

# DÉDICACE

À ma famille et à mes amis,

Je souhaite prendre un moment pour vous exprimer ma profonde gratitude et ma reconnaissance pour tout le soutien inébranlable que vous m'avez apporté tout au long de mon stage. Votre présence et votre encouragement ont été essentiels pour moi, et je me sens incroyablement chanceux de vous avoir dans ma vie.

À mes parents bien-aimés, je suis éternellement reconnaissant pour tout ce que vous avez fait pour moi. Votre amour inconditionnel, votre soutien constant et vos encouragements sans faille ont été les fondements solides sur lesquels j'ai pu m'appuyer tout au long de cette période exigeante. Votre présence réconfortante et vos sages conseils m'ont donné la force nécessaire pour affronter tous les défis et les obstacles qui se sont dressés sur ma route. Sans vous, je ne serais pas là où je suis aujourd'hui, et je vous serai toujours redevable pour cela.

À ma famille élargie et à mes amis proches, vous avez été des piliers incroyables dans ma vie. Votre soutien constant, vos encouragements sincères et vos mots réconfortants ont illuminé mes journées les plus sombres. Chacun de vous a contribué à ma croissance personnelle et professionnelle, et je suis honoré de vous compter parmi les personnes qui ont cru en moi. Votre présence a rendu ce parcours enrichissant et mémorable, et je suis reconnaissant pour chaque moment partagé à mes côtés.

Aujourd'hui, je me tiens devant vous, fier des accomplissements que j'ai réalisés pendant cette période de stage, et je veux que vous sachiez que c'est grâce à votre soutien indéfectible. Votre présence a été le moteur qui m'a propulsé vers la réussite, et je ne saurais jamais assez vous remercier pour cela.

Que ce soit par ces mots simples mais sincères, je veux vous exprimer toute ma reconnaissance et mon amour pour chacun d'entre vous. Ma famille et mes amis, vous êtes les véritables trésors de ma vie, et je suis honoré de vous avoir à mes côtés. Votre soutien inébranlable a fait de mon stage une expérience extraordinaire, et je vous serai éternellement reconnaissant.

Avec tout mon amour et ma gratitude,

HELAOUI Amine

## Remerciements

Je tiens à exprimer mes sincères remerciements à M. David Fofi, le responsable de l'équipe VIBOT au laboratoire ImViA, pour avoir accepté de m'accueillir pour mon stage au sein du plateau robotique. Sa confiance en moi et son soutien constant ont été essentiels pour la réussite de ce projet. Je tiens également à le remercier pour son accueil chaleureux et ses précieux conseils tout au long de mon stage.

Je souhaite également exprimer ma gratitude envers mes deux encadrants, M. Amine Abadi et M. Eltaief Maher, pour leur temps, leurs efforts et leur contribution précieuse à ce rapport et à ce projet. Leur expertise et leurs orientations m'ont été d'une aide précieuse et ont grandement contribué à la réussite de ce travail.

Je tiens également à remercier l'École Internationale Privée Polytechnique de Sousse (EPI Polytechnique) et tout le corps professoral pour les trois années que j'ai passées au sein de l'école. Leur enseignement de qualité et leur encadrement ont été essentiels pour mon développement académique et professionnel.

Enfin, j'aimerais exprimer ma reconnaissance envers ma famille et mes amis pour leur soutien inconditionnel tout au long de mon parcours universitaire. Leur encouragement et leur soutien moral ont été d'une importance capitale pour moi.

Mes plus sincères remerciements vont à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet. Leur soutien et leur collaboration ont été d'une valeur inestimable et ont contribué à la réussite de ce travail.

Merci encore à tous pour votre aide et votre soutien précieux.

# Table des matières

<b>Liste des figures .....</b>	<b>vi</b>
<b>Introduction générale .....</b>	<b>1</b>
<b>Chapitre 1 : Cadre de projet .....</b>	<b>3</b>
I. Introduction .....	3
II. Présentation d'organismes d'accueil .....	3
II.1. Présentation .....	3
II.2. Organisation .....	3
II.3. Localisation .....	6
II.4. Plateau robotique site le creusot .....	6
III. Problématique .....	8
IV. Conclusion .....	10
<b>Chapitre 2 : Généralités sur les robots mobiles et les commandes .....</b>	<b>11</b>
I. Introduction .....	11
II. Définition de robot .....	11
III. Historique des robots .....	11
IV. Classification selon l'autonomie .....	15
V. Classification selon le type de locomotion de robots mobiles à roue .....	15
V.1. Robot unicycle .....	16
V.2. Robot tricycle .....	16
V.3. Robot voiture .....	17
V.4. Robot omnidirectionnel .....	17
VI. Application de robot mobile .....	18
VII. Généralités sur les systèmes linéaires et non linéaires .....	19
VIII. Les commandes linéaires .....	20

IX.	Les commandes non linéaires .....	21
IX.1.	Commande par platitude .....	21
IX.2.	Commande backstepping .....	22
IX.3.	La commande linéarisante .....	23
IX.4.	Commande par Lyapunov .....	24
X.	Les commandes robustes .....	24
X.1.	La commande par mode glissant .....	25
X.2.	Commande H-infini .....	26
X.2.1.	Norme $H_\infty$ d'un système linéaire .....	26
X.3.	Commande adaptative .....	27
X.4.	La commande de rejet de perturbation .....	28
X.5.	Commande optimale .....	29
XI.	Conclusion .....	32
<b>Chapitre 3 : commande non linéaire par platitude .....</b>		<b>33</b>
I.	Introduction .....	33
II.	Notion du système plat .....	33
II.1.	Définition de la platitude dans le cadre de l'algèbre différentielle .....	33
II.2.	Algorithme de vérification de sorties plates .....	34
II.3.	Platitude et linéarisation .....	35
II.4.	Platitude et planification de trajectoires .....	35
II.5.	Platitude et suivi de trajectoire : .....	37
III.	La stratégie de contrôle basée sur la commande platitude .....	38
III.1.	Modèle cinématique sans glissement .....	38
III.2.	Vérification de sorties plates .....	40
III.3.	Simulation Matlab pour une trajectoire circulaire .....	41
III.4.	Générer une trajectoire avec courbe de Bézier .....	45

III.5. Courbe de Bézier .....	47
III.6. Simulation Matlab pour une trajectoire générée par la courbe de Bézier .....	47
IV. Interprétation générale.....	51
V. Simulation de deux trajectoires en présence de perturbation .....	51
VI. Conclusion .....	52
<b>Chapitre 4 : Commande non-linéaire robuste .....</b>	<b>53</b>
I. Introduction .....	53
II. Définition de commande par mode glissant.....	53
III. Conception de la commande par mode glissant.....	54
III.1. Choix de la surface de glissement.....	55
III.2. Conditions de convergence et d'existence .....	55
III.3. Phénomène de broutement .....	59
IV. Application de la commande non-linéaire robuste sur Turtlebot3.....	60
IV.1. Le modèle cinématique avec glissement.....	62
IV.2. Simulation de la commande non-linéaire robuste sur ros et MATLAB....	66
V. Conclusion .....	75
<b>Conclusion générale.....</b>	<b>76</b>
<b>Référence bibliographique .....</b>	<b>78</b>

## Liste des figures

Figure 1 : logo du labo ImViA .....	3
Figure 2 : L'organisation de laboratoire ImViA.....	4
Figure 3 : Formation par la recherche.....	5
Figure 4 : Plateau robotique .....	6
Figure 5 : Turtlebot3 .....	7
Figure 6 : Robot chien .....	7
Figure 7 : Chaine de bras manipulateur.....	7
Figure 8 : Robot Arlock.....	7
<i>Figure 9 : Drone Parrot.....</i>	<i>7</i>
<i>Figure 10 : Robot d'Exploration de fourré .....</i>	<i>7</i>
Figure 11 : Trajectoire non-linéaire .....	8
Figure 12 : Variations du terrain impactant la navigation des robots mobiles.....	9
Figure 13 : le premier robot mobile autonome.....	12
Figure 14 : Unimate .....	13
Figure 15 : Puma .....	13
Figure 16 : Shakey.....	13
Figure 17 : Stanford carte .....	13
Figure 18 : HilAR .....	13
Figure 19 : Robocup.....	14
Figure 20 : Sojourner.....	14
Figure 21 : Aibo .....	14
Figure 22 : Spirit .....	14
Figure 23 : Curiosity .....	14
Figure 24 : Asimo .....	15

Figure 25 : Robot Nao.....	15
Figure 26 : Robot de type unicycle .....	16
Figure 27 : Robot de type tricycle .....	16
Figure 28 : Robot de type voiture.....	17
Figure 29 : Robot de type omnidirectionnel.....	17
Figure 30 : le diagramme bloc de commande PID .....	21
Figure 31 : Schéma bloc de commande platitude .....	22
Figure 32 : Schéma bloc de la commande par backstepping .....	23
Figure 33 : Différents modes pour la trajectoire dans le plan de phase .....	25
Figure 34 : Principe de système de commande adaptative .....	27
Figure 35 : Structure classique de la commande par rejet de perturbation .....	29
Figure 36 : Le diagramme bloc de commande LQR.....	30
Figure 37 : Philosophie de la commande prédictive .....	31
Figure 38 : Génération de trajectoires dans l'espace des sorties plates.....	36
Figure 39 : Modèle de robot mobile à deux roues sans glissement .....	38
Figure 40 : Diagramme de flux processus commande par platitude.....	41
Figure 41 : Trajectoire désirée $f_1$ et trajectoire de robot sur $x$ .....	42
Figure 42 : Trajectoire désirée $f_2$ et trajectoire de robot sur $Y$ .....	42
Figure 43 : Accélération linéaire de robot.....	43
Figure 44 : Vitesse linéaire de robot .....	43
Figure 45 : Vitesse angulaire .....	44
Figure 46 : Trajectoire de robot et trajectoire désirée .....	45
Figure 47 : Polynôme de Bernstein constant (degré 0).....	46
Figure 48 : Polynôme de Bernstein Linéaire (degré 1).....	46
Figure 49 : Polynôme de Bernstein quadratique (degré 2).....	46
Figure 50 : Polynôme de Bernstein cubique (degré 3).....	46

Figure 51 : Les courbes de Bézier de différents degrés .....	47
Figure 52 : Trajectoire désirée et trajectoire de robot sur l'axe x .....	48
Figure 53 : Trajectoire désirée et trajectoire de robot sur l'axe y .....	48
Figure 54 : Erreur sur l'axe x.....	49
Figure 55 : Erreur sur l'axe y.....	49
Figure 56 : Vitesse linéaire de robot .....	50
Figure 57 : Trajectoire désirée et trajectoire de robot .....	50
Figure 58 : Trajectoire désirée et trajectoire de robot avec perturbation .....	51
Figure 59 : Trajectoire de robot2D avec perturbation .....	52
Figure 60 : Différents modes pour la trajectoire dans le plan de phase .....	54
Figure 61 : Convergence du système glissant.....	56
Figure 62 : Représentation de la commande discontinue.....	59
Figure 63 : Représentation de la fonction Sat .....	60
Figure 64 : Le dimensionnement du TurtleBot .....	61
Figure 65 : Les composants principaux du TurtleBot3 .....	62
Figure 66 : Robot mobile à deux roues soumis à des incertitudes .....	63
Figure 67 : Le Phénomène de broutement.....	65
Figure 68 : Fonction de saturation.....	66
Figure 69 : Différents types de messages dans ROS .....	68
Figure 70 : Exemple du type de message Topic .....	69
Figure 71 : Exemple du type de message Service.....	69
Figure 72 : Exemple du type de message Action .....	70
Figure 73 : diagramme de flux processus commande par platitude basé sur mode glissant.....	70
Figure 74 : Simulation de la trajectoire circulaire sur ros .....	71
Figure 75 : Simulation réelle avec perturbation de vent.....	71
Figure 76 : Simulation réelle avec perturbation de vent.....	72



Figure 77 : Trajectoire sur l'axe x .....	72
Figure 78 : Trajectoire sur l'axe y .....	73
Figure 79 : Vitesse linéaire de robot .....	73
Figure 80 : Vitesse angulaire de robot .....	73
Figure 81 : Accélération de robot .....	74
Figure 82 : Trajectoire de robot et trajectoire désirée .....	74

# Introduction générale

Les robots mobiles peuvent être utilisés dans une grande variété d'applications. Parmi les applications les plus connues, on trouve la neutralisation de bombes, l'exploration spatiale et les activités sous-marines (par exemple, l'exploration pétrolière et le remplacement des plongeurs dans les tâches dangereuses). De plus, il existe des applications de véhicules autonomes guidés pour réduire les embouteillages, la pollution et les accidents causés par l'incapacité humaine

Le développement du robot mobile connaît un essor croissant depuis quelques années grâce à la possibilité d'embarquer des cartes de commande performantes et rapides capables d'exécuter une masse de calcul considérable. La robotique répond souvent à des contraintes bien connues du monde agricole, telles que la réduction de la pénibilité de certaines tâches (traite, désherbage, taille, récolte, etc.) et l'optimisation du temps de travail pour se consacrer à d'autres fonctions comme la commercialisation et profiter de plus de temps libre. Cependant, l'importance croissante des robots dans différents domaines et la complexité des modèles robotiques nécessitent une attention particulière lors du développement des algorithmes de navigation et de guidage pour les robots.

Au cours de la dernière décennie, de nombreuses techniques de commande non linéaire ont été largement utilisées pour le suivi de trajectoires des robots mobiles, telles que la commande par platitude, la linéarisation par rétroaction et le contrôle par backstepping, etc.

Cependant, bien que les contrôleurs de poursuite offrent des performances satisfaisantes dans des scénarios idéaux, ils présentent des limitations lors de leur mise en pratique. En général, les méthodes de commande non linéaire permettent un suivi asymptotique de la trajectoire souhaitée uniquement lorsque les paramètres incertains et les perturbations sont considérés comme négligeables dans le modèle, et lorsque les états de rétroaction sont mesurés avec précision

Ces conditions sont difficiles à satisfaire dans la réalité. Par conséquent, il est nécessaire de développer un contrôleur de poursuite non linéaire robuste qui compense les effets des perturbations pour améliorer les performances du robot mobile.

Ce rapport sera divisé en 4 chapitres comme suite :

Le premier chapitre de ce projet a été consacré pour une présentation sur le laboratoire ImViA, l'organisation et aussi les activités de recherche de différente équipe, d'autre partie non avons présentons le problématique de notre Project.

Le deuxième chapitre s'est concentré sur une généralité sur les robots mobiles, mettant en évidence les avancées récentes dans le domaine du contrôle non linéaire et robuste. Nous avons examiné les différents types de robots mobiles, leurs applications, en outre, nous avons étudié en les méthodes de commande non linéaire, les commandes robustes et aussi les commandes optimales en mettant l'accent sur leur capacité à gérer des systèmes dynamiques complexes.

Le troisième chapitre examiné la commande non linéaire par platitude, une approche prometteuse pour la poursuite précise d'une trajectoire prédéfinie. Nous avons exploré les concepts fondamentaux de la platitude, mettant en évidence son utilité dans la modélisation et le contrôle des robots mobiles. Nous avons également réalisé des simulations sur MATLAB pour évaluer les performances de cette méthode de commande dans des scénarios spécifiques.

Le quatrième chapitre a abordé la commande non linéaire robuste, en se focalisant sur la première partie consacrée à la commande par mode glissant. Nous avons étudié en détail les principes de cette approche, qui permettent de garantir une stabilité et une robustesse accrues du système de contrôle. Enfin, nous avons examiné la généralité de la plateforme ROS (Robot Operating System) et du robot mobile TurtleBot, en utilisant ces outils pour réaliser des simulations réalistes de notre contrôleur.

