

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Électronique des Systèmes Embarqués

Présenté par

BAHRIZ Taha

&

AMROUCHE Mahfoud

Contrôle d'un système multi-robots par mode glissant

Proposé et Dirigé par : Professeur BENSLAMA Zoubir
Monsieur DAMANI Alla-eddine

Année Universitaire 2019-2020

Remerciements

Nos remerciements s'adressent d'abord à ALLAH le tout puissant pour les aides qui nous sont offertes pour réaliser ce travail et à son prophète MOHAMED (paix et salut sur lui).

Ce mémoire n'aurait pas été possible sans l'intervention, consciente, d'un grand nombre de personnes. Nous souhaitons ici les remercier.

Nous voudrions présenter nos chaleureux remerciements à notre encadreur Mr BENSLAMA Zoubir que nous avons pu tout au long de nos recherches bénéficier de sa compétence, et lui témoigner notre gratitude pour sa patience, sa confiance et son soutien ainsi que ses précieux conseils qu'il nous a prodigué au cours de notre travail à bon port.

Nos vifs remerciements vont également aux membres de jury pour l'intérêt de porter à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

De même nous remercions vivement nos familles et en particulier nos parents, merci pour votre contribution quotidienne qui nous avons permis de bénéficier de conditions optimales tout au long de ce travail, et toute personne qui a contribué de loin ou de près à l'élaboration de ce travail.

Nous tenons également à remercier aussi le personnel de notre département ainsi nos enseignants.

Enfin, nous ne pouvons clore cette page sans remercier Mr DAMANI Allaeddine pour nous avoir fait découvrir les excellents outils que sont ROS et Gazebo au début de notre mémoire et qui nous ont permis la réalisation de simulations de qualité.

Merci à tous et à toutes.

ملخص: شهد التحكم التعاوني لأنظمة الروبوتات المتعددة المتنقلة طفرة هائلة في السنوات الأخيرة ، بسبب التطبيقات الواسعة ، مثل الإنقاذ ، ونقل الأجسام الكبيرة ، والمراقبة ، وشبكات الاستشعار ، والنقل التعاوني ، إلخ. الفكرة هي أن المركبات ذاتية القيادة التي تعمل معًا يمكن أن تحقق نتائج أفضل. في هذه المذكرة ، نحن مهتمون بالتحكم في مجموعة من الروبوتات المتنقلة غير الشاملة بناءً على نظرية الوضع المنزلق باستخدام مصحح - مقدر SVSF. الهدف هو تصميم هذا الأمر ليناسب كل روبوت بحيث يمكن لمجموعة الروبوتات المتنقلة أداء مهمة محددة مسبقًا أو اتباع المسار المطلوب في بيئة بها عوائق ، مع الحفاظ على التكوين الهندسي المطلوب.

كلمات المفاتيح: الروبوتات المتنقلة غير الشاملة؛ نظرية الوضع المنزلق؛ مصحح – مقدر SVSF ؛ عوائق؛ ROS؛ Gazebo.

Résumé : La commande coopérative des systèmes multi-robots mobiles connaît un essor considérable ces dernières années, en raison des vastes applications, comme le sauvetage, le déplacement des objets volumineux, la surveillance, les réseaux de capteurs, le transport coopératif, etc. L'idée est que des véhicules autonomes qui collaborent entre-eux peuvent obtenir de meilleurs résultats. Dans ce mémoire, nous nous intéressons à la commande d'un groupe de robots mobiles non-holonomes basé sur la théorie de mode glissant avec un correcteur-estimateur SVSF. L'objectif est de concevoir cette commande pour adapter à chaque robot et pour que le groupe de robots mobiles puisse exécuter une tâche prédéfinie ou suivre une trajectoire désirée dans un environnement avec des obstacles, tout en maintenant une configuration géométrique souhaitée.

Mots clés : les robots non-holonyme ; mode glissant ; correcteur-estimateur SVSF ; obstacle ; ROS ; Gazebo.

Abstract: Cooperative control of mobile multi-robot systems has grown considerably in recent years, due to the wide range of applications, such as rescue, moving large objects, surveillance, sensor networks, cooperative transport, etc. The idea is that autonomous vehicles that collaborate can achieve better results. In this paper, we are interested in controlling a group of non-holonomic mobile robots based on the sliding mode theory with an SVSF estimator-corrector. The objective is to design this control to adapt to each robot, and that group of mobile robots can execute a predefined task or follow a desired trajectory in an environment with obstacles, while maintaining a desired geometrical configuration.

Keywords: non-holonomic mobile robots; sliding mode; SVSF estimator-corrector; obstacles; ROS; Gazebo.

Table des matières

Table des matières	i
Liste des tableaux	iv
Table des figures	v
Liste des abréviations	viii
Introduction générale	1
1 Systèmes multi-robots autonomes	3
1.1 Introduction	3
1.2 Approche	4
1.2.1 Structure de contrôle	5
1.2.2 Approches de contrôle	7
1.3 Les systèmes multi-robots	10
1.3.1 Commande non-linéaire	11
1.3.2 Contrôle d'un robot mobile	11
1.3.3 Modélisation	13
1.4 Évitement d'obstacles	18
1.4.1 Méthode des champs de potentiel	19
1.4.2 Méthode de la zone de déformation virtuelle (ZDV)	19
1.4.3 Choix de la méthode d'évitement d'obstacles	20
1.5 Conclusion	21
2 Contrôle de la formation leader-suiveurs de multi-robots non-holonomes	22
2.1 Introduction	22
2.2 Contrôle de formation leader-suiveur	23
2.2.1 Modèle mathématique d'un robot mobile non-holonyme	23

2.2.2	Contrôle de formation leader-suiveur et la dynamique d'erreur	24
2.3	Présentation de la commande	28
2.3.1	Contrôle par bouclage linéarisant	28
2.4	Conclusion	33
3	La commande par mode glissant (CMG)	35
3.1	Introduction	35
3.2	Systèmes à structure variable	35
3.3	Principe de la technique de mode glissant	36
3.4	Conception de la commande par mode glissant	37
3.4.1	Choix de la surface de glissement	37
3.4.2	Conditions de convergence	38
3.4.3	Synthèse de la commande	39
3.5	Contrôle de la formation leader-suiveur par technique de mode glissant	44
3.5.1	Modèle de contrôle de formation leader-suiveur	44
3.5.2	Choix des surfaces de glissements	45
3.5.3	Loi de commande par mode glissant	46
3.5.4	Choix de la fonction de Lyapunov	46
3.6	Résultats des simulations sur Matlab	47
3.7	Évitement d'obstacle par champ de potentiel	51
3.8	Architecture du système de contrôle	52
3.9	Poursuite d'un robot mobile à base d'un filtre SVSF	53
3.9.1	Le principe de fonctionnement du filtre SVSF	53
3.9.2	Formulation mathématique du filtre SVSF	54
3.10	Conclusion	56
4	Simulation et expérimentation	57
4.1	Introduction	57
4.2	ROS (Robot Operating System)	58
4.2.1	Les notions de base de ROS [43]	58
4.2.2	Les fonctions et les outils de base de ROS	60
4.2.3	TurtleBot3	61
4.3	Simulation	62
4.3.1	Étapes de construction de l'environnement de simulation	64
4.3.2	Scénario 01 : la formation sur une trajectoire circulaires sans obstacles	66
4.3.3	Scénario 02 : la formation sur une trajectoire (S) sans des obstacles .	69
4.3.4	Scénario 03 : la formation sur une trajectoire (S) avec des obstacles .	72

4.4	Réalisation et implémentation	73
4.4.1	Les composants de Robots	75
4.4.2	Les différentes vues des deux robots	76
4.4.3	Communication entre les deux robots	77
4.4.4	Calibration de la caméra	77
4.4.5	Localisation des marqueurs à l'aide d'Aruco	78
4.4.6	Expérimentation	81
4.5	Conclusion	81
	Conclusion générale	82
	Annexes	84
	Bibliographie	88

Liste des tableaux

1.1	Architecture centralisée versus distribuée[11]	6
3.1	différents types de potentiel pouvant être utilisés	51
4.1	Comparaison TurtleBot3 Burger et TurtleBot3 Waffle	62
4.2	Les conditions initiales des quatre robots pour scénario N°1	66
4.3	Les gains de la loi de contrôle MG	67
4.4	Les conditions initiales des quatre robots pour 2ème scénario	69

Table des figures

1.1	Formation de robots qui changent de comportement	3
1.2	Des comportements de formation dans la nature,(a) les groupes de poissons (b) les troupeaux d'oiseaux	4
1.3	formation (a) multi-véhicules aériens sans pilote(b) multi-véhicules sous-marins autonomes(c)systèmes de multi-robots mobiles	5
1.4	Contrôle de formation $(l - \psi)$ [7]	8
1.5	Contrôle de formation $(l - l)$ [15]	9
1.6	Exemples de différents types de tâches coopératives multi-robots	10
1.7	Schéma des interactions d'un robot avec son environnement	12
1.8	Schéma d'un asservissement d'un robot mobile	12
1.9	Repérage d'un robot mobile	13
1.10	Caractérisation du roulement sans glissement	15
1.11	(a) Robot mobile Pioneer P3-DX (b) Robot mobile Turtlebot3	17
1.12	Centre instantané de rotation d'un robot de type unicycle	17
1.13	Illustration de potentiels primitifs dont la combinaison guide les déplacements du robot. Le robot se déplacera selon les lignes de courant	19
1.14	De différents potentiels primitifs	20
1.15	Évitement d'obstacles : définition des cadres et description de la pose du problème.	20
2.1	Schéma de principe de la structure de contrôle droite	23
2.2	Robot mobile non-holonomes	24
2.3	Formation Leader-Suiveur	25
2.4	Schéma de contrôle basé sur bouclage linéarisant	30
2.5	Les trajectoires en temps réel de leader et son suiveur	32
2.6	Les erreurs de suivi pour le suiveur	32
2.8	Les vitesses angulaires du suiveur et du leader	32
2.7	Les vitesses linéaire et angulaire du suiveur	33

3.1	Systèmes à structure variable dans une coupe de l'espace d'état.	36
3.2	Différents modes de convergence pour la trajectoire d'état	37
3.3	La valeur continue U_{eq} prise par la commande lors de la commutation.	40
3.4	Représentation de la fonction « Sign».	41
3.5	Phénomène de broutement	42
3.6	Fonction Sat avec un seuil	42
3.7	Organigramme du contrôle de mouvement d'un robot mobile	44
3.8	Les trajectoires de leader et son suiveur	48
3.9	Les erreurs de suivi pour le suiveur	48
3.10	Les vitesses linéaire et angulaire du suiveur	48
3.11	Les vitesses angulaires du suiveur et du leader	49
3.12	Les trajectoires en temps réel des robots	49
3.13	Les erreurs de suivi pour le suiveur et leader	50
3.14	Transformation du SMR en se déplaçant en trajectoire sinusoïdale	50
3.15	L'architecture de contrôle suiveur	52
3.16	Le principe d'estimation du filtre SVSF	53
3.17	exemple de moment où la trajectoire d'état estimée est lissée, et quand il y a un phénomène du broutement (a) trajectoire estimée lissée ($\psi \succeq \beta$)(b) présence d'un phénomène du broutement (chattering) ($\psi \prec \beta$) [40]	54
4.1	Interaction entre les différents outils ROS	58
4.2	Relation entre les nœuds, les services et les topics[44]	59
4.3	Logiciel de simulation 3D Gazebo	60
4.4	Graphique d'un robot ROS simulé	61
4.5	Présentation de TurtleBot3	62
4.6	Schéma global de contrôle du système multi-robot Leader-Suiveur avec ROS gazebo	63
4.7	Paquet du système développé	64
4.8	(a) Simulation sur Gazebo (4 robots), (b) quelques topics relatifs aux robots . .	65
4.9	Panneaux multi-plot utilisée	65
4.10	Le graphique dynamique représentant l'état de notre système Leader-Suiveur (tous les topics et nœuds)	66
4.11	Les trajectoires de leader et ses suiveurs dans un environnement non-encombré .	67
4.12	Les erreurs de suivi pour les suiveurs	67
4.13	Les vitesses linéaires des suiveurs et leader	68
4.14	Les vitesses angulaires du suiveur et du leader	68
4.15	Les trajectoires de leader et ses suiveurs dans un environnement non-encombré .	69
4.16	Les erreurs de suivi pour les suiveurs : (a) sans filtre SVSF, (b) avec filtre	70

4.17	Les vitesses linéaires du suiveurs et leader :(a) sans filtre SVSF, (b) avec filtre .	70
4.18	Les vitesses angulaires du suiveur et du leader :(a) sans filtre SVSF, (b) avec filtre	71
4.19	Les trajectoires de leader et ses suiveurs dans un environnement encombré (avec des obstacles)	72
4.20	Les erreurs de suivi pour le suiveur	73
4.21	Schéma global de contrôle SMR (réalisation pratique et implémentation)	74
4.22	Schéma électronique de leader	75
4.23	Schéma électronique de suiveur	76
4.24	Les différentes vues du leader. (a) Vue haute et gauche, (b) Vue de dessous . .	77
4.25	Les différentes vues du suiveur (a) Vue droite, (b) Vue avant	77
4.26	(a) Damier de calibration,(b) matrice de calibration et distorsion	78
4.27	Détection de leader en utilisant Aruco	79
4.28	L'asservissement est réalisé à l'aide d'un correcteur PID	79

Liste des abréviations

CMG	:Contrôle en Mode Glissant
CIR	:Centre Instantané de Rotation
EKF	:Extended Kalman Filter
GUI	:Graphical User Interface
KF	:Kalman Filter
MC	:Mode de Convergence
MG	:Mode de Glissement
MRP	:Mode du Régime Permanent
PD	:Proportionnel, Dérivées
PID	:Proportionnel, Intégral, Dérivé
ROS	:Robot Operating System
RSG	:Roulement Sans Glissement
Rviz	:ROS visualisation
SRM	:Systèmes de Robots Mobiles
SMA	:Système Mécanique Articulé
SMR	:Système Multi-Robot
SVSF	:Smooth Variable Structure Filter
ZDV	:Zone de Déformation Virtuelle

Introduction générale

La robotique autonome est l'un des thèmes les plus importants de cette génération de recherche. Il a un large éventail d'applications, telles que la construction, la fabrication, les déchets de gestion, exploration spatiale et le transport militaire. L'un des principaux domaines de la recherche, afin de réussir des robots autonomes, est la planification du chemin. Elle est définie comme une navigation qui doit être sans collision et optimale pour que le véhicule autonome puisse manœuvrer d'une source à sa destination. Le thème de ce mémoire traite un domaine dans la robotique qui se base par le guidage (leader) et le suivi (following). Un robot mobile de suivi de robot est un robot qui suit un autre spécifique tout en mettant en œuvre simultanément l'évitement d'obstacles. Le robot ne suit que la cible robot et considère tous les autres objets comme des obstacles. Cela signifie que même s'il y a plusieurs robots qui se situent dans l'environnement, le robot doit suivre son leader spécifique et éviter les autres. Les perspectives des systèmes multi-robots se sont multipliées ces dernières années. En effet, les avantages de tels systèmes par rapport à un seul robot, notamment la flexibilité, l'adaptabilité, évolutivité et prix abordable. Faire bouger un groupe de robots en formation serait bénéfique dans de nombreuses applications du monde réel, telles que la recherche et le sauvetage, le déminage dans les missions militaires, le transport de gros objets et le convoi. Il est possible pour un utilisateur de contrôler un groupe entier de robots sans spécifier explicitement les commandes pour chacun. Dans un scénario alternatif, le robot de tête peut être doté des capteurs et des capacités de calcul plus sophistiqués pour la planification et la navigation globales, tandis que d'autres robots en formation sont simples et spécialisés. La désaccentuation d'un robot gros et coûteux réduit le risque d'échec catastrophique de la mission. L'utilisation de plusieurs robots mobiles permet la réalisation de tâches complexes. Les robots ne doivent pas nécessairement être de structure très complexe, car chacun peut être spécialisé pour une tâche particulière. La réduction du nombre de capteurs sur chaque robot dans un système multi-robot (SMR) joue un rôle majeur dans la réduction de la complexité et du coût. Un avantage potentiel des systèmes multi-robots est la réduction de l'implication humaine dans des situations dangereuses, notamment la recherche et le sauvetage, les zones minières, les champs de bataille et les missions planétaires. De telles applications soumettent

les robots à des taux de dommages et d'échec élevés. Les unités individuelles équipées d'un équipement coûteux soulèvent des problèmes économiques inacceptables. L'utilisation de plusieurs robots équipés de composants à faible coût, parmi lesquels les tâches sont réparties, est plus adaptée à telles situations. La distribution des tâches comprend la détection, le calcul et le contrôle pour le groupe dans son ensemble. Dans un cadre leader-suiveur, le leader se voit confier la tâche de navigation, y compris la planification du chemin et l'évitement d'obstacles, tandis que les tâches des suiveurs impliquent de suivre le leader, de collecter des données et de gérer la communication. Ces groupes peuvent être facilement extensibles pour accueillir plus d'unités pour une plus grande couverture de capteurs ou un mouvement de troupes. Il est souhaitable dans de nombreuses applications que le robot mobile puisse suivre un robot. Il y a eu divers efforts dans la littérature pour créer des robots de suivi. Cependant, les robots mobiles actuels de suivi ne sont pas capables de fonctionner dans les environnements non structurés. Parce que plusieurs approches principales, telles que la vision et les capteurs, ne sont pas totalement fiables dans toutes les situations, il est nécessaire d'explorer d'autres méthodes. La conception de la commande pour les robots mobiles repose sur des mesures à partir de capteurs à l'estime tels que les encodeurs de roues qui fournissent des données d'odométrie ou de position. Cependant, ces mesures se rendent totalement peu fiables après quelques mètres de navigation, en raison de la faible précision et de la dérive du codeur. Patinage des roues, l'irrégulation des surfaces et le bruit électrique rendent les lectures imprécises. Les capteurs acoustiques tels que les sonars sont chers et leurs lectures sont sensibles aux réflexions parasites. Cependant, en raison de l'efficacité et du faible coût des capteurs de vision et de leur faible coût relatif en puissance de calcul, la tendance actuelle est de concevoir des systèmes qui utilisent la vision comme capteur principal. Le traitement d'image dans le logiciel évite le besoin de systèmes complexes utilisant des capteurs d'odométrie, sonar ou laser. Dans ce mémoire, un effort a été fait pour acquérir la localisation et l'information sensorielle par la vision. Bien que cette méthode entraîne une augmentation des calculs, de meilleurs algorithmes peuvent être appliqués pour le développement. Cette recherche a consisté à amener un groupe de robots mobiles dotés de capteurs bon marché à suivre un leader avec une formation géométrique souhaitée. Ces robots sont nettement moins puissants et complexes que d'autres systèmes robotiques en termes de capteurs et de puissance de calculs.

Chapitre 1

Systèmes multi-robots autonomes

1.1 Introduction

L'objectif de ce mémoire est de réaliser une architecture de contrôle permettant le maintien de la formation des robots mobiles dans un environnement encombré (des obstacles statiques) (figure 1.1), nous souhaitons que cette architecture reste la plus générique et flexible.



FIGURE 1.1 – Formation de robots qui changent de comportement

Dans ces dernières années, la coopération de nombreux robots a fait l'objet de nombreuses recherches[47][2], qui découlent des applications massives potentielles, en particulier de nombreuses applications militaires telles que l'arpentage, l'exploration, la surveillance et le contrôle des mines marines en général, et de nombreuses applications civiles telles que le tournage de cascades cinématographiques, le soutien de mines, patrouilles de pipelines et de lignes électriques, recherche et sauvetage, pulvérisation aérienne de fermes, suppression des incendies de forêt, soutien pétrolier en mer, etc. En fait, ces systèmes présentent de nombreux

avantages potentiels sur un seul robot, notamment l'amélioration de l'efficacité des tâches en termes de temps et de qualité, la réalisation de tâches non exécutables avec un seul robot ou la flexibilité de l'exécution des tâches, qui est également hautement adaptative, peu coûteuse et facile à entretenir, etc. En tant que problème très typique du contrôle collaboratif, le contrôle de la formation de plusieurs systèmes de robots a reçu une grande attention, ce qui vise à forcer les robots à converger vers un modèle qui permet de maintenir la formation des robots mobiles dans un environnement encombré d'une ingénierie spécifique. Nous pouvons remarquer que l'idée de coopération entre des systèmes à multiples facettes est largement inspirée par la nature et le monde qui nous entoure. Dans ces systèmes très puissants, les individus suivent des leaders distants sans frapper leurs voisins, y compris, mais sans s'y limiter, les groupes de poissons dans la mer et les troupes d'oiseaux dans le ciel, comme le montre la figure(1.2).



FIGURE 1.2 – Des comportements de formation dans la nature,(a) les groupes de poissons (b) les troupes d'oiseaux

De même, comme le montre la figure(1.3), dans de nombreuses applications de robots multiples, telles que les véhicules aériens sans pilote, les véhicules sous-marins autonomes et les systèmes des robots mobiles (SRM), en utilisant la formation de multiples robots pour atteindre un objectif offrent des avantages évidents.

1.2 Approche

L'étude des systèmes multi-robots mobiles (leader-suiveurs) nous conduit essentiellement vers l'analyse des lois de coordination et de coopération entre les robots. La synchronisation, la coordination et la coopération sont souvent des sujets intimement liés, surtout dans les systèmes mécaniques où elles sont utilisées comme synonymes pour décrire le même genre de comportement. Toutefois, dans la littérature, on note parfois la présence d'un concept de coordination légèrement différent de celui de la coopération des systèmes multi-robots.

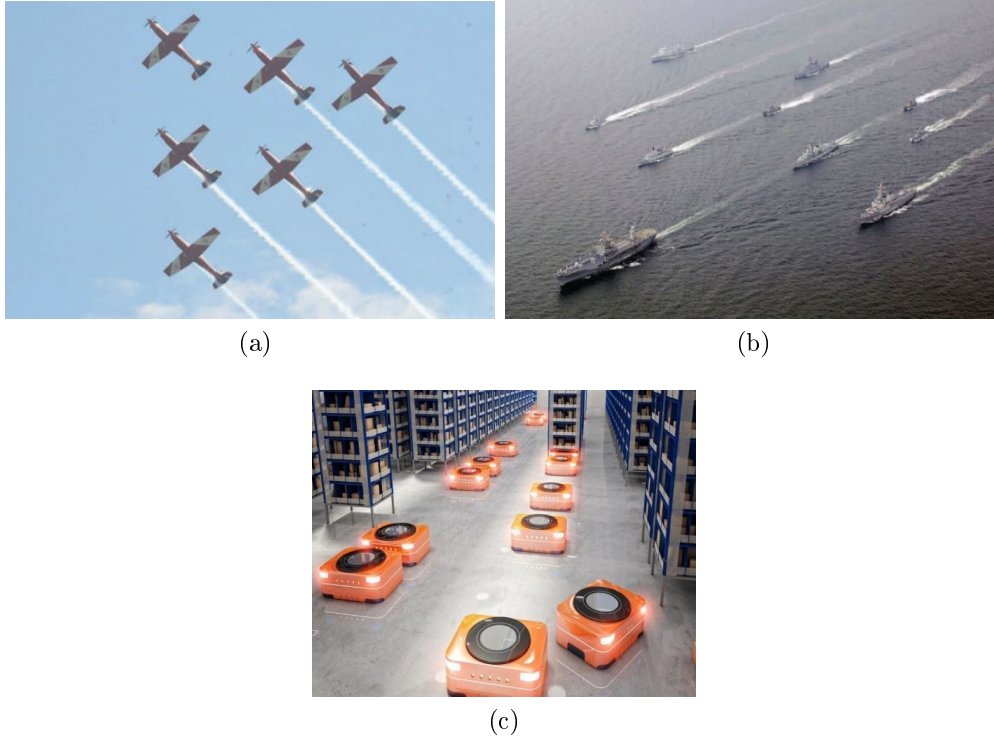


FIGURE 1.3 – formation (a) multi-véhicules aériens sans pilote (b) multi-véhicules sous-marins autonomes (c) systèmes de multi-robots mobiles

1.2.1 Structure de contrôle

Il est bien connu que la robustesse des systèmes multi-robots mobiles est strictement liée à la structure de contrôle utilisée pour organiser les robots et obtenir les comportements de formation souhaités. Dans le domaine de contrôle de la formation des robots mobiles, les structures de contrôle peuvent être identifiées comme une structure de contrôle centralisée et une structure de contrôle distribuée.

a) La structure de contrôle centralisée

Dans la structure de contrôle centralisée, une seule unité de calcul traite toutes les informations nécessaires pour atteindre les objectifs de contrôle souhaités. Par conséquent, ils peuvent idéalement donner des performances supérieures et des décisions optimales pour les membres individuels et la formation dans son ensemble [2][3]. Dans les dernières années, de nombreuses études connexes sur le contrôle de la formation de plusieurs robots mobiles avec une structure de contrôle centralisée ont été discutées. Une stratégie de coordination pour déplacer un groupe de robots dans une formation souhaitée sur une trajectoire donnée a été proposée [4]. Une approche de synchronisation pour le suivi de la trajectoire de plusieurs

robots mobiles tout en conservant des formations variables dans le temps [5]. Les avantages d'une structure centralisée incluent généralement une convergence plus rapide et une stabilité améliorée. Ces avantages s'accompagnent d'un coût financier plus élevé en raison des ressources de traitement et de communication requises par l'unité de calcul unique. Bien que ceux-ci garantissent une solution complète, les schémas de contrôle centralisé nécessitent une puissance de calcul plus élevée et ils sont moins robustes en raison de la forte dépendance à l'égard d'un seul contrôleur.

b) La structure de contrôle distribuée

La structure de contrôle distribuée est la structure la plus utilisée pour contrôler les systèmes multi-robots. Dans un système de robots mobiles avec une structure de contrôle distribuée, chaque robot agit uniquement en fonction de la connaissance de l'état des coéquipiers locaux et de l'environnement, et communication limitée entre les robots. De nombreux travaux de recherche sur le contrôle de la formation de plusieurs robots mobiles avec une structure de contrôle distribué sont basés sur les modèles simplifiés des robots mobiles. Il existe de nombreux résultats liés au contrôle de la formation à l'aide d'une structure de contrôle distribuée [6][7][8][9][10]. Les avantages de cette structure ont généralement un coût financier moindre et fonctionne mieux pour des systèmes plus grands qu'une structure centralisée. Cependant, cette structure peut entraîner une stabilité réduite et une convergence plus lente.

Modules	Architecture centralisée	Architecture distribuée
Perception et localisation	L'unité centrale utilise des capteurs centralisés et calcule la localisation absolue dans l'environnement de chaque robot. Les robots ne connaissent pas leur environnement	Les robots utilisent leurs capteurs embarqués et se localisent de façon relative par rapport à leurs consignes.
Décision	L'unité centrale décide, calcule et génère directement des variables de commande qu'elle envoie aux robots.	Les robots génèrent leurs propres consignes de type chemin, trajectoire, points de passages, etc.
Action	Les entités robotiques appliquent directement les variables de commande calculées par l'unité centrale à leurs moteurs.	Les robots assurent le respect des consignes qu'ils ont générées à travers des lois de commande avec un asservissement en boucle fermée sur ces consignes.

TABLE 1.1 – Architecture centralisée versus distribuée[11]

Alors chaque type d'architectures de contrôle présente ses avantages et ses inconvénients. Il est alors difficile, voire impossible, de privilégier un type plutôt qu'un autre sans connaître la nature de la mission confiée aux systèmes multi-robots ni le contexte de sa réalisation. En pratique, beaucoup de systèmes multi-robots n'ont pas un contrôle strictement centralisé ou strictement distribué et utilisent des architectures de contrôle hybrides centralisées/distribuées pour bénéficier des avantages des deux approches Tableau 1.1.

On a vu les deux approches principales utilisées, nous verrons dans les chapitres suivants qu'on cherche à rendre l'architecture de contrôle proposée la plus distribuée possible nous allons nous focaliser sur les architectures de contrôle distribuées.

1.2.2 Approches de contrôle

Concernant les approches de contrôle qui peuvent être grossièrement classées comme approche basée sur le comportement, approche leader-suiveur et potentiel artificiel ont été mises au point pour le contrôle de la formation de plusieurs robots mobiles[4][5][7][11][12].

a) Approche basée sur le comportement

Dans le contrôleur basé sur le comportement, chaque comportement individuel est en fait un sous-contrôleur pour atteindre un certain objectif (généralement au niveau élémentaire), par exemple, "se conformer à un motif géométrique spécifié" peut être un comportement et "éviter les obstacles" peut être un autre comportement. Si un robot s'exécute sur ces deux comportements, il finira par atteindre une combinaison d'objectifs, à savoir : « se former selon un motif géométrique spécifié tout en évitant les obstacles », cette approche [11] est généralement utilisée dans les applications de robotique comportementale.

b) Approche du potentiel artificiel

L'obstacle dans l'environnement produit une force répulsive qui éloigne le robot des obstacles, le point cible produit une force attractive qui tire le robot vers le point cible. La force de répulsion et la force d'attraction sont décrites comme une fonction potentielle répulsive et une fonction potentielle attractive. Généralement, ces deux types de fonctions potentielles sont utilisés ensemble dans des applications pratiques pour satisfaire la convergence, sans collision et sans obstacle. En fait, le robot se déplacera dans la direction qui minimise l'énergie potentielle. À nos connaissances, le concept de potentiel artificiel est d'abord proposé par J. Borenstein et Y. Koren [13]. Depuis lors, de nombreux résultats [14] ont étudié le problème de génération de modèles pour plusieurs robots mobiles dans un cadre de champ potentiel. Mis en œuvre la ségrégation de groupe dans un cadre potentiel de terrain. La formation et la

reconfigurable de motifs dans un système de robots mobiles multiples utilisant la théorie de la bifurcation pour démontrer que par un simple changement de paramètre une formation de robots peut être faite pour modifier leur configuration gratuitement. Une étape importante dans les systèmes d'ingénierie réels consiste à s'assurer que la formation peut se former de manière fiable.

c) Approche leader-suiveur

Un robot agit en tant que leader qui génère la trajectoire de référence pour le groupe de robots, et les autres robots du groupe agissent en tant que suiveurs qui doivent conserver la séparation et la position relative souhaitées par rapport au leader [7][2][15]. En fait, une fois que le mouvement du leader est donné, la séparation et l'orientation souhaitées du suiveur avec le leader peuvent être obtenues en choisissant une loi de commande locale sur chaque suiveur en fonction de sa dynamique de position relative. Ensuite, la stabilité de la formation est également garantie, c'est-à-dire que l'ensemble du groupe peut atteindre et maintenir la formation souhaitée. Sur la base de l'observation, le problème de contrôle de la formation peut être essentiellement considéré comme une extension naturelle du problème traditionnel de suivi de trajectoire. Seuls quelques chercheurs ont examiné le problème de suivi de trajectoire lorsqu'ils ont traité le problème de formation multi-robots. [8][2] ils ont présenté une méthode de contrôle linéaire à rétroaction pour la formation de robots mobiles non-holonomes naviguant sur un terrain avec des obstacles en utilisant l'approche leader-suiveur, et ont proposé deux algorithmes de contrôle : le contrôle $(l - \psi)$ (séparation-orientation) et le contrôle $(l - l)$ (séparation séparation). Dans le cas $(l - \psi)$ les variables de configuration

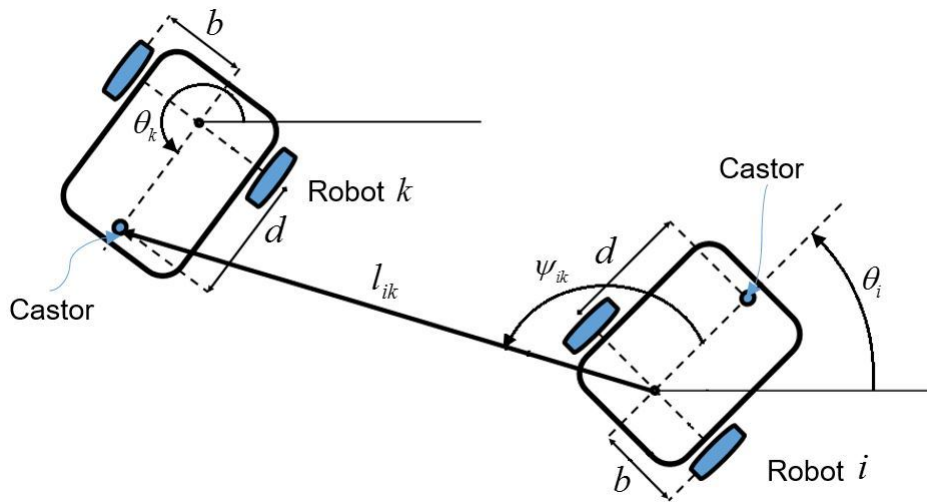


FIGURE 1.4 – Contrôle de formation $(l - \psi)$ [7]

du robot commandé sont donnés par $(l_{ik}, \psi_{ik}, \theta_{ik})$ la séparation souhaitée l_{ik}^d et l'angle relatif

ψ_{ik}^d entre le robot leader et le robot suiveur, comme il est indiqué sur la figure(1.4) pour deux robots mobiles à roues non-holonomes et dans le cas $l - l$ le contrôle est donne par $(l_{ik}, l_{jk}, \theta_k)$, où i, j se réfère à l'indice pour leader robot/robot et k se réfère à l'indice pour le robot contrôlé la séparation souhaitée à ses deux leaders. Le but est de contrôler et de maintenir les séparations souhaitées l_{ik}^d et l_{jk}^d entre le suiveur et ses deux leaders, comme le montre la figure(1.5) Pour une longue chaîne de robots mobiles avec une formation particulière, le 1er robot de la chaîne prévoit un chemin optimal, en évitant les obstacles. Sa vitesse et les informations de trajectoire sont transmises à la 1ère génération du robots dans la chaîne puis ils agissent à leur tour comme les robots de leader pour les générations suivantes en bas de la chaîne.

