-identical controllers. In *American Control Conference, 2004. Proceedings of the 2004*, pages 2769 – 2776 vol.3.
- [Kim and Khosla, 1992] Kim, J.-O. and Khosla, P. K. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 8.
- [Komoriya and Tanie, 1989] Komoriya, K. and Tanie, K. (1989). Trajectory design and control of a wheel-type mobile robot using b-spline curve. In *IROS*.
- [Lafferriere et al., 2004] Lafferriere, G., Caughman, J., and Williams, A. (2004). Graph theoretic methods in the stability of vehicle formations. In *American Control Conference*, pages vol. 4, pp. 3729–3734.
- [Levine and Athans, 1966] Levine, W. and Athans, M. (1966). On the optimal error regulation of a string of moving vehicles. In *IEEE Transactions on Automatic Control*, pages Volume 11, NO. 3, pp 355 – 361.
- [Lewis and Tan, 1997] Lewis, M. A. and Tan, K.-H. (1997). High precision formation control of mobile robots using virtual structures. *Journal of Autonomous Robots*.

- [Mariottini et al., 2007] Mariottini, G. L., Morbidi, F., Prattichizzo, D., Pappas, G. J., and Daniilidis, K. (2007). Leader-follower formations : Uncalibrated vision-based localization and control. In *IEEE International Conference on Robotics and Automation*.
- [MATKO et al., 2008] MATKO, D., KLANCAR, G., BLAZIC, S., GECHTER, F., Simonin, O., CONDET, J.-M., and GRUER, P. (2008). Control strategy for wheeled mobile robots platoons. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.
- [Micaelli and Samson, 1993] Micaelli, A. and Samson, C. (1993). Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Technical report, INRIA.
- [Monteiro and Bicho, 2002] Monteiro, S. and Bicho, E. (2002). A dynamical systems approach to behavior-based formation control. In *ICRA*.
- [Olfati-Saber and Murray, 2002] Olfati-Saber, R. and Murray, R. M. (2002). Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*.
- [Sabattini et al., 2009] Sabattini, L., Secchi, C., and Fantuzzi, C. (2009). Potential based control strategy for arbitrary shape formations of mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [Samson, 1995] Samson, C. (Jan 1995). Control of chained systems application to path following and time-varying point-stabilization of mobile robots. In *IEEE Transactions on Automatic Control*, pages Volume :40, Issue : 1.
- [Sastry and Isidori, 1989] Sastry, S. S. and Isidori, A. (1989). Adaptive control of linear-linear systems. *IEEE Trans. Automat. Contr.*, 34 :1123–1131.
- [Scheuer and Fraichard, 1997] Scheuer, A. and Fraichard, T. (1997). Continuous-curvature path planning for car-like vehicles. In *International Conference on Intelligent Robots and Systems*.
- [Scheuer et al., 2009] Scheuer, A., Simonin, O., and Charpillet, F. (2009). Safe longitudinal platoons of vehicles without communication. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 70–75, Kobe (JP). <http://www.loria.fr/~scheuer/Platoon>.
- [Sheikholeslam and Desoer, 1993] Sheikholeslam, S. and Desoer, C. A. (1993). Longitudinal control of a platoon of vehicles with no communication of lead vehicle information : A system level study. In *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*.
- [Shladover et al., 1991] Shladover, S., Desoer, C., Hedrick, J., Tomizuka, M., Walrand, J., Zhang, W.-B., McMahon, D., Peng, H., Sheikholeslam, S., and McKeown, N. (1991). Automated vehicle control developments in the path program. *IEEE Transactions on Vehicular Technology*, 40(1) :114 – 130.
- [Simonin et al., 2011] Simonin, O., Hureauux, T., and Charpillet, F. (2011). Interactive surface for bio-inspired robotics, re-examining foraging models. In *23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*.
- [Süli and Mayers, 2003] Süli, E. and Mayers, D. (2003). *An Introduction to Numerical Analysis*. Cambridge University Press.