Après avoir analysé en détail les différentes facettes de notre sujet d'étude, il est évident que la rédaction de notre rapport doit être clôturée par une conclusion générale et des perspectives éclairantes

# Chapitre 1 : Cadre de projet

## I. Introduction

Ce chapitre présente le laboratoire ImViA, spécialisé en imagerie, vision et intelligence artificielle. Nous explorerons son organisation interne, les différentes équipes de recherche et leurs activités. Il s'agit d'une introduction à l'environnement dynamique du laboratoire et présente la problématique de notre projet.

## II. Présentation d'organismes d'accueil

### II.1. Présentation

Le laboratoire Imagerie et Vision Artificielle (ImViA) est une entité de recherche qui a évolué à partir du Laboratoire Electronique, Informatique et Image (Le2i) affilié à l'Université de Bourgogne de 1996 à 2018, aux Arts et Métiers ParisTech de 2014 à 2017 et au CNRS de 2001 à 2017. Le laboratoire ImViA se concentre sur la thématique de la Vision Artificielle et regroupe environ cinquante enseignants-chercheurs.



*Figure 1 : logo du labo ImViA*

### II.2. Organisation

Il est structuré en trois équipes distinctes comme représente la figure 2, à savoir CORES, IFTIM et VIBOT.

#### **L'équipe CORES :**

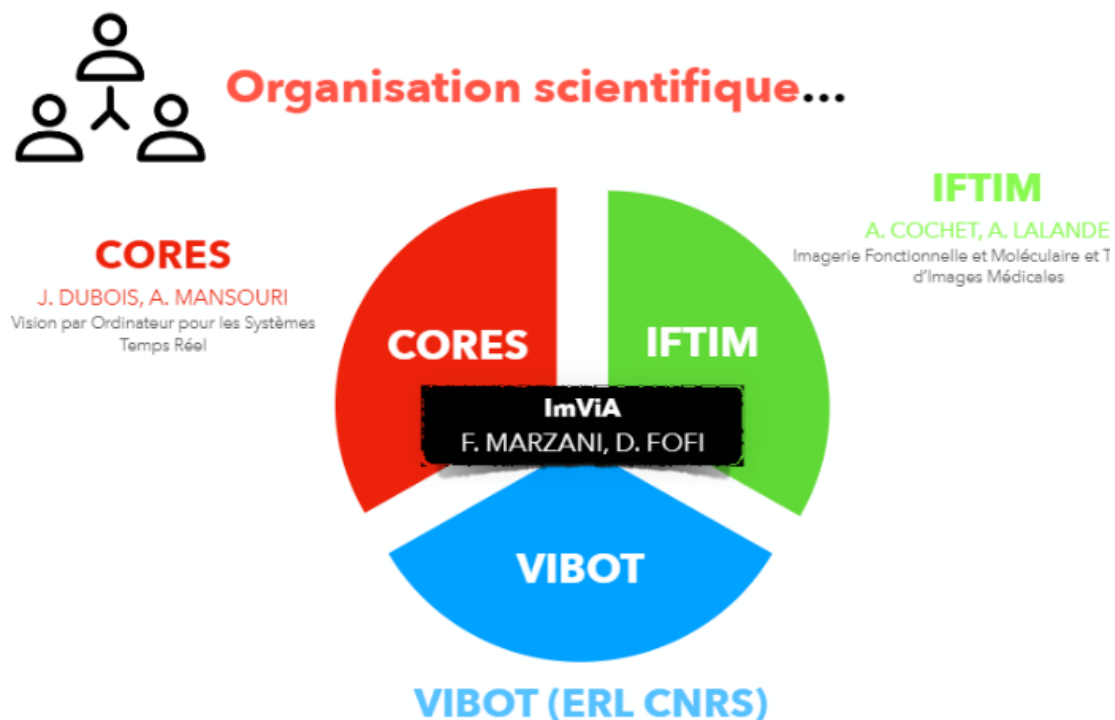
(COgnitive REasoning Systems) se spécialise dans les domaines de la perception visuelle, de la modélisation cognitive et des systèmes de raisonnement. Les recherches menées au sein de cette équipe visent à développer des méthodes avancées de traitement d'images et de vidéos, ainsi que des modèles cognitifs pour l'analyse et la compréhension des scènes visuelles.

### L'équipe IFTIM :

(Ingénierie de la Formation et Technologies de l'Information et de la Mesure) se concentre sur les applications de la Vision Artificielle dans le domaine de l'ingénierie de la formation. Les recherches effectuées au sein de cette équipe portent sur le développement de technologies et d'outils innovants pour l'analyse de mouvement, la réalité virtuelle et augmentée, ainsi que la mesure et l'évaluation des compétences.

### L'équipe VIBOT :

Les activités de l'équipe VIBOT sont dédiées à la vision pour la robotique et elle entend répondre aux questions traditionnellement posées en robotique mobile, telles que la localisation, la cartographie, l'estimation de trajectoire, la reconnaissance d'objets et la détection d'obstacles, en étendant leur champ d'application aux environnements complexes (géométriquement, cinématiquement, photométriquement ou radio métriquement), pour atteindre ces objectifs, l'équipe VIBOT est structurée autour de deux thèmes principaux de recherche : l'imagerie non conventionnelle pour la robotique et la reconstruction et l'analyse de scènes .

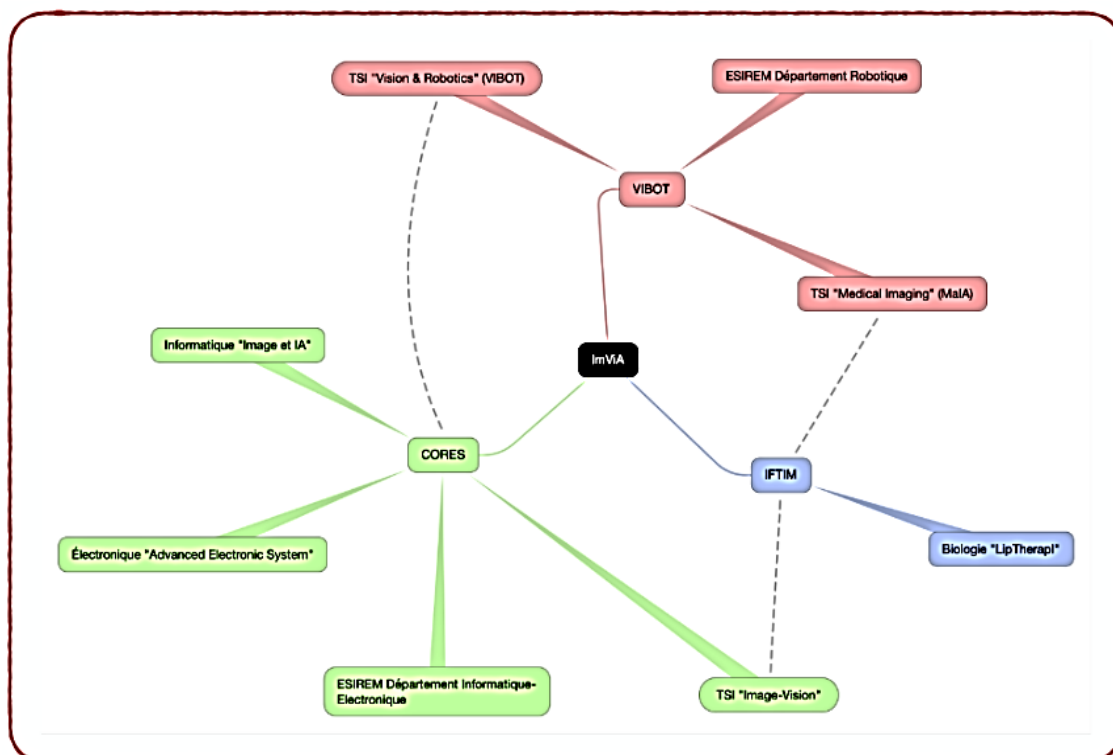


*Figure 2 : L'organisation de laboratoire ImViA*

La complémentarité des trois équipes et leur convergence mathématique permettent de créer une unité de recherche solide, basée sur des fondements scientifiques communs.

Ces équipes ont marqué leur activité de recherche grâce à des réalisations significatives, telles que :

- L'obtention de labellisations de projets de recherche de renom tels que ANR, H2020, PHC, FUI, COST, démontrant ainsi leur rayonnement tant au niveau national qu'international.
- Un taux élevé de publication d'articles de qualité, témoignant de la contribution des équipes à la production scientifique dans le domaine de la Vision Artificielle.
- L'équipement IMAPPI (Imagerie Médicale), une infrastructure technologique de pointe dédiée à l'imagerie médicale.
- Les masters Erasmus Mundus en Vision pour la Robotique (VIBOT) et en Imagerie Médicale (MaIA), offrant une formation de haut niveau dans ces domaines.
- La plateforme Vision 2D/2D+/3D de Dijon, qui constitue un environnement de recherche et de développement avancé pour les études en vision par ordinateur.
- Le plateau Robotique du Creusot, un espace dédié à la réalisation d'expérimentations et de tests en robotique.



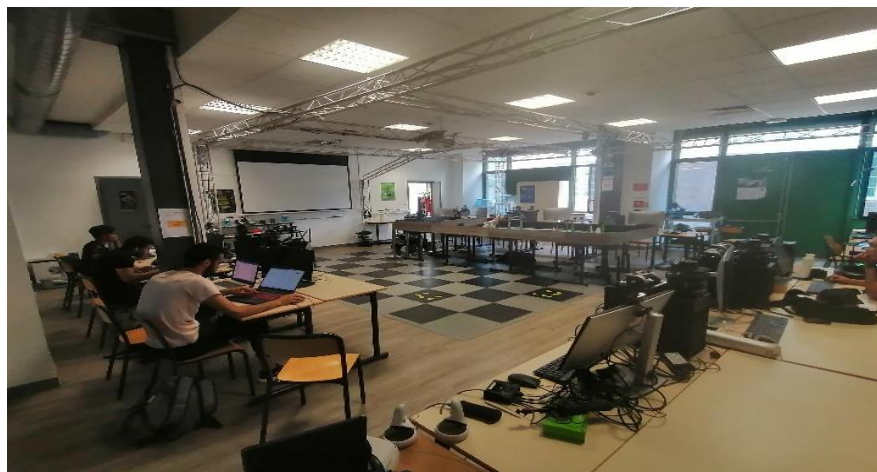
*Figure 3 : Formation par la recherche*

### II.3. Localisation

Le laboratoire ImViA est présent sur les sites de Dijon et du Creusot, permettant une concentration géographique des axes de recherche. Les trois équipes du laboratoire offrent une complémentarité et une spécificité thématique, avec une majorité des activités à Dijon et une troisième équipe basée principalement au Creusot. Cette répartition favorise la collaboration, l'expertise spécialisée et l'utilisation optimale des ressources et infrastructures disponibles sur chaque site.

### II.4. Plateau robotique site le creusot

Le plateau robotique du Creusot (voir la figure 4) est une infrastructure majeure du laboratoire ImViA. Situé sur le site du Creusot, il offre un environnement propice à la recherche et au développement dans le domaine de la robotique. Ce plateau est équipé de ressources technologiques avancées, telles que des robots industriels, des systèmes de vision, des capteurs et des interfaces de contrôle. Il permet aux chercheurs et aux étudiants d'explorer et de tester des applications robotiques dans des conditions réelles. Grâce à cette plateforme, le laboratoire ImViA peut réaliser des expérimentations, des validations et des démonstrations concrètes de ses travaux de recherche, contribuant ainsi à l'avancement des connaissances en robotique et à la formation des futurs experts dans ce domaine.



*Figure 4 : Plateau robotique*

Le plateau robotique du Creusot dispose d'une variété de robots pour soutenir les activités de recherche et de développement. Parmi les robots disponibles, on retrouve les drones AR Drone 2.0 et Tello Drone, qui permettent d'explorer les applications aériennes comme représente les figures (5 ,6,7,8 ,9,10) les différents robots utilisés dans le laboratoire. Les TurtleBot 3 Burger sont des robots mobiles utilisés pour la navigation et

la cartographie de l'environnement. Les Robot Ranger, quant à eux, offrent des fonctionnalités polyvalentes pour la recherche en robotique. Enfin, le plateau comprend également des bras manipulateurs et des robots chiens, qui sont des plateformes d'étude pour la manipulation d'objets et l'interaction sociale voir les figures suivantes. Ces différents robots offrent aux chercheurs et aux étudiants la possibilité d'explorer une large gamme d'applications et de développer des compétences en robotique dans un environnement pratique et innovant.



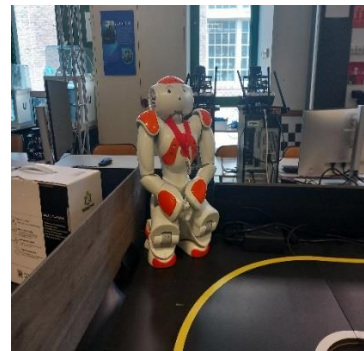
*Figure 5 : Turtlebot3*



*Figure 6 : Robot chien*



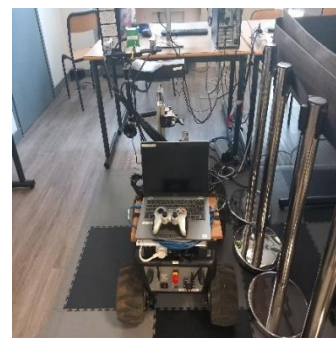
*Figure 7 : Chaine de bras  
manipulateur*