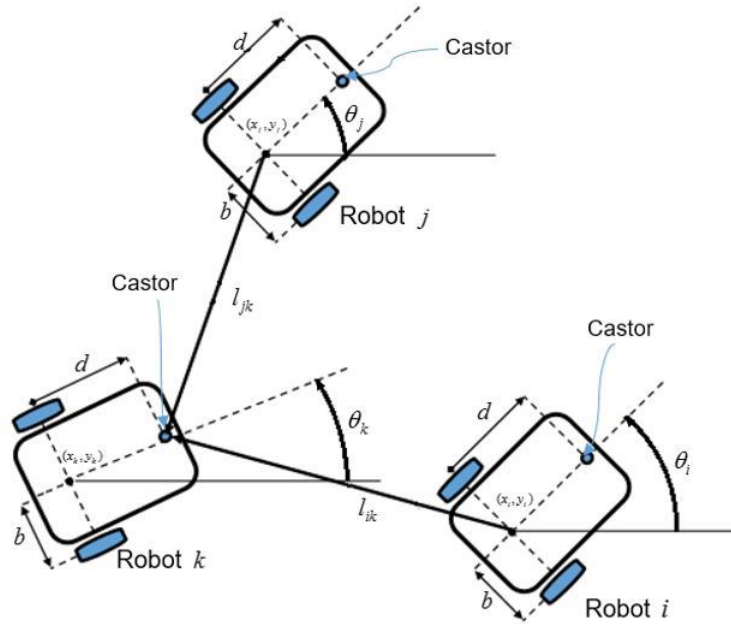


FIGURE 1.5 – Contrôle de formation ($l - l$)[15]

D'autres approches sont également utilisées pour le contrôle de la formation, à savoir l'approche de la théorie des graphes [7] et l'apprentissage par renforcement [16], etc. Les approches de suivi des dirigeants ont fait l'objet d'une attention croissante. Bien qu'elles soient critiquées pour un «point d'échec unique» [9], l'approche leader-suiveur est toujours devenue populaire car elle peut être explorée empiriquement et la stabilité de la formation interne peut être théoriquement garantie.

1.3 Les systèmes multi-robots

De nos jours, des recherches approfondies se concentrent sur les systèmes multi-robots. Ces systèmes offrent de nombreux avantages, car ils ont un potentiel élevé pour résoudre une multitude de problèmes tels que les entrepôts industriels, la surveillance et les missions de recherche et le sauvetage. Les systèmes multi-robots offrent une fiabilité et des performances élevées pour les tâches complexes, même si leur conception est simple. En 1989, Brooks et Flynn du laboratoire (MIT's Artificial Intelligence) ont proposé ce qu'ils ont appelé une idée radicale dans le domaine de l'exploration du système solaire : remplacer un grand robot explorateur par une collection de petits robots mobiles appelés "**rovers**" [17]. Ils ont estimé que le coût par kilogramme des "rovers" serait fortement réduit. Diminuer un peu la fiabilité pour chaque rover mobile serait acceptable, du moment que l'échec d'un rover unique ne mettrait pas en péril toute la mission. En effet, la mission pourrait être prévue avec une estimation de la fiabilité inférieure à 100%.



FIGURE 1.6 – Exemples de différents types de tâches coopératives multi-robots

Après l'atterrissage, soit ensemble ou en petits groupes, les rovers se disperseront en couvrant une large surface. Bien qu'aujourd'hui l'application des systèmes multi-robots n'est pas très répandue notamment dans l'espace. Cette idée radicale présente la principale motivation pour l'utilisation de systèmes multi robots, en effet, à plusieurs ils peuvent se disperser plus rapidement et couvrir un large espace pour les missions de reconnaissance, ou bien transporter des objets dispersés dans une zone de stockage. Ils peuvent même être utilisés dans des missions sous-marines comme la cartographie des fonds des mers, la récupération des balises échoués ou encore les boites noires suite au crash d'avions (figure1.6).

1.3.1 Commande non-linéaire

En adoptant cette idée, différentes méthodes de contrôle ont été développées pour les systèmes multi-robots, à savoir le contrôle prédictif [18], le contrôle adaptatif [19], linéarisation par bouclage [7] et contrôle du mode glissant [9][20], pour n'en nommer que quelques-uns. En tant qu'outil de conception à rétroaction non-linéaire, la méthode de contrôle en mode glissant (CMG) est robuste contre les incertitudes, certaines techniques basées sur CMG ont été étudiées pour le problème de formation-contrôle, c'est-à-dire un CMG adaptatif [21], un CMG intégral [22], CMG de premier ordre [23] et CMG de second ordre [20]. Dans les applications réelles, les systèmes de contrôle souffrent inévitablement d'incertitude. Les robots peuvent souffrir dans un essai d'incertitude, c'est-à-dire des fluctuations des paramètres internes, des dynamiques atypiques et des perturbations externes. Enchaînée à de tels doutes, la dynamique de formation devient incertaine. L'incertitude de formation dégrade non seulement les performances de configuration, mais peut également nuire à la stabilité de la configuration. Cependant, la méthode CMG, en tant qu'outil robuste et robuste, a une nature invariante aux incertitudes correspondantes lorsqu'un système entre en mode glissant. Malheureusement, l'incertitude de la composition est de nature inégalée et ne peut être contrôlée par la stabilité du CMG [22]. Leur présence remet grandement en cause le problème du contrôle de configuration.

1.3.2 Contrôle d'un robot mobile

Le contrôle d'un robot mobile nécessite naturellement la perception totale ou partielle de l'environnement ou il se trouve (figure 1.7), ainsi que sa localisation dans cet environnement, afin de décider de l'action à entreprendre. On retient alors trois éléments principaux indispensables au contrôle du robot mobile : la perception, la décision et l'action.

a) La perception

Dans le robot mobile les modules de perception sont chargés de la transformation des données, provenant des capteurs (caméra, le capteur GPS, ultrason, infrarouge, laser etc.), en données symboliques. Il contient également des modules pour atteindre un point ou suivre une trajectoire, ou alors éviter un obstacle gênant, le robot mobile dispose généralement de capteurs permettant de récupérer des stimuli appropriés afin d'alimenter un modèle et produire une représentation de cet environnement.

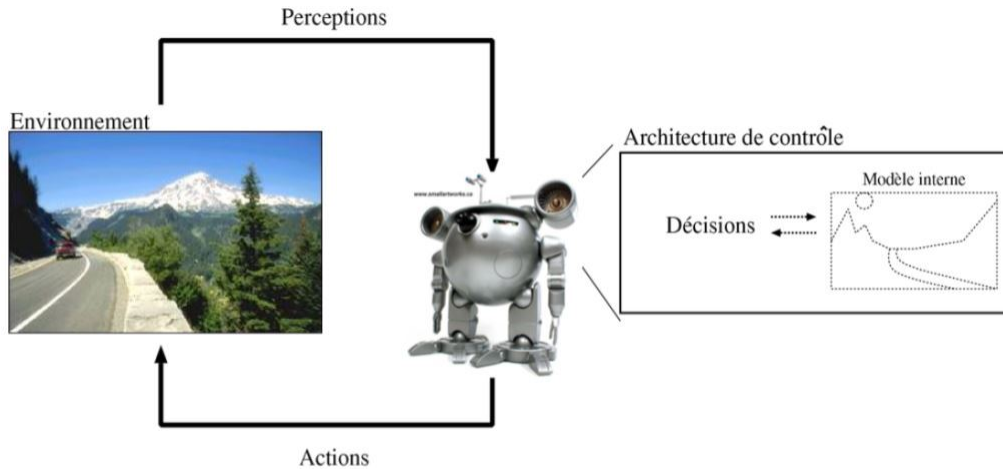


FIGURE 1.7 – Schéma des interactions d'un robot avec son environnement

b) La décision

L'autonomie du robot mobile est une faculté qui lui permet de s'adapter ou de prendre une décision afin d'atteindre un point de coordonnées connues dans un environnement peut connu. La décision se fait selon en fonction de la tâche à accomplirait du contexte d'exécution en fonction des informations locales de l'environnement délivrées par ses propres capteurs et de sa communication avec ses voisins (détermine par des éléments de perception).

c) L'action

Le robot mobile peut être représenté par un schéma d'asservissement en automatique. Et le bloc loi de commande génère la commande à appliquer aux actionneurs du bloc actionneur en fonction des informations fournies par les capteurs (la perception) et consigne (décision) (figure1.8).

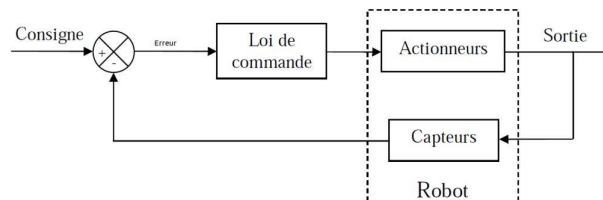


FIGURE 1.8 – Schéma d'un asservissement d'un robot mobile

1.3.3 Modélisation

Pour modéliser le déplacement d'un robot mobile en tenant compte des contraintes de roulement sans glissement de ses roues, un repère convenable doit être judicieusement choisi. Chaque roue individuelle contribue au mouvement du robot, et en même temps, impose des contraintes au mouvement du robot. Les roues sont liées entre elles en fonction de la géométrie du châssis du robot, et donc leurs contraintes se combinent pour former des contraintes sur le mouvement global du châssis du robot (figure 1.9).

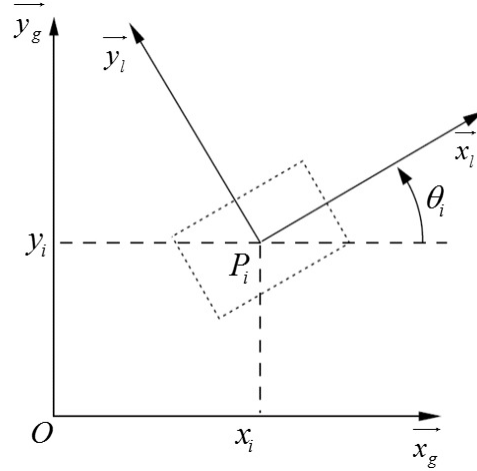


FIGURE 1.9 – Repérage d'un robot mobile

Il est possible de considérer que le robot se déplace sur un plan horizontal avec deux (2) degrés de mobilité : le premier correspond à un mouvement longitudinal (déplacement à l'avant ou à l'arrière) tandis que le deuxième est celui de la rotation autour d'un axe vertical. On a le repère global (repère absolu) $R_g(O, x_g, y_g, z_g)$ et un repère local (repère relatif) $R_l(P_i, x_l, y_l, z_l)$ est associé au robot mobile i l'origine P_i correspond au centre du robot i , à position du robot et son orientation sont définis en exprimant la relation entre le repère local et un repère spécifiée par (x_i, y_i) correspond aux coordonnées du point P_i dans le repère global $R_g(O, x_g, y_g, z_g)$ et son orientation θ_i est l'angle compris entre l'axe x_g et x_l . Nous pouvons décrire la pose du robot comme un vecteur avec ces trois éléments :

$$\xi_i = \begin{pmatrix} x_i \\ y_i \\ \theta_i \end{pmatrix} \quad (1.1)$$

La configuration [24] d'un système mécanique est connue quand la position de tous ses points dans un repère donné est connue. Alors que pour un bras manipulateur cette notion est définie sans ambiguïté par les positions angulaires des différentes articulations, on peut,

dans le cas d'un robot mobile, donner une vision plus ou moins fine de la configuration, comme on le verra par la suite. Dans tous les cas, on définit la configuration du robot mobile par un vecteur :

$$q = \begin{pmatrix} q_1 \\ q_2 \\ . \\ . \\ q_n \end{pmatrix} \quad (1.2)$$

De n coordonnées appelées coordonnées généralisées. La configuration est ainsi définie sur un espace N de dimension n , appelée l'espace des configurations. La matrice $R(\theta_i)$ est la matrice de rotation [24] permettant d'exprimer la variation des rotations du robot i dans le repère global $R_g(O, x_g, y_g, z_g)$ est définie comme suit :

$$R(\theta_i) = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 \\ \sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.3)$$

a) Contraintes non-holonomes et roulement sans glissement

Les robots mobiles non-holonomes ont une structure et un mouvement très similaire aux véhicules de notre vie courant : voiture particulière, les bus ... ; En générale leur structure mécanique est simple. Ils ont des roues motrices, des roues directrices et des roues porteuses, même on peut trouver des roues avec deux ou trois fonction combinées, dans de bonnes conditions, ces roues ont un roulement sans glissement (RSG) sur le sol, c'est-à-dire que la vitesse relative de la roue par rapport au sol au point de contact est nulle. Théoriquement, pour vérifier cette condition, il faut réunir les hypothèses suivantes :

- Le contact entre la roue et le sol est ponctuel.
- Un rayon fixe r pour les roues qui sont indéformables.
- Le sol est parfaitement plan.

Mais en pratique :

- Le contact se fait sur une surface.
- Hypothèse d'indéformabilité : fausse avec roues équipées de pneus.
- Contact continu avec le sol : essentiel pour l'odométrie d'un robot.

Mathématiquement, on peut traduire la condition de RSG sur une roue. Soit $P_r(x, y, r)$ le centre de la roue, $Q_r(x, y, 0)$ le point de contact de la roue avec le sol, φ est l'angle de rotation propre de la roue et θ l'angle entre le plan de la roue et le plan $(O, \vec{x}, \vec{y})$ comme est indiqué

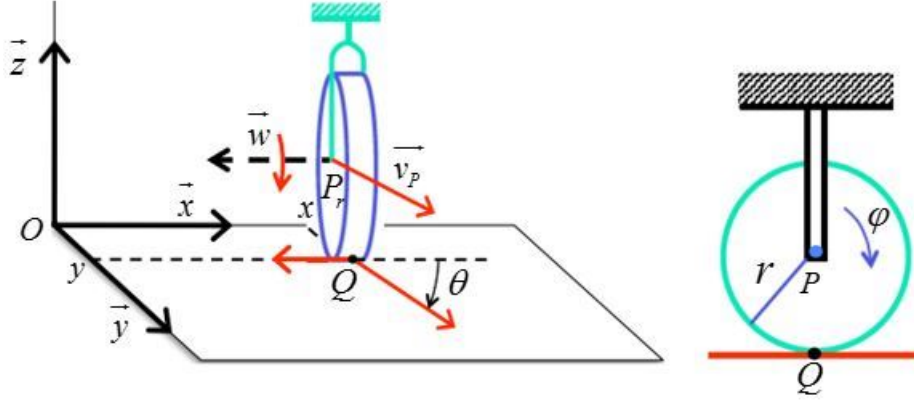


FIGURE 1.10 – Caractérisation du roulement sans glissement

sur la figure(1.10). La nullité de la vitesse relative $\vec{v}_Q$ roue/sol au point de contact permet d'obtenir une relation vectorielle entre la vitesse $\vec{v}_{P_r}$ du centre P_r de la roue et le vecteur de vitesse de rotation $\vec{\omega}$ de la roue :

$$\vec{v}_Q = \vec{v}_{P_r} + \vec{\omega} \times \overrightarrow{P_r Q} = 0 \quad (1.4)$$

Si on utilise l'expression des points $P_r Q$, on trouve :

$$\dot{x}\vec{x} + \dot{y}\vec{y} + (\dot{\theta}\vec{z} + \dot{\varphi}(\vec{x}\sin\theta - \vec{y}\cos\theta))\Lambda(-r\vec{x}) = \vec{0} \quad (1.5)$$

Si on calcule le produit vectoriel de l'équation (1.5), on a que :

$$(\dot{x} + r\dot{\varphi}\cos\theta)\vec{x} + (\dot{y} + r\dot{\varphi}\sin\theta)\vec{y} = \vec{0} \quad (1.6)$$

Ceci nous donne le système de contraintes scalaires suivant :

$$\begin{aligned} \dot{x} + r\dot{\varphi}\cos\theta &= 0 \\ \dot{y} + r\dot{\varphi}\sin\theta &= 0 \end{aligned} \quad (1.7)$$

Avec $\dot{\varphi}$ est la vitesse angulaire de roue et r le rayon de roue.

On peut transformer ces contraintes pour faire apparaître les composantes de vitesse dans le plan de la roue d'une part et perpendiculairement à la roue d'autre part :

— Contrainte de glissement :

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \quad (1.8)$$

— Contrainte de roulement :

$$\dot{x}\cos\theta + \dot{y}\sin\theta = r\dot{\varphi} \quad (1.9)$$

les équations(1.8) et (1.9) traduisent le fait que $\overrightarrow{v_{P_r}}$ ait dans le plan de la roue et ait pour module, il s'appelle contraintes non-holonomes. Et le modèle cinématique d'un robot mobile, qui peut être obtenu à partir des contraintes non-holonomes, peut être décrit comme :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (1.10)$$

Ce qui s'écrit, sous forme matricielle :

$$\dot{\xi} = C(q)u \quad (1.11)$$

Avec :

$$C(q) = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \quad (1.12)$$

Et $u = \begin{pmatrix} v & \omega \end{pmatrix}^T$ la commande

Dans le cadre de notre projet, nous nous intéresserons plus particulièrement aux contraintes de non-holonyme caractérisant entre autres les déplacements d'un robot mobile de type unicycle.

b) Modèle cinématique d'un robot unicycle

Les robots mobiles utilisés dans notre application sont des robots mobiles de type unicycle. Le robot mobile unicycle est composé de deux roues motrices indépendantes à l'arrière du corps principal du robot et une roue non commandée à l'avant ou à l'arrière (roue folle) (figure1.11(a)) cette dernière est destinée uniquement à assurer la stabilité du robot. Il existe d'autres robots qui peuvent être classés dans la catégorie des robots mobiles de type unicycle, notamment le robot mobile avec deux roues motrices indépendantes au milieu du corps principal du robot (figure1.11(b)).

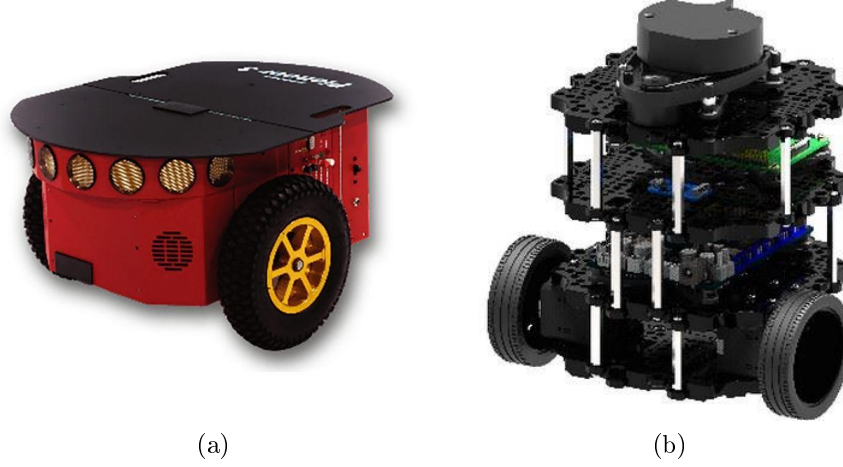


FIGURE 1.11 – (a) Robot mobile Pioneer P3-DX (b) Robot mobile Turtlebot3

Pour modéliser le déplacement du robot mobile en tenant compte de ces contraintes ; la contrainte de roulement pure représente le fait que chaque roue maintient un point de contact Q avec le sol tel que représenté sur la figure(1.10). Il n'y a pas de glissement de la roue dans son axe longitudinal y_l ni de dérapage dans son axe orthogonal x_l . Les vitesses des points de contact dans le cadre du robot sont liées aux vitesses de roue :

$$\begin{aligned} v_D &= -r\dot{\phi}_D \\ v_G &= r\dot{\phi}_G \end{aligned} \quad (1.13)$$

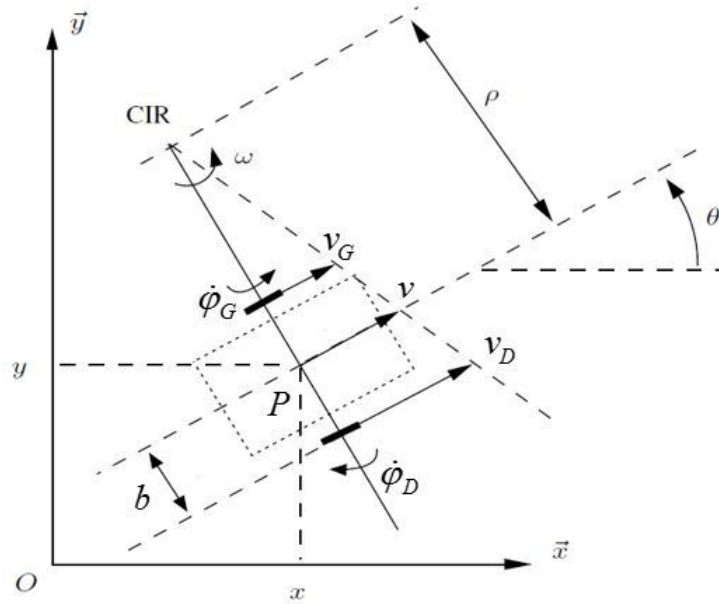


FIGURE 1.12 – Centre instantané de rotation d'un robot de type unicycle

Les roues ont le même axe de rotation et le centre instantané de rotation (CIR) du robot est un point de cet axe (figure 1.12). Soit ρ le rayon de courbure de la trajectoire du robot, b l'entraxe et ω la vitesse de rotation du robot autour du CIR. Les vitesses des roues droite et gauche sont :

$$\begin{aligned} v_D &= -r\dot{\varphi}_D = (\rho + b)\omega \\ v_G &= r\dot{\varphi}_G = (\rho - b)\omega \end{aligned} \quad (1.14)$$

Ce qui permet de définir ρ et ω à partir des vitesses des roues :

$$\begin{aligned} \rho &= b \frac{\dot{\varphi}_D - \dot{\varphi}_G}{\dot{\varphi}_D + \dot{\varphi}_G} \\ \omega &= -r \frac{\dot{\varphi}_D + \dot{\varphi}_G}{2b} \end{aligned} \quad (1.15)$$

Pour élaborer une stratégie plus fine de déplacement, il est intéressant de savoir comment la posture du robot est reliée à la commande de ses roues.

— Le Modèle d'actionnement

La modélisation d'actionnement est l'étude du mouvement des systèmes mécaniques sans considérer les forces qui affectent le mouvement. Le but principal de la modélisation d'actionnement est de relier la posture du robot à la commande de ses roues donc de représenter les vitesses du robot en fonction des vitesses des roues motrices ainsi que des paramètres géométriques du robot. A partir de ces deux équations (1.8) et (1.9) on peut alors déduire le modèle d'actionnement de notre robot. Le mouvement d'un robot à conduite différentielle est caractérisé par deux contraintes non-holonomes. On peut alors déduire le modèle d'actionnement de notre robot par :

$$\begin{aligned} v = \dot{x}_l &= \frac{v_D + v_G}{2} = r \frac{\dot{\varphi}_G - \dot{\varphi}_D}{2} \\ \omega = \dot{\theta} &= \frac{v_D - v_G}{2b} = -r \frac{\dot{\varphi}_D + \dot{\varphi}_G}{2b} \end{aligned} \quad (1.16)$$

Généralement, on définit le contrôle nécessaire du robot en établissant les vitesses linéaire et angulaire $u = (v, \omega)^T$ le contrôle des roues se déduit alors grâce au modèle décrit par l'équation (1.13). Le modèle cinématique inverse permet, quant à lui, de passer des vitesses opérationnelles v et ω aux vitesses de chaque roue. On admet les équations suivantes :

$$\begin{aligned} \dot{\varphi}_D = \omega_D &= -\frac{v + b\omega}{r} \\ \dot{\varphi}_G = \omega_G &= \frac{v - b\omega}{r} \end{aligned} \quad (1.17)$$

1.4 Évitement d'obstacles

L'évitement d'obstacles est un comportement quasiment essentiel dans un SMR. Il lui permet de naviguer en présence d'obstacles tout en gérant les écarts entre le modèle interne

du robot et le monde réel. Parmi ces méthodes il y a l'évitement d'obstacles par : champs de potentiel et la zone de déformation virtuelle. L'efficacité des deux méthodes que nous présentons dans ce mémoire a été prouvée dans les différents travaux de recherche. L'efficacité est conditionnée par l'existence d'une perception correcte de l'environnement (nature du capteur : laser, sonar, caméra).

1.4.1 Méthode des champs de potentiel

Dans la méthode d'évitement d'obstacles par champs de potentiels, on assimile le robot à une particule se déplace suivant les lignes de courant d'un potentiel créé en fonction de l'environnement perçu par le robot [25]. Ce potentiel traduit différents objectifs tels que l'évitement d'obstacles ou une direction de déplacement préférée. Il est calculé par sommation de différentes primitives de potentiels traduisant chacun de ces objectifs (figure1.13).

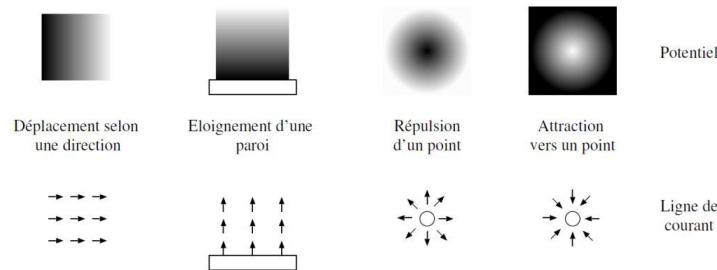


FIGURE 1.13 – Illustration de potentiels primitifs dont la combinaison guide les déplacements du robot. Le robot se déplacera selon les lignes de courant

Ces différents potentiels peuvent avoir une étendue spatiale limitée ou non (par exemple, n'avoir une influence que près des obstacles) et leur intensité peut dépendre ou non de la distance. Le gradient de ce potentiel donne en chaque point de l'espace la direction du déplacement du robot (figure1.13). Comme ce gradient, et non la valeur absolue du potentiel, qui nous intéresse, il est possible de calculer directement en chaque point sa valeur par une simple somme vectorielle en ajoutant les valeurs issues des différents potentiels primitifs. Ainsi, pour un robot se déplaçant en ligne droite dans un espace ouvert et évitant les obstacles qu'il peut rencontrer, nous obtenons par exemple les lignes de courant illustrées (figure1.14).

1.4.2 Méthode de la zone de déformation virtuelle (ZDV)

La méthode d'évitement d'obstacles par ZDV a été proposée initialement A. Cacitti et R. Zapata [26] pour le cas des manipulateurs mobiles. Cette méthode repose sur la théorie d'évitement réflexe. Cette théorie permet de se rapprocher au mieux d'un comportement

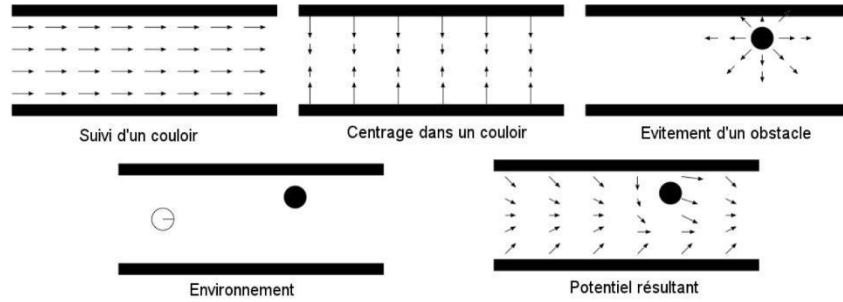


FIGURE 1.14 – De différents potentiels primitifs

humain, car le contrôle du mouvement d'un robot mobile dans un environnement complexe nécessite une intelligence et une autonomie. L'idée principale est de définir l'interaction robot/environnement comme une ZDV qui contourne le véhicule (figure 1.15). La déformation de la zone de sécurité est due à l'intrusion d'une information provenant d'un capteur de proximité.

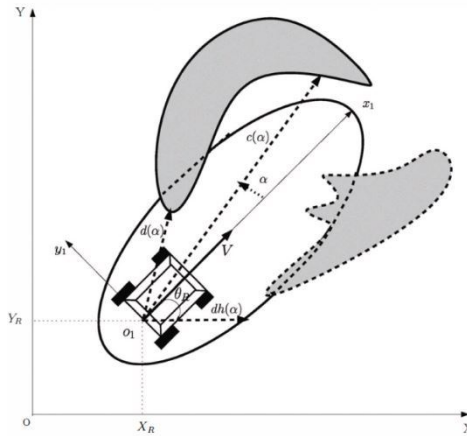


FIGURE 1.15 – Évitement d'obstacles : définition des cadres et description de la pose du problème.

1.4.3 Choix de la méthode d'évitement d'obstacles

L'étude bibliographique établie sur les deux différentes méthodes d'évitement d'obstacles, nous a permis d'opter pour le choix de la stratégie d'évitement par champs de potentiel. Ce choix s'explique par le fait que cette méthode est réactive et qu'elle génère des actions réflexes rapides. Ceci permet d'assurer une autonomie importante au robot dans le SMR.

1.5 Conclusion

Ce chapitre nous a permis en premier lieu d’avoir une introduction générale à la robotique mobile et à ses applications et les principales spécifiées des SMR. Étant donné que l’objectif de mémoire est de réaliser une architecture de contrôle hybride permettant le maintien de formation de groupe de robots mobiles dans un environnement encombré est de commuter entre la partie commande et la partie évitement d’obstacles, nous avons d’abord vu les différentes structures de contrôle existant dans la littérature, elles sont classifiées selon leur méthode d’implantation : centralisées ou distribuées leurs avantages et leurs inconvénients. Ce chapitre est aussi l’occasion de présenter les approches principales de contrôle et leur métrologie qui partage des problématiques communes à la coopération d’un groupe de robots mobiles d’une manière plus spécifique et que l’approche leader-suiveur est toujours devenue populaire car elles peu être explorée empiriquement et la stabilité de la formation interne peut être théoriquement garantie.

Chapitre 2

Contrôle de la formation leader-suiveurs de multi-robots non-holonomes

2.1 Introduction

Dans certaines applications, le problème du contrôle de la formation peut être essentiellement considéré comme une extension naturelle du problème traditionnel de suivi de trajectoire. Seuls quelques chercheurs ont examiné le problème de suivi du trajectoire lorsqu'ils traitent le problème de la formation multi-robots [47], dans ce chapitre, nous étendrons le problème traditionnel du contrôle de suivi du trajectoire pour un seul robot à un problème du contrôle de formation pour plusieurs robots mobiles. Cela a été étudié ces dernières années (voir [9][20][23][27]), on peut citer des méthodes de la commande par «Backstepping», cette méthode se base sur la construction progressive de fonctions des Lyapunov formée à partir des erreurs de poursuite, la commande par bouclage linéarisant exacte (connue également sous le nom de « feedback linearization ») [28] au sens entrée-état ou bien au sens entrée-sortie, Defoort.M, Floquet.T, Kokosy.A et Perruquetti.W (2008) ont présenté une méthode de commande non linéaire, qui est également très répandue dans la littérature et la commande par mode glissant [9][20][23]. Cette méthode se base sur la conception de contrôleurs à structure variable (VSC « Variable Structure Controllers » [29]) pour amener le système à suivre une dynamique bien définie. Parmi les méthodes ci-dessus, nous allons utiliser dans ce projet, cette dernière technique (mode glissant) de commande qui est très populaire auprès des automaticiens, grâce à sa simplicité d'implémentation et à sa grande robustesse vis-à-vis les incertitudes paramétriques, les erreurs de modélisation et les perturbations et la stabilité du système peuvent être garanties par la théorie de la stabilité de Lyapunov [30].