-
- [Tanner, 2004] Tanner, H. (2004). On the controllability of nearest neighbor interconnections. In *IEEE Conference on Decision and Control*, pages vol. 3, pp.2467–2472.
- [Tom Robinson, 2010] Tom Robinson, Eric Chan, E. C. (2010). Operating platoons on public motorways : An introduction to the sartre platooning programme.
- [Vedova et al., 2012] Vedova, M. L. D., Rubagotti, M., Facchinetti, T., and Ferrara, A. (2012). Platooning control of autonomous nonholonomic mobile robots in a human-robot coexisting environment. In *American Control Conference*.
- [Wei-bin Zhang, 1990] Wei-bin Zhang, R. E. P. (May 1990). An intelligent roadway reference system for vehicle lateral guidance/control. In *American Control Conference*, pages 281 – 286.
- [Xiang and Bräunl, 2010] Xiang, J. and Bräunl, T. (2010). String formations of multiple vehicles via pursuit strategy. *IET Control Theory and Applications*, 4 :1027–1038.
- [Yazbeck et al., 2013] Yazbeck, J., Scheuer, A., and Charpillet, F. (2013). Near-to-near decentralized algorithm for vehicle platooning. In *Control Architecture of Robots*.
- [Yazbeck et al., 2014] Yazbeck, J., Scheuer, A., and Charpillet, F. (2014). Decentralized near-to-near approach for vehicle platooning based on memorization and heuristic search. In *International Conference on Robotics and Automation*.
- [Yazbeck et al., 2011] Yazbeck, J., Scheuer, A., Simonin, O., and Charpillet, F. (2011). Improving near-to-near lateral control of platoons without communication. In *IROS*, pages 4103–4108.
- [Yi and Chong, 2005] Yi, S.-Y. and Chong, K.-T. (2005). Impedance control for a vehicle platoon system. *Mechatronics*, 15(5) :627–638.

Résumé

Dans cette thèse, nous nous intéressons au problème du suivi en convoi, désigné en anglais par le terme *platooning*, où un train de robots essaie de suivre un chemin décrit par le leader. Ce chemin, n'étant pas prédéfini mais généré au cours du suivi, est inconnu de tous les robots suiveurs. Dans ce travail, nous choisissons une approche décentralisée locale où chaque robot du convoi observe son voisinage et calcule son contrôle de façon à avoir un suivi stable (absence d'oscillations) et précis (erreur latérale aussi faible que possible). Cette thèse étudie plus précisément le comportement latéral d'un robot du convoi et propose deux contrôleurs s'appuyant sur la mémorisation du chemin suivi par son prédécesseur. Un premier algorithme de contrôle *Memo-LAT* (Memorization and Look-Ahead Target) calcule une commande latérale continue en utilisant une loi de contrôle analytique. La stabilité de *Memo-LAT* n'étant pas toujours garantie, nous proposons l'algorithme de contrôle *NOC* (Non-Oscillatory Convergence) qui prend en compte la courbure du chemin à suivre dans le calcul du comportement latéral. *NOC* combine une approche géométrique avec une recherche heuristique pour calculer une commande discrète permettant au robot de suivre avec précision le chemin de son prédécesseur sans oscillation.

Mots-clés : robotique mobile, navigation autonome, contrôle décentralisé local, suivi en file, systèmes de transport intelligent.

Abstract

This thesis deals with the platooning problem which aims to concieve a control algorithm allowing a convoy of vehicles to follow their leader's path. This path, which is initially undefined and unknown to all the following robots, is generated as the leader moves. In this thesis, we choose a local decentralized approach in which each robot of the platoon uses its local perceptions to compute its own commands aiming to achieve a stable (no oscillations) and precise (with a lateral error as small as possible) platooning. More precisely, this thesis studies the lateral behavior of a platoon's robot and introduces two controllers based on the memorization of the robot's predecessor's path. The first algorithm, *Memo-LAT* (Memorization and Look-Ahead Target), computes a continuous lateral command using an analytic control law. As the stability of *Memo-LAT* is not always guaranteed, we present *NOC* (Non-Oscillatory Convergence), a control algorithm which takes into account the path's curvature in the robot's lateral behavior's computation. *NOC* combines a geometric approach to a heuristic search method to compute a discrete command allowing the robot to follow precisely and without oscillations its predecessor's path.

Keywords : mobile robotics, autonomous navigation, local decentralized control, platooning, intelligent transportation systems.