*Figure 8 : Robot Arlock*



*Figure 9 : Drone Parrot*

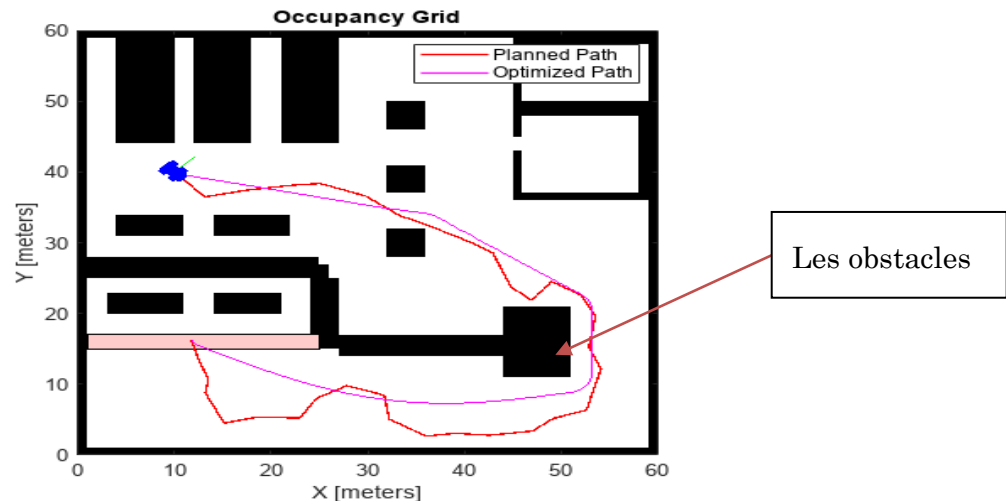


*Figure 10 : Robot d'Exploration de fourré*



### III. Problématique

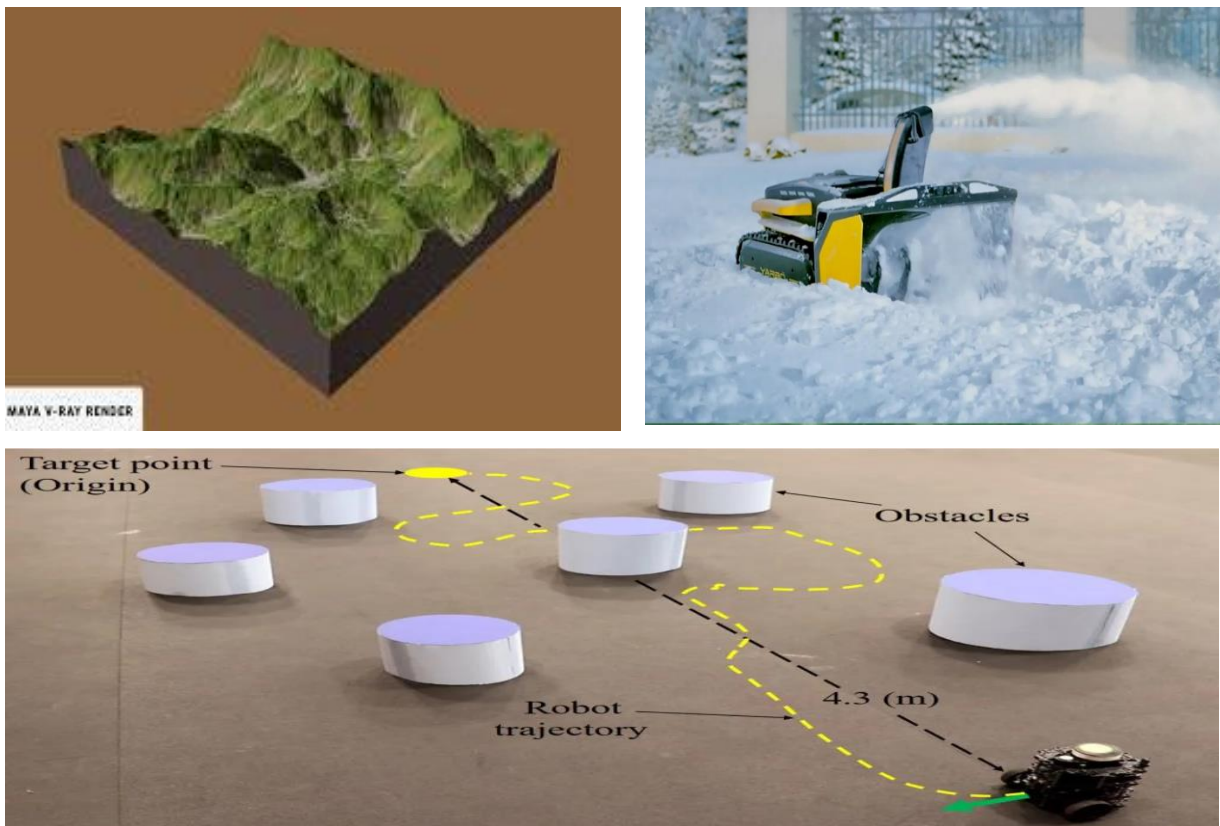
La figure (11) montre une trajectoire optimisée (optimized path) où le robot tente de la suivre c'est la trajectoire planifier par le robot (planned path), mais il y a plusieurs moments où il dévie de la trajectoire prévue .



*Figure 11 : Trajectoire non-linéaire*

Comment développer une commande non linéaire robuste pour un robot mobile afin d'améliorer sa stabilité et sa manœuvrabilité en présence de perturbations externes ? Cette problématique consiste à développer des commandes capables de suivre des trajectoires de référence malgré l'existence de perturbations externes comme (voire la figure 12) :

- Les obstacles tels que des rochers, des arbres, des murs ou des objets divers peuvent entraver la progression d'un robot mobile, La capacité à détecter et à éviter ces obstacles est essentielle pour naviguer avec succès dans un environnement encombré ;
- Les variations du terrain comme les collines, les montagnes, les terrains rocheux, les sols inégaux, la glace, le verglas les sols mouillés peuvent présenter des défis majeurs pour les robots mobiles. Ils doivent être capables de gérer des pentes, des changements de hauteur et des surfaces irrégulières pour maintenir leur stabilité et leur mobilité ;
- Les conditions environnementales changeantes, Les variations des conditions météorologiques telles que la pluie, la neige, le vent fort ou les tempêtes peuvent rendre le terrain plus difficile pour un robot mobile. Ils peuvent affecter la visibilité, la traction et la stabilité du robot ;



**Figure 12 : Variations du terrain impactant la navigation des robots mobiles**

Le développement d'une telle commande implique la mise en œuvre de différentes techniques de contrôle non linéaire, adaptées aux caractéristiques spécifiques du robot mobile.

Parmi les techniques de commande non linéaire qui peuvent être explorées, on retrouve la commande par platitude, la commande backstepping, la commande linéarisant, la commande par Lyapunov, ainsi que des commandes robustes comme la commande à glissement de mode, la commande  $H^\infty$ , la commande adaptative et la commande par rejet de perturbations.

Le développement d'une commande non linéaire robuste pour un robot mobile nécessite une sélection judicieuse des techniques de commande appropriées, une évaluation comparative de leurs performances et une mise en œuvre efficace à l'aide de simulations et de tests réels. Cela permettra d'améliorer la stabilité, la manœuvrabilité et la résistance du robot mobile aux perturbations externes, ouvrant ainsi de nouvelles perspectives d'application dans des environnements réels et exigeants.

## IV. Conclusion

Dans ce chapitre, on a présenté le laboratoire ImViA , est spécialisé en imagerie, vision et intelligence artificielle. Nous avons présenté aussi les différentes équipes de recherche et leurs activités. Une présentation des différents robots existants dans le plateau robotique et la problématique de notre projet, qui s'articule sur la commande de ces robots, fait l'objet de la dernière partie de ce chapitre.

# Chapitre 2 : Généralités sur les robots mobiles et les commandes

## I. Introduction

Ce chapitre présente les notions fondamentales des robots mobiles et une généralité sur les commandes : linéaire, non-linéaire, optimale et robuste. Les robots mobiles sont des systèmes capables de se déplacer de manière autonome dans leur environnement. Les commandes permettent de contrôler leur comportement et leurs mouvements.

## II. Définition de robot

Etymologiquement, le mot robot vient du tchèque (robota) qui signifie travail ; il désigne un système automatique asservi à une unité de commande informatisée. Un robot est un ensemble de mobiles associés dont les mouvements sont commandés numériquement et synchronisés. Chaque mouvement élémentaire est un axe de commande numérique. Cependant un robot n'est qu'une machine programmable qui ne fait qu'exécuter ce que l'homme lui a appris.

## III. Historique des robots

Le concept de robots mobiles autonomes a émergé à la fin des années 1960 à partir de deux sources assez différentes : premièrement, les recherches menées par le Stanford Research Institute sur la possibilité de doter les machines de la capacité de raisonner et de réagir logiquement aux événements extérieurs. Nous avons donc construit le Shakey, une machine à roues qui se connecte à un ordinateur et possède une caméra qui lui permet de prendre des images de son environnement. Elle évolue dans un univers de cubes et de pyramides de différentes tailles et couleurs. Le travail de Shakey consiste à ramasser un objet et à le transporter ailleurs, quel que soit l'endroit ; chaque mission dure près de cinquante minutes voire figure 16.

D'autre part, l'industrie nucléaire a besoin de machines capables de fonctionner à distance dans des environnements encombrés et inaccessibles à l'homme. General Electric a ensuite développé des quadrupèdes pour tenter de résoudre ce problème, tandis que des revers lunaires et martiens ont été construits pour explorer des planètes sans intervention humaine.

Laboratoires, industriels, informaticiens et mécaniciens continueront de travailler en parallèle pour les années à venir.

Du côté industriel, on en tire de la télé-opération et des éléments de la robotique classique, et du côté informatique, on assiste à d'énormes avancées dans le domaine de l'intelligence artificielle. Ainsi, à la fin des années 1970, trois grands pôles géographiques (France, Japon, États-Unis) se sont imposés. La synthèse de tous les travaux réalisés auparavant a abouti aux robots mobiles autonomes (allant des robots domestiques aux robots militaires) [1].

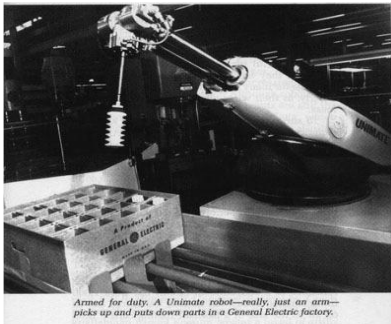
C'est dans cet environnement de plus en plus automatisé que se fait sentir le besoin d'outils capables non seulement d'effectuer des tâches répétitives, même celles que l'homme ne peut pas (porter de lourdes charges, découpe ultra-précise, etc.), mais aussi de démontrer une certaine autonomie de mouvement dans les environnements. Hostile à l'homme. On voit maintenant son application sur les chantiers de construction, comme le désamiantage des bâtiments, la décontamination radioactive, les expérimentations en milieux dangereux, etc.

- XVIIIe siècle : création des automates. En 1738, Jacques Vaucanson conçoit un canard en cuivre articulé qui pouvait boire, manger, bavarder et digérer comme un vrai animal.
- 1921: Karel Capek, RUR (Rossum's Universal Robots).
- 1940 : Isaac Asimov rédige une série d'histoires au sujet des robots. Trois lois de la robotique.
- En 1948, Grey Walter invente le premier robot mobile autonome, une tortue se déplaçant vers les sources lumineuses qu'elle perçoit (Figure13). Par contre, le robot n'est pas programmable. L'introduction des robots était seulement possible dans les années 50 avec la création de transistors et de circuits intégrés

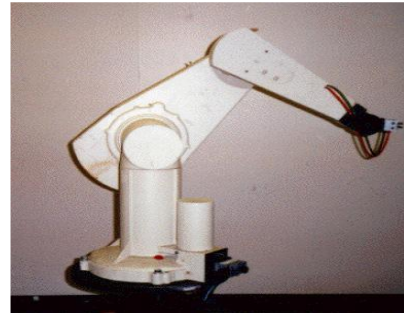


*Figure 13 : le premier robot mobile autonome*

- 1961 : Premier robot industriel mis en place dans une usine de General Motors UNIMATE (tubes cathodiques de télévision). (Figure 14)
- 1972 : Nissan ouvre la première chaîne de production complètement robotisée.
- 1978 : PUMA (Programmable Universal Machine for Assembly) développé par General Motors. (Figure15)



*Figure 14 : Unimate*



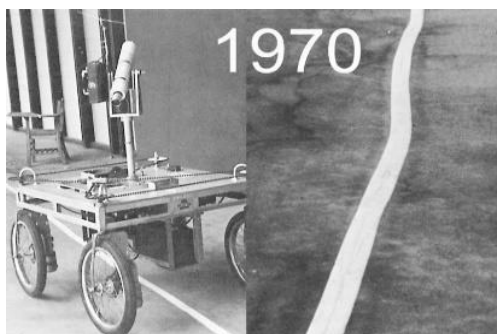
*Figure 15 : Puma*

- 1960-64 : Ouverture des laboratoires d'Intelligence Artificielle au M.I.T., Stanford Research Institute (SRI), Université de Stanford, Université d'Edinburgh.
- Fin des années 1960 : Mise en place de « Shakey » premier robot mobile intégrant perception, planification et exécution. (Figure 16)



*Figure 16 : Shakey*

- 1970 : Stanford Carte. (Figure 17)
- 1977 : premier robot mobile français HILARE au LAAS (CNR Toulouse). (Figure18)



*Figure 17 : Stanford carte*



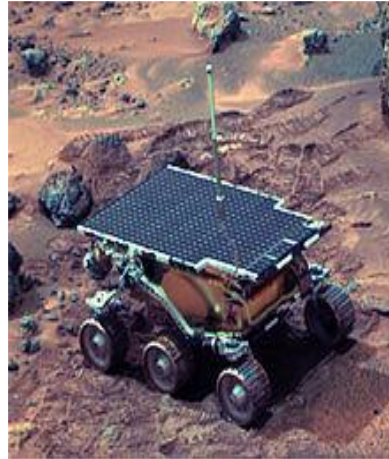
*Figure 18 : HilAR*



- 1992 : Mise en place de la compétition annuelle AAAI sur la robotique mobile.
- 1995 : Mise en place de la Robocup. (Figure 19)
- 1997 : premier robot mobile extra planétaire sur Mars. (Figure 20)
- 1999 : Lancement de Aibo. (Figure21)



*Figure 19 : Robocup*

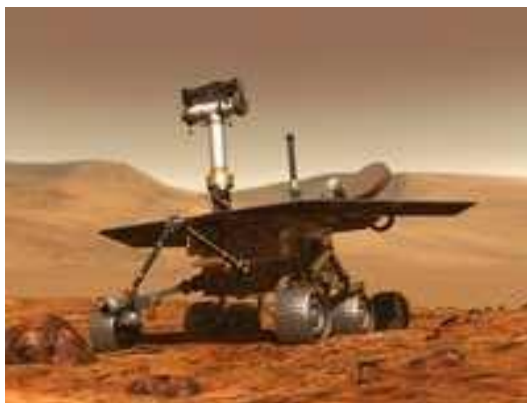


*Figure 20 : Sojourner*



*Figure 21 : Aibo*

- 2003 : Projet « Mars Exploration Rover » (Spirit & Opportunity). (Figure22)
- 2009 : projet « Mars Science Laboratoire » succédant au projet Rover, envoi prévu de Curiosity fin 2011. (Figur23)



*Figure 22 : Spirit*

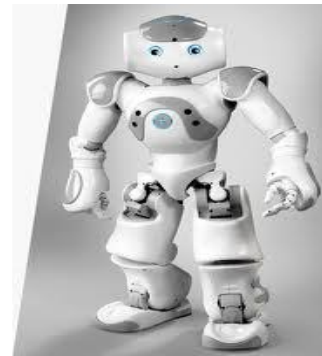


*Figure 23 : Curiosity*

- 2000 : Lancement d'Asimo. (Figure 24)
- 2006 : le projet Aibo n'est plus assez rentable, fin de la production.
- 2009 : robot Nao initialement développé par la société Aldebaran Robotiques, une start-up Française située à Paris. (Figure 25)



*Figure 24 : Asimo*



*Figure 25 : Robot Nao*

#### IV. Classification selon l'autonomie

Une classification est proposée dans la littérature qui définit le degré d'autonomie du robot mobile.

- Véhicule télécommandé par un opérateur qui lui impose chaque tâche élémentaire réalisée.
- Véhicule télécommandé au sens de la tâche à réaliser. Le véhicule contrôle automatiquement ses actions.
- Véhicule semi-autonome réalisant sans l'aide de l'opérateur des tâches prédéfinies.

Véhicule autonome qui réalise des tâches semi-définies. Ce type de véhicule pose des problèmes d'un niveau de complexité élevé de représentation des connaissances, de capacité décisionnelle et de génération de plans qui sont résolus à bord dans la mesure du possible.

L'ensemble des problèmes particuliers liés à la conception de tels robots sont :

- La conception mécanique liée à la mobilité.
- La détermination de la position et de la latitude (orientation).
- La détermination du chemin optimal pour atteindre le lieu de la tâche.

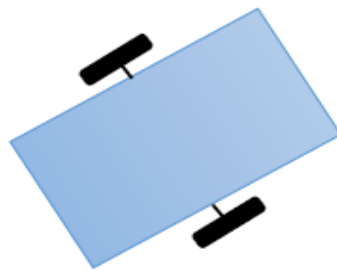
#### V. Classification selon le type de locomotion de robots mobiles à roue

La mobilité par roues est la structure mécanique la plus utilisée. Ce type de robot assure un déplacement avec une accélération et une vitesse rapide mais nécessite un sol relativement plat. On distingue plusieurs classes de robots à roues déterminées principalement par la position et le nombre de roues utilisées. Nous citerons ici les quatre classes principales de robots à roues [2].