Cependant, dans notre système de formation leader-suiveur, les trajectoires des suiveurs ne sont généralement pas prédéfinies, elles sont décidées par leur leader en temps réel. Ce

chapitre étend le contrôle de suivi de trajectoire pour un seul robot mobile non-holonyme au contrôle de formation pour plusieurs robots mobiles non-holonomes basé sur la technique de mode glissant, dans lequel le suiveur peut suivre son leader en temps réel par le contrôleur cinématique proposé.

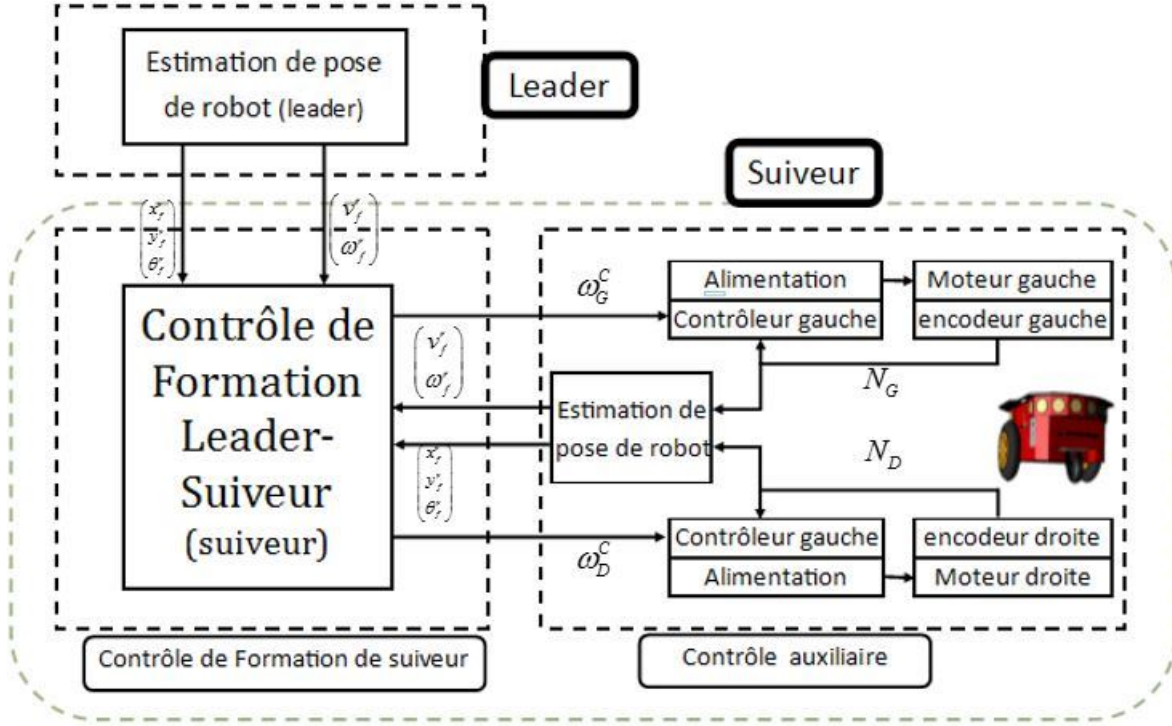


FIGURE 2.1 – Schéma de principe de la structure de contrôle droite

Enfin, en raison de la contrainte non-holonyme de chaque robot et l'objectif de contrôle de la formation leader-suiveur, une loi auxiliaire de contrôle de la vitesse angulaire des roues est développée pour garantir la stabilité asymptotique globale des adeptes et pour garantir davantage la stabilité asymptotique locale de toute la formation (figure 2.1).

2.2 Contrôle de formation leader-suiveur

2.2.1 Modèle mathématique d'un robot mobile non-holonyme

Considérons un groupe de n robots mobiles non-holonomes. Pour simplifier, nous supposons que chaque robot a la même structure mécanique que celle illustrée à la figure (2.1). La posture du $i^{\text{ème}}$ ($1 \leq i \leq n$) robot (nommé robot R_i) dans un cadre cartésien OXY est spécifiée par $p_i = \begin{pmatrix} x_i & y_i & \theta_i \end{pmatrix}^T$, avec (x_i, y_i) désigne la coordonnée de l'avant du robot R_i , θ_i est l'angle de cap de robot.

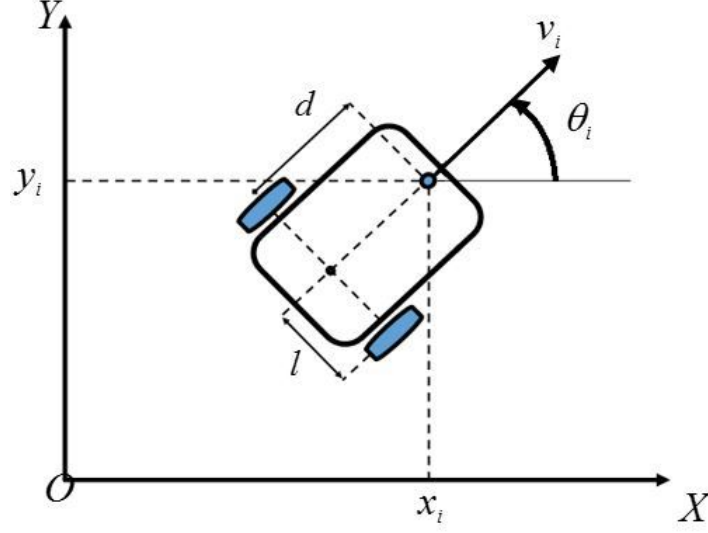


FIGURE 2.2 – Robot mobile non-holonomes

Le robot mobile à deux roues motrices illustré à la figure(2.2) est un exemple typique de systèmes mécaniques non-holonomes. Sous l'hypothèse du roulement pur et du non-glissement paragraphe (1.3.3), la contrainte cinématique du robot mobile non-holonyme R_i est donnée comme :

$$\dot{y}_i \cos \theta_i - \dot{x}_i \sin \theta_i = d \dot{\theta}_i \quad (2.1)$$

(d) est la distance de l'essieu arrière à l'avant du robot.

A partir de la contrainte cinématique et les deux équations (1.1) et (2.1), le modèle cinématique du robot mobile non-holonyme R_i peut s'écrire :

$$\dot{p} = \begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_i \\ \omega_i \end{pmatrix} \quad (2.2)$$

Avec v_i et ω_i sont la vitesse linéaire et la vitesse angulaire de robot R_i .

2.2.2 Contrôle de formation leader-suiveur et la dynamique d'erreur

a) Contrôle de formation leader-suiveur

D'après le schéma de contrôle de la formation représenté sur la figure(2.3), le robot R_f suit son leader R_l avec la séparation souhaitée l_{lf}^d , le relèvement souhaité ψ_{lf}^d et l'orientation souhaitée θ_f^d . Ici $\theta_{lf}^d = \theta_l$. Notez que $p_l = \begin{pmatrix} x_l & y_l & \theta_l \end{pmatrix}^T$ est la posture réelle du robot leader

R_l et (v_l, ω_l) sont ses entrées de contrôle, $p_f = \begin{pmatrix} x_f & y_f & \theta_f \end{pmatrix}^T$ est la posture réelle du robot suiveur R_f et (v_f, ω_f) sont ses entrées de contrôle, la position souhaitée du suiveur dans le système des coordonnées du leader est (x_d, y_d) , et la position réelle dans le système des coordonnées globales $p_f^d = (x_c, y_c)$ et en utilisant les équations (1.3) et on obtient l'équations suivantes :

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} x_l \\ y_l \end{pmatrix} + \begin{pmatrix} \cos \theta_l & -\sin \theta_l \\ \sin \theta_l & \cos \theta_l \end{pmatrix} \begin{pmatrix} x_d \\ y_d \end{pmatrix} \quad (2.3)$$

Avec $l_{lf}^d = \sqrt{x_d^2 + y_d^2}$ et $\psi_{lf}^d = \arctan(\frac{y_d}{x_d}) - \theta_l + \pi$

Grâce à la relation géométrique entre les robots [40], il est facile d'obtenir que la position souhaitée p_f^d du robot suiveur satisfait :

$$p_f^d = \begin{pmatrix} x_l - d \cos \theta_l + l_{lf}^d \cos(\psi_{lf}^d + \theta_l) \\ x_l - d \sin \theta_l + l_{lf}^d \sin(\psi_{lf}^d + \theta_l) \\ \theta_l \end{pmatrix} \quad (2.4)$$

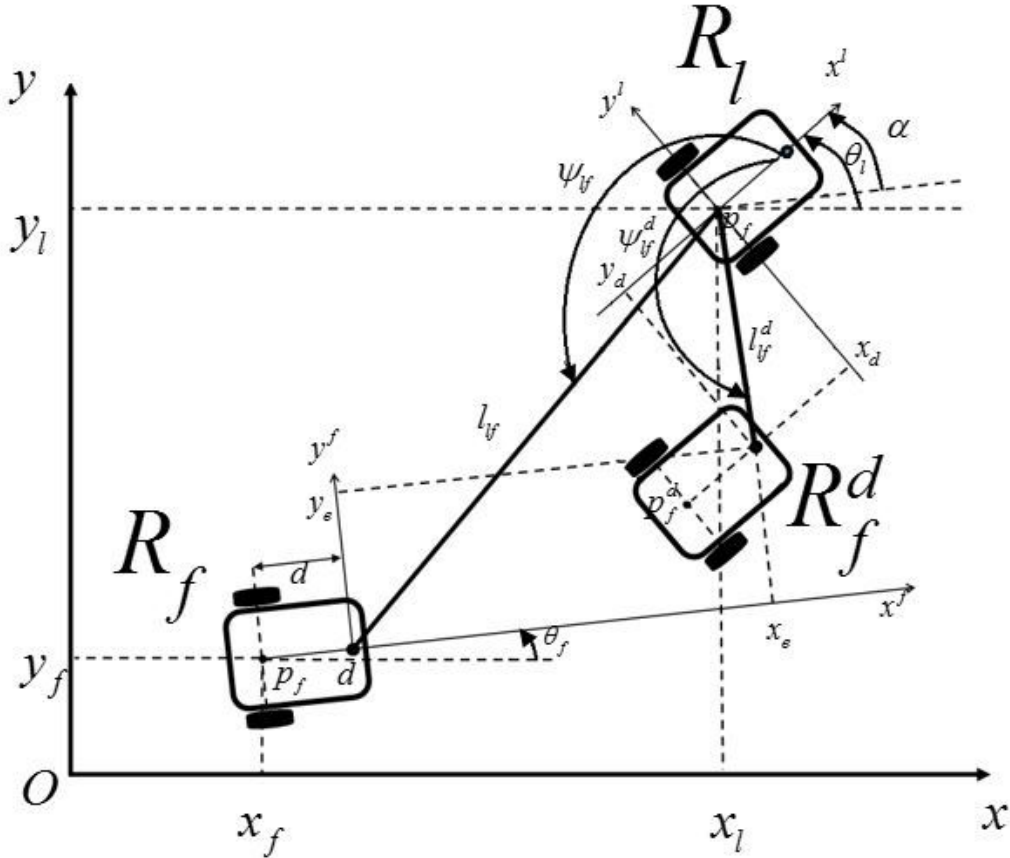


FIGURE 2.3 – Formation Leader-Suiveur

La posture réelle p_f du suiveur R_f satisfait

$$p_f = \begin{pmatrix} x_l - d \cos \theta_l + l_{lf} \cos(\psi_{lf} + \theta_l) \\ x_l - d \sin \theta_l + l_{lf} \sin(\psi_{lf} + \theta_l) \\ \theta_l \end{pmatrix} \quad (2.5)$$

Prenant la dérivée temporelle de l'équation (2.3); (x_d, y_d) sont des valeurs constantes et en utilisant les équations (1.10) et en appliquant des identités trigonométriques simples qui sont obtenues comme suit :

$$\begin{aligned} \dot{x}_c &= \dot{x}_l - x_d \dot{\theta}_l \sin \theta_l - y_d \dot{\theta}_l \cos \theta_l \\ \dot{y}_c &= \dot{y}_l + y_d \dot{\theta}_l \cos \theta_l - x_d \dot{\theta}_l \sin \theta_l \\ \dot{x}_c &= v_l \cos \theta_l - x_d \omega_l \sin \theta_l - y_d \omega_l \cos \theta_l \\ \dot{y}_c &= v_l \sin \theta_l + x_d \omega_l \cos \theta_l - y_d \omega_l \sin \theta_l \\ \dot{x}_c &= (v_l - y_d \omega_l) \cos \theta_l - x_d \omega_l \sin \theta_l \\ \dot{y}_c &= (v_l - y_d \omega_l) \sin \theta_l + x_d \omega_l \cos \theta_l \\ \begin{pmatrix} \dot{x}_c \\ \dot{y}_c \end{pmatrix} &= \begin{pmatrix} \cos \theta_l & -\sin \theta_l \\ \sin \theta_l & \cos \theta_l \end{pmatrix} \begin{pmatrix} v_l - y_d \omega_l \\ x_d \omega_l \end{pmatrix} \end{aligned} \quad (2.6)$$

Si la distance entre le point $O_f(x_{of}, y_{of}, \theta_f)$ (avant du robot) et le point C_f est d dans le système de coordonnées global, La relation peut être présentée comme suit :

$$\begin{aligned} x_{of} &= x_f + d \cos \theta_f \\ y_{of} &= y_f + d \sin \theta_f \end{aligned} \quad (2.7)$$

Prenant la dérivée temporelle de l'équation (2.7) et en utilisant les équations (2.2) nous obtenons comme suit :

$$\begin{aligned} \dot{x}_{of} &= \dot{x}_f - d \dot{\theta}_f \sin \theta_f \\ \dot{y}_{of} &= \dot{y}_f + d \dot{\theta}_f \cos \theta_f \\ \dot{x}_{of} &= v_f \cos \theta_f - d \dot{\theta}_f \sin \theta_f \\ \dot{y}_{of} &= v_f \sin \theta_f + d \dot{\theta}_f \cos \theta_f \end{aligned} \quad (2.8)$$

En utilisant les équations (2.6), (2.8) et (1.3), l'erreur de suivi pour la formation leader-suiveur dans le système de coordonnées du suiveur est obtenue comme :

$$\begin{pmatrix} x_{fe} \\ y_{fe} \end{pmatrix} = \begin{pmatrix} \cos \theta_f & \sin \theta_f \\ -\sin \theta_f & \cos \theta_f \end{pmatrix} \begin{pmatrix} x_c - x_{of} \\ y_c - y_{of} \end{pmatrix} \quad (2.9)$$

De plus, la différence angulaire entre le leader et le suiveur est notée comme (2.10).

$$\alpha = \theta_l - \theta_f \text{ et } \dot{\alpha} = \omega_l - \omega_f \quad (2.10)$$

Notre objectif est d'atteindre et de maintenir la formation leader-suiveur souhaitée, c'est-à-dire de maintenir la séparation-séparation et la séparation-roulement souhaitées entre le leader et le suiveur. Par conséquent, la tâche de contrôle consiste à concevoir les entrées de contrôle (v_f, ω_f) pour le robot suiveur $R_f (1 \leq f \leq n)$ de telle sorte que :

$$\lim_{t \rightarrow \infty} x_{fe} = 0, \lim_{t \rightarrow \infty} y_{fe} = 0 \text{ où } |\alpha| < \delta \ (\delta > 0) \quad (2.11)$$

$$\text{Donc } \lim_{t \rightarrow \infty} (p_f^d - p_f) = 0$$

b) La dynamique d'erreur de la formation leader-suiveur

Dans l'approche leader-suiveur, la vitesse angulaire et linéaire du leader est donnée, nous n'aurons besoin que de contrôler la vitesse angulaire et linéaire du suiveur pour maintenir la séparation relative et le relèvement relatif entre eux afin de satisfaire la formation souhaitée. Par conséquent, le leader suivant le contrôle de la formation de plusieurs robots mobiles basé peut être considéré comme une extension du problème de suivi du robot mobile non-holonyme [31].

En utilisant les équations (2.6), (2.8) et en appliquant des identités trigonométriques simples, l'erreur de suivi pour la formation leader-suiveur est obtenue comme : Supposons que (x_d, y_d) sont des valeurs constantes, Il s'ensuit alors que $(\dot{x}_d, \dot{y}_d)$. En prenant la dérivée temporelle de (2.9)

$$\begin{aligned} x_{fe} &= x_c \cos \theta_f - x_{of} \cos \theta_f + y_c \sin \theta_f - y_{of} \sin \theta_f \\ y_{fe} &= -x_c \sin \theta_f + x_{of} \sin \theta_f + y_c \cos \theta_f - y_{of} \cos \theta_f \\ \dot{x}_{fe} &= \dot{x}_c \cos \theta_f + \dot{y}_c \sin \theta_f - x_c \omega_f \sin \theta_f + y_c \omega_f \cos \theta_f \\ &\quad - \dot{x}_{of} \cos \theta_f - \dot{y}_{of} \sin \theta_f + x_{of} \omega_f \sin \theta_f - y_{of} \omega_f \cos \theta_f \\ \dot{y}_{fe} &= -\dot{x}_c \sin \theta_f + \dot{y}_c \cos \theta_f - x_c \omega_f \cos \theta_f - y_c \omega_f \sin \theta_f \\ &\quad + \dot{x}_{of} \sin \theta_f - \dot{y}_{of} \cos \theta_f + x_{of} \omega_f \cos \theta_f + y_{of} \omega_f \sin \theta_f \\ \dot{x}_{fe} &= (v_l - y_d \omega_l) \cos \alpha - x_d \omega_l \sin \alpha - v_f \\ &\quad + \omega_f (-x_c \sin \theta_f + x_{of} \sin \theta_f + y_c \cos \theta_f - y_{of} \cos \theta_f) \\ \dot{y}_{fe} &= (v_l - y_d \omega_l) \sin \alpha + x_d \omega_l \cos \alpha \\ &\quad - \omega_f (d + x_c \cos \theta_f - x_{of} \cos \theta_f + y_c \sin \theta_f - y_{of} \sin \theta_f) \end{aligned}$$

Ainsi, on peut obtenir la dynamique d'erreur du robot mobile comme suit :

$$\begin{pmatrix} \dot{x}_{fe} \\ \dot{y}_{fe} \\ \dot{\theta}_{fe} \end{pmatrix} = \begin{pmatrix} (v_l - y_d \omega_l) \cos \alpha - x_d \omega_l \sin \alpha - v_f + \omega_f y_e \\ (v_l - y_d \omega_l) \sin \alpha + x_d \omega_l \cos \alpha - \omega_f (d + x_e) \\ \omega_f^d - \omega_f \end{pmatrix} \quad (2.12)$$

2.3 Présentation de la commande

La modélisation des systèmes réels est plus ou moins précise. En effet, elle se fait en négligeant certains non linéarités et perturbations, souvent même, en réduisant le nombre de variables d'état du système. Et c'est à partir de cette représentation approximative de la réalité physique qu'on souhaite construire le plus simplement possible une commande robuste et performante. Lorsque la partie commandée du processus est faiblement perturbée, les algorithmes de commande classiques. L'une des méthodes de commande non linéaires les plus connues est la commande par bouclage linéarisant exacte au sens entrée-état ou bien au sens entrée-sortie, peuvent s'avérer suffisants si les exigences sur la précision et la performance du système ne sont pas trop strictes, nous utiliserons cette méthode dans la 1ere partie, pour les robots avec plates-formes non-holonomes contrôlées par la vitesse linéaire et angulaire. Les lois de commande sont motivées par des idées du domaine bien établi par bouclage linéarisant [32]. Cela signifie que nous pouvons réguler deux sorties. La cinématique du robot peut être résumée comme un modèle d'unicycle.

2.3.1 Contrôle par bouclage linéarisant

Le bouclage linéarisant est une approche de la conception de commande non linéaire qui a suscité un grand intérêt de recherche ces dernières années. L'idée centrale de cette approche est de transformer algébriquement une dynamique de système non linéaire en une dynamique (entièrement ou partiellement) linéaire, afin que des techniques de contrôle linéaire puissent être appliquées. Cela diffère entièrement de la linéarisation conventionnelle en ce que le bouclage linéarisant est obtenu par des transformations d'état et une rétroaction exacte, plutôt que par des approximations linéaires de la dynamique.

L'idée de simplifier la forme de la dynamique d'un système en choisissant une représentation d'état différente n'est pas totalement inconnue. En mécanique, par exemple, il est bien connu que la forme et la complexité d'un modèle de système dépendent considérablement du choix des référentiels ou des systèmes de coordonnées. Les techniques de bouclage linéarisant peuvent être considérées comme des moyens de transformer des modèles de système originaux

en modèles équivalents d'une forme plus simple. Ainsi, ils peuvent également être utilisés dans le développement de contrôleurs non linéaires robustes ou adaptatifs [32].

Nous avons considéré un robot mobile décrit par l'équation d'état (2.13)

$$\dot{x} = f(x, u) \quad x \in \mathbb{R}^n \quad u \in \mathbb{R}^m \quad (2.13)$$

L'idée est d'annuler les non-linéarités et d'imposer une dynamique linéaire souhaitée, peut-être simplement appliquée à une classe de systèmes non linéaires décrits par la forme dite compagnon ou forme canonique de contrôlabilité. On dit qu'un système est sous forme compagnon si sa dynamique est représentée par :

$$\begin{aligned} \dot{x}^{(k)} &= A(X)u + b(X) \\ y &= h(X) \end{aligned} \quad (2.14)$$

Où u est l'entrée de contrôle scalaire, $X = \left[x, \dot{x}, \dots, x^{(k-1)} \right]^T$ est le vecteur d'état, avec k le degré relatif (le nombre de fois de dérivé pour faire apparaître l'entrée u), $A(X)$ et $b(X)$ sont des fonctions non linéaires. La forme est unique dans le fait que, bien que des dérivées de x apparaissent dans cette équation, aucune dérivée de l'entrée u n'est présente. Notez que, dans la représentation de l'espace d'états, (2.14) peut être écrite :

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ \dots \\ x_{k-1} \\ x_k \end{bmatrix} = \begin{bmatrix} x_2 \\ \dots \\ x_k \\ A(x)u + b(x) \end{bmatrix} \quad (2.15)$$

Pour les systèmes qui peuvent être exprimés sous la forme canonique de contrôlabilité, en utilisant l'entrée de contrôle (en supposant que $A(X)$ est différent de zéro condition de singularité [32]) :

$$u = A^{-1}(X) [v - b(X)] \quad (2.16)$$

On peut annuler les non-linéarités et obtenir la relation simple entrée-sortie (forme intégrateur multiple) :

$$v = x^{(k)}$$

Avec le vecteur v est notre nouvelle entrée (figure 2.4) et de même dimension que u qui lui-même de même dimension que y , de cela, nous pouvons considérer notre système bouclé de commande comme suit :

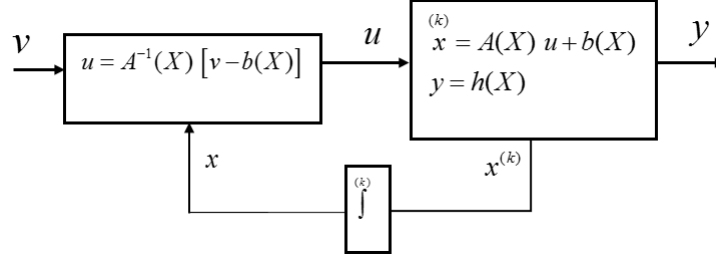


FIGURE 2.4 – Schéma de contrôle basé sur bouclage linéarisant

Nous utiliserons un régulateur de type PD (proportionnel et dérivées), l'utilisation de tel régulateur. Il est nécessaire de disposer des dérivées de sorties. Comme nous avons supposé avoir accès à toutes les variables d'état x du système une expression formelle de ces dérivées en fonction de x s'obtient facilement en utilisant les équations d'état.

a) Régulateur proportionnel et dérivées

On se propose tout d'abord de stabiliser ce système par un régulateur proportionnel et dérivées du type :

$$v = \alpha_0(w - y) + \alpha_1(\dot{w} - \dot{y}) + \dots + \alpha_{k-1}\left(\overset{(k-1)}{w} - \overset{(k-1)}{y}\right) + \overset{(k)}{w} \quad (2.17)$$

Où w est la consigne souhaitée pour y . Et w peut dépendre du temps comme notre système de leader-suiveur. Le fait que ce régulateur nécessite les dérivées de y n'est pas un problème dans le cadre défini par le bouclage linéarisant. En effet, toutes ces dérivées peuvent être écrites comme des fonctions analytiques de l'état x du système et de l'entrée u . En ce qui concerne la consigne $w(t)$, elle est choisie par l'utilisateur et une expression analytique de $w(t)$ peut-être supposée connue (par exemple $w(t) = \sin(t)$). Ainsi, le calcul des dérivées de w ce fait de façon formelle et aucune sensibilité de l'opérateur dérivation par rapport au bruit n'est craindre.

$$\overset{(k)}{x} = v = \alpha_0(w - y) + \alpha_1(\dot{w} - \dot{y}) + \dots + \alpha_{k-1}\left(\overset{(k-1)}{w} - \overset{(k-1)}{y}\right) + \overset{(k)}{w} \quad (2.18)$$

Si l'on définit l'erreur e entre la consigne w et la sortie y par $e = w - y$, cette équation devient :

$$\overset{(k)}{e} + \alpha_{k-1}\overset{(k-1)}{e} + \dots + \alpha_1\dot{e} + \alpha_0e = 0 \quad (2.19)$$

Cette équation différentielle est appelée dynamique de l'erreur.

b) Application de la commande au robot mobile non-holonyme

Après avoir présenté la théorie de la commande par le bouclage linéarisant et comme nous avons vu dans le paragraphe (2.2.2) le contrôle de leader-suiveur et l'erreur dynamique, et en raison des caractéristiques du système dynamique d'erreur (2.12) (1.1), le contrôle par le bouclage linéarisant peut être utilisé pour s'attaquer à ce modèle (2.12). En outre, en raison de la contrainte non-holonyme de chaque robot et de l'objectif de contrôle de la formation leader-suiveur. Par conséquent, on peut écrire le système dynamique d'erreur comme suit :

$$\begin{pmatrix} \dot{x}_{fe} \\ \dot{y}_{fe} \end{pmatrix} = \begin{pmatrix} -1 & y_e \\ 0 & -(d + x_e) \end{pmatrix} \begin{pmatrix} v_f \\ \omega_f \end{pmatrix} + \begin{pmatrix} (v_l - y_d \omega_l) \cos \alpha - x_d \omega_l \sin \alpha \\ (v_l - y_d \omega_l) \sin \alpha + x_d \omega_l \cos \alpha \end{pmatrix}$$

Considérer les entrées de commande de bouclage linéarisant suivantes :

$$\begin{pmatrix} v_f \\ \omega_f \end{pmatrix} = \begin{pmatrix} -1 & y_e \\ 0 & -(d + x_e) \end{pmatrix}^{-1} \left[V - \begin{pmatrix} (v_l - y_d \omega_l) \cos \alpha - x_d \omega_l \sin \alpha \\ (v_l - y_d \omega_l) \sin \alpha + x_d \omega_l \cos \alpha \end{pmatrix} \right] \quad (2.20)$$

Où V est la nouvelle entrée, notre système bouclé se réécrit :

$$V = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} \dot{x}_{fe} \\ \dot{y}_{fe} \end{pmatrix}$$

Pour prouver que le système de contrôle de suivi de trajectoire (2.12) sous la loi du contrôleur (2.20) est asymptotiquement stable et que l'erreur de suivi converge vers zéro, nous choisissons le contrôleur PD paragraphe (a).

c) Résultats des simulations sur Matlab

Dans cette simulation, l'efficacité des contrôleurs de suivi proposés (2.20), considérer un leader et un suiveur. Supposons que la distance d de l'essieu arrière à l'avant du robot soit de (0.2) , la vitesse linéaire et la vitesse angulaire du robot leader sont respectivement de (1,0.5), et la posture initiale du robot leader est à (0,0,0). Afin de conserver la formation de leader-suiveur, le suiveur doit conserver la séparation souhaitée $l^d = 1.5$ et l'orientation souhaitée ($\psi^d = \frac{7\pi}{6}$) avec son leader. Par calcul (équation 2.4), il est facile d'obtenir que la posture souhaitée du suiveur est à (-1.499, -0.750, 0). Supposons que la posture réelle initiale du suiveur soit à (-2, -4, 0). Ensuite, l'erreur initiale est (0.502, 3.25, 0).

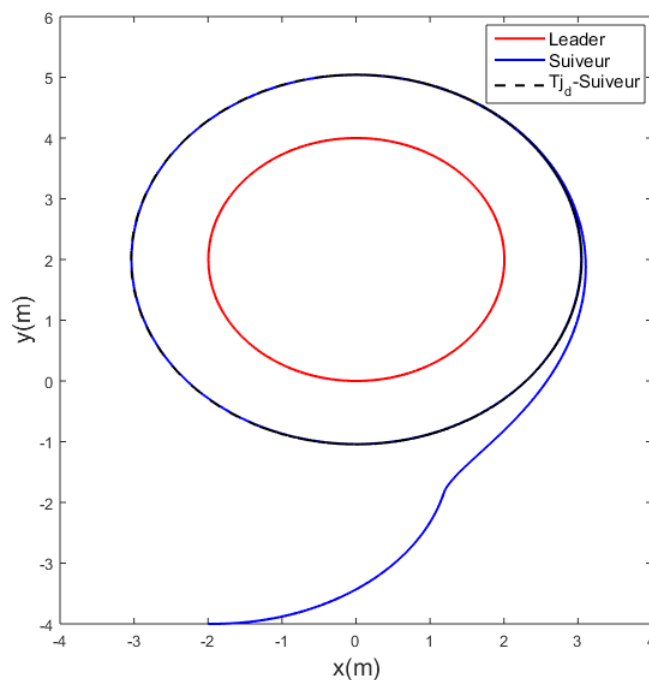


FIGURE 2.5 – Les trajectoires en temps réel de leader et son suiveur

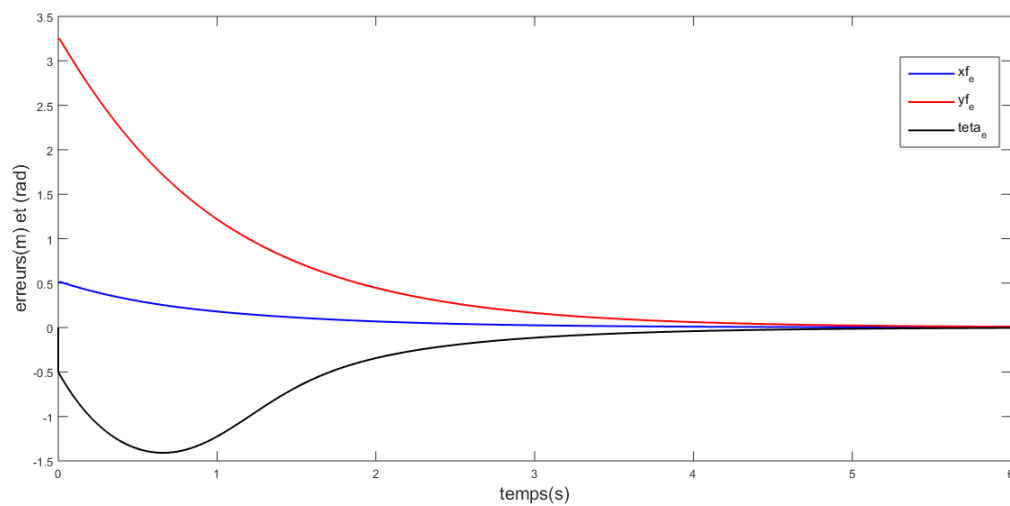
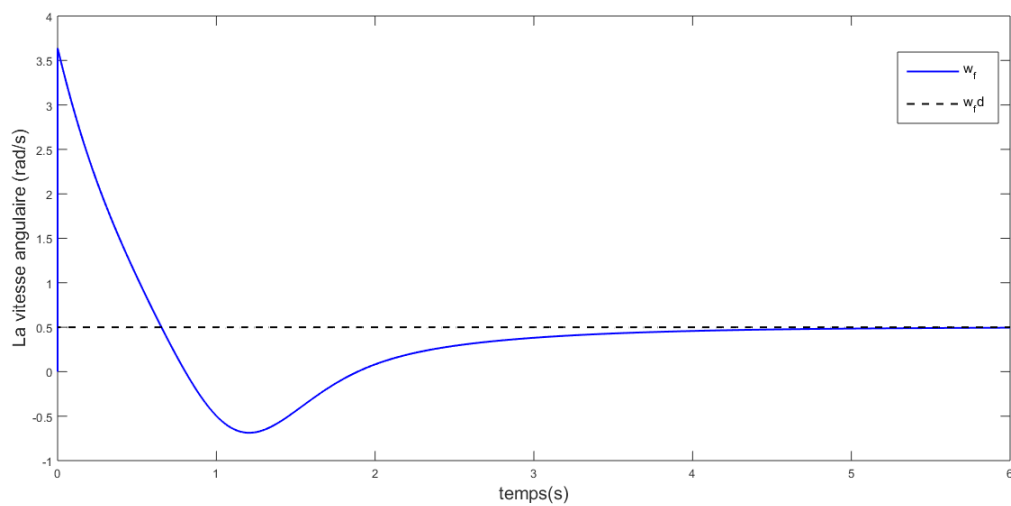


FIGURE 2.6 – Les erreurs de suivi pour le suiveur



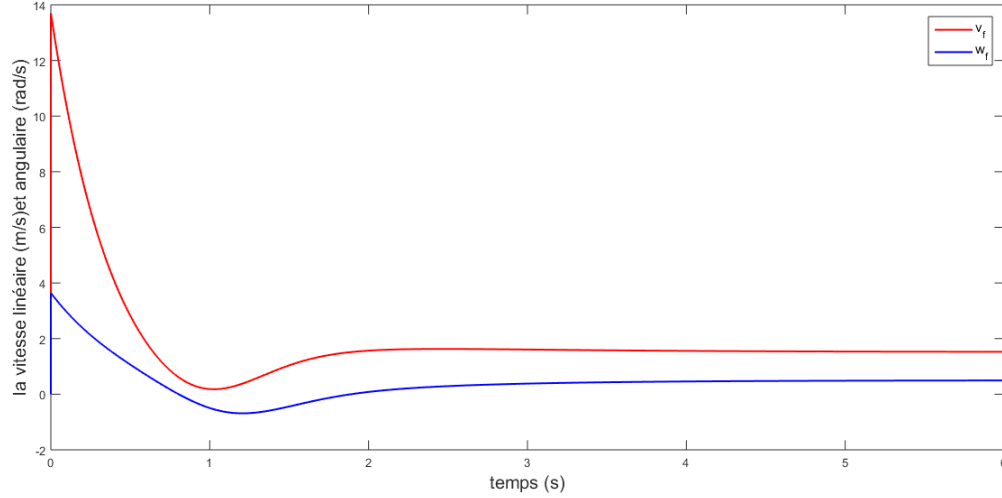


FIGURE 2.7 – Les vitesses linéaire et angulaire du suiveur

d) Discussion des résultats

La figure(2.5) est la trajectoire des robots. Elle montre que le suiveur peut bien suivre le leader et maintenir la séparation et le roulement souhaités avec le leader. La figure(2.6) est les erreurs du suiveur. Elle montre que les erreurs de suivi du suiveur convergent vers zéro après ($t = 5s$). La figure(2.7) est la vitesse angulaire et la vitesse linéaire du suiveur. À partir de cette dernière, le suiveur commence avec une grande vitesse linéaire et une grande vitesse angulaire. Cela implique que l'accélération linéaire initiale et l'accélération angulaire sont très importantes, c'est-à-dire que la force et le couple de suiveur sont très importants. La figure(2.8) est la vitesse angulaire ω_f du suiveur et la vitesse angulaire du leader ($\omega_l = \omega_f^d$). Elle montre que la vitesse angulaire de suiveur peut converger vers la vitesse angulaire du leader après ($t = 5s$).