### V.1. Robot unicycle

Un robot de type uni cycle est actionné par deux roues indépendantes (figure 26), il possède des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. C'est un robot non-holonome. En effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion. Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites [2].



*Figure 26 : Robot de type unicycle*

### V.2. Robot tricycle

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable, placée sur l'axe longitudinal (figure 27). Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable. C'est un robot non-holonome. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée.

Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable.



*Figure 27 : Robot de type tricycle*

### V.3. Robot voiture

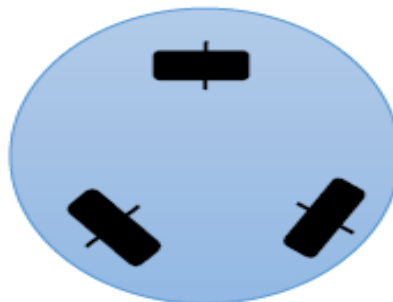
Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe (figure 28). Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire. Toutes les autres propriétés du robot voiture sont identiques au robot tricycle [2].



*Figure 28 : Robot de type voiture*

### V.4. Robot omnidirectionnel

Un robot omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral (figure 29). L'énorme avantage du robot omnidirectionnel est qu'il est holonome puis qu'il peut se déplacer dans toutes les directions [2].



*Figure 29 : Robot de type omnidirectionnel*

Il existe plusieurs types de robots mobiles, chacun avec ses avantages et ses inconvénients en fonction de leur conception, de leur fonctionnalité et de leur application :

- Unicycle :
  - Avantage : Stable, Rotation sur soi-même, Complexité mécanique faible
  - Inconvénients : Non holonome
- Tricycle :
  - Avantage : Complexité mécanique modérée.
  - Inconvénients : Non holonome, Peu stable, Pas de rotation sur soi-même.
- Voiture :
  - Avantage : Stable, Complexité mécanique modérée.
  - Inconvénients : Non holonome, Pas de rotation sur soi même
- Omnidirectionnel
  - Avantage : Holonome, Stable, Rotation sur soi-même.
  - Inconvénients : Complexité mécanique importante.

## VI. Application de robot mobile

Les robots mobiles sont des machines autonomes qui ont la capacité de se déplacer et d'interagir avec leur environnement. Grâce à leur polyvalence et à leur adaptabilité, ils sont utilisés dans divers domaines pour optimiser les opérations, automatiser les tâches répétitives et apporter des solutions novatrices. Que ce soit dans la logistique, l'industrie, les soins de santé, l'agriculture ou la sécurité, les robots mobiles offrent des avantages considérables en termes d'efficacité, de précision et de réduction des coûts. Leur utilisation continue d'évoluer et de se développer pour répondre aux besoins croissants de notre société en matière d'automatisation et de soutien technologique

- Industrie nucléaire : surveillance de sites, manipulation de matériaux radioactifs ;
- Sécurité civile : neutralisation d'activité terroriste, déminage, surveillance de munitions ;
- Militaire : surveillance, patrouille, manipulation de munitions ;
- Chimique : surveillance de site, manipulation de matériaux toxiques
- Médecine : assistance d'urgence, aide aux handicapés physiques, aux aveugles ;
- Espace : exploration ;

- Industriel : convoyage, surveillance ;
- Sous-marine : pose de câbles, recherche de nodules, inspection des fonds marins ;
- Agricole : cueillette de fruits, traite, moisson

## VII. Généralités sur les systèmes linéaires et non linéaires

Un système physique est linéaire si la relation entre les grandeurs d'entrée et la ou les grandeurs de sortie est une équation différentielle linéaire à coefficients constants, là où on peut appliquer le théorème de superposition.

Les systèmes non linéaires, par opposition aux systèmes linéaires, sont les systèmes physiques qui sont régis par des équations différentielles à coefficients variables (équations non linéaires), décrivant l'évolution temporelle des variables constitutives du système sous l'action d'un nombre fini de variables indépendantes appelées entrées ou variables de commande, ou simplement commandes, que l'on peut choisir librement pour réaliser certains objectifs [3] et [4]. Les entrées peuvent être choisies en boucle ouverte, c'est à dire ne dépendant que du temps, ou en boucle fermée, c'est à dire comme des fonctions des variables mesurées, appelés observations, qui rendent compte de l'état du système à chaque instant.

Un système est non linéaire s'il ne vérifie pas le principe de superposition, les conditions de proportionnalité et d'additivité ne s'appliquent plus aux systèmes non linéaires [3] et [4]. Lors de l'étude des systèmes non linéaires on se heurte à plusieurs difficultés.

- L'analyse par des fonctions de transfert est impossible ;
- La notion des pôles disparaît ;
- Un système non linéaire possède en général plusieurs points d'équilibre et l'étude de leur stabilité est plus complexe que dans le cas linéaire pour lequel le concept de stabilité est global.

Si le système à commander présente des faibles non linéarités, dans ce cas il suffit de les ignorer et les traitant comme des perturbations affectant un modèle linéaire du système. Cette approche peut s'appliquer à une grande classe de systèmes et elle permet de profiter de l'arsenal de la théorie du contrôle des systèmes linéaires qui est maintenant assez complet et maîtrisé [3] et [4]. En outre, certains phénomènes sont

difficilement linéarisables (saturations, seuils, etc.), et seule une approche purement non linéaire permet d'expliquer des comportements globaux.

La forme la plus utilisée pour la représentation des systèmes non linéaires est la suivante [3] :

$$\dot{x} = f(x(t), u(t)), \forall t \geq 0$$

Où  $t$  est le temps,  $x(t) \in R^n$  est le vecteur d'état,  $u(t) \in R^m$  est le vecteur de commande ou d'entrée.

## VIII. Les commandes linéaires

Les commandes linéaires sont une approche classique utilisée pour contrôler les robots mobiles. Elles se basent sur des modèles mathématiques linéaires pour décrire le comportement du système et générer des trajectoires ou des actions appropriées. Les commandes linéaires sont souvent simples à mettre en œuvre et offrent une bonne stabilité, mais elles peuvent être limitées pour des systèmes non linéaires ou présentant des comportements complexes.

La commande PID (Proportionnelle, Intégrale, Dérivée) est en effet l'une des techniques de commande les plus connues et les plus utilisées pour contrôler un système linéaire.

Le régulateur PID, également connu sous le nom de correcteur PID (proportionnel, intégral, dérivé), est un système de contrôle largement utilisé dans l'industrie pour améliorer les performances des systèmes asservis en boucle fermée. Il est capable de réguler et de corriger plusieurs grandeurs physiques.

Le régulateur PID est composé de trois termes principaux : le terme proportionnel (P), le terme intégral (I) et le terme dérivé (D). Chaque terme joue un rôle spécifique dans le contrôle du système figure (30).

- Le terme proportionnel (P) : ajuste la sortie du régulateur en fonction de l'erreur actuelle, c'est-à-dire la différence entre la valeur de consigne (la valeur souhaitée) et la valeur réelle du système. Plus l'erreur est grande, plus la correction proportionnelle est importante.
- Le terme intégral (I) : agit en fonction de l'accumulation des erreurs passées. Il corrige les erreurs systématiques persistantes en ajustant la sortie du régulateur proportionnellement à la somme cumulée des erreurs au fil du temps. Cela permet de réduire l'erreur en régime permanent.

- Le terme dérivé (D) : réagit à la vitesse à laquelle l'erreur change. Il est sensible à la pente de l'erreur et produit une correction en proportion de cette variation. Le terme dérivé est particulièrement utile pour anticiper les changements rapides de l'erreur et réagir rapidement pour les atténuer.

En combinant ces trois termes, le régulateur PID peut fournir un contrôle précis et réactif du système. La combinaison des termes proportionnel, intégral et dérivé permet d'optimiser les performances en réduisant le temps de réponse, l'erreur statique et les oscillations indésirables du système.

Dans l'industrie, le régulateur PID est utilisé dans de nombreux domaines, tels que la régulation de la température, la régulation de la pression, la régulation du débit, le positionnement précis des moteurs, etc. Son utilisation généralisée s'explique par sa simplicité de mise en œuvre, sa flexibilité et ses performances éprouvées dans de nombreuses applications la figure suivante (30) représente le diagramme bloc de commande PID.

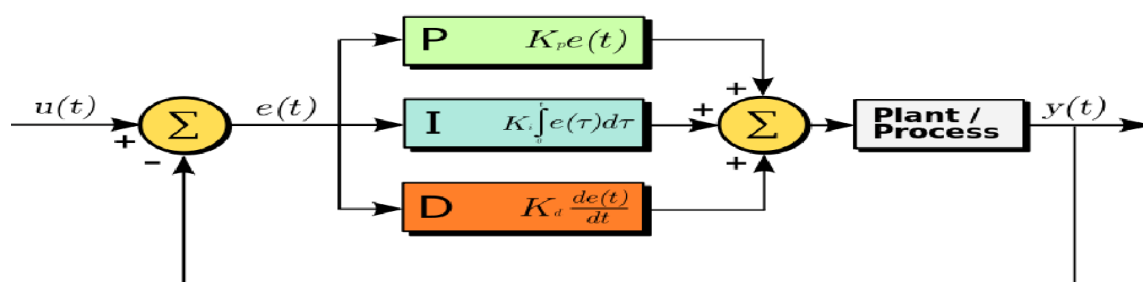


Figure 30 : le diagramme bloc de commande PID

## IX. Les commandes non linéaires

Le contrôle de systèmes non linéaires est un domaine de recherche passionnant qui a connu une forte croissance ces dernières années. Les systèmes non linéaires sont présents dans de nombreuses applications industrielles telles que les robots, les avions, les voitures autonomes et les systèmes de production. Les commandes non linéaires ont l'avantage de pouvoir traiter des systèmes non linéaires complexes et offrir de bonnes performances de suivi de référence.

### IX.1. Commande par platitude

Un système est considéré comme "plat" si ses sorties et ses entrées peuvent être exprimées comme des fonctions algébriques d'un ensemble fini de sorties plates et de leurs dérivées.

### IX.1.1. Notion du système plat

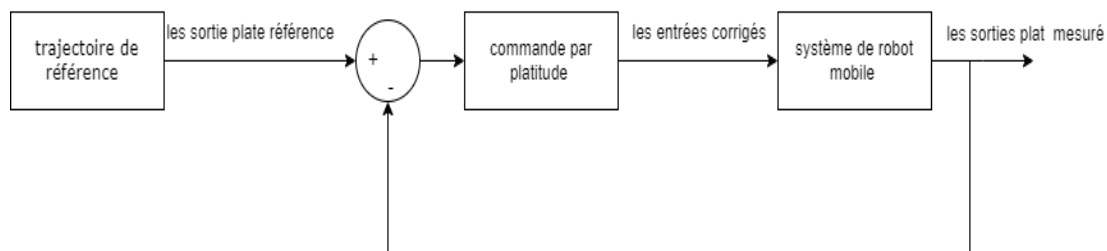
La notion de platitude est une propriété caractérisant une classe de systèmes non linéaires. Elle a été introduite et étudiée dans le cadre de l'algèbre différentielle et de la géométrie différentielle.

En algèbre différentielle, la platitude est étudiée en relation avec les distributions intégrables. Un système est dit plat s'il existe une distribution intégrable qui permet d'exprimer toutes les variables d'état en fonction d'un nombre fini de dérivées de la sortie plate, tout en vérifiant les équations différentielles du système. Cette notion est liée à la possibilité de trouver un système linéaire équivalent par bouclage dynamique endogène, où les variables d'état peuvent être exprimées en fonction de la sortie plate et d'un nombre fini de ses dérivées [5].

En géométrie différentielle, la platitude est étudiée en relation avec les transformations de Lie. Un système est dit plat s'il existe une transformation de Lie-Bäcklund qui permet de transformer le système non linéaire en un système linéaire commandable. Cette transformation établit une équivalence entre le système non linéaire et le système linéaire commandable, facilitant ainsi l'application des techniques de commande linéaire.

Les concepts d'équivalence par bouclage dynamique endogène et d'équivalence de Lie-Bäcklund sont utilisés pour décrire cette équivalence entre les systèmes non linéaires et les systèmes linéaires commandables dans les cadres de l'algèbre différentielle et de la géométrie différentielle respectivement [5] figure (31).

Il est important de noter que les systèmes plats sont une généralisation des systèmes linéaires commandables dans le sens où tout système linéaire commandable est plat.

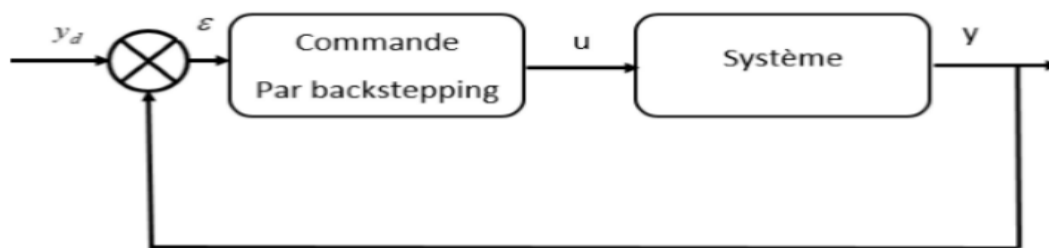


*Figure 31 : Schéma bloc de commande platitude*

### IX.2. Commande backstepping

La technologie Backstepping propose une méthode de synthèse de système, qui est conçue pour être utilisée pour des systèmes non linéaires avec des formes triangulaires.

Il est basé sur une cascade qui décompose l'ensemble du système de contrôle (généralement de haut niveau) en sous-systèmes de contrôle de premier ordre. Pour chaque sous-système, la loi dite de commande virtuelle est calculée. Ce dernier servira de référence pour le sous-système suivant jusqu'à ce que la loi de commande de l'ensemble du système soit obtenue. La dernière étape résume les lois de commande à appliquer. Elle doit toujours assurer la stabilité globale du système d'indemnisation lorsqu'il est poursuivi ou ajusté. Cette méthode convient aux systèmes avec des triangles dits inférieurs voire la figure (32).



*Figure 32 : Schéma bloc de la commande par backstepping*

### IX.3. La commande linéarisante

La commande linéarisante est une technique de commande pour les systèmes non linéaires qui consiste à concevoir une commande qui transforme le système en un système linéaire équivalent (figure 33). Pour résoudre le modèle dynamique d'un robot mobile à deux roues, nous pouvons utiliser la commande linéarisant en suivant les étapes suivantes :

- **Modélisation dynamique du robot mobile à deux roues :** Il est nécessaire de décrire le modèle dynamique du robot mobile à deux roues en utilisant des équations différentielles non linéaires. Cette modélisation prend en compte les mouvements du robot, la dynamique des roues et les forces externes appliquées sur le robot.
- **Linéarisation du modèle dynamique :** La linéarisation consiste à trouver une transformation de coordonnées qui permet de décrire le système non linéaire sous une forme linéaire. Cette étape peut être réalisée par une approximation de Taylor au premier ordre, en évaluant la dérivée partielle des équations différentielles non linéaires.
- **Conception de la commande linéaire :** Une fois que le modèle dynamique a été linéarisé, il est possible d'appliquer une commande linéaire pour stabiliser le système.



- Différentes techniques de commande linéaire peuvent être utilisées, telles que la commande par retour d'état, la commande prédictive, la commande proportionnelle-intégrale (PI), ou la commande proportionnelle-intégrale-dérivée (PID).
- Simulation et validation : La commande linéarisant doit être validée par des simulations sur ordinateur avant d'être appliquée au robot mobile réel. Les simulations permettent de vérifier que la commande linéaire est capable de stabiliser le système en présence de perturbations externes.

### IX.4. Commande par Lyapunov

La commande par Lyapunov est une méthode de conception de commande pour les systèmes dynamiques qui consiste à utiliser une fonction de Lyapunov pour concevoir un contrôleur qui assure la stabilité du système. La fonction de Lyapunov est une fonction mathématique qui mesure l'énergie du système et qui est choisie pour être décroissante le long des trajectoires du système, c'est-à-dire qu'elle diminue au fil du temps.

#### IX.4.1. Les étapes de la méthode de commande par Lyapunov

La méthode de commande par Lyapunov implique les étapes suivantes :

- Sélectionnez une fonction de Lyapunov candidate qui mesure l'énergie du système et qui est décroissante le long des trajectoires du système.
- Calculez la dérivée de cette fonction le long des trajectoires du système en utilisant l'équation du système dynamique.
- Concevez un contrôleur qui utilise cette dérivée pour générer des commandes qui assurent que la fonction de Lyapunov candidate décroît le long des trajectoires du système.
- Vérifiez que le système est asymptotiquement stable.

## X. Les commandes robustes

La commande robuste est une approche de contrôle qui vise à garantir la stabilité et la performance d'un système, même en présence d'incertitudes ou de variations dans les paramètres du système ou de perturbations externes. Elle est conçue pour rendre le système moins sensible à ces incertitudes et perturbations, assurant ainsi un fonctionnement fiable dans un large éventail de conditions.

## X.1. La commande par mode glissant

La technique des modes glissants consiste à amener la trajectoire d'état d'un système vers la surface en mode glissant et de la faire commuter à l'aide d'une logique de commutation appropriée autour de celle-ci jusqu'au point d'équilibre voire figure (33), d'où le phénomène de mode glissant [7].

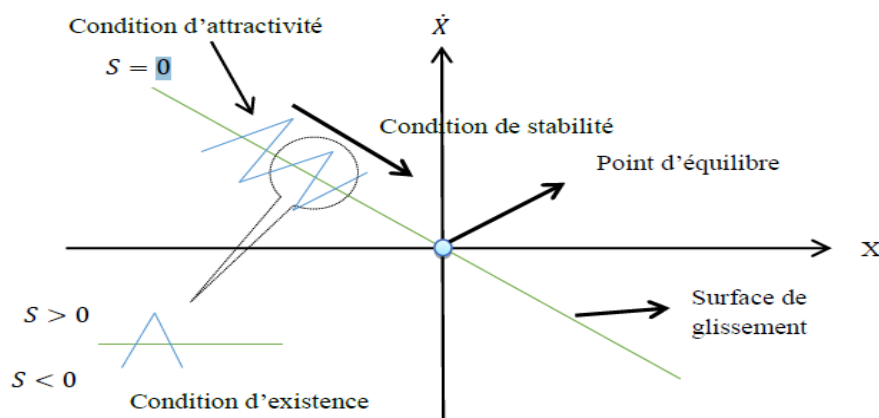
Parmi les propriétés des modes glissants :

- Le processus de glissement est d'ordre réduit en comparaison au système original.
- La dynamique du système en mode de glissement est déterminée uniquement par le choix des coefficients de la surface de glissement.
- La robustesse vis-à-vis de la variation de certains types de paramètres.
- La trajectoire dans le plan de phase est constituée de trois parties distinctes.

Le Mode de convergence « MC » : C'est le mode durant lequel la variable à régler se déplace à partir de n'importe quel point initial dans le plan de phase et tend vers la surface de commutation  $s(x, y) = 0$ . 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  $s(x, y) = 0$ .

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 [7].



*Figure 33 : Différents modes pour la trajectoire dans le plan de phase*

### X.1.1. Conception de l'algorithme de commande par mode glissant

La conception de l'algorithme de commande par mode glissant prend en compte les problèmes de stabilité et de bonnes performances de façon systématique dans son approche, qui s'effectue principalement en trois étapes complémentaires définies par [7] :

- Choix des surfaces de glissement ;
- Définition des conditions d'existence et de convergence du régime glissant ;
- Détermination de la loi de commande.

### X.2. Commande H-infini

La commande H-infini est une méthode de conception de contrôleur robuste pour les systèmes linéaires ou non linéaires soumis à des perturbations, des incertitudes et des erreurs de modélisation. Le but de la commande H-infini est de minimiser l'erreur de suivi de référence tout en garantissant une stabilité interne et une atténuation des perturbations. Cette méthode est basée sur la théorie des systèmes linéaires H-infini, qui a été développée dans les années 1980.

Pour appliquer la commande H-infini à un système non linéaire, on doit d'abord le linéariser autour d'un point de fonctionnement nominal et obtenir un modèle linéaire du système. Ensuite, on peut utiliser la méthode de synthèse de contrôleur H-infini pour concevoir un contrôleur qui minimise l'erreur de suivi de référence tout en garantissant une stabilité interne et une atténuation des perturbations pour le modèle linéaire.

Une approche possible pour la conception de la commande H-infini pour les systèmes non linéaires consiste à utiliser des techniques de commande non linéaire, telles que la commande par retour d'état non linéaire ou la commande par platitude, pour réguler le système non linéaire autour du point nominal. Ensuite, on peut utiliser la méthode de synthèse de contrôleur H-infini pour concevoir un contrôleur qui minimise l'erreur de suivi de référence pour le système régulé non linéaire.

#### X.2.1. Norme $H_\infty$ d'un système linéaire

Soit un système linéaire invariant décrit par la représentation d'état [8] :

$$\dot{x}(t) = A x(t) + B u(t)$$

$$y(t) = C x(t) + D u(t)$$

$$A, B, C, D \in \mathbb{R}^{n \times m}$$

La matrice de transfert correspondante s'écrit  $G(s) = C(sI - A)^{-1}B$ . On suppose que le système est stable. L'ensemble des matrices de transfert  $G(s)$  correspond à un système stable est noté usuellement  $\mathcal{RH}_{\infty}^{n \times m}$

On définit la norme  $H_{\infty}$  notée  $\|G(s)\|_{\infty}$  de la manière suivante :

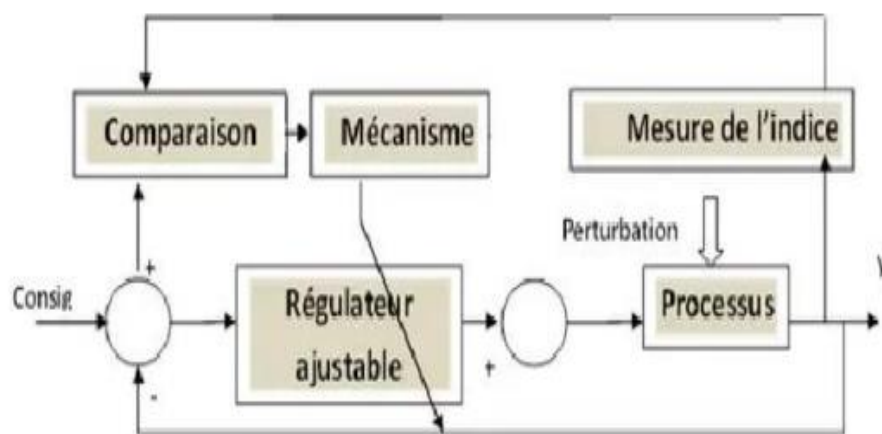
$$\|G(s)\|_{\infty} = \sup_{\omega \in \mathbb{R}} \sigma(G(j\omega))$$

$\|G(s)\|_{\infty}$  est la valeur la plus élevée du gain du système sur l'ensemble des pulsations (pour un système mono-variable), c'est la valeur la plus élevée ( $|G(j\omega)|$ ). Il n'existe pas de méthode explicite pour calculer la norme  $H_{\infty}$  d'une matrice de transfert. Il est possible de faire recours à des méthodes itératives [8].

### X.3. Commande adaptative

La commande adaptative est un ensemble de techniques utilisées pour l'ajustement automatique en temps réel des régulateurs des boucles de commande afin de réaliser ou maintenir un certain niveau de performances quand les paramètres du procédé à commander sont inconnus soit variantes dans le temps [1].

Un système de commande adaptative mesure un certain indice de performance (IP) du système à commander à partir de l'écart entre l'indice de performance désiré et l'indice de performance mesuré. Le mécanisme d'adaptation commande certains paramètres du système ajustable ou introduit un signal supplémentaire de commande d'après une certaine stratégie afin de minimiser l'IP la figure (34) représente le principe général d'un système dans une plage donnée de commande adaptative [1].



*Figure 34 : Principe de système de commande adaptative*

## **X.4. La commande de rejet de perturbation**

La commande de rejet de perturbation est une technique de commande qui permet de contrôler un système en présence de perturbations externes. Elle est utilisée pour maintenir le système à un état stable malgré les variations et les perturbations qui peuvent affecter le système.

### **X.4.1. Le principe de fonctionnement de la commande de rejet de perturbation.**

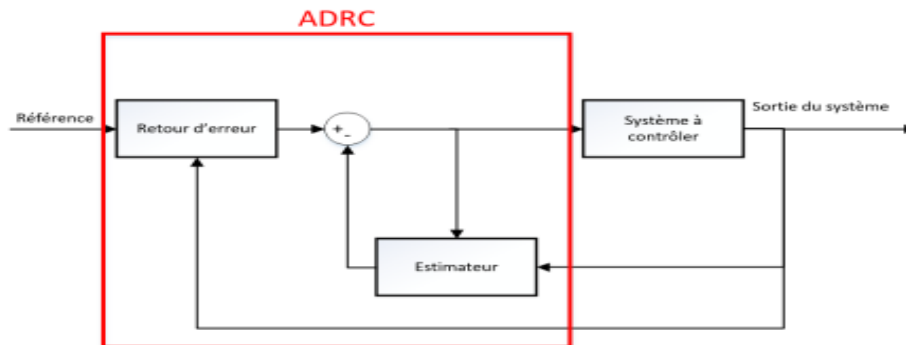
Le principe de fonctionnement de la commande de rejet de perturbation consiste à mesurer les perturbations externes qui affectent le système et à utiliser ces mesures pour générer une commande qui compense les effets des perturbations. La commande est calculée en temps réel à partir des mesures de perturbation et de l'état actuel du système.

Le processus de commande de rejet de perturbation peut être divisé en plusieurs étapes :

- **Mesure des perturbations** : Des capteurs sont utilisés pour mesurer les perturbations externes qui affectent le système. Les mesures de perturbation sont utilisées pour calculer la commande de rejet de perturbation.
- **Estimation de l'état du système** : Des capteurs sont utilisés pour mesurer l'état actuel du système, tel que la position, la vitesse, l'accélération, etc. Cette information est utilisée pour calculer la commande de rejet de perturbation en temps réel.
- **Calcul de la commande** : Les mesures de perturbation et l'état du système sont utilisés pour calculer la commande de rejet de perturbation en temps réel. La commande est conçue pour compenser les effets des perturbations externes et maintenir le système à un état stable malgré les perturbations.
- **Application de la commande** : La commande est appliquée au système pour corriger les erreurs causées par les perturbations et maintenir la performance du système.

Le principe de fonctionnement de la commande de rejet de perturbation est donc basé sur la mesure des perturbations externes et l'utilisation de ces mesures pour calculer une commande qui compense les effets des perturbations. Cette technique permet de maintenir la stabilité et la performance du système malgré les perturbations externes.

La structure classique de la commande par rejet de perturbation représentée par la figure (35).



*Figure 35 : Structure classique de la commande par rejet de perturbation*

## X.5. Commande optimale

Les commandes optimales, sont un domaine de l'ingénierie et des mathématiques qui étudie la manière de contrôler un système dynamique de manière à optimiser un critère spécifique. L'objectif principal des commandes optimales est de déterminer la meilleure trajectoire de contrôle pour atteindre un certain état final tout en optimisant une fonction de coût prédéterminée.

La théorie des commandes optimales s'appuie sur des outils mathématiques tels que le calcul variationnel et les équations aux dérivées partielles. Elle est utilisée dans de nombreux domaines, tels que l'ingénierie, l'économie, la physique, la biologie et l'aérospatiale, pour résoudre des problèmes de contrôle complexes.

### X.5.1. Commande Linéaire Quadratique LQR

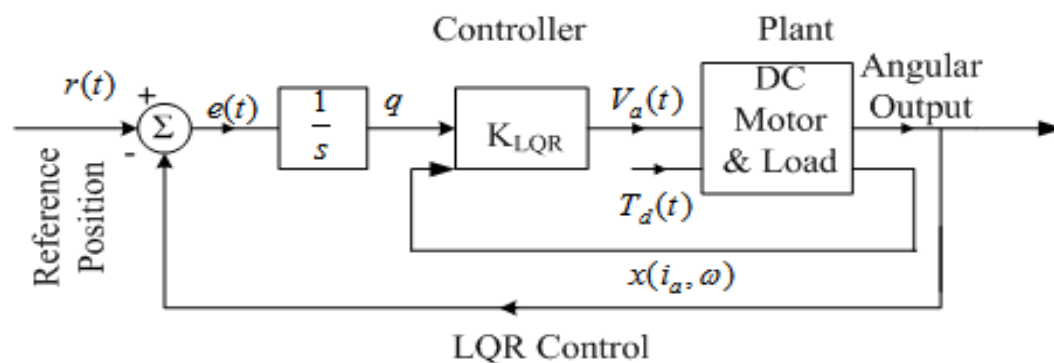
On parle de commande linéaire quadratique : LQ ou LQR pour (linear Quadratic regulator), le système est linéaire et la commande est quadratique, la commande optimale est un retour d'état sous la forme :  $U(t) = -kx(t)$

On veut amener  $x$  à un état désiré :

- A moindre coût.
- Bien réagir aux perturbations.
- Contrôler  $U$  linéairement grâce à  $x$  de manière optimale.

Le problème de commande LQR peut être formulé mathématiquement comme une équation d'état linéaire avec une fonction de coût quadratique. Les étapes typiques pour implémenter une commande LQR sont les suivantes :

- Modélisation du système : Le système dynamique doit être modélisé sous forme d'équations d'état linéaires.
- Définition de la fonction de coût : Une fonction de coût quadratique est définie, généralement sous la forme d'une somme pondérée des erreurs d'état et des commandes.
- Calcul des gains de rétroaction : Les gains de rétroaction optimaux sont calculés à l'aide de techniques d'optimisation. La méthode classique pour résoudre ce problème consiste à utiliser l'équation de Riccati algébrique, qui fournit une solution analytique pour les gains de rétroaction.
- Calcul de la commande : La commande de contrôle optimale est calculée en multipliant les gains de rétroaction par l'état actuel du système.



**Figure 36 : Le diagramme bloc de commande LQR**

Les figure (36) représente le diagramme bloc de commande LQR. La commande prédictive (MPC).

La commande prédictive à base de modèle c'est une approche de commande avancée. De toutes les commandes qui portent cette appellation, la commande prédictive est la plus utilisée dans l'industrie, aussi bien pour les qualités techniques et ses performances ainsi que sa facilité de mise en œuvre. Son champ d'application s'étend à tous les domaines industriels, notamment aux cas où la régulation PID n'est pas efficace. [14]

- **Concepts de base :**

Toutes les techniques proposées dans la littérature ont en commun les concepts suivants :

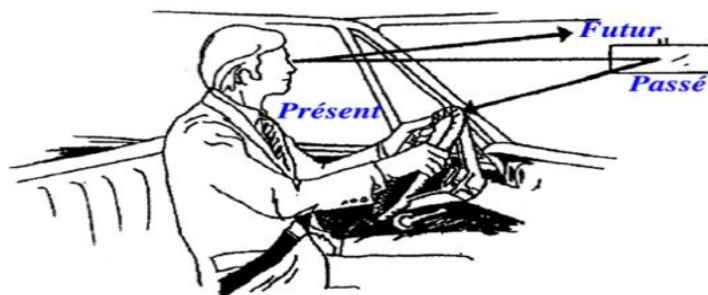
- Utilisation d'un modèle pour construire la prédiction des signaux intéressants : États ou sorties.