2.4 Conclusion

Dans ce chapitre, l'approche de suivi d'une trajectoire pour un robot mobile non-holonyme (suiveur) a été développée, cette trajectoire n'est généralement pas prédéfinie, qui est décidées par son leader en temps réel, c'est la commande par bouclage linéarisant avec des travaux de simulation. Nous avons montré que cette approche n'assure pas de bons résultats de suivi le leader. Néanmoins, dans le cas, particulièrement, lorsque la partie commandée est soumise à de fortes non linéarités et à des variations temporelles, il faut concevoir des algorithmes de commandes assurant la robustesse du comportement du processus vis-à-vis des incertitudes sur les paramètres et leurs variations. La méthode de mode de glissement est rendue

très populaire auprès des automaticiens, grâce sa grande robustesse vis-à-vis les incertitudes paramétriques, les erreurs de modélisation et les perturbations. En utilisant la technique de mode glissant, les contrôleurs de suivi peuvent être simples et la stabilité du système peut être garantie par la théorie de la stabilité de Lyapunov.

Chapitre 3

La commande par mode glissant (CMG)

3.1 Introduction

La commande par mode de glissement consiste à concevoir une loi de commande qui puisse guider le vecteur d'état d'un système donné vers un hyperplan $S(x) = 0$, appelé la surface de glissement, le vecteur d'état possédera alors une dynamique stable qui dépendra directement du choix de la surface de glissement, et qui fera en sorte que le vecteur d'état convergera vers le point d'équilibre.

Nous pouvons constater que la commande par mode de glissement est divisée en deux étapes, ou deux modes : Dans une première phase, le vecteur d'état doit atteindre la surface de glissement, nous parlons alors du mode d'atteinte, ou «reaching mode» en anglais. Dans une deuxième phase, et après avoir atteint la surface de glissement $S(x) = 0$, le vecteur d'état doit glisser sur cette surface jusqu'à atteindre le point d'équilibre, nous parlons alors du mode glissant, ou de « sliding mode».

3.2 Systèmes à structure variable

La théorie des systèmes à structure variable, initialisée au début du 20ème siècle par **Nikol'ski**. Il a pris son essor dans les années 60 par Emelyanov et A.F Philipov. Pour de tels systèmes, la structure du contrôleur utilisé peut changer d'une façon discontinue entre deux ou plusieurs structures. C'est le cas, par exemple, des circuits de conversion de puissance, où le système est gouverné par une équation différentielle différente pour chaque position de l'interrupteur.

Un modèle serait du type $\dot{x} = f(x, t)$, où le champ de vecteurs f prend diverses valeurs f^i selon la région de l'espace d'état où l'on se trouve (figure 3.1).

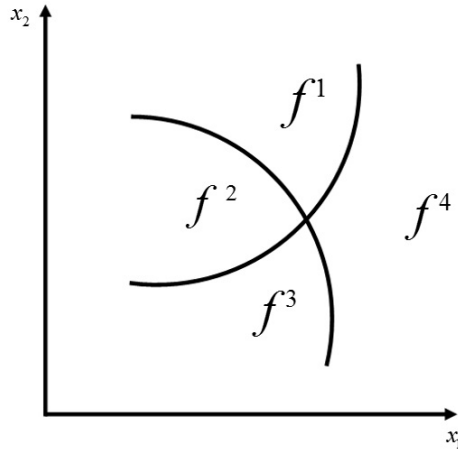


FIGURE 3.1 – Systèmes à structure variable dans une coupe de l'espace d'état.

La théorie des systèmes à structure variable a donné naissance aux notions de régimes glissants qui se sont par la suite étendus aux systèmes discontinus réalisés artificiellement, c'est-à-dire, des systèmes de type $\dot{x} = f(x, t, u)$, avec f une fonction non linéaire continue, et pour lesquels on définit un retour d'état discontinu à l'aide de la variable u . Cette approche a pour but de maintenir l'état représentatif de l'évolution du système sur une variété S définie au préalable, appelée : surface de glissement [33].

3.3 Principe de la technique de mode glissant

Le principe de la commande par mode glissant et de contraindre le système à atteindre la surface donnée appelée : surface de glissement, en fonction des objectifs de la commande fixant la dynamique en boucle fermée : c'est le mode de convergence. Puis par la synthèse d'une commande discontinue qui permet aux trajectoires du système à atteindre et rester sur cette surface : c'est la surface de glissement et le mouvement qui se produit le long de celle-ci est appelé mouvement de glissement. La trajectoire dans le plan de phase est constituée de trois parties distinctes :

- Le mode de convergence « MC » : C'est le mode durant lequel l'état du système se déplace à partir de n'importe quel point initial dans le plan de phase et tend vers la surface de commutation $S(x) = 0$. Ce mode est caractérisé par la loi de commande et le critère de convergence.
- Le mode de glissement « MG » : C'est le mode durant lequel la variable d'état a atteint la surface de glissement et tend vers l'origine du plan de phase. La dynamique de ce mode est caractérisée par le choix de la surface de glissement $S(x) = 0$.
- Le mode du régime permanent « MRP » : Ce mode est ajouté pour l'étude de la

réponse du système autour de son point d'équilibre (origine du plan de phase), il est caractérisé par la qualité et les performances de la commande.

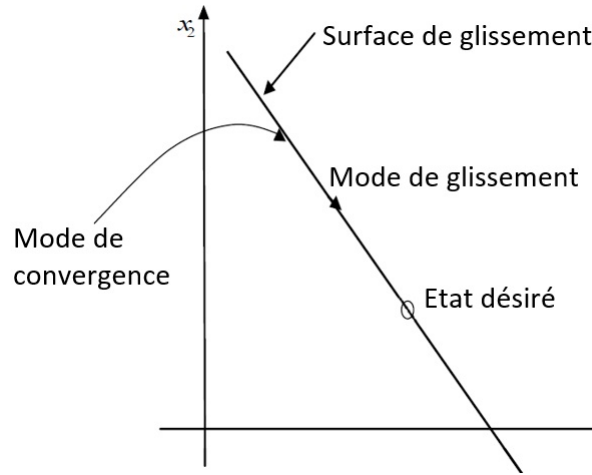


FIGURE 3.2 – Différents modes de convergence pour la trajectoire d'état

3.4 Conception de la commande par mode glissant

Les avantages de la commande par mode glissant sont importants et multiples, entre autres, la haute précision, la bonne stabilité, la simplicité, l'invariance, la robustesse...etc. Ceci lui permet d'être particulièrement adaptée pour les systèmes ayant des modèles imprécis.

La conception de la commande peut être effectuée en trois étapes principales très dépendantes l'une de l'autre :

- Choix de la surface ;
- Établissement des conditions d'existence ;
- Détermination de la loi de commande.

3.4.1 Choix de la surface de glissement

La surface de glissement peut être linéaire ou non linéaire, elle est construite de telle sorte que le système ait toujours une dynamique désirée et il est généralement choisie avec un degré relatif égal (paragraphe 2.3.1) à un (la dérivée de la surface $S(x)$ par rapport au temps fait apparaître explicitement la commande u)[30].

Considérons une classe de systèmes non linéaires (équation 3.2), le vecteur de surface a la même dimension que le vecteur de commande u .

$$\dot{X} = A(X, t)u + b(X, t) \quad (3.1)$$

Où $X = (x_1, x_2, \dots, x_n)^T$ appartient à X est le vecteur d'état de système avec $u = (u_1, u_2, \dots, u_m)^T$ est le vecteur de commande et A, b sont deux fonctions non linéaires et b est inversible. En fait, tout système commandable peut être ramené à la forme canonique de (3.1). Avec ces notations, nous pouvons choisir la fonction de glissement suivante :

$$S_j(x) = \sum_{i=0}^n c_{ij} e_i \quad (3.2)$$

Où : $s_i(x) = 0$ est la fonction de commutation, elle sépare l'espace d'état en deux parties disjointes $s(x, t) \prec 0$ et $s(x, t) \succ 0$;

$S(x) = (s_1(x), s_2(x), \dots, s_m(x))$: Vecteur de la surface de glissements ;

$X_{di} = (x_{d1}, x_{d2}, \dots, x_{dn})^T$: Vecteur de référence (l'état désiré) ;

$e_i = x_{di} - x_i$: L'erreur entre la variable à commander et sa référence ;

$e_i = (e_1, e_2, \dots, e_n)^T$: Vecteur de l'écart ;

c_{ij} : Paramètres de la surface de glissement ;

C : Matrice d'éléments $m \times n$ paramètres de la surface (matrice de pente de la droite de surface).

Le choix des paramètres de la surface se fait de façon satisfaisant les performances désirées. **Slotine** propose une forme d'équation générale pour déterminer la surface de glissement qui assure la convergence d'une variable vers sa valeur désirée.

$$S(x) = \left(\frac{\partial}{\partial t} + \lambda \right)^{r-1} e(x) \quad (3.3)$$

Avec : $e_i = x_{di} - x_i$;

λ : Une constante positive ;

r : Degré relatif, égal au nombre de fois qu'il faut dériver la sortie pour faire apparaître la commande ;

$S(x) = 0$ est une équation différentielle linéaire dont l'unique solution est : $e(x) = 0$, ceci implique que l'état du système est égal à l'état désiré.

3.4.2 Conditions de convergence

Les conditions d'existence et de convergence sont les critères qui permettent aux différentes dynamiques du système de converger vers la surface de glissement et d'y rester.

a) La fonction discrète de commutation

Cette approche est la plus ancienne. Elle est proposée par Emelyanov et Utkin. Il s'agit de donner à la surface une dynamique convergente vers zéro. Elle est donnée par :

$$U = \begin{cases} \dot{S}(x) \succ 0 & \text{si } S(x, t) \prec 0 \\ \dot{S}(x) \prec 0 & \text{si } S(x, t) \succ 0 \end{cases} \quad (3.4)$$

Cette condition (3.4) peut être formulée comme suit :

$$\dot{S}(x)S(x) \prec 0$$

Elle est globale mais ne garantit pas en revanche un temps d'accès fini. Cette condition est toutefois difficile à utiliser pour faire la synthèse de la loi de la commande.

b) La fonction de Lyapunov

La fonction de **Lyapunov** est une fonction scalaire positive $V(x) \succ 0$ pour les variables d'état du système. La loi de commande doit faire décroître cette fonction $\dot{V}(x) \prec 0$. L'idée est de choisir une fonction scalaire $S(x)$ pour garantir l'attraction de la variable à contrôler vers sa valeur de référence et de concevoir une commande u telle que le carré de la surface correspond à une fonction de Lyapunov. Nous définissons la fonction de Lyapunov comme suit :

$$V(x) = \frac{1}{2} S^T(x) S(x) \quad (3.5)$$

La dérivée de cette fonction (3.5) est :

$$\dot{V}(x) = \dot{S}(x) S(x)$$

Pour que la fonction de Lyapunov décroisse, il suffit d'assurer que sa dérivée est négative. Ceci est vérifié si $\dot{S}(x) S(x) \prec 0$. Cette équation montre que le carré de la distance vers la surface, mesuré par $S^T(x) S(x)$, diminue tout le temps, contraignant la trajectoire du système à se diriger vers la surface des deux côtés. Cette condition suppose un régime glissant idéal.

3.4.3 Synthèse de la commande

Dans ce mémoire, nous avons visé l'élaboration d'une loi de commande par mode glissant assurant à la fois robustesse, performance et simplicité. Cette commande ainsi établie permet de résoudre les problèmes liés au phénomène de broutement et à la réduction de l'ordre du système en boucle fermée. Conception de la commande par mode de glissement, Lorsque le régime glissant atteint la dynamique du système qui est indépendante de la loi de commande qui n'a pour but de maintenir les conditions de glissement (l'attractivité de la surface), c'est pour cette raison que la surface est déterminée indépendamment de la commande. La

structure de la commande comporte deux parties, une première concernant la linéarisation exacte U_{eq} et une deuxième stabilisante ΔU :

$$U(t) = \Delta U + U_{eq} \quad (3.6)$$

U_{eq} : Correspond à la commande équivalente proposée par Philippov et Utkin, elle sert à maintenir la variable à contrôler sur la surface de glissement $S(x)$. La commande équivalente est déduite en considérant que le dérivé de la surface est nul $\dot{S}(x) = 0$. Elle peut être interprétée comme étant un retour d'état particulier jouant le rôle d'un signal de commande appliqué sur le système à commande lors de la commutation rapide entre les valeurs U_{min} et U_{max} [34] (figure 3.3).

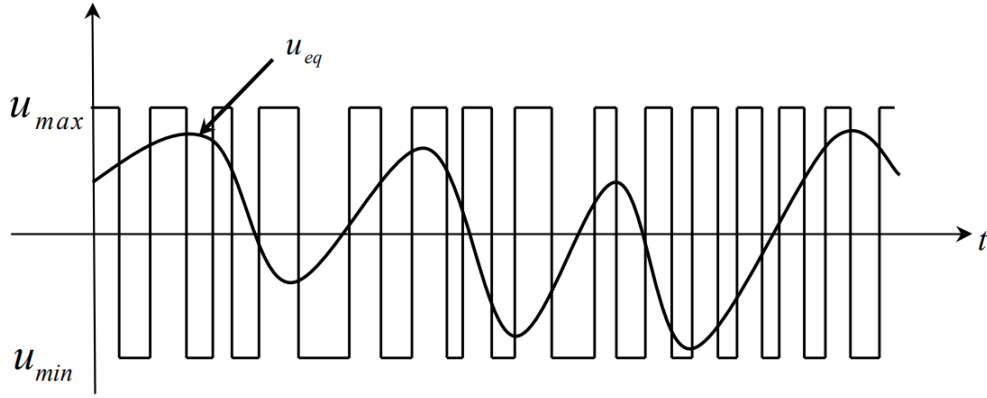


FIGURE 3.3 – La valeur continue U_{eq} prise par la commande lors de la commutation.

ΔU : est déterminée pour vérifier la condition de convergence. Afin de mettre en évidence le développement précédent, on considère le système d'état (3.1) . On cherche à déterminer l'expression analogique de la commande U . La dérivée de la surface $S(x)$ est :

$$\dot{S} = \frac{dS}{dt} = \frac{\partial S}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial S}{\partial x} \{A(x, t)U_{eq} + b(x, t)\} + \frac{\partial S}{\partial x} \{A(x, t)\Delta U\} \quad (3.7)$$

En mode de glissement et en régime permanent, la dérivée de la surface est nulle (car la surface est égale à zéro). Ainsi, nous obtenons :

$$U_{eq} = - \left\{ \frac{\partial S}{\partial x} A(x, t) \right\}^{-1} \left\{ \frac{\partial S}{\partial x} b(x, t) \right\} \Delta U \quad (3.8)$$

Pour que la commande équivalente puisse prendre une valeur finie, il faut que :

$$\frac{\partial S}{\partial x} A(x, t) \neq 0 \quad (3.9)$$

Durant le mode de convergence, en remplaçant le terme par sa valeur (3.8) dans l'équation (3.7). Donc, nous obtenons une nouvelle expression de la dérivée de la surface, soit :

$$\dot{S} = \frac{\partial S}{\partial x} \{A(x, t) \Delta U\} \quad (3.10)$$

Le problème revient à trouver ΔU tel quel :

$$S \dot{S} = S \frac{\partial S}{\partial x} \{A(x, t) \Delta U\} \prec 0 \quad (3.11)$$

La solution la plus simple est de choisir sous la forme de relais (figure 3.4). Dans ce cas, la commande (Loi d'arrivée avec une vitesse constante) s'écrit comme suit :

$$\Delta U = k_x \text{sing}(S(x)) \quad (3.12)$$

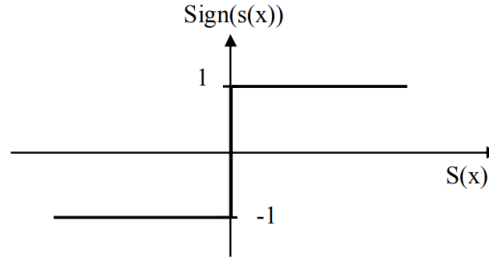


FIGURE 3.4 – Représentation de la fonction « Sign ».

En remplaçant l'expression (3.12) dans (3.11) on obtient :

$$S \dot{S} = \frac{\partial S}{\partial x} A(x, t) k |S| \prec 0 \quad (3.13)$$

Où le facteur $\frac{\partial S}{\partial x} A(x, t)$ est toujours négatif pour la classe des systèmes que nous considérons. Le gain k est choisi positif pour satisfaire la condition (3.13). Le choix de ce gain est très influent sur la commande car, s'il est très petit le temps de réponse sera très long et s'il est choisi très grand, nous aurons des fortes oscillations au niveau de l'organe de la commande. Ces oscillations peuvent exciter les dynamiques négligées (phénomène de broutement (figure 3.5)).

a) Élimination du phénomène du broutement (chattering)

L'un des principaux inconvénients du réglage par mode de glissement est le phénomène du chattering. Et dans le but de réduire ces oscillations, plusieurs solutions ont été apportées comme par exemple : remplacer la fonction «sign» par une fonction de saturation «sat» caractérisée par un seuil (figure 3.6).

La fonction est définie par :

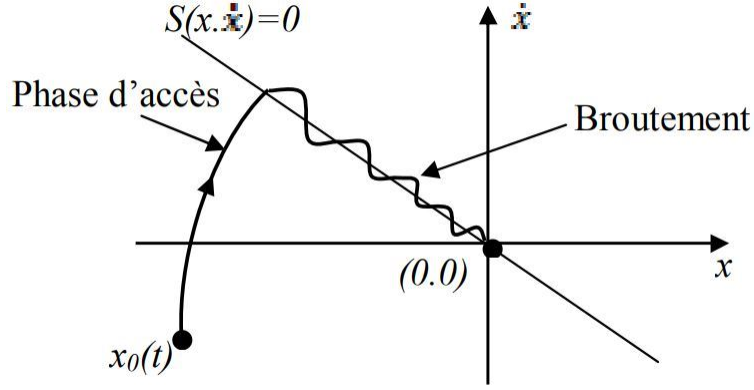


FIGURE 3.5 – Phénomène de broutement

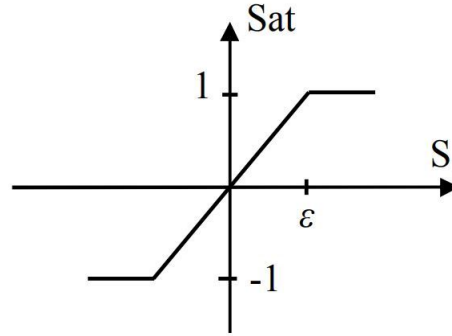


FIGURE 3.6 – Fonction Sat avec un seuil

$$sat(S) = \begin{cases} -1 & si \quad S \prec -\epsilon \\ \frac{S}{\epsilon} & si \quad |S| \prec \epsilon \\ 1 & si \quad S \succ \epsilon \end{cases} \quad (3.14)$$

Cependant, la convergence du vecteur d'état reste dans un voisinage de la surface de glissement, dont l'amplitude dépend directement de ϵ [30].

b) Loi d'arrivée avec une vitesse constante et à action proportionnelle

Une approche intéressante introduite par **Gao** et **Hung** [30] consiste à concevoir des lois de convergence non linéaires qui permettent une adaptation dynamique d'une composante de la commande en fonction de la variation de la fonction de glissement. Il est théoriquement possible de réduire les commutations de hautes fréquences.

En régime établi, sans toutefois affecter le temps de convergence ni même l'erreur de poursuite du système. Gao et Hung ont proposé notamment deux lois d'atteinte possibles pour réduire les commutations. La première loi d'atteinte contient un terme proportionnel à

la fonction de glissement qui permet au système d'atteindre plus rapidement la surface de glissement lorsque celui-ci en est éloigné :

$$\dot{S}(x) = -\varepsilon S(x) - k \operatorname{sing}(S(x)) \quad (3.15)$$

Dans la deuxième loi de convergence figure une puissance fractionnaire de la fonction de glissement qui multiplie le signe de celle-ci, comme suit :

$$\dot{S}(x) = -k |S(x)|^\alpha \operatorname{sing}(S(x)) \quad (3.16)$$

Où α est un réel strictement compris entre 0 et 1. Il est par ailleurs démontré par Gao et Hung que la loi de convergence (3.16) engendre un temps d'atteinte fini de la surface de glissement.

c) Commande continue par Bande de surface (quasi-sliding mode)

D'après la partie précédente, nous constatons l'intérêt de concevoir des lois de convergence qui puissent s'adapter dynamiquement aux fluctuations de la fonction de glissement. Néanmoins, il est important que la loi de convergence conçue maintienne la stabilité du système au sens de Lyapunov, en assurant aussi des transitions douces au niveau de la commande. Pour répondre à toutes ces exigences, nous proposons dans cette partie une loi de portée par morceaux, donnée par l'expression :

$$\dot{S}(x) = -\varepsilon S(x) - k f(S(x)) \quad (3.17)$$

Avec :

$$f(S) = \begin{cases} |S|^\alpha \operatorname{sing}(S) & \text{si } |S| \leq \varphi \\ \frac{S \varphi^\alpha}{S} & \text{si } |S| > \varphi \end{cases}$$

Où α est un réel strictement compris entre 0 et 1, et ε, φ, k des entiers naturels strictement positifs. Le choix de ces paramètres dépend de plusieurs variantes, à savoir la dynamique du système, l'ampleur des incertitudes et les bornes des perturbations éventuelles qui s'exercent sur le système [30].

3.5 Contrôle de la formation leader-suiveur par technique de mode glissant

Cette partie examine la formation du contrôle des robots mobiles non-holonomes. Le problème de formation est converti en modèle d'erreur basé sur la structure leader-suiveurs. Un contrôleur de mode glissant, qui se révèle être asymptotiquement stable en temps fini par la théorie de la stabilité de Lyapunov, il est présenté dans cette partie. De plus, une loi de portée continue est conçue pour réduire les bavardages causés par les délais de calcul et les limitations de contrôle. Les résultats de la simulation sur Matlab vérifient la faisabilité et l'efficacité de la stratégie de contrôle.

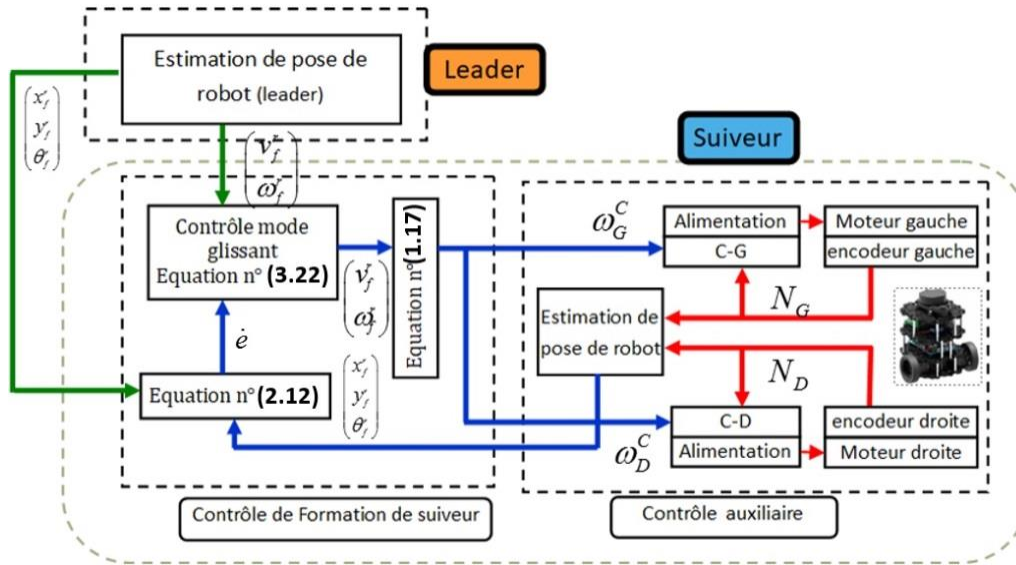


FIGURE 3.7 – Organigramme du contrôle de mouvement d'un robot mobile

3.5.1 Modèle de contrôle de formation leader-suiveur

Après avoir présenté la théorie de la commande par Le mode glissant et comme nous avons vu dans le paragraphe (2.2.2) le contrôle de leader-suiveur et l'erreur dynamique et en raison des caractéristiques du système dynamique d'erreur (2.12) et l'erreur (2.9) (1.1). Il peut traiter efficacement l'installation contrôlée avec un degré relatif égal (dans notre cas $r = 1$) à un, par rapport à l'entrée de commande. Théoriquement, la technique peut rendre le mode glissant variable et sa dérivée temporelle convergente vers l'origine malgré les incertitudes et les perturbations paragraphe (3.4.3). Ainsi, cette technique peut être considérée comme une solution pour les manœuvres de formation du SMR.

L'erreur de suivi de suiveur peut être exprimée comme dans le système de coordonnées du suiveur par l'équation (2.9) :

$$\begin{pmatrix} x_{fe} \\ y_{fe} \end{pmatrix} = \begin{pmatrix} \cos \theta_f & \sin \theta_f \\ -\sin \theta_f & \cos \theta_f \end{pmatrix} \begin{pmatrix} x_c - x_{of} \\ y_c - y_{of} \end{pmatrix} \quad (3.18)$$

Alors, l'équation d'erreur dynamique peut être obtenue par l'équation (2.12) :

$$\begin{pmatrix} \dot{x}_{fe} \\ \dot{y}_{fe} \\ \dot{\theta}_{fe} \end{pmatrix} = \begin{pmatrix} (v_l - y_d \omega_l) \cos \alpha - x_d \omega_l \sin \alpha - v_f + \omega_f y_e \\ (v_l - y_d \omega_l) \sin \alpha + x_d \omega_l \cos \alpha - \omega_f (d + x_e) \\ \omega_f^d - \omega_f \end{pmatrix}$$

La destination de la conception de la loi de commande est de trouver les entrées de commande appropriées $(v_f, \omega_f)^T$ pour atteindre le résultat souhaité (2.11)

3.5.2 Choix des surfaces de glissements

Notre but consiste à générer une loi de commande $(v_f, \omega_f)^T$ en utilisant la théorie du contrôle par mode glissant. Les états considérés pour le contrôle du suiveur sont (x, y, θ) . Les surfaces de glissement seront alors (voir [46]) :

$$s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = R_\alpha^{C_d} \begin{pmatrix} x_e \\ y_e \end{pmatrix} \quad (3.19)$$

Avec $R_\alpha^{C_d} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix}$: matrice de rotation en fonction de $\alpha = \theta_l - \theta_f$ pour orienter le robot vers l'orientation désirée :

$$\begin{pmatrix} S_1 \\ S_2 \end{pmatrix} = \begin{pmatrix} x_e \cos \alpha + y_e \sin \alpha \\ -x_e \sin \alpha + y_e \cos \alpha \end{pmatrix} \quad (3.20)$$

Les dérivées successives de S_1 et S_2 seront :

$$\dot{S}_1 = \dot{x}_e \cos \alpha - x_e \dot{\alpha} \sin \alpha + \dot{y}_e \sin \alpha + y_e \dot{\alpha} \cos \alpha$$

$$\dot{S}_2 = -\dot{x}_e \sin \alpha - x_e \dot{\alpha} \cos \alpha + \dot{y}_e \cos \alpha - y_e \dot{\alpha} \sin \alpha$$

$$\begin{aligned} \dot{S}_1 &= v_l - y_d \omega_l + (-v_f + \omega_f y_e) \cos \alpha - (\omega_f d + \omega_f x_e) \sin \alpha \\ &\quad + y_e (\omega_l - \omega_f) \cos \alpha - x_e (\omega_l - \omega_f) \sin \alpha \end{aligned}$$

$$\begin{aligned} \dot{S}_2 &= x_d \omega_l + (v_f - \omega_f y_e) \sin \alpha - (\omega_f d + \omega_f x_e) \cos \alpha \\ &\quad - y_e (\omega_l - \omega_f) \sin \alpha - x_e (\omega_l - \omega_f) \cos \alpha \end{aligned}$$

$$\dot{S}_1 = -v_f \cos \alpha - \omega_f d \sin \alpha + v_l - y_d \omega_l + \omega_l (-x_e \sin \alpha + y_e \cos \alpha)$$

$$\dot{S}_2 = v_f \sin \alpha - \omega_f d \cos \alpha + x_d \omega_l - \omega_l (x_e \cos \alpha + y_e \sin \alpha)$$

Après le développement de calculs :

$$\begin{aligned}\dot{S}_1 &= -v_f \cos \alpha - \omega_f d \sin \alpha + v_l - y_d \omega_l + \omega_l S_2 \\ \dot{S}_2 &= v_f \sin \alpha - \omega_f d \cos \alpha + x_d \omega_l - \omega_l S_1\end{aligned}$$

Commande de contrôle $(v_f, \omega_f)^T$:

$$\begin{aligned}v_f &= (v_l - y_d \omega_l) \cos \alpha - x_d \omega_l \sin \alpha + \omega_l S_2 \cos \alpha + \omega_l S_1 \sin \alpha - \dot{S}_1 \cos \alpha + \dot{S}_2 \sin \alpha \\ \omega_f &= \left[(v_l - y_d \omega_l) \sin \alpha + x_d \omega_l \cos \alpha + \omega_l S_2 \sin \alpha - \omega_l S_1 \cos \alpha - \dot{S}_1 \sin \alpha - \dot{S}_2 \cos \alpha \right] / d \\ v_f &= (v_l - y_d \omega_l) \cos \alpha - x_d \omega_l \sin \alpha + \omega_l y_e - \dot{S}_1 \cos \alpha + \dot{S}_2 \sin \alpha \\ \omega_f &= \left[(v_l - y_d \omega_l) \sin \alpha + x_d \omega_l \cos \alpha - \omega_l x_e - \dot{S}_1 \sin \alpha - \dot{S}_2 \cos \alpha \right] / d\end{aligned}\quad (3.21)$$

3.5.3 Loi de commande par mode glissant

Une loi de commande est adoptée pour que chaque état atteigne la surface de glissement . En comparaison avec la fonction de commutation (3.17).

$$\dot{S}(x) = \begin{cases} -\varepsilon_1 S_1(x) - k_1 f_1(S_1(x)) \\ -\varepsilon_2 S_2(x) - k_2 f_2(S_2(x)) \end{cases}\quad (3.22)$$

avec :

$$f_i(S_i) = \begin{cases} |S_i|^{\alpha_i} \text{sing}(S_i) & \text{si } |S_i| \leq \varphi \\ \frac{S_i \varphi^{\alpha_i}}{|S_i|} & \text{si } |S_i| > \varphi \end{cases}$$

Ainsi, nous pouvons obtenir le contrôleur de mode glissant (3.23) à partir de l'équation (3.18), (3.21) et (3.22) :

$$\begin{cases} v_f = (v_l - y_d \omega_l) \cos \alpha - x_d \omega_l \sin \alpha + \omega_l y_e + (\varepsilon_1 S_1(x) + k_1 f_1(S_1(x))) \cos \alpha \\ \quad - (\varepsilon_2 S_2(x) + k_2 f_2(S_2(x))) \sin \alpha \\ \omega_f = [(v_l - y_d \omega_l) \sin \alpha + x_d \omega_l \cos \alpha - \omega_l x_e + (\varepsilon_1 S_1(x) + k_1 f_1(S_1(x))) \sin \alpha] / d \\ \quad + [(\varepsilon_2 S_2(x) + k_2 f_2(S_2(x))) \cos \alpha] / d \end{cases}\quad (3.23)$$

3.5.4 Choix de la fonction de Lyapunov

Le choix de fonction de Lyapunov est donné par l'équation (3.5)

$$V_i(x) = \frac{1}{2} S_i^2(x)$$

Avec ($i = 1, 2$)

L'utilisation de la loi commande (3.22) peut mettre le système asymptotiquement stable et l'erreur (3.18) de posture du robot (Suiveur) converge vers zéros dans un temps fini. La dérivée de cette fonction est donnée par :

$$\dot{V}_i = S_i \dot{S}_i = -\varepsilon_i S_i^2 - k_i f_i(S_i) S_i \quad (3.24)$$

si $|S_i| \leq \varphi$ La dérivée de cette fonction est donnée par :

$$\dot{V}_i = S_i \dot{S}_i = -\varepsilon_i S_i^2 - k_i |S_i|^{\alpha_i} \text{sing}(S_i) S_i < 0$$

si $|S_i| > \varphi$ La dérivée de cette fonction est donnée par :

$$\dot{V}_i = S_i \dot{S}_i = -\varepsilon_i S_i^2 - k_i \frac{\varphi^{\alpha_i}}{|S_i|} S_i^2 < 0$$

Où α est un réel strictement compris entre 0 et 1, et ε, φ, k des entiers naturels strictement positifs.