- Connaissance de la trajectoire à suivre sur un horizon au moins aussi long que l'horizon De prédiction.
- Minimisation d'un critère le plus souvent quadratique incluant l'erreur de poursuite et l'effort de commande.
- Utilisation d'un algorithme de résolution produisant en temps réel la solution optimale /ou sous optimale/admissible.
- Application du premier élément de la séquence de commande calculée.
- Répétition de la procédure à la période d'échantillonnage suivante selon le principe de L'horizon fuyant. [15]

- **Principes de la commande prédictive :**

Fondamentalement, l'algorithme de commande prédictive prend en compte le comportement futur du système afin d'élaborer une commande permettant le meilleur suivi d'une trajectoire connue. [16]

Le principe de la commande prédictive consiste à créer pour le système à commander un effet anticipatif par rapport à une trajectoire à suivre connue à l'avance, en se basant sur la prédiction du comportement futur du système et en minimisant l'écart de ces prédictions par rapport à la trajectoire au sens d'une certaine fonction coût, tout en respectant des contraintes de fonctionnement. Cette idée est simple et pratiquée de façon assez systématique dans la vie quotidienne. Par exemple, le conducteur d'un véhicule connaît la trajectoire de référence désirée à l'avance (la route) sur un horizon de commande fini (son champ visuel), et en prenant en compte les caractéristiques de la voiture (modèle mental du comportement du véhicule), il décide quelles actions (accélérer, freiner ou tourner le volant) il faut réaliser afin de suivre la trajectoire désirée. Seule la première action de conduite est exécutée à chaque instant, et la procédure est répétée à nouveau pour les prochaines actions. [17]



*Figure 37 : Philosophie de la commande prédictive*



La figure (37) résume le principe de la commande prédictive dans le cas général d'un problème de commande optimale sous contraintes sur un horizon glissant.

## **XI. Conclusion**

En conclusion, ce chapitre nous a permis de comprendre les différents types de robots mobiles, leur domaine d'utilisation et leur classification. Nous avons également examiné les techniques de commande linéaire, non linéaire et robuste utilisées dans la robotique mobile. Ces connaissances nous serviront de base solide pour la suite du parcours, où nous approfondirons nos compétences en matière de commande.

Dans le prochain chapitre, nous approfondirons la commande par platitude, qui simplifie la commande des robots mobiles en transformant les variables d'état en variables de sortie contrôlables. Nous explorerons également la génération de trajectoires à l'aide de cette commande et par courbe de Bézier.

# Chapitre 3 : commande non linéaire par platitude

## I. Introduction

Le contrôle précis des trajectoires est un aspect essentiel de nombreux systèmes dynamiques, qu'il s'agisse de robots, de véhicules autonomes ou d'autres applications. Dans ce chapitre, nous abordons deux concepts fondamentaux qui jouent un rôle clé dans la génération et le contrôle des trajectoires : la commande par platitude et les courbes de Bézier.

## II. Notion du système plat

La platitude est une propriété caractérisant une classe de systèmes non linéaires. Elle a été définie dans le cadre de l'algèbre différentielle, puis dans le cadre de la géométrie différentielle.

Le concept de la platitude introduit une notion d'équivalence entre un système non linéaire et un système linéaire commandable. Cette équivalence porte le nom d'équivalence par bouclage dynamique endogène dans le cadre de l'algèbre différentielle et d'équivalence de Lie-Bäcklund dans le cadre de la géométrie différentielle, [8].

Il est important de noter que les systèmes plats sont une généralisation de systèmes linéaires commandables dans le sens où tout système linéaire commandable est plat.

### II.1. Définition de la platitude dans le cadre de l'algèbre différentielle

Dans le contexte de l'algèbre différentielle, un système est vu comme un champ de vecteurs généré par un ensemble de variables (états et commandes). La caractéristique essentielle des systèmes plats est qu'il existe un vecteur de sorties plates de dimension égale à la dimension du vecteur de commande, tel que ses composantes soient des fonctions différentielles des variables du système, et telle que toute variable du système (on ne fait pas de distinction entre les variables d'état et les entrées du processus) puisse s'exprimer à partir de ces sorties et d'un nombre fini de leurs dérivées, sans intégration d'équations différentielles.

Pour définir les systèmes plats, considérons le système non linéaire régi par l'équation différentielle suivante :

$$\dot{x} = f(x, u) \quad (\text{III.1})$$

Où  $x \in \mathbb{R}^m$  est l'état,  $u \in \mathbb{R}^m$  est l'entrée et  $f$  est une fonction régulière de classe  $C^\infty$  de  $x$  et de  $u$  dont le rang de la matrice Jacobéenne  $\frac{\partial f}{\partial u}$  est égal à  $m$  (c'est-à-dire que le système admet effectivement  $m$  commandes indépendantes).

Un système non linéaire modélisé par (II.1) est dit différentiellement plat si, et seulement s'il existe un vecteur de sorties plates  $F \in \mathbb{R}^m$  différentiellement indépendantes, de dimensions égales à celle du vecteur de commande  $u$ , dépendant de  $x$  et de  $u$  et d'un nombre fini  $r$  de ses dérivées :

$$y = \phi(x, \dot{u}, \dots, u^{(s)}) \quad (\text{III.2})$$

Tel que :

$$x = X(F, \dot{F}, \dots, F^r) \quad (\text{III.3})$$

$$u = U(F, \dot{F}, \dots, F^r, F^{r+1})$$

Où  $\phi: \mathbb{R}^n \times (\mathbb{R}^m)^{s+1} \rightarrow \mathbb{R}^m$ ,  $X: (\mathbb{R}^m)^{r+1} \rightarrow \mathbb{R}^n$   $U: (\mathbb{R}^m)^{r+2} \rightarrow \mathbb{R}^m$  fonction régulière.  $r$  Est le degré relatif.

Cela revient à dire que tout le comportement dynamique du système peut être décrit par le comportement dynamique de la sortie plate  $F$  [8].

A  $y(t)$  définie pour  $t \in [0, T]$ , les trajectoires sont de la forme :

$$\begin{aligned} x(t) &= X(F(t), \dot{F}(t), \dots, F^r(t)) \\ u(t) &= U(F(t), \dot{F}(t), \dots, F^r(t), F^{r+1}(t)) \end{aligned} \quad (\text{III.4})$$

Où  $r$  est un entier

## II.2. Algorithme de vérification de sorties plates

Il existe plusieurs méthodes de vérification de sorties plates, mais on va présenter la méthode la plus utilisée. Nous présentons deux sorties plates, c'est-à-dire deux commandes [8].

Nous voulons savoir si le couple  $(F_1, F_2)$  peut jouer le rôle de sorties plates d'un système possédant un état de dimensions  $n$ . On procède comme suit :

1. Dériver  $F_1$  jusqu'à faire apparaître une combinaison des commandes.
2. Nous notons  $n_1$  le nombre de dérivations nécessaires.
3. Dériver  $F_2$  jusqu'à faire apparaître une autre combinaison des commandes.
4. Nous notons  $F_2$  le nombre de dérivations nécessaires.

Puis :

si  $n_1 + n_2 = n$ , le système admet  $(y_1, y_2)$  pour sorties plates. Sinon,  $(y_1, y_2)$  n'est pas un couple de sorties plates pour le système.

### II.3. Platitude et linéarisation

La notion de platitude d'un système non linéaire est une propriété mathématique puissante qui permet de simplifier le contrôle de systèmes complexes en les ramenant à des systèmes linéaires commandables. Elle repose sur la possibilité de transformer le système non linéaire en un système équivalent qui peut être décrit par un modèle linéaire, en utilisant des techniques de linéarisation par difféomorphisme et de bouclage statique.

Le processus de linéarisation par difféomorphisme implique la recherche d'une transformation bijective, appelée difféomorphisme, qui transforme les variables d'état du système non linéaire en de nouvelles variables, de telle sorte que le système sous cette nouvelle forme devient linéaire. Ce difféomorphisme permet d'effectuer un changement de coordonnées pour exprimer le système dans un espace où il est plus facilement linéarisable[8].

Une fois que le système a été linéarisé par difféomorphisme, il peut encore comporter des non-linéarités statiques. Le bouclage statique intervient alors pour éliminer ces non-linéarités en utilisant des fonctions de rétroaction qui rendent le système linéairement commandable. Le principe est de construire des lois de commande qui annulent les non-linéarités statiques, transformant ainsi le système en un système linéaire avec une dynamique commandable.

L'avantage d'avoir un système équivalent linéaire commandable est qu'il offre une plus grande facilité d'analyse et de conception de lois de commande. Les techniques bien établies de contrôle linéaire peuvent être appliquées pour stabiliser le système, atteindre des performances souhaitées et suivre des trajectoires spécifiques.

### II.4. Platitude et planification de trajectoires

Pour revenir à votre demande initiale, le problème de planification de trajectoires pour un système non linéaire est en effet complexe. Il nécessite généralement une approche itérative pour trouver une trajectoire  $t \rightarrow (x(t), u(t))$  qui satisfait les équations (II.1) du système, ainsi que les conditions initiales et finales [8].

$$x(t_0) = x_0, u(t_0) = u_0 \quad (\text{III.5})$$

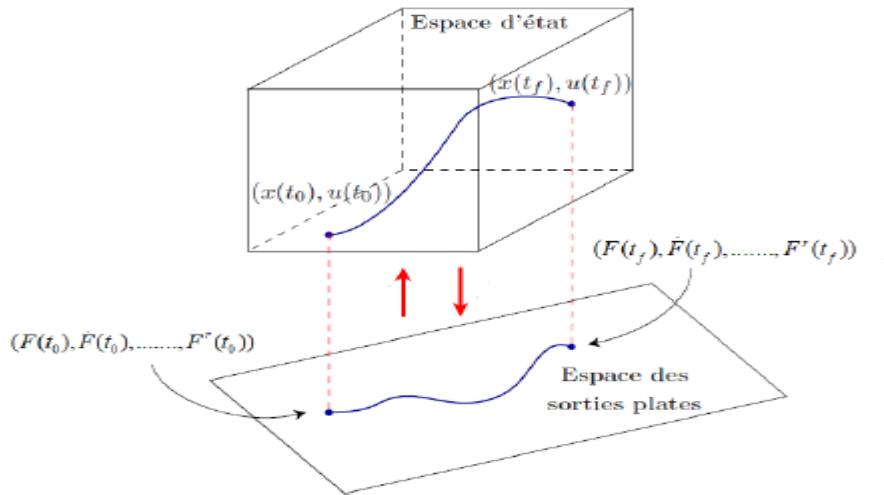
$$x(t_f) = x_f, u(t_f) = u_f$$

Le processus itératif implique de fixer une commande initiale  $t \rightarrow u_0(t)$ , d'intégrer les équations du système à partir des conditions initiales pour obtenir la solution à l'instant final  $t_f$ ,

Puis de développer une procédure pour choisir une nouvelle commande  $t \rightarrow u_1(t)$  qui se rapproche des conditions finales souhaitées. Ce processus est répété jusqu'à ce qu'une trajectoire satisfaisante soit trouvée. La résolution numérique des équations différentielles est généralement utilisée pour résoudre ce problème. Cependant, grâce à la propriété de la platitude, il est possible d'établir une relation directe entre les trajectoires des sorties plates et celles de l'état et des commandes du système. Plus précisément, à partir des trajectoires de référence des sorties plates  $F_d(t)$  et de leurs dérivées successives, il est possible de déduire explicitement la trajectoire de référence de l'état  $x_d(t)$  et les commandes nominales correspondantes  $u_d(t)$ . Cette propriété de la platitude permet d'éviter la résolution itérative des équations différentielles et facilite la détermination directe des trajectoires de référence en se basant sur les sorties plates de référence. Voir figure (38).

$$x_d(t) = \Gamma_1(F_d(t), \dot{F}_d(t), \dots, F_d^{r-1}(t)) \quad (\text{III.6})$$

$$u_d(t) = \Gamma_2(F_d(t), \dot{F}_d(t), \dots, F_d^r(t)) \quad (\text{III.7})$$



**Figure 38 : Génération de trajectoires dans l'espace des sorties plates**

Ainsi, pour la planification de trajectoires, nous cherchons à définir une trajectoire continue  $t \rightarrow F_d(t)$  qui est au moins  $r + 1$  fois dérivable sur l'intervalle  $[t_0, t_f]$  et qui satisfait les conditions initiales et finales pour les sorties plates  $(F_d(t), \dot{F}_d(t), \dots, F_d^{(r)}(t))$ .

Les trajectoires générées dans l'espace des sorties plates peuvent ensuite être converties en trajectoires dans l'espace d'état en utilisant les relations précédentes (III.6) et (III.7).

Une caractéristique intéressante est que la construction de la trajectoire  $t \rightarrow F_d(t)$  ne nécessite pas de résoudre des équations différentielles ni d'effectuer des intégrations. Au lieu de cela, il est possible de la construire simplement en utilisant une interpolation polynomiale. Cela signifie que nous pouvons spécifier les valeurs de  $F_d(t)$  à des instants de temps discrétisés, puis utiliser une interpolation polynomiale pour obtenir une trajectoire continue entre ces points. Cette approche facilite la génération de trajectoires désirées dans l'espace des sorties plates, offrant ainsi une méthode efficace pour la planification de trajectoires.

## II.5. Platitude et suivi de trajectoire :

Dans le domaine de la planification de trajectoires, il est possible d'adopter une approche en boucle ouverte ou en boucle fermée, en fonction des connaissances disponibles sur le système et des exigences de performance. [8].

Dans une approche en boucle ouverte, la trajectoire de référence est planifiée à l'avance, en utilisant uniquement le modèle dynamique du système et les informations de temps. Cette planification de trajectoire anticipe la réaction du système en fonction de ses caractéristiques et de son comportement connus. L'objectif est d'atteindre la trajectoire souhaitée en fonction de ces informations anticipées.

Cependant, si la modélisation du système n'est pas suffisamment précise ou si des perturbations externes importantes affectent le système, une approche en boucle ouverte seule peut ne pas être suffisante pour atteindre de manière précise la trajectoire de référence. C'est là que l'approche en boucle fermée entre en jeu.

Dans une approche en boucle fermée, des capteurs sont utilisés pour mesurer l'état du système en temps réel. Ces mesures sont comparées à la trajectoire de référence anticipée, ce qui permet de quantifier l'écart entre la trajectoire réelle et la trajectoire souhaitée. Une loi de commande en boucle fermée est alors conçue pour ajuster les actions de contrôle en fonction de cet écart, afin de réduire progressivement l'écart et de permettre au système de suivre de manière plus précise la trajectoire de référence.

Donc l'objectif de la boucle fermée est de minimiser l'écart entre la trajectoire réelle et la trajectoire de référence, en utilisant les informations en temps réel pour prendre des décisions de contrôle appropriées. La conception de la loi de commande en boucle fermée

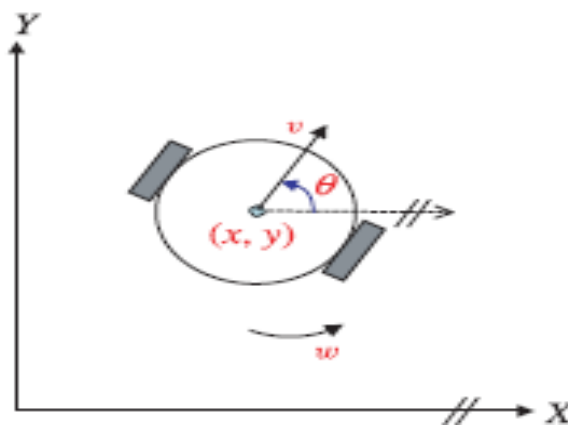
dépend du système spécifique, de la précision de la modélisation, des perturbations attendues et des exigences de performance.