3.6 Résultats des simulations sur Matlab

Les simulations sont à présent faites pour évaluer l'effectivité de la méthode d'étude proposée (mode glissant) ; le robot de type unicycle est illustré à la figure(3.8) . Le robot est utilisé pour suivre son leader : 1ère le leader se déplace sur une trajectoire prédéfinie, qui veut atteindre cette trajectoire par le bouclage linéarisant. L'efficacité des contrôleurs de suiveur proposés (3.21) basée sur la fonction de commutation (3.22) est utilisée avec ($\varepsilon_1 = 2, \varepsilon_2 = 2, k_1 = 3, k_2 = 2, \alpha_1 = 0.8, \alpha_2 = 0.8, \varphi = 0.25$).

1er Cas le leader se déplace sur une trajectoire circulaire : la vitesse linéaire et la vitesse angulaire du robot leader sont respectivement de (1, 0.5), la posture initiale du robot leader est à (0, 0, 0), Afin de conserver la formation de leader-suiveur, le suiveur doit conserver la séparation souhaitée ($l^d = 1.5$) et le orientation souhaité ($\psi^d = \frac{7\pi}{6}$) avec son leader, Par calcul (2.4), il est facile d'obtenir que la posture souhaitée du suiveur est à (-1.499, -0.750, 0). Supposons que la posture réelle initiale du suiveur soit à (-2, -4, 0). L'erreur initiale est (0.502, 3.25, 0).

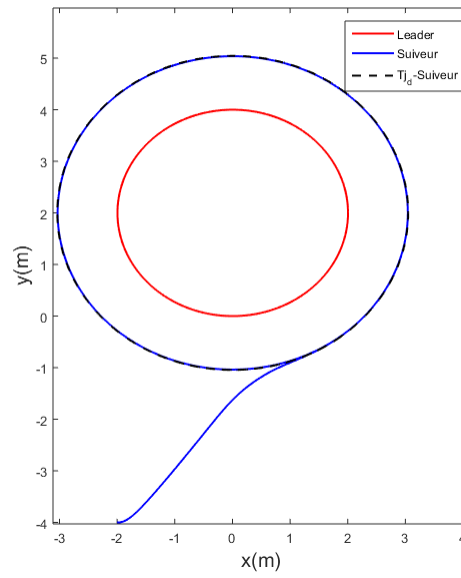


FIGURE 3.8 – Les trajectoires de leader et son suiveur

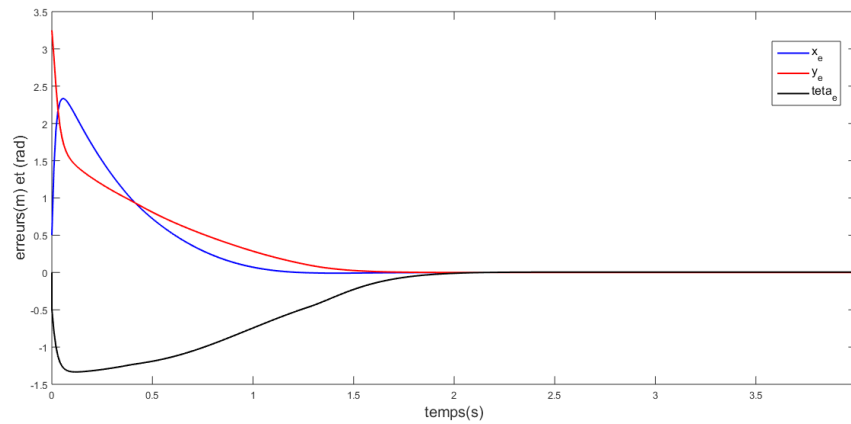


FIGURE 3.9 – Les erreurs de suivi pour le suiveur

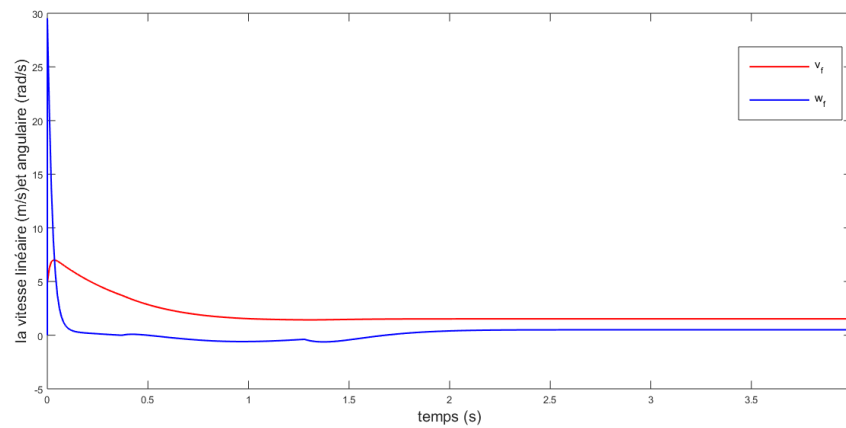


FIGURE 3.10 – Les vitesses linéaire et angulaire du suiveur

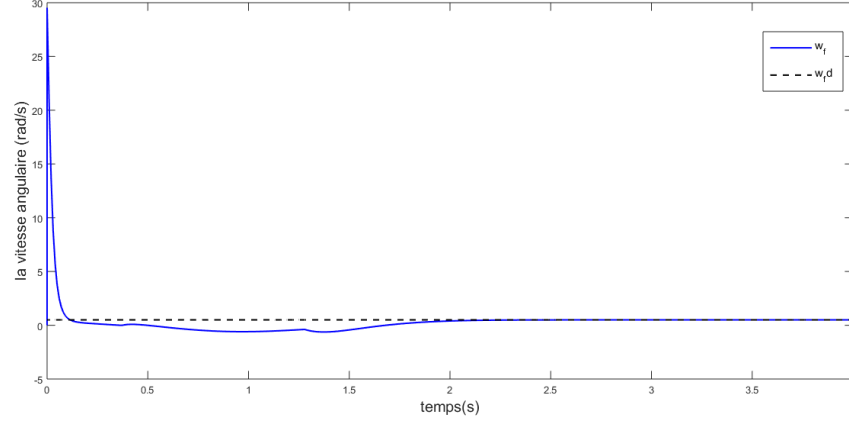


FIGURE 3.11 – Les vitesses angulaires du suiveur et du leader

On peut voir qu'un suivi parfait est atteint, le robot est capable de suivre le leader. La figure(3.10) montre la vitesse du robot qui atteint la valeur fixe de $(1.5m/s)$ au temps $(t = 2s)$. La figure(3.9) montre l'erreur de poursuite d'une trajectoire ; toutes convergent vers zéros après un temps $(t = 2s)$.

2ème Cas le leader veut atteindre une trajectoire circulaire prédéfinie contrôler (de Rayon $(3m)$, centre $(0,0)$) par le bouclage linéarisant : la posture initiale du robot leader est à $(1,2,0)$. Afin de conserver la formation de leader-suiveur, le suiveur doit conserver la séparation souhaitée $(l^d = 1.5)$ et le orientation souhaité $(\psi^d = \frac{7\pi}{6})$ avec son leader, Par calcul (2.4), il est facile d'obtenir que la posture souhaitée du suiveur est à $(-0.499, 1.250, 0)$. Supposons que la posture réelle initiale du suiveur soit à $(0,0,0)$. L'erreur initiale est à $(-0.499, 1.250, 0)$.

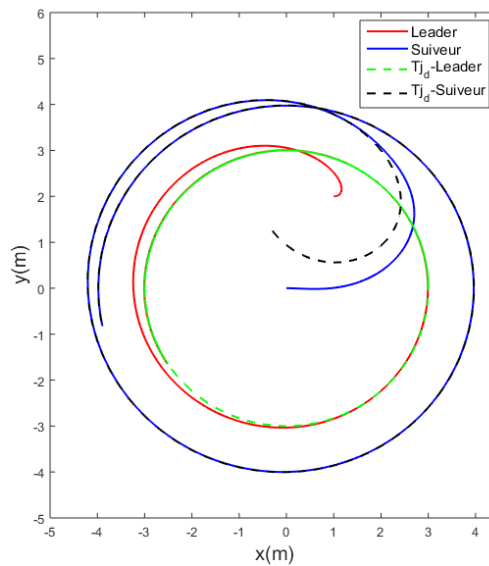


FIGURE 3.12 – Les trajectoires en temps réel des robots

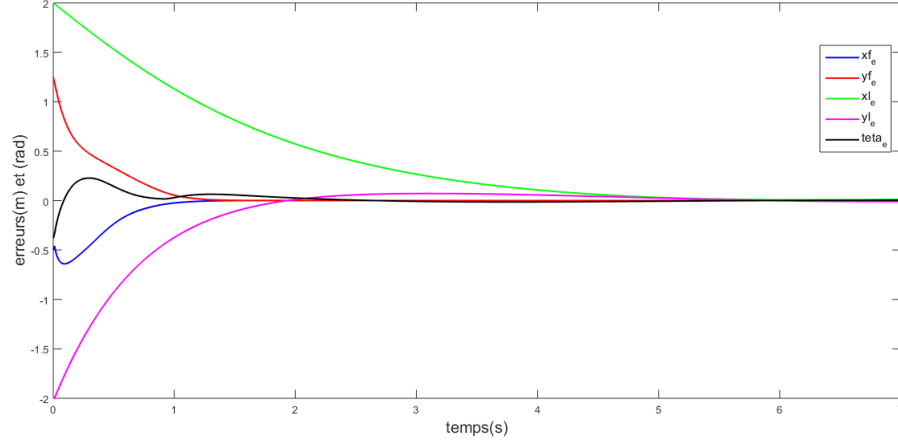


FIGURE 3.13 – Les erreurs de suivi pour le suiveur et leader

3ème Cas mouvement de transformation dans une trajectoire sinusoïdale la figure(3.14) montre les quatre robots créant une formation en forme composée lors du déplacement le long d'une trajectoire sinusoïdale. Tout d'abord, les robots forment une chaîne tout en se déplaçant dans une trajectoire trapèze. Ils transforment ensuite la chaîne en un coin. Concernant cette tâche, les postures initiales des trois robots suiveurs et un leader sont fixées par : Leader $(1, 2, 0)$; Suiveur1 $(-5, -5, \frac{\pi}{4})$; Suiveur2 $(-1, -2, \frac{\pi}{3})$; Suiveur3 $(-5, 5, 0)$.

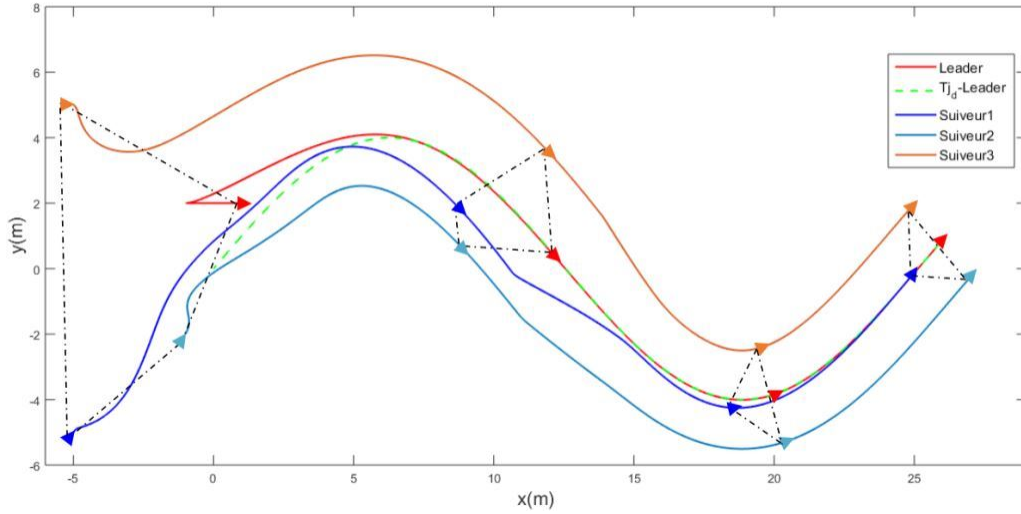


FIGURE 3.14 – Transformation du SMR en se déplaçant en trajectoire sinusoïdale

- si $(0 < t \leq 7s)$ les états souhaités des suiveurs sont fixés par : Suiveur1 $(l_1^d = 3 ; \psi_1^d = \frac{9\pi}{8})$, Suiveur2 $(l_2^d = 3 ; \psi_2^d = \frac{5\pi}{4})$, Suiveur3 $(l_3^d = 3 ; \psi_3^d = \frac{3\pi}{4})$ pour créer la formation de Trapèze ;
- si $(t > 7s)$ les états souhaités des suiveurs sont fixés par : Suiveur1 $(l_1^d = 1.5 ; \psi_1^d = \pi)$, Suiveur2 $(l_2^d = 1.5 ; \psi_2^d = \frac{3\pi}{2})$, Suiveur3 $(l_3^d = 1.5 ; \psi_3^d = \frac{\pi}{2})$ pour créer la formation

de cordes ;

Les variables d'état et les entrées de contrôle sont également similaires à la tâche de formation de la figure(3.12) selon le contrôleur (3.23).

3.7 Évitement d'obstacle par champ de potentiel

Le robot n'a pas d'information globale sur l'environnement ; aucune carte de l'environnement n'est mise en mémoire et aucune information préalable n'est connue concernant les obstacles ; La méthode du champ potentiel artificiel[36] consiste à imaginer que le robot peut se comporter comme une particule électrique qui peut être attirée ou repoussée par d'autres objets suivant le signe de leur charge électrique. Il s'agit d'une approche réactive du guidage dans laquelle le chemin n'est pas planifié à l'avance. En physique, nous avons la relation (3.25) :

$$f = -grad(V(p)) \quad (3.25)$$

Où p est position du centre du robot dans l'espace, V un potentiel imaginé par le robot et f le vecteur de vitesse à suivre. Les obstacles seront représentés par des potentiels exerçant une force répulsive sur le robot tandis que l'objectif à suivre exercera une force attractive. Dans une situation où plusieurs robots doivent rester groupés tout en évitant les collisions, nous pouvons utiliser un potentiel de répulsion proche et un potentiel attractif éloigné Le Tableau 3.1 donne plusieurs types de potentiel pouvant être utilisés :

Potentiel	$V(p)$	$-grad(V(p))$
Attractive canonique	$\ p - \hat{p} \ $	$-\frac{p-\hat{p}}{\ p-\hat{p}\ }$
Attractive quadratique	$\ p - \hat{p} \ ^2$	$-2(p - \hat{p})$
Répulsive	$\frac{1}{\ p-\hat{q}\ }$	$\frac{p-\hat{q}}{\ p-\hat{q}\ ^3}$
Uniforme	$-\hat{v}^T p$	$\hat{v}^T$

TABLE 3.1 – différents types de potentiel pouvant être utilisés

Dans ce tableau, $\hat{p}$ représente un point attractif, $\hat{q}$ un point répulsif et $\hat{v}$ une vitesse souhaitée pour le robot. Nous modélisons le comportement souhaité de notre robot par le potentiel (3.26) :

$$V(p) = -\hat{v}^T p + \| p - \hat{p} \|^2 + \frac{1}{\| p - \hat{q} \|} \quad (3.26)$$

Où $-\hat{v}^T p$ le potentiel représente la consigne de vitesse, le potentiel $\|p - \hat{p}\|^2$ rend la position souhaitée $\hat{p}$ attractive et le potentiel $\frac{1}{\|p - \hat{q}\|}$ rend l'obstacle $\hat{q}$ repoussant. Le calcul de gradient de $V(p)$ est :

$$w(p, t) = -grad(V(p)) = - \left(\frac{\partial V(p)}{\partial p} \right)^T$$

$$w(p, t) = \hat{v} - 2(p - \hat{p}) + \frac{p - \hat{q}}{\|p - \hat{q}\|^3} \quad (3.27)$$

3.8 Architecture du système de contrôle

Un Système Mécanique Articulé (SMA) contrôlé par ses deux moteurs moteur D et moteur G qui désignent respectivement les moteurs droite et gauche. ω_d et ω_g sont respectivement les modules pour la commande des roues motorisées droites et gauches. Ce module utilise les mesures provenant des encodeurs E_d et E_g . Le télémètre laser est le module qui contrôle le télémètre laser. Il délivre des informations pour le niveau supérieur qui est une liste de distances polaires dont chacune dépend d'angle de mesure.

Le module champs potentiels fournit au robot un comportement réflexif pour éviter des obstacles, une zone de sécurité qui assure une navigation sans risque (figure 3.15).

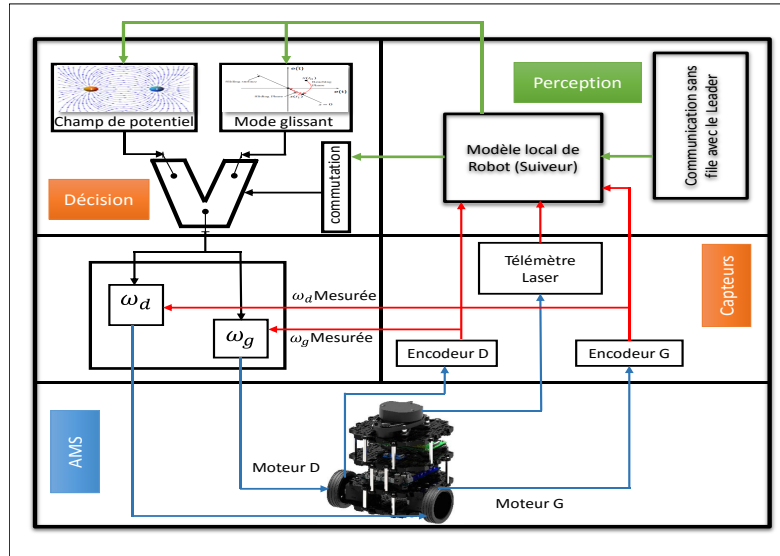


FIGURE 3.15 – L'architecture de contrôle suiveur

La Commutation assure la commutation entre deux fonctionnalités : le contrôle de formation et l'évitement d'obstacles pour que les suiveurs exécutent la formation désirée en évitant des obstacles.

Le module **Wifi ou Radio** pour la communication sans fil entre le leader et le suiveur, le leader envoie ses vitesses linéaire et angulaire et sa position au suiveur.

Avec cette architecture, les suiveurs robotisés peuvent maintenir de formation des robots mobiles dans un environnement encombré (à éviter les obstacles en utilisant le pilotage par champs potentiels) et à atteindre la formation (en utilisant le contrôle par mode glissant).

3.9 Poursuite d'un robot mobile à base d'un filtre SVSF

Le problème d'estimation de la pose d'un objet en temps réel (dans notre cas c'est le robot mobile Leader) pour chaque observation, est un enjeu important pour l'asservissement robotique. De nombreux systèmes d'estimation de pose se basent sur un filtre de Kalman (KF) ou Kalman étendu (EKF) qui fournissent une bonne estimation. Cependant, il peut provoquer une instabilité d'estimation en raison des incertitudes de modélisation non linéaire. Le filtre SVSF représente une bonne alternative, éventuellement, dans le cadre de la poursuite de trajectoire des robots[37] [38][39].

3.9.1 Le principe de fonctionnement du filtre SVSF

Le SVSF (Smooth Variable Structure Filter) a été introduit en 2007 par Saeid R. Habibi [40]. Ce filtre est basé sur les techniques de commande et d'estimation en mode glissant. Il est formulé dans un mode prédicteur-correcteur, où la commutation de gain est utilisée pour s'assurer que les estimations convergent vers des valeurs d'état réel. Comme le montre la figure(3.16).

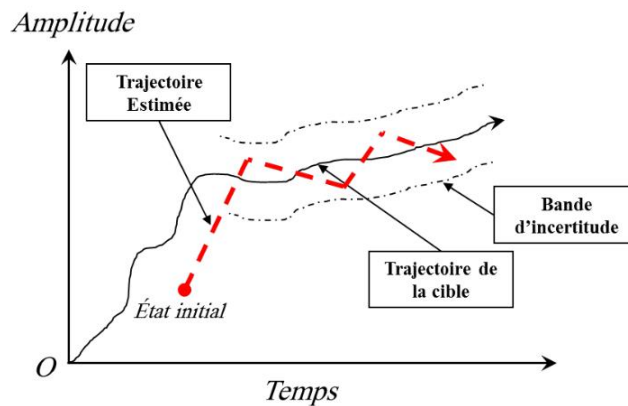


FIGURE 3.16 – Le principe d'estimation du filtre SVSF

Le SVSF utilise un sous-espace existant et une couche limite de lissage pour maintenir les estimations limitées dans une région de la trajectoire d'état réel. Le processus d'estimation

est résumé dans les équations (3.28) - (3.37). Le SVSF est totalement différent de l'EKF qui n'a ni matrice de covariance ni ne l'utilise. Le SVSF utilise un gain de commutation (gain du filtre SVSF) pour faire converger les estimations à l'intérieur d'une bande autour de la trajectoire réelle. Cette bande est appelée ; sous-espace d'existence (figure 4.19).

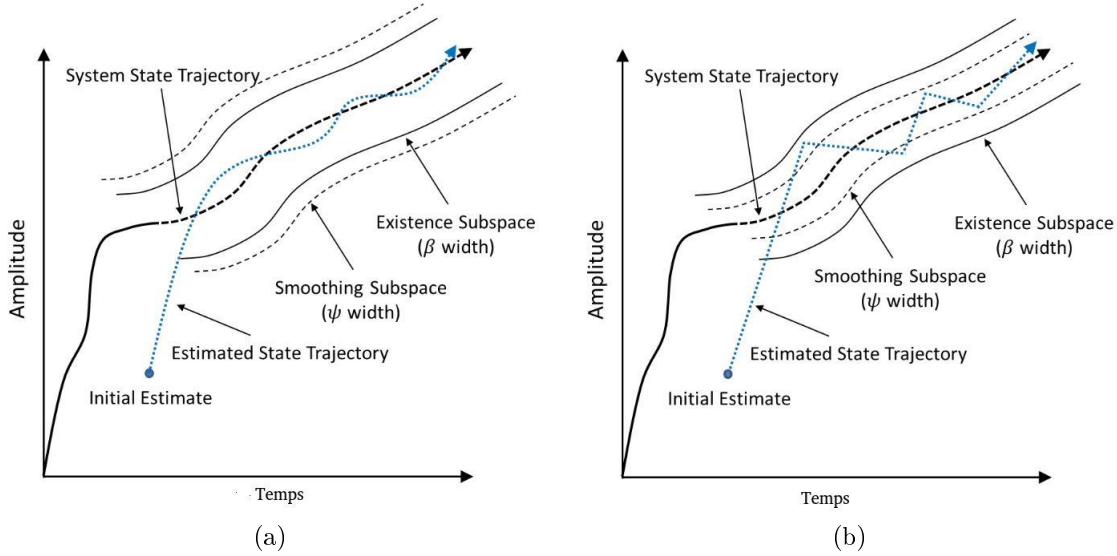


FIGURE 3.17 – exemple de moment où la trajectoire d'état estimée est lissée, et quand il y a un phénomène du broutement (a) trajectoire estimée lissée ($\psi \succeq \beta$) (b) présence d'un phénomène du broutement (chattering) ($\psi \prec \beta$) [40]

3.9.2 Formulation mathématique du filtre SVSF

En 2003, le filtre à structure variable (VSF) a été introduit comme une nouvelle méthode prédictor-correcteur utilisée pour l'estimation des états et des paramètres. Un terme correctif est ensuite appliqué pour calculer l'estimation d'état a posteriori, et le processus d'estimation est répété de manière itérative. Le SVSF était plus tard dérivé du VSF, et utilise un calcul de gain plus simple et moins complexe. Dans cette forme, le SVSF est stable et robuste à la modélisation des incertitudes et du bruit [41].

- Soit un système, linéaire ou non linéaire, modélisé sous sa forme d'état suivante :

$$\begin{cases} x_{k+1} = F(x_k, u_k, w_k) \\ z_{k+1} = H.x_k + v_k \end{cases} \quad (3.28)$$

- Le filtre SVSF passe par trois (03) étapes :

(1) *Initialisation* ; (2) *Prédiction* ; (3) *Mise à jour*

(1) Initialisation

$$\begin{cases} \hat{x}_{0|0} = x_0 \\ E_{0|0} = E_0 \end{cases} \quad (3.29)$$

(2) Prédiction

1. L'estimation de l'état à priori est prédite en utilisant le modèle estimé du système :

$$\hat{x}_{k+1|k} = \hat{F}(x_{k|k}, u_k) \quad (3.30)$$

2. L'équation (3.30) est utilisée pour calculer l'estimation à priori des mesures :

$$\hat{z}_{k+1|k} = \hat{H}.\hat{x}_{k+1|k} \quad (3.31)$$

3. Calcul de l'erreur à priori des mesures :

$$E_{k+1|k} = z_{k+1} - \hat{z}_{k+1|k} \quad (3.32)$$

(3) Mise à jour

4. Pour l'estimation d'état, le gain de correction SVSF est calculé par :

$$K_{k+1} = \hat{H}^+ (|E_{k+1|k}|_{abs} + \gamma \cdot |E_{k|k}|_{abs}) \circ \text{Sign}(E_{k+1|k}) \quad (3.33)$$

Où :

- γ est une matrice diagonale dont les éléments $0 \leq \gamma_{ii} \leq 1$, elle détermine la vitesse de convergence (convergence rate) choisie par l'utilisateur ;
- L'opérateur+indique la pseudo-inverse d'une matrice ;
- L'opérateur $\circ$ représente le produit élément par élément entre deux vecteurs.

5. L'estimation de l'état à posteriori à l'instant k+1 est calculée comme suit :

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \quad (3.34)$$

6. L'estimation de la mesure à posteriori à l'instant k+1 est calculé comme suit :

$$\hat{z}_{k+1|k+1} = \hat{H}.\hat{x}_{k+1|k+1} \quad (3.35)$$

7. Calcul de l'erreur à posteriori des mesures à l'instant k+1 est calculée comme suit :

$$E_{k+1|k+1} = z_{k+1} - \hat{z}_{k+1|k+1} \quad (3.36)$$

- Deux variables critiques dans le processus d'estimation du SVSF :

- L'erreur d'estimation à priori $E_{k+1|k}$;
- L'erreur d'estimation à posteriori $E_{k+1|k+1}$.

Le gain de correction discontinu K_{k+1} est formulé de sorte à garantir la convergence et la stabilité du processus d'estimation, c'est-à-dire, l'état estimé ne diverge pas et reste toujours au voisinage de l'état réel à l'intérieur d'une bande d'incertitude. En utilisant la condition de stabilité du Lyapunov. Cependant l'utilisation du gain K_{k+1} avec la fonction signe introduit des broutements (chattering) éventuellement de haute fréquence. Afin de remédier à ce phénomène, l'utilisation d'une forme de saturation à l'intérieur de la bande d'incertitudes peut atténuer ces broutements.

– Alors, l'équation (3.33) devient :

$$K_{k+1} = \hat{H}^+ (|E_{k+1|k}|_{abs} + \gamma \cdot |E_{k|k}|_{abs}) \circ Sat (E_{k+1|k}, \Psi) \quad (3.37)$$

- La forme de saturation utilisée dans l'équation (3.37) est définie dans l'équation (3.14).

3.10 Conclusion

Dans ce chapitre, une seconde approche qui nous propose une autre nouvelle basée sur les théories de mode glissant. Cette approche montre la robustesse et la performance dans tout divers formation pour un groupe de robots qui suivent leur leader, cette méthode est basée sur la théorie de la commande par mode glissant, l'avantage principal de cette technique par rapport à la plupart des approches existantes dans la littérature est de permettre au robot de pouvoir suivre le leader, tout en considérant la fonction de Lyapunov dans le domaine de l'espace d'état dans le but de permettre au système d'être asymptotiquement stable. L'approche proposée du mode glissant est composée d'une loi de commande basée sur la fonction de commutation (3.22) et nous avons décrit le principe de fonctionnement du filtre SVSF et développer son algorithme appliqué pour estimer la position de robot mobile (Leader) dans un environnement encombré (avec obstacles (en utilisant le pilote par champs potentiels équation (3.27) pour éviter ces obstacles). Dans le but de faire accélérer l'état du système pour rejoindre la surface dans un temps fini, les résultats des simulations entreprises (Matlab) ont démontré la robustesse de l'approche proposée et permettent non seulement de garantir la précision élevée de suivi, mais aussi de garder une grande stabilité du robot mobile.

Chapitre 4

Simulation et expérimentation

4.1 Introduction

Dans le premier chapitre, nous avons abordé les différents types de structures de formations et les approches de contrôles des systèmes multi-robots. Dans le second, la modélisation et la commande des robots unicycles. Et dans le troisième, nous avons étudié la commande en mode glissant, l'évitement des obstacles et le filtre correcteur-prédicteur SVSF.

A la base de toutes ces informations, nous avons choisi un scénario spécifique pour notre projet : « contrôle d'un système multi-robot par mode glissant ». Ce système multi-robots sera composé de quatre (4) agents de type TurtleBot (unicycle), le premier étant leader, les trois (3) autres seront suiveurs.

« Le leader » est équipé d'un laser pour l'assister dans sa navigation autonome, et d'un point d'accès sans fil pour partager son odométrie avec les suiveurs.

Ces derniers suiveurs sont équipés d'un récepteur, (afin de recevoir les informations envoyées par le leader pour leur permettre de suivre sa trajectoire), et d'un laser est utilisé pour éviter les obstacles d'une manière autonome.

Le présent chapitre, consiste à : définir l'environnement du logiciel nécessaire, en occurrence ROS, pour concrétiser le scénario ci-dessus. Puis d'écrire la mise en œuvre de l'environnement de simulation en 3D ainsi que les résultats de simulation.

Enfin, détailler la mise en œuvre de l'environnement réel, construction de deux (2) robots unicycles (leader+suiveur), calibrage de la camera, l'implémentation des codes de navigation, communication et détection visuelle. Le tout, sera concrétisé par un test réel.