### III. La stratégie de contrôle basée sur la commande platitude

La conception de contrôle et la planification de trajectoire pour les systèmes plans sont relativement faciles car la trajectoire peut être définie en termes de sorties planes tandis que la commande requise peut être obtenue en utilisant la propriété la commande platitude, pour prouver comment le modèle cinématique du robot mobile est différentiellement plate, nous choisissons la position cartésienne du centre du robot  $(x, y)$  comme sorties planes.

Afin de concevoir une difféomorphisme entre les sorties planes et leurs dérivées, ainsi que les états d'origine, nous utilisons la prolongation d'entrée. La prolongation est une méthode essentielle utilisée pour étendre le vecteur représentant l'état en y ajoutant certains paramètres du système qui permettent de décrire un système particulier comme un système différentiellement plat. Une façon très courante de prolonger consiste à prolonger l'entrée, où l'entrée devient également un état. Cette propriété est utilisée dans la génération de trajectoires optimales et les lois de contrôle de suivi.

Maintenant, en effectuant une prolongation de  $v$  en tant qu'état supplémentaire, nous décrivons les systèmes prolongés par :



*Figure 39 : Modèle de robot mobile à deux roues sans glissement*

#### III.1. Modèle cinématique sans glissement

Dans cette section, nous obtenons le modèle cinématique d'un différentiel robot mobile à deux roues illustrées à la figure 39. En coordonnées cartésiennes, la configuration du système est donnée par

$$q = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (\text{III.8})$$

Où  $(x, y)$  est la position centrale du robot et  $\theta$  est la orientation par rapport à l'axe X. De l'hypothèse d'une condition d'absence de glissade sur les roues, la contrainte non holonomie est donnée par :

$$C(q)\dot{q} = 0 \quad (\text{III.9})$$

Ou

$$C(q) = [\sin \theta, -\cos \theta, 0] \quad (\text{III.10})$$

Avec une matrice  $S(q)$  qui couvre l'espace nul de  $C(q)$ , il est possible de représenter le modèle cinématique du mobile robot comme

$$\dot{q} = S(q)v \quad (\text{III.11})$$

Avec

$$s(q) = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}, v = \begin{bmatrix} v \\ w \end{bmatrix} \quad (\text{III.12})$$

Ici,  $v$  et  $w$  représentent les vitesses avant et angulaire du robot mobile, respectivement. On peut écrire le modèle cinématique comme suit :

$$\begin{aligned} \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= w \end{aligned} \quad (\text{III.13})$$

La relation entre le vecteur de commande,  $w$  et  $v$ , et les dérivées les plus élevées de la sortie plane n'est pas inversible. Ce problème expose clairement un obstacle à la réalisation de la linéarisation du feedback statique. Pour surmonter ce fait, l'entrée de commande  $v$  est considérée comme un état supplémentaire pour le modèle cinématique (6), par conséquent, le nouveau système étendu est défini comme suit :

$$\begin{aligned} \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \end{aligned} \quad (\text{III.14})$$

$$\dot{\theta} = w$$

$$u1 = \dot{v} \quad (\text{III.15})$$

$$u2 = \dot{\theta}$$

où le système d'état du robot mobile est  $X = [x ; y ; v ; \theta]^T$  et la nouvelle entrée de



commande est définie par  $u1 = \dot{v}$  et  $u2 = \dot{\theta}$ . La relation inversible entre les entrées  $u1$  et  $v$  et  $u2$  et  $w$

En choisissant les sorties planes :

$$F = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (\text{III.16})$$

Toutes les entrées et les variables d'état peuvent être exprimés en termes de sorties planes et de leurs dérivées. Avec  $(x ; y) = (f_1 ; f_2)$ .

$$\dot{v} = \sqrt{\dot{f}_1^2 + \dot{f}_2^2}, \theta = \arctan\left(\frac{\dot{f}_2}{\dot{f}_1}\right) \quad (\text{III.17})$$

Les entrées  $u1$  et  $u2$  peuvent être définies de la manière suivante :

$$u1 = \dot{v} = \frac{\dot{f}_1 \ddot{f}_1 + \dot{f}_2 \ddot{f}_2}{\sqrt{\dot{f}_1^2 + \dot{f}_2^2}} \quad (\text{III.18})$$

$$u2 = \dot{\theta} = \frac{\dot{f}_1 \ddot{f}_2 - \dot{f}_2 \ddot{f}_1}{\dot{f}_1^2 + \dot{f}_2^2} \quad (\text{III.19})$$

### III.2. Vérification de sorties plates

Nous voulons savoir si le couple  $(F_1, F_2)$  peut jouer le rôle de sorties plates d'un système possédant un état de dimension  $n$ .

$$f_1 = x$$

$$\dot{f}_1 = v \cos \theta$$

$$\ddot{f}_1 = \dot{v} \cos \theta - \dot{\theta} v \sin \theta = u1 \cos \theta - u2 v \sin \theta \quad (\text{III.20})$$

on obtient  $n_1 = 2$

$$f_2 = y$$

$$\dot{f}_2 = v \sin \theta$$

$$\ddot{f}_2 = \dot{v} \sin \theta + \dot{\theta} v \cos \theta = u1 \sin \theta + u2 v \cos \theta \quad (\text{III.21})$$

on obtient  $n_2 = 2$

Donc :

$$n_1 + n_2 = n = 4$$

Le système admet  $(f_1, f_2)$  pour sorties plates.

### III.3. Simulation Matlab pour une trajectoire circulaire

Pour analyser le comportement du système, une simulation a été réalisée sur MATLAB en utilisant les sorties planes calculées précédemment. Les paramètres du modèle cinématique du robot mobile ont été définis, et les conditions initiales ont été fixées pour la position en x, la position en y, la vitesse et l'angle. La durée de la simulation a été spécifiée, ainsi que le pas de temps, voire la figure (40) représente le diagramme flux processus.

On a pris en considération les caractéristiques de la trajectoire circulaire définie comme suit :

- Rayon = 5;
  - Acceleration angular = 2;
  - $f_{1\text{ désirée}} = \text{Rayon} \times \cos(\text{acceleration\_angulaire} \times t)$
  - $f_{2\text{ désirée}} = \text{Rayon} \times \sin(\text{acceleration\_angulaire} \times t)$
- (III.22)

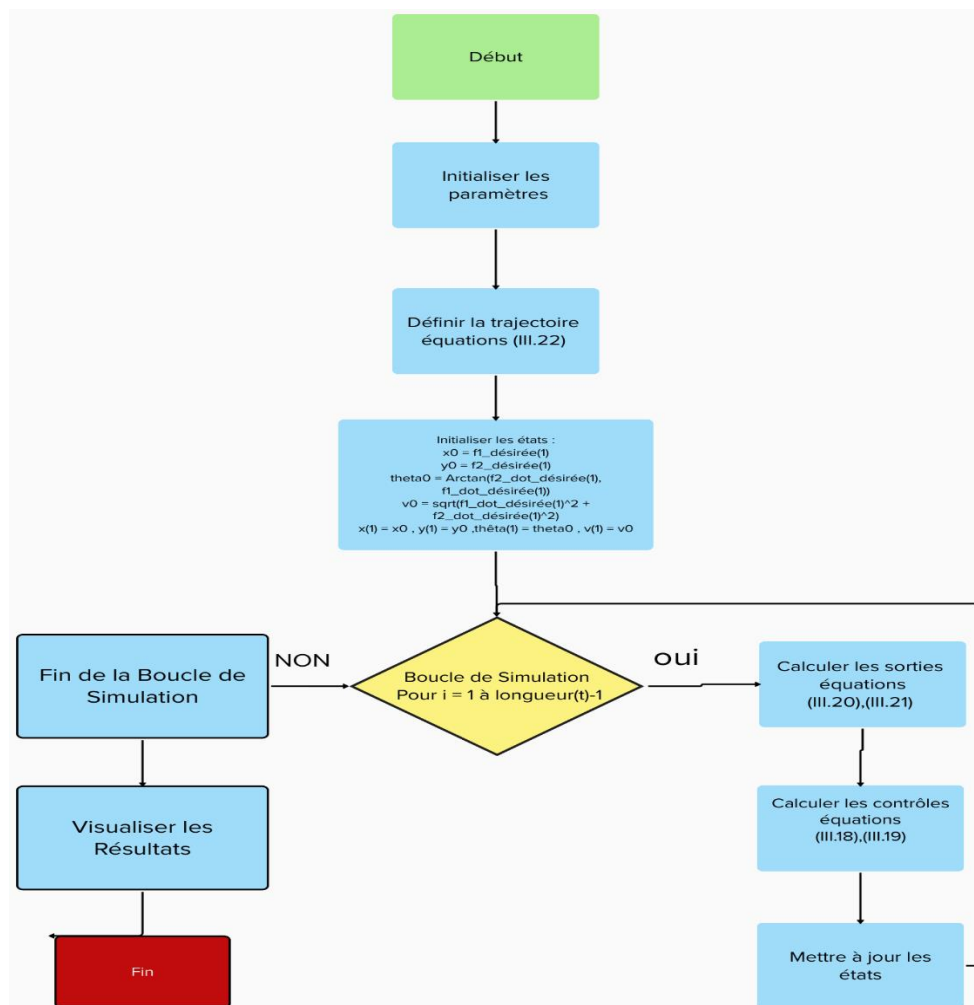
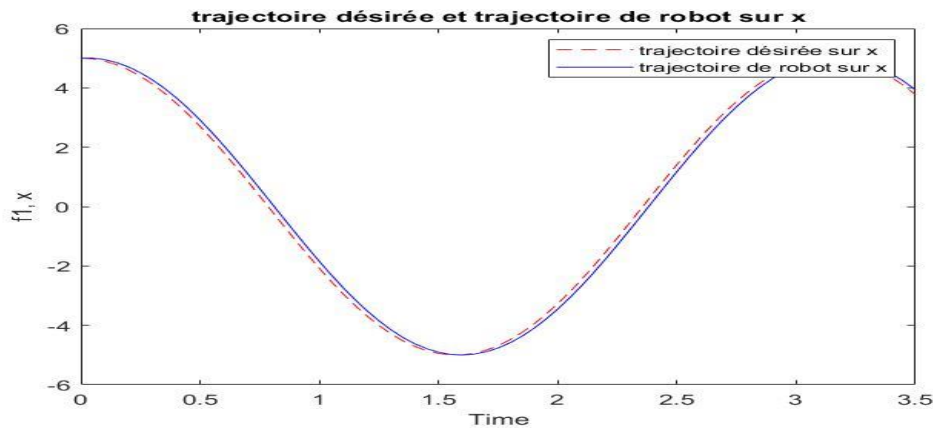


Figure 40 : Diagramme de flux processus commande par platitude

Les figures suivantes présentent les résultats de la simulation du système de contrôle basé sur la planéité du robot mobile qui suit une trajectoire circulaire. Chaque figure représente l'évolution d'une variable spécifique au fil du temps.

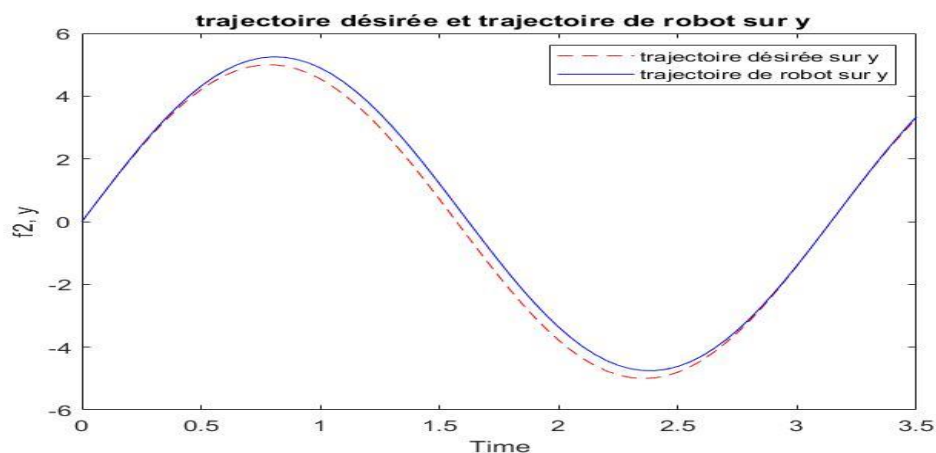


**Figure 41 : Trajectoire désirée  $f1$  et trajectoire de robot sur  $x$**

La figure (41) représente le tracé des fonctions  $f1$  et  $x$  en fonction du temps. La fonction  $f1$  représente la trajectoire désirée sur l'axe  $x$ , tandis que la fonction  $x$  représente la trajectoire réelle suivie par le robot. On peut observer que le robot parvient à suivre la trajectoire désirée avec une erreur négligeable.

L'axe des abscisses (temps en seconde) indique la progression du temps, tandis que l'axe des ordonnées ( $f1$  et  $x$ ) représente la position sur l'axe  $x$ . La fonction  $f1$  décrit le chemin que le robot est censé suivre, et la fonction  $x$  illustre la trajectoire réelle suivie par le robot.

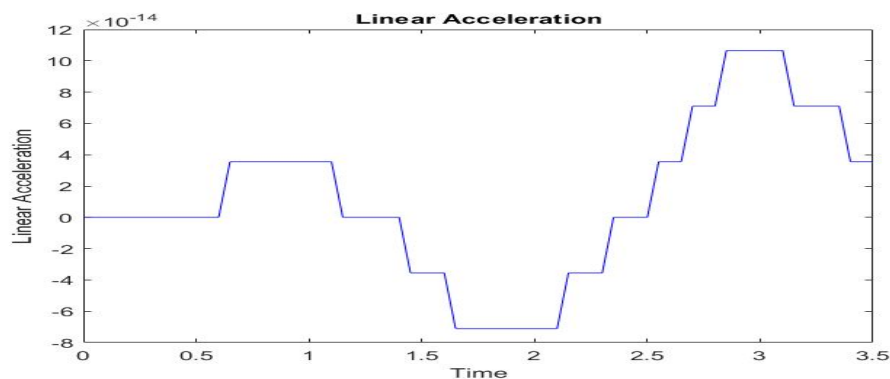
Cette figure montre que le robot est capable de suivre la trajectoire souhaitée avec une grande précision.



**Figure 42 : Trajectoire désirée  $f2$  et trajectoire de robot sur  $y$**

La figure (42) représente le tracé des fonctions  $f_2$  et  $y$  en fonction du temps. La fonction  $f_2$  représente la trajectoire désirée sur l'axe  $y$ , tandis que la fonction  $y$  représente la trajectoire réelle suivie par le robot. On peut observer que le robot parvient à suivre la trajectoire désirée avec une erreur négligeable.