4.2 ROS (Robot Operating System)

« ROS » est un méta système d'exploitation, il se situe, entre le système d'exploitation et le « middleware ». Il fournit des bibliothèques et des outils pour développer le software des robots. ROS offre des services permettant l'abstraction matérielle ou encore la gestion des communications inter-processus, afin d'augmenter la réutilisabilité du code et le partage simplifié sur d'autres plateformes. Une grande majorité des distributions (ROS supportent Ubuntu). Cependant, il existe des supports pour Fedora ou encore Mac OS X. Concernant le développement du logiciel, les principaux langages de la programmation utilisés sont C++ et Python [43].

La figure(4.1) montre les composants principaux de ROS et leurs interconnexions. Les blocs rouges correspondent aux parties différentes entre la simulation et l'environnement réel [42].

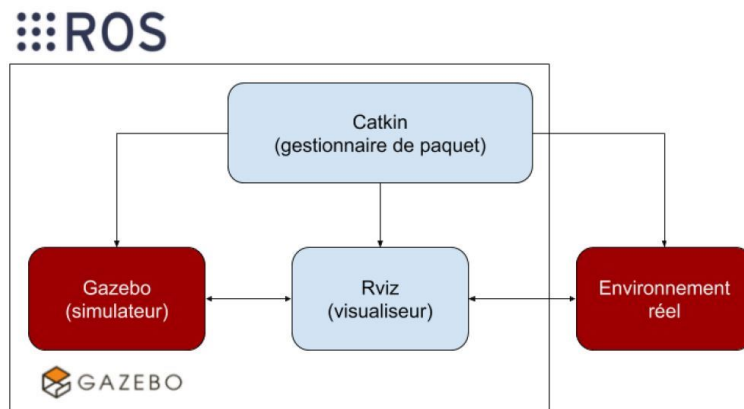


FIGURE 4.1 – Interaction entre les différents outils ROS

4.2.1 Les notions de base de ROS [43]

a) Création de l'environnement

Pour la création de l'espace de travail, il faut d'abord créer le répertoire associé par la commande :

```
~$ mkdir -p ~/catkin_ws/src
```

La commande « *catkin_make* » est un outil pratique pour travailler avec les espaces de travail catkin. Il créera un lien CMakeLists.txt dans le dossier «src».

```
~/catkin_ws$ catkin_make
```

b) Création d'un paquet

Le paquet c'est l'unité principale d'organisation logicielle de ROS. Il contient des nœuds, des bibliothèques externes, des données et des fichiers de configuration xml. Pour créer un nouveau paquet et ses dépendances, il faut utiliser la commande « *catkin_create_pkg* », tout en spécifiant le nom du paquet et chacune de ses dépendances .

```
~/catkin_ws/src$ catkin_create_pkg <package_name> [depend1] [depend2] [depend3]
```

c) Nœud (node)

On peut définir un nœud comme étant un fichier exécutable dans un paquet ROS. Les nœuds ROS utilisent une librairie client pour communiquer avec les autres nœuds. Les nœuds peuvent publier ou souscrire à des topics. Les nœuds peuvent fournir ou utiliser un service. Les librairies clientes sont « *rospy* » pour python et « *roscpp* » pour C++.

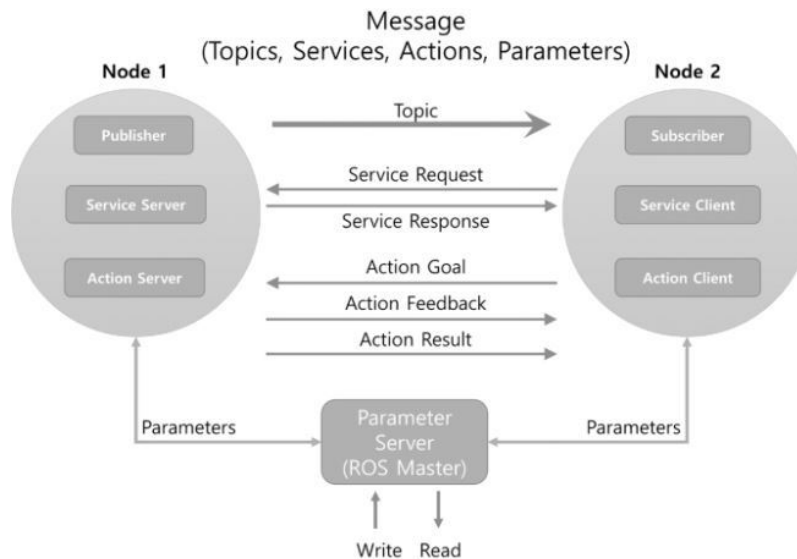


FIGURE 4.2 – Relation entre les nœuds, les services et les topics[44]

d) Topic

Un topic est un système de transport d'information basé sur le système de l'abonnement / publication (subscribe / publish). Un ou plusieurs nœuds pourront publier de l'information sur un topic et un ou plusieurs nœuds pourront lire une information. Pour obtenir la liste des topics actifs on utilise la commande :

```
:-$ rostopic list
```

e) **Master**

Le Master est un service de déclaration et d'enregistrement des nœuds, il leur permet ainsi se connaître et d'échanger des informations.

f) **roscore**

Roscore est constitué d'un master, d'un rosout et d'un serveur de paramètres. Il doit être lancé systématiquement, mais une seule fois.

4.2.2 Les fonctions et les outils de base de ROS

a) **Catkin**

Catkin est le constructeur officiel de ROS. Il comprend des outils comme des macros « CMake », ou des scripts « Python » pour une meilleure association du code.

b) **roslaunch**

Est un outil qui permet de lancement facile de plusieurs nœuds ROS localement et à distance via SSH, et la définition des paramètres sur le serveur de paramètres.

c) **Rviz**

Rviz (ROS visualisation) est un outil de visualisation 3D. Il permet de collecter les différentes informations des nœuds afin de présenter un aspect visuel des valeurs.

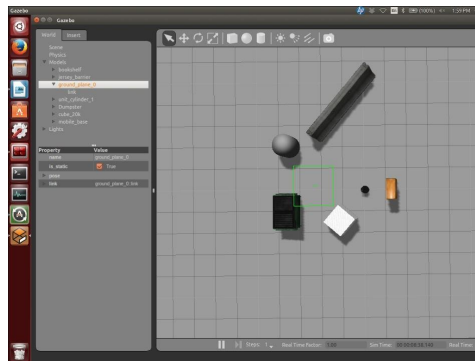
d) **Gazebo**

FIGURE 4.3 – Logiciel de simulation 3D Gazebo

La simulation de robot est un outil essentiel dans la boîte à outils de chaque roboticien. Un simulateur bien conçu permet de tester rapidement des algorithmes, de concevoir des robots, d'effectuer des tests de régression et de former un système d'intelligence artificiel à l'aide de scénarios réalistes. Gazebo offre la possibilité de simuler avec précision et efficacité des populations de robots dans des environnements intérieurs et extérieurs complexes. Gazebo est facile à utiliser, c'est un moteur de simulation 3D robuste et il présente des graphiques de haute qualité et des interfaces programmatiques et graphiques pratiques. De plus encore, Gazebo est gratuit avec une communauté dynamique [45].

e) **rqt**

rqt comprend deux (02) outils principaux :

- **rqt_graph** : c'est une interface graphique permettant l'analyse du graphe d'applications et les transferts de données via les topics.
- **rqt_multiplot** : c'est une interface de contrôle incrémentable. Elle se base sur le système de plugins de la bibliothèque de GUI qt.

```
~$ rosrn rqt_multiplot rqt_multiplot
```

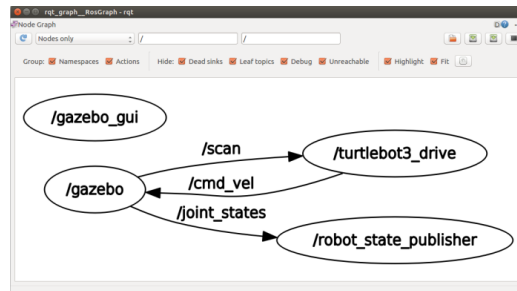


FIGURE 4.4 – Graphe d'un robot ROS simulé

4.2.3 TurtleBot3

Plusieurs robots potentiels ont été trouvés pouvant réaliser les tâches nécessaires à notre projet. Le choix s'est porté sur le TurtleBot3 Burger (figure4.5), car ce dernier répond aux besoins de notre projet.

TurtleBot3 Burger est un robot mobile de la nouvelle génération modulaire, compact et personnalisable. Les caractéristiques techniques du robot TurtleBot3 [45] sont présentées dans le tableau4.1

Caractéristiques	Waffle	Burger
Vitesse Maximale de translation	0,26 m/s	0,22 m/s
Vitesse Maximale de rotation	1,82 rad/s	2,84rad/s
DYNAMIXEL	XM430-W210-T	XM430-W250-T
SBC	Rasberry Pi 3	Rasberry Pi 3
Contrôleur Embarqué ARM	OpenCR (32-bit)	OpenCR (32-bit))
Capteurs	Rasberry Pi Camera360° LiDAR Gy+Acc+Mag 9 Axes	360° LiDAR Gy+Acc+Mag 9 Axes

TABLE 4.1 – Comparaison TurtleBot3 Burger et TurtleBot3 Waffle

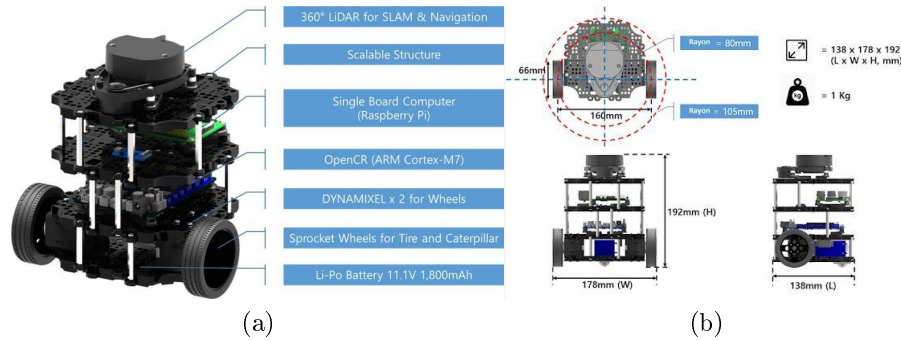


FIGURE 4.5 – Présentation de TurtleBot3

4.3 Simulation

La simulation s'est faite avec un PC portable modèle EliteBook-Revolve-810-G2 de disque SSD M.2 256 Go avec SDRAM DDR3L(1600 MHz) de 8 Go, fonctionnant sous **ubuntu16.04** [46]. Cette version de Linux est interfacée avec **ROS Kinatic** [43].

Nous avons utilisé quatre (04) robots de type TurtleBot3 burger. Le schéma global de l'environnement et de contrôle du système multi-robot Leader-Suiveur construit avec ROS (Gazebo) est détaillé sur la figure(4.6).

Nous avons considéré trois (03) scénarios pour tester les performances de l'algorithme de contrôle de formation, de filtre correcteur-estimateur (SVSF) et l'algorithme d'évitement des obstacles sur Gazebo. la construction de l'environnement de simulation sur la figure(4.6).

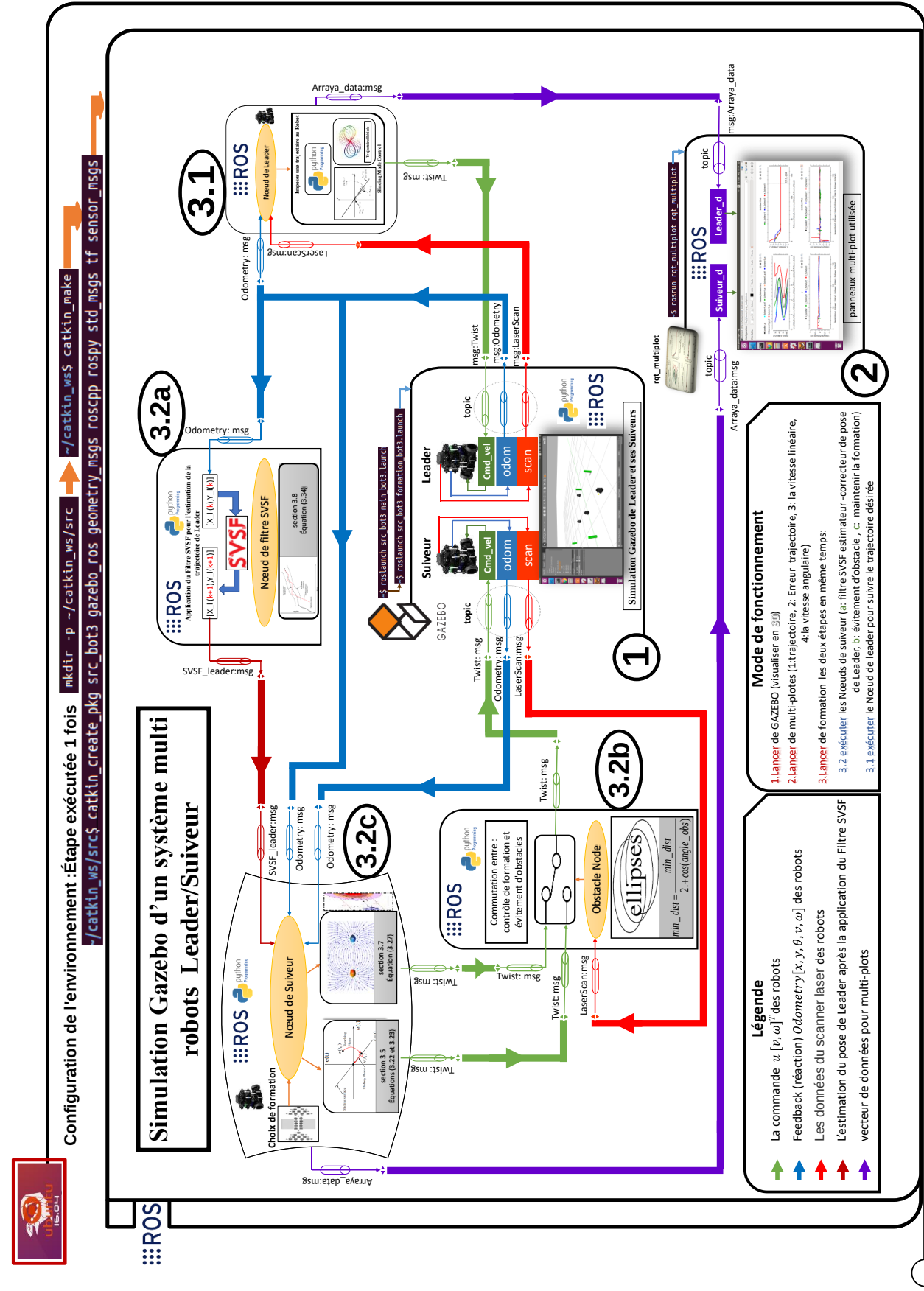


FIGURE 4.6 – Schéma global de contrôle du système multi-robot Leader-Suiveur avec ROS gazebo

4.3.1 Étapes de construction de l'environnement de simulation

- Création de paquet

Nous avons utilisé la commande « `catkin_create_pkg` » en spécifiant le nom du paquet et chacune des dépendances

```
~/catkin_ws/src$ catkin_create_pkg src_bot3 gazebo_ros geometry_msgs roscpp rospy std_msgs tf sensor_msgs
```

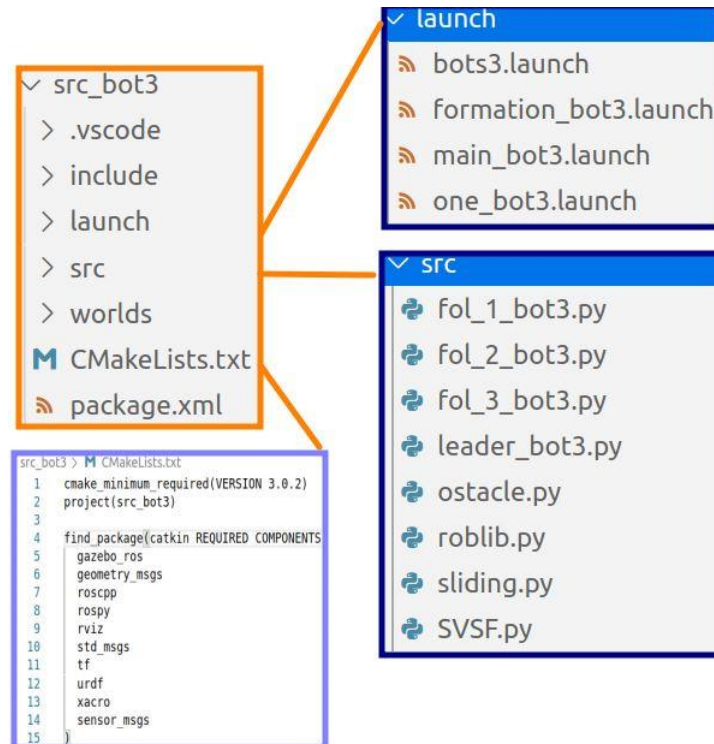


FIGURE 4.7 – Paquet du système développé

- Les nœuds

La liste des nœuds actifs dans le système est :

```
m@m-HP-EliteBook-Revolve-810-G2: ~
m@m-HP-EliteBook-Revolve-810-G2:~$ rostopic list
/Leader/robot_state_publisher
/Leader_29194_1595883146795
/SVSF_29452_1595883154007
/Suiveur_1/robot_state_publisher
/Suiveur_1_29376_1595883151677
/gazebo
/rosout
m@m-HP-EliteBook-Revolve-810-G2:~$
```

- Les fichiers launch

Nous avons deux fichiers launch dans notre simulation. Le premier, pour lancer la simulation dans Gazebo avec un environnement prédéfini, et le second concerne le lancement de la formation multi-robots. Ce dernier contient tous les nœuds (figure4.7).

```
~$ roslaunch src_bot3 main_bot3.launch
```

```
~$ roslaunch src_bot3 formation_bot3.launch
```

- Gazebo et topics

Notre système multi robots a été simulé sur gazebo (figure4.8).

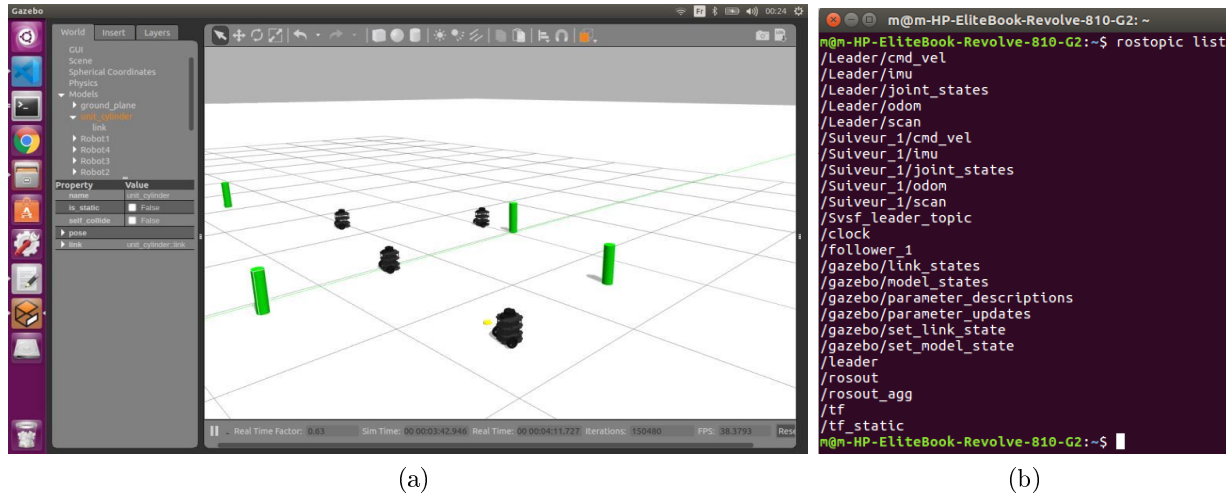


FIGURE 4.8 – (a) Simulation sur Gazebo (4 robots), (b) quelques topics relatifs aux robots

- **rqt_multiplot** : nous avons configuré 4 plots dans notre projet. Ils reçoivent les données des nœuds « Leader.py, Suiveur1.py, Suiveur2.py et Suiveur3.py » (Plot (01) : trajectoire des robots, Plot (02) : l'erreur de position par rapport au leader, Plot (03) : vitesse linéaire des quatre robots, 4 : vitesse angulaire des quatre robots).

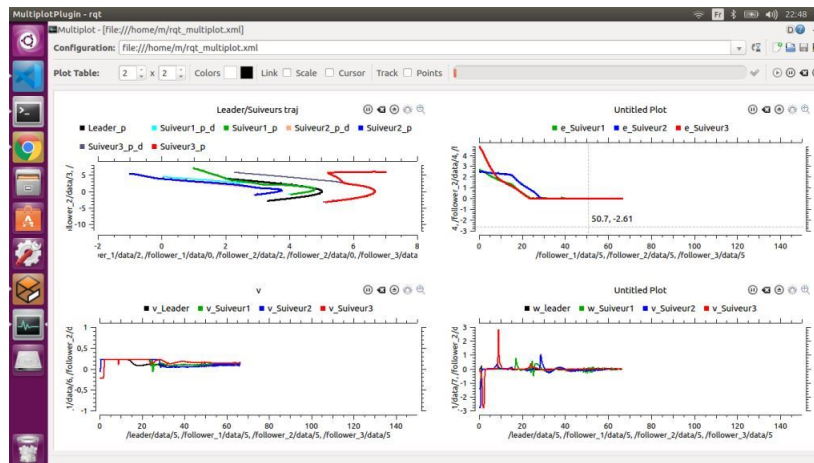


FIGURE 4.9 – Panneaux multi-plot utilisée

- rqt_graph

Le graphique dynamique qui représente l'état de notre SMR est sur la figure(4.10).

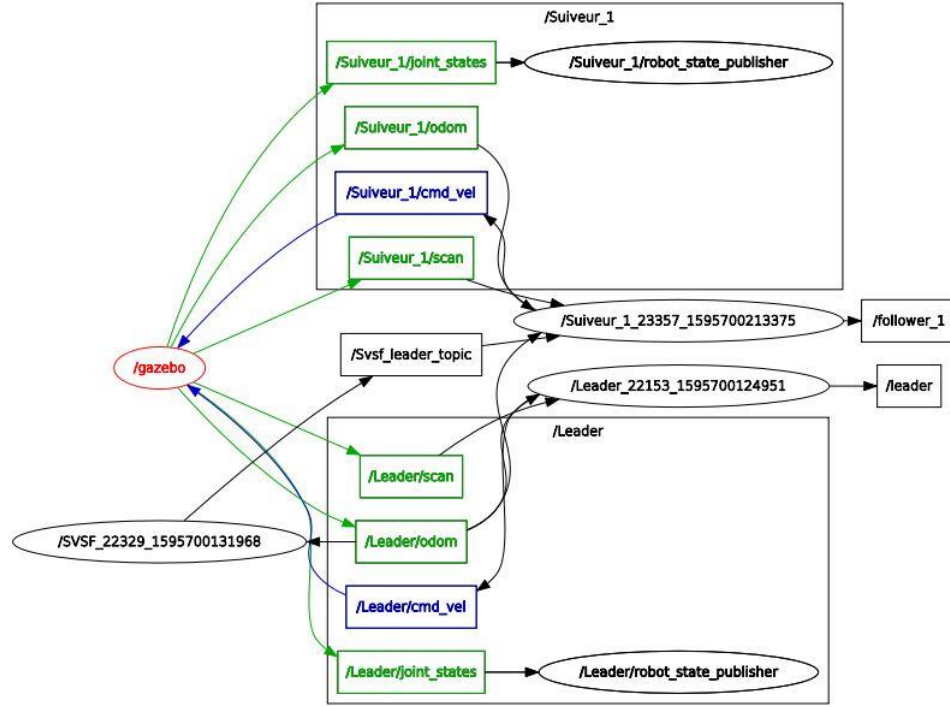


FIGURE 4.10 – Le graphique dynamique représentant l'état de notre système Leader-Suiveur (tous les topics et nœuds)

4.3.2 Scénario 01 : la formation sur une trajectoire circulaires sans obstacles

Les quatre (04) robots utilisés sont de type TurtleBot3 burger , ils sont dans un environnement non-encombré. Le leader veut atteindre une trajectoire circulaire prédéfinie, et chaque suiveur doit conserver la séparation et l'orientation souhaitées.

Les conditions initiales des quatre robots sont présentées sur le tableau 4.2. Ce test permet de valider notre algorithme de contrôle par le mode glissant (équation 3.23) avec la fonction de commutation $f_i(S_i)$ (équation 3.22), et les gains de la loi de contrôle en mode glissant (3.23) sont sur le Tableau 4.3

	Position initiale	Séparation souhaitée (m)	Orientation souhaité (rad)
Leader	$x = 2m, y = 1m, \theta = -1rad$	Cercle (c 3,3 et r =3)	
Suiveur 1	$x = 0m, y = 0.5m, \theta = 1.2rad$	1.5	$\frac{8\pi}{7}$
Suiveur 2	$x = 2m, y = -2m, \theta = -1rad$	1.5	$\frac{3\pi}{2}$
Suiveur 3	$x = 0m, y = 3m, \theta = 0rad$	1.5	$\frac{\pi}{2}$

TABLE 4.2 – Les conditions initiales des quatre robots pour scénario N°1

Coefficient	k_1	k_2	ε_1	ε_2	α_1	α_2	φ
valeur	1	0.6	0.04	0.04	0.6	0.6	0.1

TABLE 4.3 – Les gains de la loi de contrôle MG

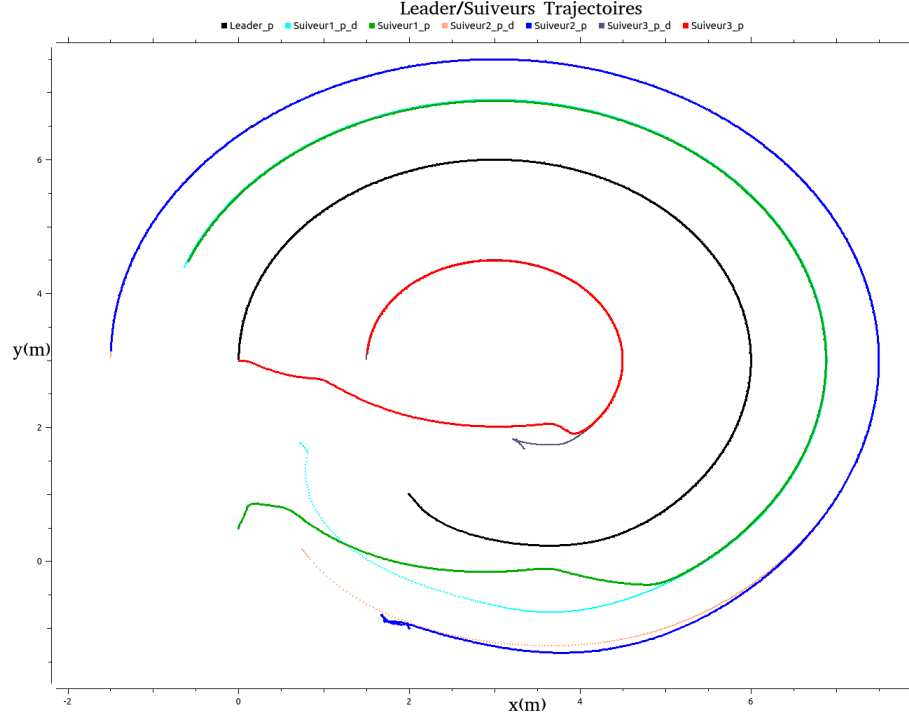


FIGURE 4.11 – Les trajectoires de leader et ses suiveurs dans un environnement non-encombré

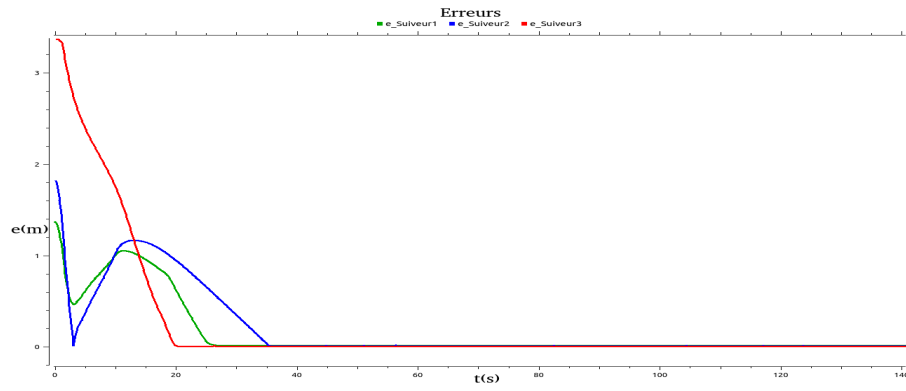


FIGURE 4.12 – Les erreurs de suivi pour les suiveurs

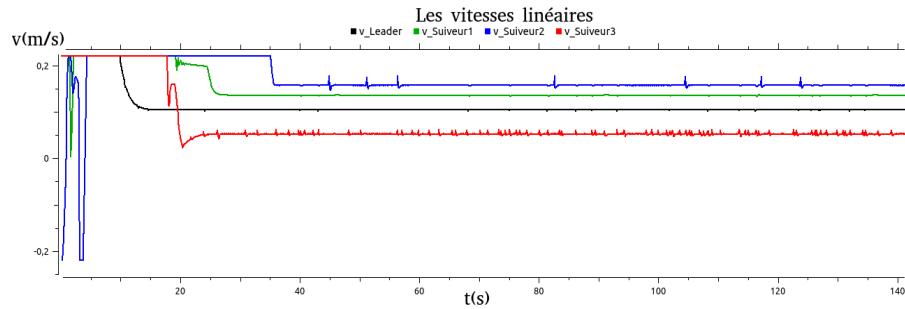


FIGURE 4.13 – Les vitesses linéaires des suiveurs et leader

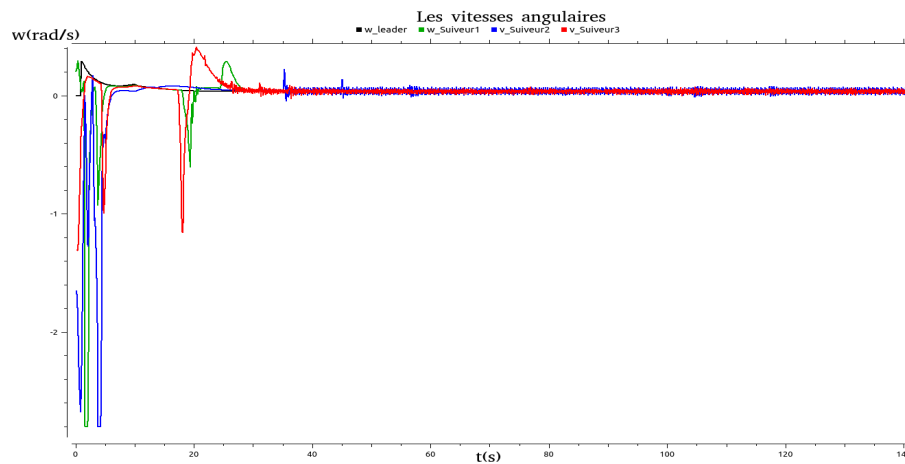


FIGURE 4.14 – Les vitesses angulaires du suiveur et du leader

Discussion des résultats

- La figure(4.11) montre une bonne adéquation entre les trajectoires des suiveurs (le bleu, le rouge et le vert) et celle du leader, tout en maintenant les séparations et orientations prédéfinies.
- La figure(4.12) représente les erreurs en position des suiveurs par rapport au leader. Nous constatons que les erreurs de suivi convergent toutes vers zéro à temps fini, ce résultat démontre clairement la robustesse du contrôleur par mode glissant.
- La figure(4.13-4.14)représente les vitesses linéaires et angulaires des robots. Dans ces figures, nous constatons que les suiveurs suivent deux (02) phases : la première est une phase à grande vitesse pour permettre aux suiveurs de rejoindre le leader le plus rapidement possible, et la seconde c'est la synchronisation des vitesses après avoir atteint la distance et l'orientation imposées.

4.3.3 Scénario 02 : la formation sur une trajectoire (S) sans des obstacles

Le but de ce scénario est la comparaison entre le comportement des robots avec et sans l'utilisation du filtre correcteur-predicteur SVSF.

Les conditions de l'environnement et de contrôle sont les mêmes que le scénario 01. La trajectoire prédéfinie est sinusoïdale (trajectoire (S)).

Dans le scénario 02-b, nous avons rajouté le filtre correcteur-prédicteur SVSF (équation 3.34), où les conditions initiales des quatre robots utilisées et le gain de commutation de SVSF sont présentés sur le tableau 4.4

	Position initiale	Séparation souhaitée (m)	Orientation souhaité (rad)	gain de commutation (Ψ)
Leader	$x = 2m, y = 4m, \theta = -0.7rad$	Trajectoire (S)		
Suiveur 1	$x = 1m, y = 7m, \theta = -0.7rad$	2	$\frac{8\pi}{7}$	0.08
Suiveur 2	$x = -1m, y = 5.5m, \theta = -1rad$	1.8	$\frac{4\pi}{3}$	0.08
Suiveur 3	$x = 7m, y = 6m, \theta = 0rad$	1.8	$\frac{2\pi}{3}$	0.08

TABLE 4.4 – Les conditions initiales des quatre robots pour 2ème scénario

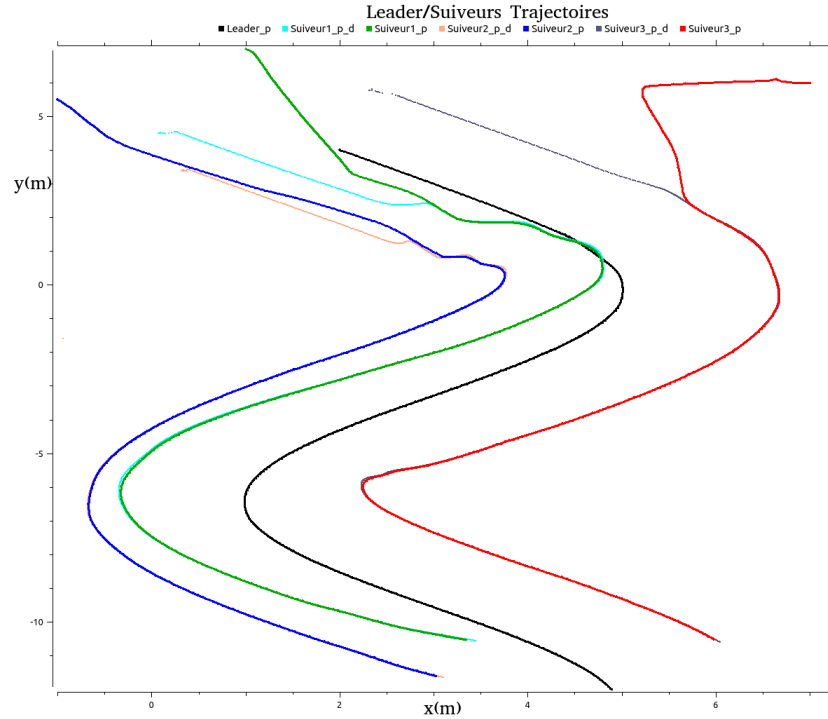


FIGURE 4.15 – Les trajectoires de leader et ses suiveurs dans un environnement non-encombré

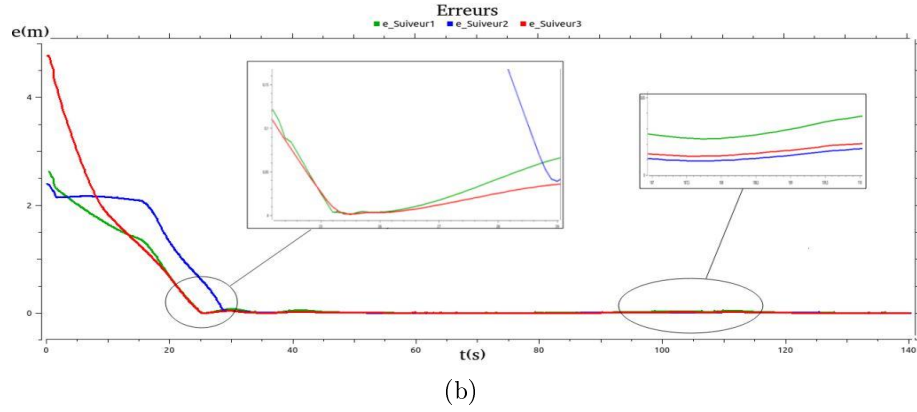
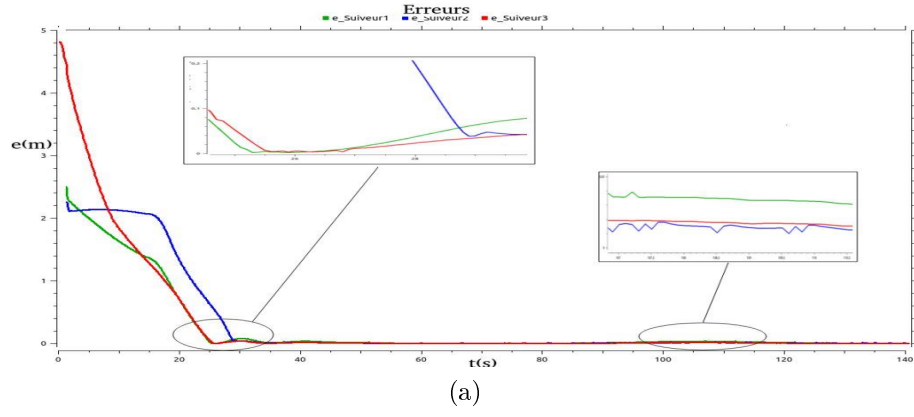


FIGURE 4.16 – Les erreurs de suivi pour les suiveurs : (a) sans filtre SVSF, (b) avec filtre

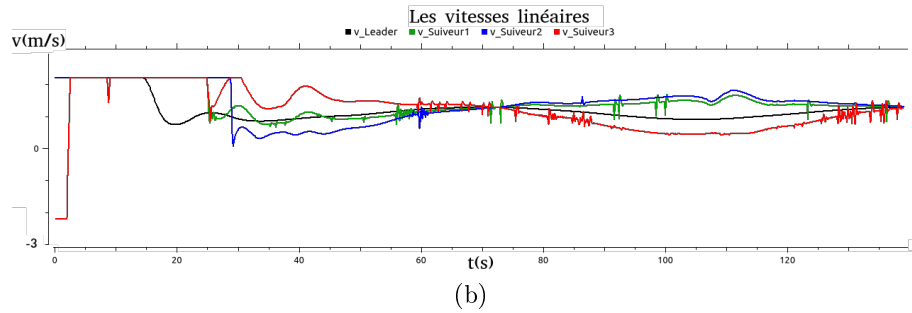
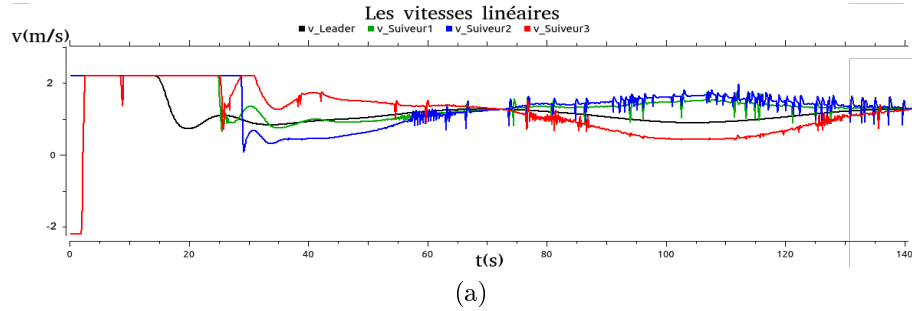


FIGURE 4.17 – Les vitesses linéaires du suiveurs et leader : (a) sans filtre SVSF, (b) avec filtre

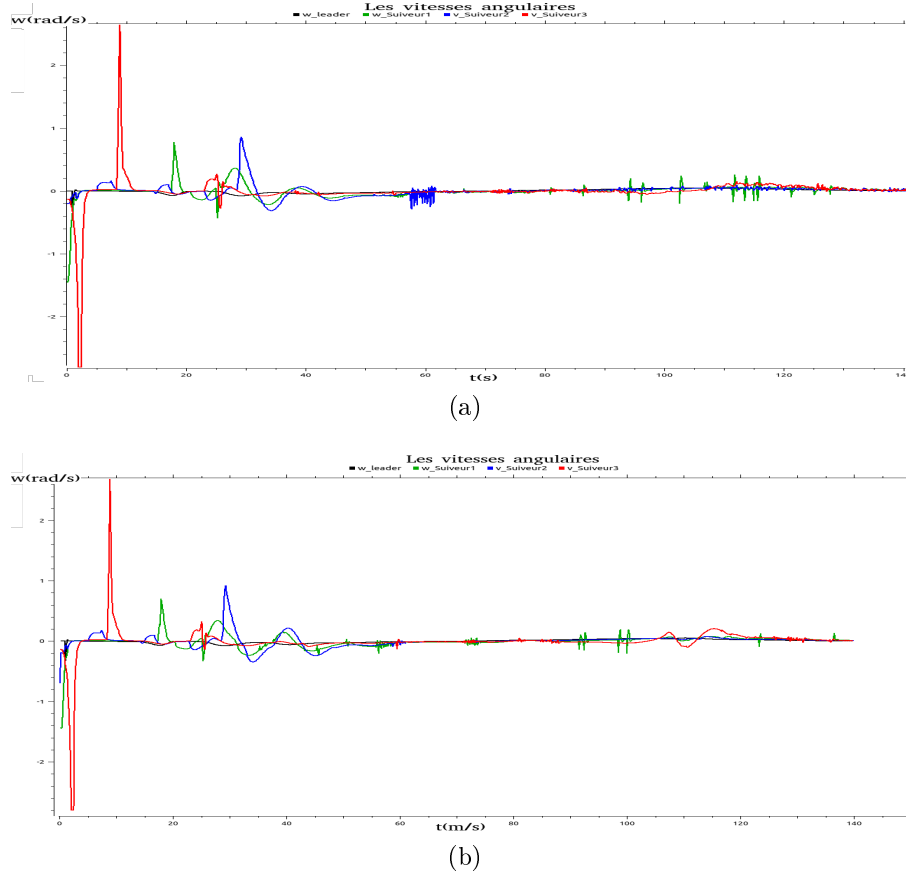


FIGURE 4.18 – Les vitesses angulaires du suiveur et du leader : (a) sans filtre SVSF, (b) avec filtre

Discussion des résultats

- La figure(4.15) montre une bonne adéquation entre les trajectoires des suiveurs (le bleu, le rouge et le vert) et celle du leader, tout en maintenant les séparations et orientations prédéfinies.
- Dans les figures(4.17 - 4.18), nous constatons une diminution nette du phénomène de broutement (chattering), ce qui démontre l'efficacité du filtre correcteur-predicteur SVSF.
- La figure(4.16) représente les erreurs de position des suiveurs. Elle montre que les erreurs de suivi convergent vers le zéro à temps fini dans les deux cas (a et b), en plus dans cette figure(4.16-b) (avec filtre SVSF) les erreurs de suivi convergent plus rapidement que dans la figure(4.16-a), se qui démontre la robustesse du filtre correcteur-estimateur SVSF.

4.3.4 Scénario 03 : la formation sur une trajectoire (S) avec des obstacles

Dans ce scénario, les quatre robots évoluent en formation sur une trajectoire sinusoïdale (S) encombrée (avec des obstacles). Nous avons utilisé le même contrôleur et filtre correcteur-estimateur SVSF que dans le scénario N°2 partie (b).

Le but de ce scénario est d'imposer au suiveur une distance et une orientation par rapport au leader, tout en évitant les obstacles placés dans leurs passages.

Les conditions initiales des quatre robots utilisées et le gain de commutation de SVSF sur le tableau 4.4

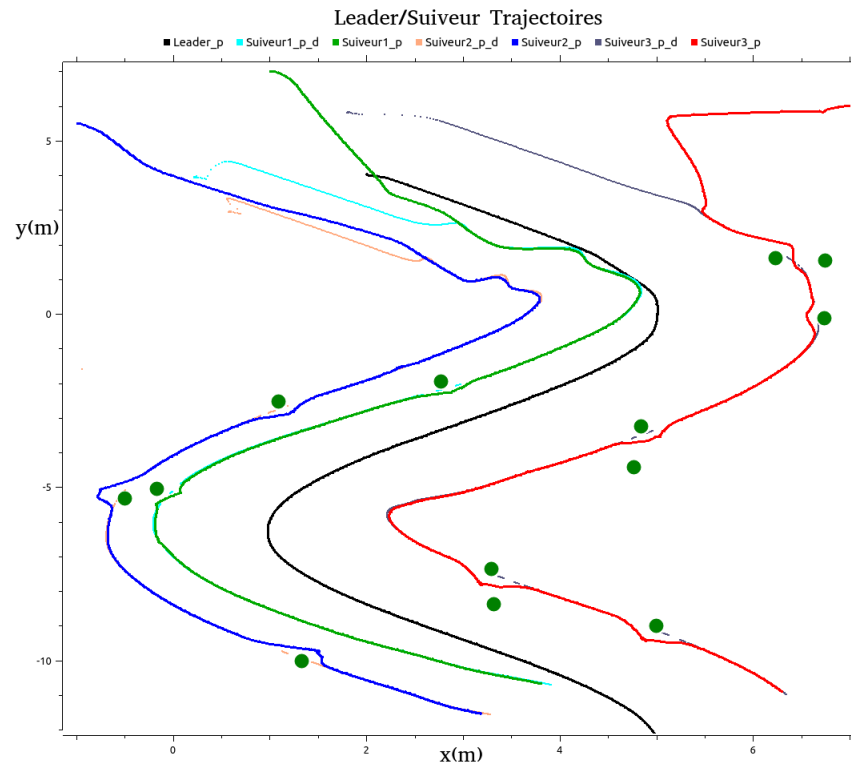


FIGURE 4.19 – Les trajectoires de leader et ses suiveurs dans un environnement encombré (avec des obstacles)

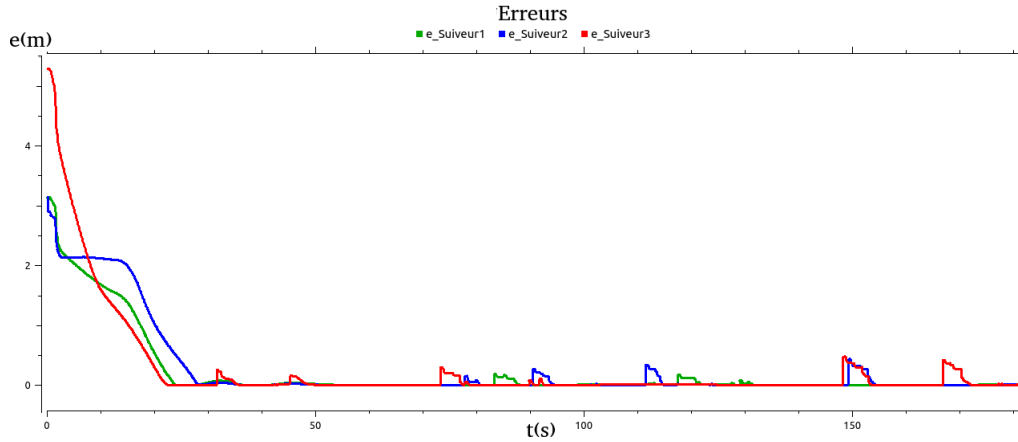


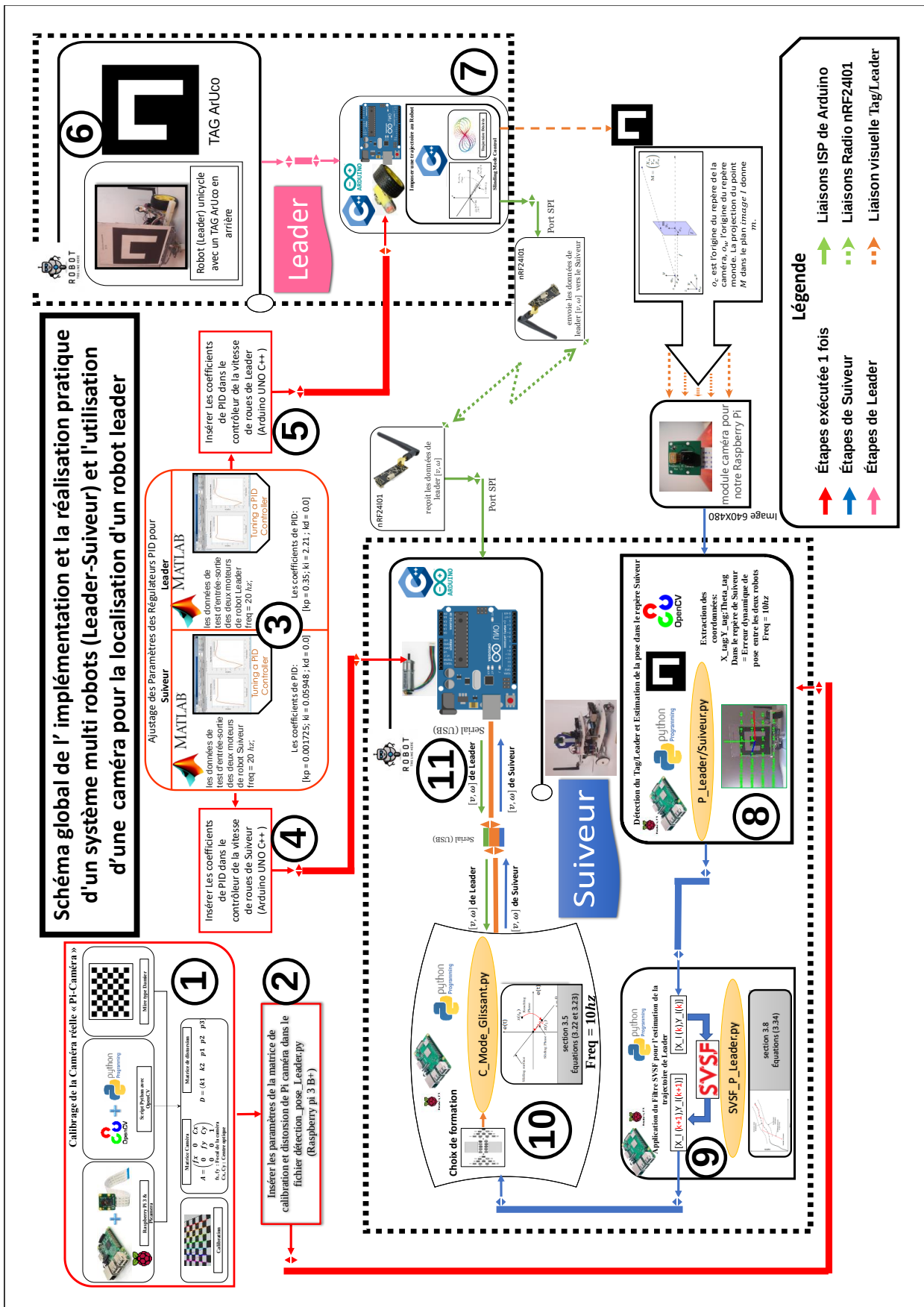
FIGURE 4.20 – Les erreurs de suivi pour le suiveur

Discussion des résultats

- Dans la figure(4.19), nous remarquons que le leader (noir) est en mode planification supervisée de grande vitesse. Les robots suiveurs (bleu, rouge et vert) sont en mode réactif tendant à rejoindre leurs positions dans la formation. Les quatre robots naviguent en formation triangle. Pendant cette navigation en formation, nous avons constaté que les suiveurs, après leur rencontre avec des obstacles, quittent leurs trajectoires dans la formation, le temps d'éviter les obstacles en mode réactif (en évitant des obstacles et en utilisant le pilote par champs potentiels (3.27)). Une fois cet obstacle est évité avec succès, les robots en formation reviennent rapidement et continuent leur avancement sur la trajectoire imposée jusqu'à ce que le leader atteigne sa destination finale.
- La figure(4.20) montre que les erreurs de suivi des suiveurs convergent vers zéro dans tous les cas. Les écarts constatés, sont dus aux évitement des obstacles rencontrés.
- Les deux figures de ce scénario montrent la robustesse de notre contrôleur de formation leader-suiveur par le mode glissant, le filtre correcteur-estimateur SVSF et surtout l'efficacité de notre algorithme d'évitement des obstacles par champs potentiels.

4.4 Réalisation et implémentation

Cette partie décrit la mise en place de la plateforme expérimentale que nous avons développée afin de valider nos propositions de stratégies de contrôle de formation Leader-Suiveur dans la première partie. Nous avons réalisé un groupe de deux robots mobiles. le Schéma globale de l'implémentation et réalisation pratique de ce groupe de robots (Leader-Suiveur (figure4.21)). Tandis que les simulations et les tests des différents scénarios ont été simulés sur le Gazebo. Le travail a surtout concerné la conception et le développement d'une commu-



nication filaire USB (entre Raspberry pi et Arduino) et communication sans fil par le module radio nRF24l01, la détection du Tag/Leader (figure4.24) et l'estimation de la pose dans le repère suiveur se font à l'aide d'un capteur caméra monté sur le Raspberry pi 3B+ de suiveur, afin d'assurer une communication explicite entre les deux robots. Dans la première étape, la conception mécanique du robot (Suiveur) mobile a été réalisée sur le logiciel **OpenSCAD** afin de l'imprimer à l'aide d'imprimante 3D (figure4.25).

Enfin, nos efforts se sont orientés vers l'intégration mécatronique d'un groupe de deux robots. Nous avons développé les modules de contrôle-commande pour chaque robot.

4.4.1 Les composants de Robots

La partie commande sur le suiveur est basée essentiellement sur un Raspberry pi 3 B+ (Annexe) avec son camera et Arduino UNO embarqué sur le robot suiveur, et d'autre Arduino UNO embarqué sur le leader (figure4.21). L'unité de commande de chaque robot est capable de gérer les flux de communication Radio avec RF24l01+ pour communiquer les vitesses de leader vers le suiveur.

Leader

- Robot 2WD (chassais + deux moteurs dc)
- Arduino UNO
- nRF24l01+ : communication radio
- L298N : module contrôleur de moteur DC
- Deux MH-Sensor : capteur de compteur de moteur infrarouge H

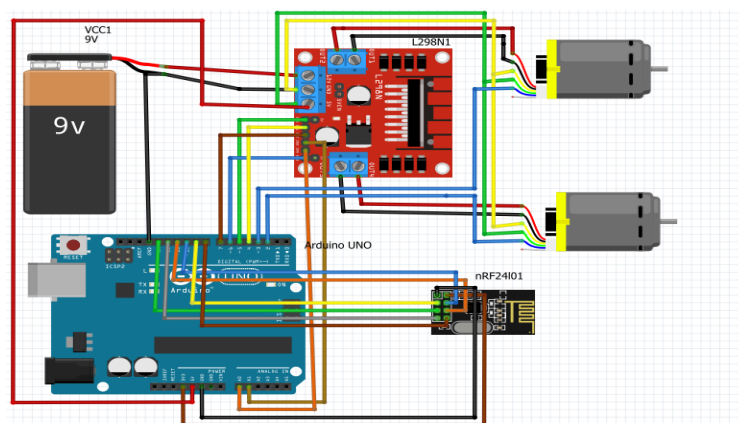


FIGURE 4.22 – Schéma électronique de leader

Suiveur

- Raspberry pi 3b+
- Camera vesion 1.3 de Raspberry PI
- Arduino UNO
- nRF24l01+ : communication radio
- L298N : module contrôleur de moteur DC avec deux moteurs dc ref : ig220019x00015r

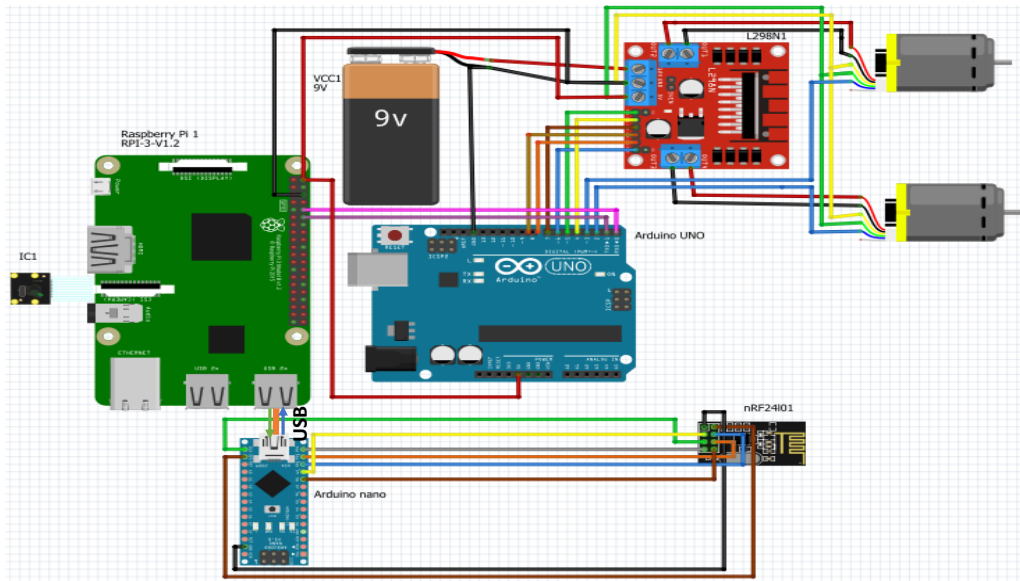


FIGURE 4.23 – Schéma électronique de suiveur

4.4.2 Les différentes vues des deux robots

Les roues motrices utilisées sur le suiveur sont indéformables de «4cm» de diamètre avec un noyau de «4mm» de diamètre adaptées à l'arbre du moteur pour lequel elles ont été conçues en l'occurrence le moteur et que nous avons utilisé pour la motorisation du robot. Ce sont des moteurs qui s'alimentent avec du 9V ayant une vitesse nominale de 15000 tr/mn, équipés d'encodeurs et de réducteurs avec un rapport de «19.22 :1». Dans le cadre de notre application, nous les avons fixés à des supports qui ont été fixés à leur tour au châssis du robot de telle sorte que les arbres des deux moteurs aient le même axe de rotation. Et pour assurer l'équilibre du robot, nous avons utilisé une roue libre. Les moteurs et les roues montés sur le chassie sont représentés sur la figure(4.25).

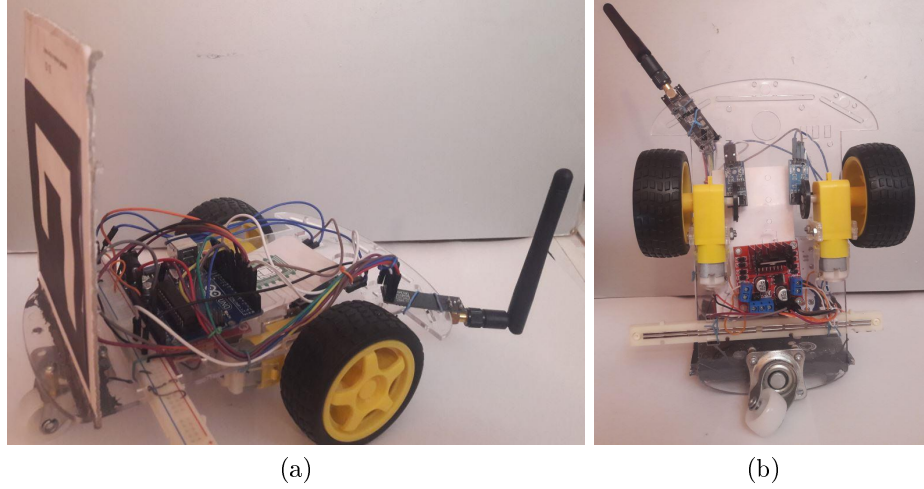


FIGURE 4.24 – Les différentes vues du leader. (a) Vue haute et gauche, (b) Vue de dessous

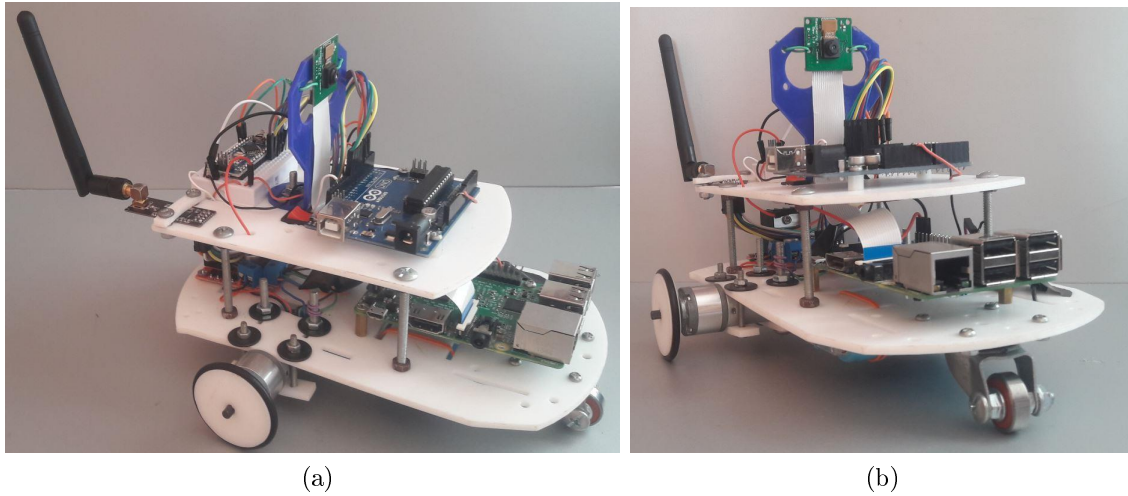


FIGURE 4.25 – Les différentes vues du suiveur (a) Vue droite, (b) Vue avant

4.4.3 Communication entre les deux robots

Le processus d'évaluation de la situation et de calcul du contrôle est centré sur le suiveur, qui reçoit les vitesses d'estimation $[v, \omega]^T$ de leader qui sont nécessaires au pilotage de la formation (voir équation 3.23). La communication entre les deux robots se produit rapidement par le module nRF24l01+ (Annexe).

4.4.4 Calibration de la caméra

Le robot suiveur équipé d'une caméra de Raspberry pi version 1.3 (Annexe). Ce capteur est particulièrement adapté à la navigation par robot mobile.

Le calibrage permet d'estimer la géométrie interne et celle externe de la caméra, quoique souvent, seule la géométrie interne est recherchée. La géométrie externe est la position et l'orientation de la caméra par rapport à l'objet de calibrage.

Nous avons fait la calibration en passant par [47] qui fait la calibration grâce à un damier (figure 4.26). Le programme permet de faire plusieurs photos du damier dans des différentes positions (pour réduire le bruit au maximum), ce qui permet d'avoir à la fin un fichier avec calibration de la caméra (la valeur de ses paramètres intrinsèques).

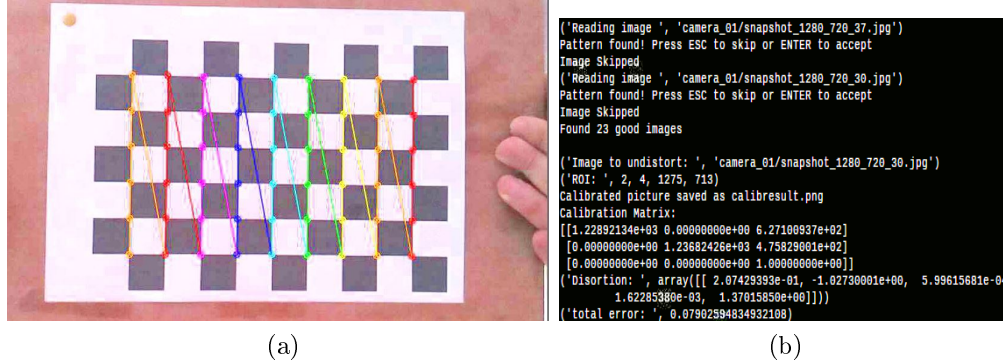


FIGURE 4.26 – (a) Damier de calibration, (b) matrice de calibration et distorsion

4.4.5 Localisation des marqueurs à l'aide d'Aruco

Nous avons choisi d'utiliser des codes dans GitHub de T.fiorenzani voir [47] pour faire la visualisation et la localisation de leader. Nous nous sommes servis ces paquets qui permettent, en utilisant OpenCV, de détecter un marqueur (Tag ArUco de poursuite voir [48]) qui collée sur le Leader et de renvoyer son identifiant et sa pose (c'est-à-dire un vecteur 3D pour la position et un quaternion pour l'orientation). Pour se servir d'Aruco, il faut au préalable calibrer la caméra qui filme les marqueurs, dont nous avons présenté la technique utilisée dans la partie précédente. Puis, il faut définir la taille de marqueur qui va être collé au leader (figure 4.27).

Le principal avantage de ces marqueurs est qu'un seul marqueur fournit suffisamment de correspondances (ses quatre coins) pour obtenir la position relative de Leader dans le repère camera donc le Suiveur. De plus, la codification binaire interne, les rend particulièrement robustes, permettant la possibilité d'appliquer des techniques de détection et de correction d'erreurs.

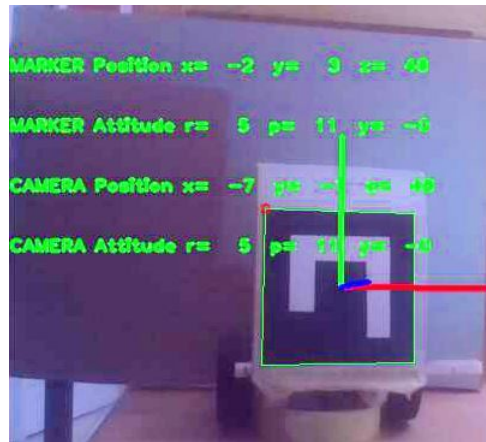


FIGURE 4.27 – Détection de leader en utilisant Aruco

a) **Asservissement de la vitesse angulaire des roues par PID (Proportionnel, Intégral, Dérivé)**

Le moteur DC fonctionne quand on lui envoie un courant. Le rotor tourne plus ou moins vite selon celui-ci, puis s'arrête dès lors qu'il n'est plus alimenté. Cependant, on ne connaît ni la distance angulaire parcourue, ni la vitesse de rotation. Aussi, si l'on veut réaliser un asservissement en vitesse, il faut transformer la consigne pour envoyer un courant pendant le temps nécessaire pour atteindre l'objectif.

L'asservissement est réalisé à l'aide d'un correcteur PID (à effets proportionnel, intégral, et dérivé). Cet asservissement peut être modélisé par la figure(4.28), la consigne (en Tr/min) à envoyer au moteur, et K_i , K_d , K_p des coefficients à modifier à la main jusqu'à trouver les coefficients les plus satisfaisants.

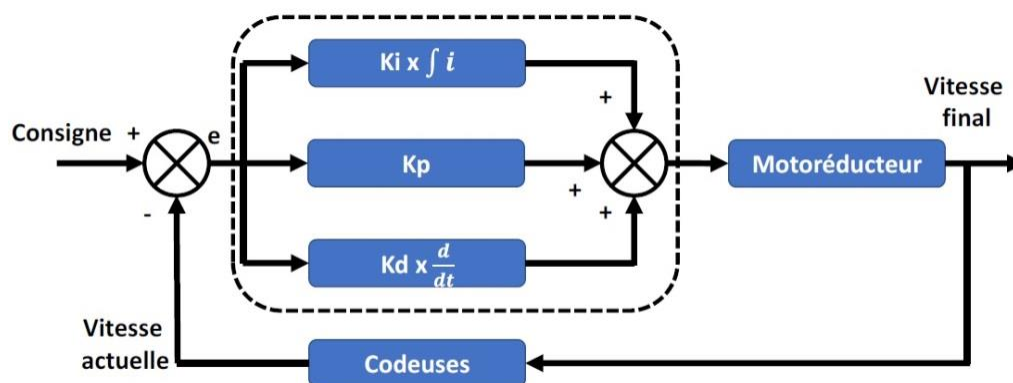


FIGURE 4.28 – L'asservissement est réalisé à l'aide d'un correcteur PID

b) Échantillonnage

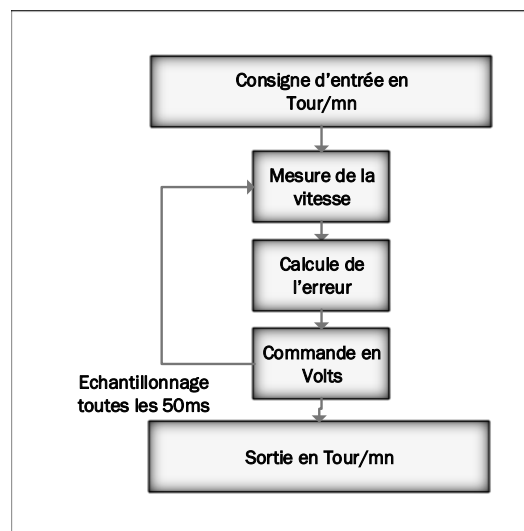
L'échantillonnage est une technique utilisée pour le traitement du signal et permet de dire à l'Arduino de calculer tous les intervalles donnés afin de ne pas saturer la puce de calculs inutiles. Ainsi, au lieu de faire des mesures continues et de ralentir le microprocesseur, celui-ci effectue ses calculs par exemple toutes les 50ms.

c) Les interruptions logicielles

Pour compter le nombre des tours pendant que le moteur tourne, on utilise une interruption qui va se déclencher à chaque transition de l'encodeur et viendra simplement incrémenter un compteur sur l'Arduino (int 0 sur pin 2 et int1 sur pin3). L'encodeur est intégré et constitué de deux capteurs hall (équivalents magnétique des capteurs optiques) qui permettent de lire ces interruptions sous forme de front montants et descendants. Deux capteurs, deux fonctions d'interruptions.

d) Asservissement en vitesse

La fonction « asservissement », appelée par la fonction d'échantillonnage



e) Commande et alimentation du moteur

La carte Arduino dispose des sorties qui peuvent être utilisées en mode PWM, c'est à dire en modulation de largeur d'impulsion. Ce sont des signaux logiques binaires de fréquence constante (500Hz) mais de rapport cyclique variable, et pour commander le moteur dans les deux sens, on utilise un pont en H intégré de type L298N voir l'annexe (système composé

de quatre transistors qui permet de contrôler électroniquement le sens du courant ou de la tension).

4.4.6 Expérimentation

La partie expérimentation sert à vérifier et valider l'efficacité de notre contrôleur basé sur la commande en mode glissant, afin de maintenir une formation SMR souhaitée, et la robustesse du filtre correcteur-prédicteur SVSF.

Après le montage complet des deux robots, le calibrage de la camera et la mise en place du robot leader tagué (TAG ArUco), il nous reste que l'implémentation de notre algorithme et l'observation du comportement du suiveur dans un environnement réel.

Malheureusement, l'expérimentation n'a pas pu être effectuée à cause de la pandémie COVID-19 et du confinement. Étant donné que la plupart des équipements et pièces nécessaires à la construction de nos robots sont importés, et avec la fermeture des frontières, les dénicher est devenu très compliqué. Nous avons pu les trouver, mais trop tard pour effectuer des testes concluant pour notre projet.

4.5 Conclusion

Le quatrième chapitre présente en premier lieu la simulation de trois (3) scénarios sur quatre (4) robots de type TurtleBot3 contrôlés par l'algorithme de commande en mode glissant avec ou sans correcteur-estimateur SVSF, et cela dans des différents environnements (avec ou sans obstacles). La formation simulée (formation Leader-Suiveur) est très réaliste. Le TurtleBot3 sur Gazebo a été le meilleur choix, car il répond mieux aux besoins de notre projet. A la fin de ce chapitre nous avons présenté la conception et la réalisation de la plateforme expérimentale pour la concrétisation du système coopératif multi-robot que nous avons conçue et réalisée. Les résultats obtenus montrent l'efficacité du système conçu sur les points de vue matériel et logiciel. Nous avons montré dans ce chapitre la robustesse et la performance de la commande par mode glissant dans plusieurs formations multi-robots (leader-suiveurs). La cohérence de la formation souhaitée s'est vue augmenté en implémentant un filtre correcteur-estimateur SVSF.

Enfin, nous avons présenté l'architecture matérielle du système développé, la fabrication et le montage complet des deux robots (leader tagué (ArUco) + suiveur) et le calibrage de la camera.

Conclusion générale

Ce mémoire présente un cadre de contrôle multi-robots en utilisant l'approche leader suiveur. L'objectif principal de ce mémoire est d'étudier la faisabilité de développer un système de robot de suivi de leader utilisant un système contrôle par mode glissant. Afin d'atteindre cet objectif, les objectifs suivants ont été atteints dans cette thèse :

1. Tester les résultats sur Matlab et Ros.
2. Développer la conception de l'algorithme qui est capable d'éviter simultanément obstacles et suivre le leader désigné dans un environnement non structuré.
3. Simulation de suivi, lorsqu'il n'y a pas d'obstacle dans l'environnement.
4. Simulation de suivi, lorsqu'il y a un obstacle dans l'environnement.
5. Simulation de suivi avec l'utilisation du filtre de SVSF (Smooth Variable Structure Filter) ;
6. Calibrage du caméra et détection d'obstacle.
7. Communication leader suiveur par le module radio.

Dans ce mémoire on a bien défini les système multi-robots et on a présenté les structures de contrôle afin de choisir la structure hybride qui était la plus efficace. Après on a exposé les approches de contrôle qui sont disponible et nous avons choisi l'approche de leader suiveur qui est le sujet principale du mémoire. Pour le suivi nous avons défini le mode glissant et son principe, nous avons développé son algorithme après les résultats de simulations sur Matlab ont montré la robustesse de ce contrôle.

Dans le monde de la robotique, gazebo est le plus connu des logiciels et le plus performant dans les simulations en 3D destiné à mettre en oeuvre des scénarios réalistes. Nous y avons construit l'environnement qui a permis de concrétiser ce projet. Le Robot utilisé est turtlebot, nous avons effectué plusieurs scénarios (évitement d'obstacle par champ potentiel, filtre correcteur-estimateur SVSF, suivi de trajectoire), afin de démontrer la robustesse du contrôle par le mode glissant et le filtre SVSF en utilisant python. Les résultats de simulation sur Ros et Matlab ont démontré l'efficacité et la stabilité des robots et l'algorithme de suivi en utilisant l'approche leader suiveur qui a aussi prouvé leur bon fonctionnement. Pour la partie

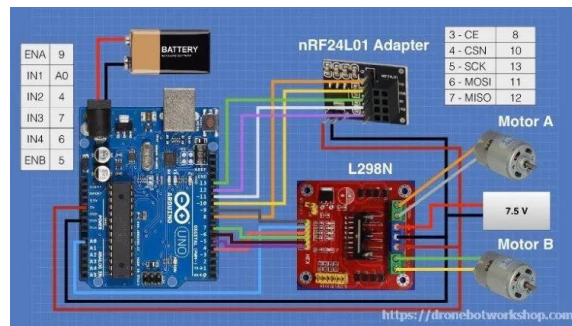
pratique on a fait une implémentation du caméra pour la poursuite visuelle basée sur des tags de type ArUco et de raspberry pi sur le suiveur et une communication radio entre le leader et le suiveur a été réalisée. Une autre entre la carte arduino et le raspberry pi pour un changement des informations les résultats sont moins fiables que la simulation à cause du temps et la situation sanitaire du pays qui a un grand impact dans le travail sur le mémoire.

Les perspectives de ce modeste travail sont nombreuses jusqu'à la rédaction. Nous pouvons citer quelques unes qui sont réalisables (à court et à moyen termes) ci-dessous :

- Exploiter les réseaux de neurones pour remédier au problème de Chattering, tout en gardant les avantages de la commande par le mode glissant.
- Améliorer l'algorithme de détection visuelle du tag ArUco afin de le rendre plus robuste aux occlusions partielles comme l'algorithme PDAT (Parallel Detection And Tracking). Cet algorithme, entre autre, permet de détecter un tag par ses points clés même avec plus de 50% d'occlusion.
- Rendre l'algorithme de filtre correcteur-estimateur plus universel en utilisant l'algorithme IMM (Inter-acting Multiple Model), afin de permettre au suiveur de poursuivre n'importe quel type d'UGV (leader).
- Rendre l'algorithme de commutation entre deux fonctionnalités : le contrôle de formation et l'évitement d'obstacles en utilisant la logique floue.

Annexes

Les connexions entre Arduino Uno-nRF24l01-Module L298N



La carte Arduino Uno est basée sur un ATmega328 cadencé à 16 MHz. C'est la plus simple et la plus économique carte à microcontrôleur d'Arduino. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires.

Elle peut se programmer avec le logiciel Arduino. Le contrôleur ATmega328 contient un bootloader qui permet de modifier le programme sans passer par un programmeur.

mémoire flash : 32 kB, mémoire SRAM : 2 kB, mémoire EEPROM : 1 kB, 14 broches d'E/S dont 6 PWM, 6 entrées analogiques 10 bits

bus série, I2C et SPI

Le module radio NRF24L01 est un émetteur-récepteur permettant d'établir une connexion et de transférer des données d'un appareil à un autre via des ondes radio.

Ils possèdent une antenne intégrée et fonctionnent en modulation GFSK avec 125 canaux possibles .

La vitesse de fonctionnement maximale est de 2 Mbps (Mégabits par secondes), avec une faible consommation électrique.

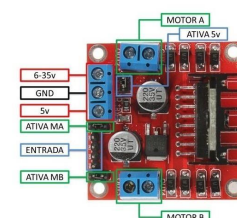
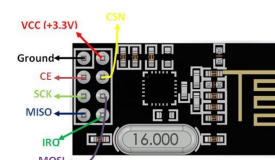
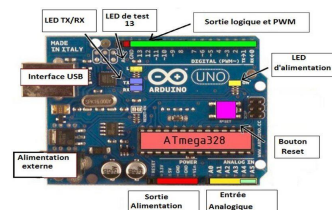
Tension d'alimentation : 1.9 à 3.6 V

Interface SPI jusqu'à 10 Mb/s (tolérant 5 V)

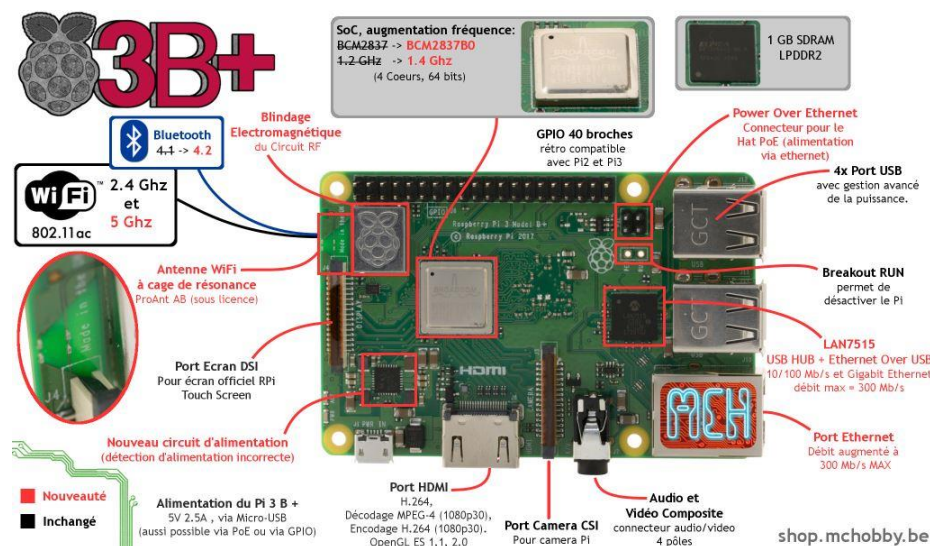
Le module L298N est un double pont-H, ou H-Bridge et peut contrôler des moteurs dans une plage de tension entre 5 et 35 VDC.

Dimensions : 43mm x 43mm x 27mm, Tension logique : 5V, Courant de commande logique : 0 à 36 mA, Courant Max : 2A

Puissance Max : 25W (L298N à 75 °C), Température de fonctionnement : -20°C à +125°C



Raspberry Pi 3B+



Grâce à ses caractéristiques et à son processeur ARM, la carte Raspberry Pi 3 fonctionne avec la distribution Linux de notre choix (ou même Android ou Firefox OS par exemple) installé sur une carte micro SD. Une solution économique et très pratique pour satisfaire notre application. La façon de son emploi n'a pas de limites, que ce soit comme plateforme de développement (puisque'il est compatible avec de très nombreux langages Perl, Ruby, Python, Java, Lua...), comme mini serveur web ou comme véritable média center efficace et tout à fait apte à décoder de la HD en 1080p (distribution OpenELEC optimisée pour Kodi / XBMC recommandée). Sans aucune limite, le Raspberry Pi 3 B est le support idéal pour exprimer vos projets...

Le modèle Raspberry Pi 3 type B+ est composé de composants fiables :

un CPU 64 bit quad core ARM Cortex-A53 intégré et cadencé à 1,4 GHz

un contrôleur graphique Broadcom Videocore IV

— 1 Go de mémoire vive pour assurer la fluidité de votre système.

— 1 lecteur micro SD / SDHC.

— 4 sorties USB 2.0.

— 1 port RJ45 (Ethernet 10/100/300 Mbps).

— 1 port HDMI.

— 1 audio Jack 3,5 mm.

— GPIO 40 broches + 4 nouvelles broches pour le PoE.

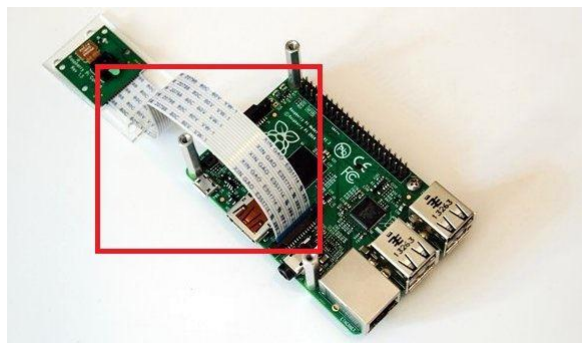
— Compatible Wifi 802.11 b/g/n/ac double bande 2,4 Ghz et 5 Ghz et Bluetooth 4.2 (Bluetooth Classique et LE).

— Une alimentation électrique d'au moins 2,5A est fortement recommandée.

Le module caméra pour Raspberry Pi

Le module reçu est une rev 1.3, le capteur Omnivision (OV5647) nous fait tout de suite penser à celui présent dans de nombreux smartphones (ressemblance flagrante avec le capteur d'un ancien

Samsung ACE). Il a une définition maximale de 5 millions de pixels et est capable de filmer en 1080p à 30 images/s ou en 720p à 60 images/s. Il est monté sur une petite carte qui se connecte au Raspberry Pi via une nappe (un bus de type CSI-2), ce qui laisse l'avantage de rendre disponible les ports USB de la carte peu nombreux.



Bibliographie

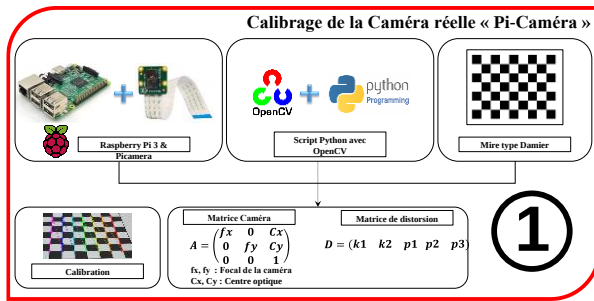
- [1] Z.PENG ,« Contribution à la commande d'un Groupe de Robots Mobiles Non-holonomes à Roues », thèse Doctorat Automatique, L'ecolle Centrale de Lille,2013, p. 156.
- [2] J. P. Desai, J. P. Ostrowski, et V. Kumar, « Modeling and Control of Formations of Nonholonomic Mobile Robots », IEEE Trans. Robot. Autom., vol. 17, no 6, p. 6, 2001.
- [3] Y.Bouterra « Commande Distribuée et Synchronisation de Robots Industriels Coopératifs»,thèse Doctorat Robotique,l'Université'Orléans,2012,p 143 .
- [4] M.Egerstedt,X.Hu « Formation Constrained Multi-Agent Control», IEEE Trans. Robot. Autom., vol. 17, no 6, p.ages 947-951, 2001.
- [5] Dong Sun, Can Wang, Wen Shang, et Gang Feng, « A Synchronization Approach to Trajectory Tracking of Multiple Mobile Robots While Maintaining Time-Varying Formations », IEEE Trans. Robot., vol. 25, no 5, p. 1074-1086, oct. 2009, doi : 10.1109/TRO.2009.2027384.
- [6] S.Wee et al. «Variable formation control of multiple robots via VRc and formation switching to accommodate mlarge heading changes by leader robot ».vol 11,no 6,p1-12,2019 .
- [7] J. P. Desai, J. Ostrowski, et V. Kumar, « Controlling formations of multiple mobile robots », in Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), Leuven, Belgium, 1998, vol. 4, p. 2864-2869, doi : 10.1109/ROBOT.1998.680621.
- [8] H. M. Ha, A. D. Nguyen, et Q. P. Ha, « Controlling formations of multiple mobile robots with inter-robot collision avoidance »,University of Technology Sydeney. 7p.2007
- [9] G. Orlando, E. Frontoni, A. Mancini, et P. Zingaretti, « Sliding mode control for vision based leader following », p. 6.
- [10] S.-G. Wee, Y. Dai, T. H. Kang, et S.-G. Lee, « Variable formation control of multiple robots via VRc and formation switching to accommodate large heading changes by

- leader robot », *Adv. Mech. Eng.*, vol. 11, no 6, p. 168781401985733, juin 2019, doi : 10.1177/1687814019857339.
- [11] L.ADOUANE.« Architectures de contrôle comportementales et réactives pour coopération d'un groupe de robots mobiles».thèse Doctorat AUTOMATIQUE.L'Université de franche-comté.2005.p 247.
- [12] T.-T. Yang, « Formation Control and Obstacle Avoidance for Multiple Mobile Robots : Formation Control and Obstacle Avoidance for Multiple Mobile Robots », *Acta Autom. Sin.*, vol. 34, no 5, p. 588-592, mai 2008, doi : 10.3724/SP.J.1004.2008.00588.
- [13] J. Borenstein et Y. Koren, « Real-time obstacle avoidance for fast mobile robots », *IEEE Trans. Syst. Man Cybern.*, vol. 19, no 5, p. 1179-1187, sept. 1989, doi : 10.1109/21.44033.
- [14] Mong-ying A. Hsieh et V. Kumar, « Pattern generation with multiple robots », in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, Orlando, FL, USA, 2006, p. 2442-2447, doi : 10.1109/ROBOT.2006.1642068.
- [15] A.Widyotriatmo et al « Implementation of Leader-Follower Formation Control of a Team of Nonholonomic Mobile Robots », *International journal of computers*,vol 12,n 6,pages 871-885.2017 .
- [16] A. Rawat et K. Karlapalem, « Multi-Robot Formation Control Using Reinforcement Learning », p. 5.
- [17] R. A. Brooks et A. M. Flynn, « FAST, CHEAP AND OUT OF CONTROL : A ROBOT INVASION OF THE SOLAR SYSTEM », *The British interplanetary Society*, vol 42 , pages 478-485. 9,1989.
- [18] T. Howard, M. Pivtoraiko, R. A. Knepper, et A. Kelly, « Model-Predictive Motion Planning : Several Key Developments for Autonomous Mobile Robots », *IEEE Robot. Autom. Mag.*, vol. 21, no 1, p. 64-73, mars 2014, doi : 10.1109/MRA.2013.2294914.
- [19] K.-B. Zhou, X.-K. Wu, M.-F. Ge, C.-D. Liang, et B.-L. Hu, « Neural-Adaptive Finite-Time Formation Tracking Control of Multiple Nonholonomic Agents With a Time-Varying Target », *IEEE Access*, vol. 8, p. 62943-62953, 2020, doi : 10.1109/ACCESS.2020.2980894.
- [20] J. Lü et Y. Qi, « Leader-following Formation Control for Multi-agent Systems with Sliding Mode Estimator under fixed Topologies », in *Proceedings of the 2nd International Conference on Computer Science and Application Engineering - CSAE '18*, Hohhot, China, 2018, p. 1-5, doi : 10.1145/3207677.3278048.

-
- [21] Das et al. «Adaptive sliding mode formation control of multipl.pdf ».archives of Control Sciences,vol 24,no 4, pages 515-543, 2014 .
- [22] D. Qian et Y. Xi, « Leader–Follower Formation Maneuvers for Multi-Robot Systems via Derivative and Integral Terminal Sliding Mode », Appl. Sci., vol. 8, no 7, p. 1045, juin 2018, doi : 10.3390/app8071045.
- [23] M. Defoort, T. Floquet, A. Kokosy, et W. Perruquetti, « Sliding-Mode Formation Control for Cooperative Autonomous Mobile Robots », IEEE Trans. Ind. Electron., vol. 55, no 11, p. 3944-3953, nov. 2008, doi : 10.1109/TIE.2008.2002717.
- [24] G. Campion, G. Bastin, et B. D’Andrea-Novel, « Structural properties and classification of kinematic and dynamic models of wheeled mobile robots », in [1993] Proceedings IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 1993, p. 462-469, doi : 10.1109/ROBOT.1993.292023.
- [25] D. Filliat, « Robotique Mobile »,Engineering school,ENSTA ParisTech, 176 p, 2011.
- [26] A. Cacitti et R. Zapata, « Reactive behaviours of mobile manipulators based on the DVZ approach », in Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), Seoul, South Korea, 2001, vol. 1, p. 680-685, doi : 10.1109/ROBOT.2001.932629.
- [27] M. A. Moqbel Obaid, A. R. Husain, et A. A. Mohammed Al-kubati, « Robust Backstepping Tracking Control of Mobile Robot Based on Nonlinear Disturbance Observer », Int. J. Electr. Comput. Eng. IJECE, vol. 6, no 2, p. 901, avr. 2016, doi : 10.11591/ijece.v6i2.9594.
- [28] J. Hermosillo et S. Sekhavat, « Feedback control of a bi-steerable car using flatness application to trajectory tracking », in Proceedings of the 2003 American Control Conference, 2003., Denver, CO, USA, 2003, vol. 4, p. 3567-3572, doi : 10.1109/ACC.2003.1244101.
- [29] R. A. DeCarlo, S. H. Zak, et G. P. Matthews, « Variable structure control of nonlinear multivariable systems : a tutorial », Proc. IEEE, vol. 76, no 3, p. 212-232, mars 1988, doi : 10.1109/5.4400.
- [30] C.FALLAHA, « Etude de la Commande par Mode de Glissement sur les Systèmes Mono et Multi Variables »,Ecole de Technologie Supérieure Université du QUEBEC, p. 109, 2007.
- [31] Y. Kanayama, Y. Kimura, F. Miyazaki, et T. Noguchi, « A stable tracking control method for an autonomous mobile robot », in Proceedings., IEEE International Conference

- on Robotics and Automation, Cincinnati, OH, USA, 1990, p. 384-389, doi : 10.1109/ROBOT.1990.126006.
- [32] J.-J. E. Slotine et W. Li, « Applied nonlinear control », Englewood Cliffs, N.J : Prentice Hall, P 259, 1991.
- [33] S.HAJRI, « Commande Dynamique par Mode Glissant Application à la Robustification des Processus Complexes », thèse Doctorat en Automatique et Informatique, Université des sciences et technologies de LILLE, p. 226, 1997.
- [34] R.TITRAOUI, « Commande par mode glissant d'un appareil à vol et atterrissage vertical ». GENIE ELECTRIQUE, Université Mohamed Boudiaf-m'sila, 100 p, 2016 .
- [35] W.Zheng et Y.Jia, « Leader-follower Formation Control of Mobile Robots with Sliding Mode » J.Robot., vol.4, n° 1, p.4, 2017.
- [36] L. Jaulin, « RobMOOC », p. 103. 2020
- [37] L. Lapierre, R. Zapata, et P. Lepinay, « Simultaneous Path Following and Obstacle Avoidance Control of a Unicycle-type Robot », in Proceedings 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, avr. 2007, p. 2617-2622, doi : 10.1109/ROBOT.2007.363860.
- [38] A.Gadsden and S.Habibi, « Target Tracking Using the Smooth Variable Structure Filter », Proceedings of the ASME 2009 Dynamic Systems and Control Conference DSCC2009, October 12-14, 2009, Hollywood, California, USA. DSCC2009-2632.
- [39] Habibi, S., Burton, R., and Chinniah, Y., 2002. "Estimation Using a New Variable Structure Filter". Proceedings of the American Control Conference, Anchorage, May 8-10, 2002.
- [40] Habibi SR. The smooth variable structure filter. In : IEEE Conference. Vol. 95. 2007. p. 1026-1059.
- [41] Habibi, S., and Burton, R., 2003. "The Variable Structure Filter". Journal of Dynamic Systems, Measurement, and Control, ASME, Vol. 125, pp. 287-293.
- [42] O. Stasse, « Robot Operating System Introduction », p. 88. 2016
- [43] « fr - ROS Wiki ». <http://wiki.ros.org/fr> (consulté le avril. 05, 2020).
- [44] A. Koubaa, « INTRODUCTION TO ROBOTICS (CS460) », Introd. Robot., p. 90.
- [45] Y.PYO et al, « ROS robot programming », ROBOTICS CO.Ltd, Seoul Korea, 487 p, 2017

- [46] « Installer Ubuntu 16.04— Lea Linux». [https ://lealinux.org/documentations/Installer_Ubuntu_16.04](https://lealinux.org/documentations/Installer_Ubuntu_16.04) (consulté le janvier. 18, 2020).
- [47] « T.fiorenzani» [https ://github.com/tizianofiorenzani/how_do_drones_work](https://github.com/tizianofiorenzani/how_do_drones_work) (consulté le mai. 12, 2020).
- [48] « Online ArUco markers generator» [https ://chev.me/arucogen](https://chev.me/arucogen) (consulté le mai. 13, 2020).



Insérer les paramètres de la matrice de calibration et distorsion de Pi caméra dans le fichier `detection_pose_Leader.py` (Raspberry pi 3 B+)

2

Schéma global de l'implémentation et la réalisation pratique d'un système multi robots (Leader-Suiveur) et l'utilisation d'une caméra pour la localisation d'un robot leader

Insérer Les coefficients de PID dans le contrôleur de la vitesse de roues de Suiveur (Arduino UNO C++)

4

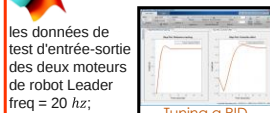
Ajustage des Paramètres des Régulateurs PID pour Suiveur Leader

MATLAB



Les coefficients de PID: [kp = 0.001725; ki = 0.05948; kd = 0.0]

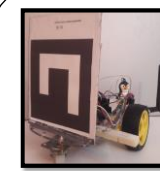
MATLAB



Les coefficients de PID: [kp = 0.35; ki = 2.21; kd = 0.0]

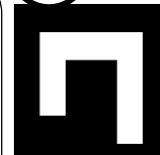
Insérer Les coefficients de PID dans le contrôleur de la vitesse de roues de Leader (Arduino UNO C++)

5



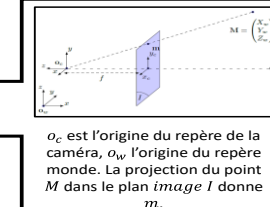
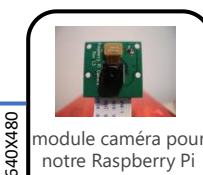
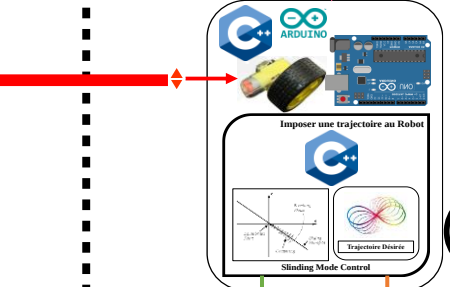
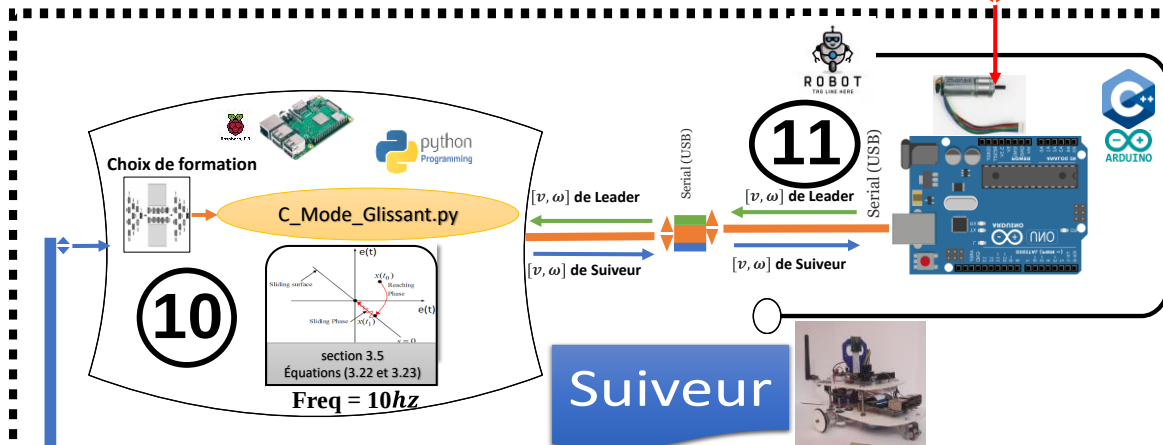
Robot (Leader) unicycle avec un TAG ArUco en arrière

6



TAG ArUco

Leader



Légende

- Étapes exécutée 1 fois
- Étapes de Suiveur
- Étapes de Leader
- Liaisons ISP de Arduino
- Liaisons Radio nRF24I01
- Liaison visuelle Tag/Leader

Configuration de l'environnement : Étape exécutée 1 fois

`mkdir -p ~/catkin_ws/src` → `~/catkin_ws$ catkin_make`

`~/catkin_ws/src$ catkin_create_pkg src_bot3 gazebo_ros geometry_msgs roscpp rospy std_msgs tf sensor_msgs`

ROS

Simulation Gazebo d'un système multi robots Leader/Suiveur