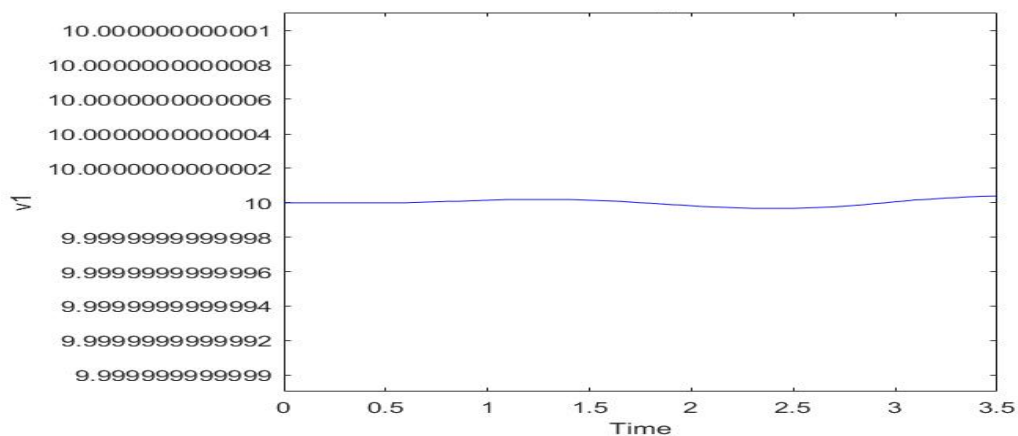
L'axe des abscisses (temps en seconde) indique la progression du temps, tandis que l'axe des ordonnées ( $f_2$  et  $y$ ) représente la position sur l'axe  $y$ . La fonction  $f_2$  décrit le chemin que le robot est censé suivre, et la fonction  $y$  illustre la trajectoire réelle suivie par le robot.



**Figure 43 : Accélération linéaire de robot**

La figure (43) représente le graphique de l'accélération du robot en fonction du temps. L'axe des abscisses (temps) indique la progression du temps, tandis que l'axe des ordonnées représente l'accélération du robot.

En observant la figure, on peut noter que l'accélération du robot varie au fil du temps. Il peut y avoir des périodes où l'accélération est constante, indiquant un mouvement uniformément accéléré, et d'autres moments où l'accélération peut changer de manière non linéaire, suggérant des changements de vitesse ou de direction.

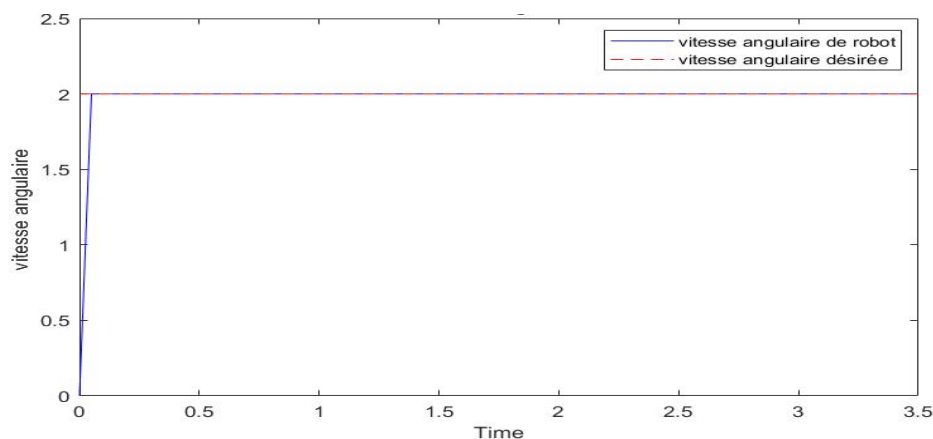


**Figure 44 : Vitesse linéaire de robot**

La figure (44) représente le graphique de la vitesse linéaire du robot en fonction du temps. L'axe des abscisses (temps) indique la progression du temps, tandis que l'axe des ordonnées représente la vitesse linéaire du robot.

En observant la figure, on peut constater que la vitesse linéaire du robot varie peu au fil du temps. Elle reste presque constante sur toute la période de temps étudiée. Cela suggère que le robot se déplace à une vitesse relativement stable et uniforme, sans subir de variations significatives de sa vitesse pendant cette période.

Une vitesse linéaire presque constante peut indiquer que le robot suit une trajectoire régulière et sans à-coups. Il est important de noter que la stabilité de la vitesse linéaire peut être un résultat d'un contrôle bien conçu et efficace du robot.

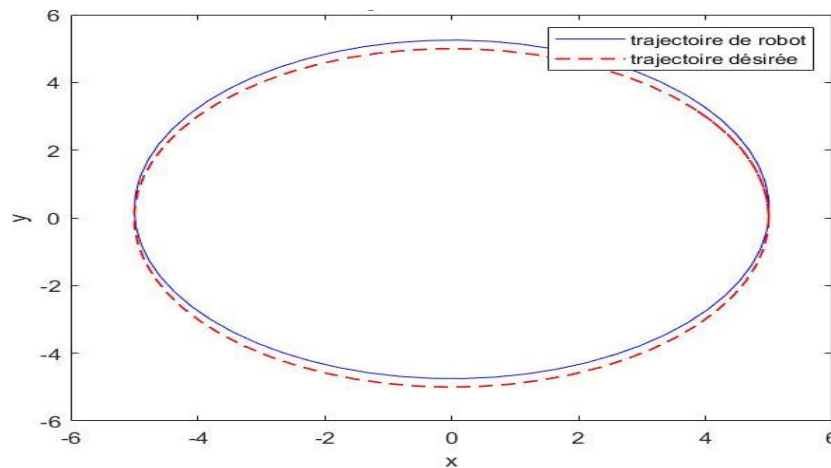


**Figure 45 : Vitesse angulaire**

La figure (45) représente le graphique de la vitesse angulaire du robot en fonction du temps. L'axe des abscisses (temps) indique la progression du temps, tandis que l'axe des ordonnées représente la vitesse angulaire du robot.

D'après la description, on peut observer deux phases distinctes dans le graphique. Dans la première phase, qui dure environ 0.1 seconde, la vitesse angulaire du robot augmente rapidement pour atteindre la vitesse angulaire désirée. Cette phase indique que le robot a réussi à atteindre la vitesse angulaire souhaitée en un court laps de temps, ce qui démontre une capacité de réactivité et de contrôle dynamique.

Dans la deuxième phase, après avoir atteint la vitesse angulaire désirée, la courbe de la vitesse angulaire reste constante jusqu'à la fin de la simulation. Cela suggère que le robot est capable de maintenir de manière précise et stable la vitesse angulaire souhaitée tout au long de cette période de temps.



**Figure 46 : Trajectoire de robot et trajectoire désirée**

La figure (46) représente les trajectoires du robot et de la trajectoire désirée en fonction du temps. Les courbes des deux trajectoires sont très proches l'une de l'autre, ce qui indique que le robot suit presque exactement la trajectoire désirée.

L'erreur entre la trajectoire du robot et la trajectoire désirée est négligeable, ce qui signifie que le robot est capable de suivre la trajectoire souhaitée avec une précision élevée.

### III.4. Générer une trajectoire avec courbe de Bézier

Une courbe de Bézier est une courbe lisse définie par un ensemble de points de contrôle. Elle tire son nom du mathématicien français Pierre Bézier, qui a développé ces courbes dans les années 1960 lorsqu'il travaillait chez Renault pour concevoir des carrosseries de voiture.

Une courbe de Bézier est généralement représentée en utilisant des polynômes de Bernstein, qui sont des polynômes spéciaux qui permettent d'interpoler les points de contrôle de manière douce et régulière.

#### III.4.1. Les polynômes de Bernstein

Également connue sous le nom de système de Bernstein, est un ensemble des polynômes utilisé pour représenter les courbes ou les surfaces dans le domaine de l'interpolation et de l'approximation. Les polynômes de Bernstein forment une base polynomiale pour les polynômes de degré  $n$  sur l'intervalle  $[0, 1]$

Ce qui signifie qu'ils sont linéairement indépendants et peuvent représenter n'importe quel polynôme de degré  $n$  sur cet intervalle. Ils sont également positifs et la somme de

tous les polynômes de Bernstein pour un degré donné est égale à 1 sur cet intervalle, ce qui en fait une base utile pour l'interpolation et l'approximation.

Les polynômes de Bernstein de degré  $n$  sur l'intervalle  $[0, 1]$  sont définis explicitement comme suit :

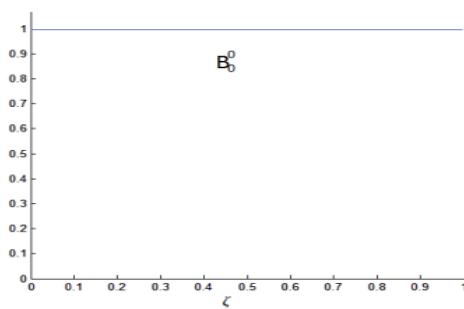
$$B_n^i(t) = C_n^i t^i (1-t)^{n-i} \quad \forall i = 0, \dots, n \quad (\text{III.22})$$

Avec les coefficients binomiaux  $C_n^i$  donnés par :

$$C_n^i = \begin{cases} \frac{n!}{i!(n-i)!} & 0 \leq i \leq n \\ 0 & \text{si non} \end{cases} \quad (\text{III.23})$$

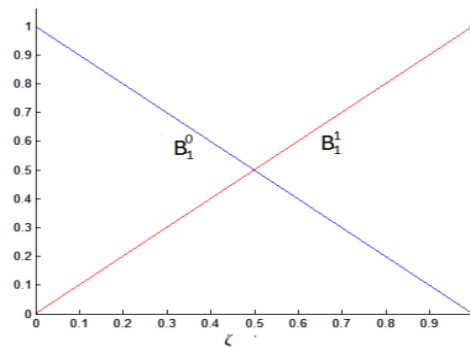
Un exemple de polynômes de Bernstein constants, linéaires, quadratiques et cubiques est présenté dans les figures suivantes.

$$B_0^0(t) = 1$$



**Figure 47 : Polynôme de Bernstein constant (degré 0)**

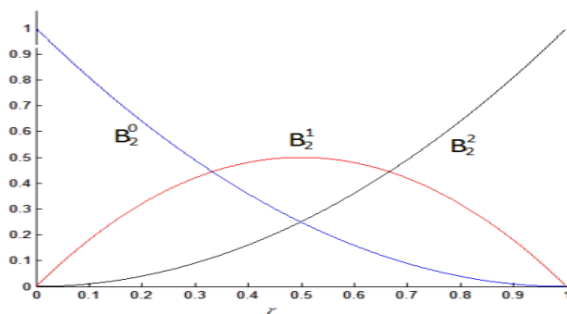
$$B_1^0(t) = (1-t), B_1^1(t) = t$$



**Figure 48 : Polynôme de Bernstein Linéaire (degré 1)**

$$B_2^0(t) = (1-t)^2, B_2^1(t) = 2t(1-t),$$

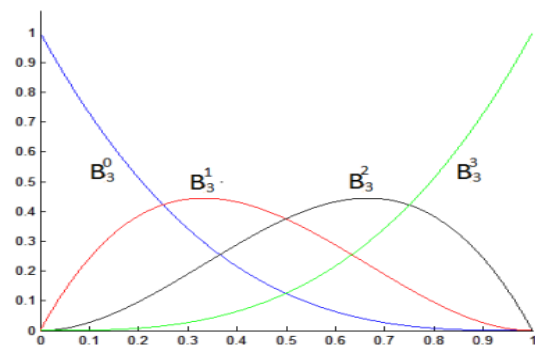
$$B_2^2(t) = t^2$$



**Figure 49 : Polynôme de Bernstein quadratique (degré 2)**

$$B_3^0(t) = (1-t)^3, B_3^1(t) = 3t(1-t)^2,$$

$$B_3^2(t) = 3t^2(1-t), B_3^3(t) = t^3$$



**Figure 50 : Polynôme de Bernstein cubique (degré 3)**

### III.5. Courbe de Bézier

- Définition : Étant donné  $(n + 1)$  points distincts  $P_0, P_1, \dots, P_n$  dans l'espace, la courbe de Bézier (de degré  $n$ ) définie à partir de ces points est la courbe paramétrique  $C_n$  définie par :

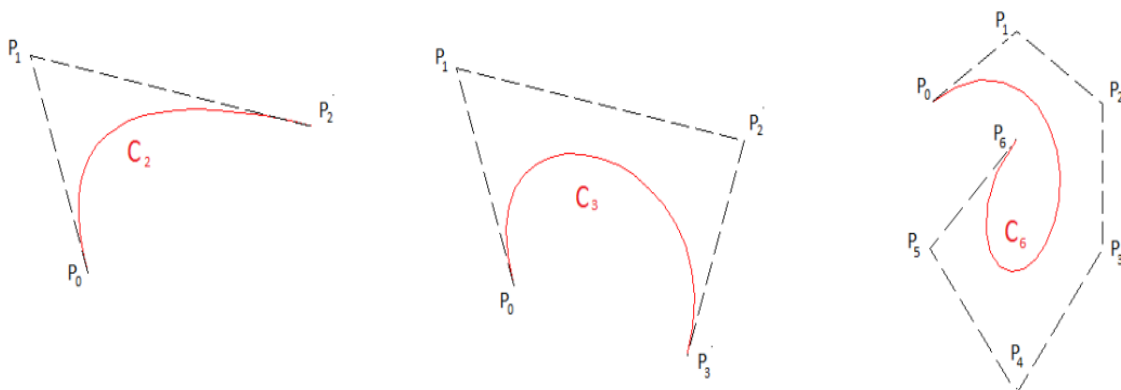
$$C_n = \sum_{i=0}^n B_n^i(t) P_i = 0$$

Où  $t$  est un paramètre dans l'intervalle  $[0, 1]$ ,

$B_n^i(t)$  sont les polynômes de Bernstein correspondants à l'indice  $i$  et au degré  $n$ , et  $P_i$  sont les points de contrôle correspondants.

La courbe de Bézier  $C_n$  est obtenue en interpolant les points de contrôle  $P_0, P_1, \dots, P_p$  en utilisant les polynômes de Bernstein comme fonctions de base. Cette courbe est définie pour  $t$  allant de 0 à 1, et elle relie les points de contrôle de manière fluide et continue.

Les courbes de Bézier de différents degrés sont présentées dans la figure (51) :



*Figure 51 : Les courbes de Bézier de différents degrés*

### III.6. Simulation Matlab pour une trajectoire générée par la courbe de Bézier

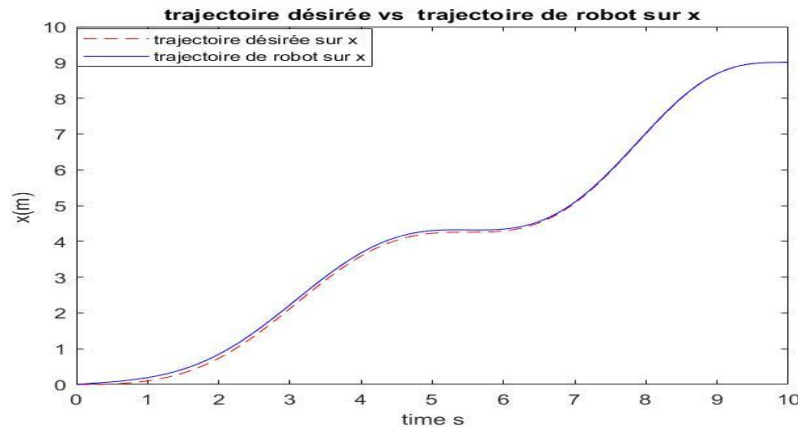
Dans cette section nous allons générer une trajectoire en utilisant la méthode de courbe de Bézier et tenter de la suivre avec le robot mobile contrôlé par la commande platitude, nous avons réalisé une simulation avec logiciel MATLAB, et la trajectoire à suivre est une trajectoire d'ordre 12 est on va prendre les points de contrôle comme suit :



$P_{x0}=0$  ;  $P_{x1}=0$  ;  $P_{x2}=0$  ;  $P_{x3}=1$  ;  $P_{x4}=0$  ;  $P_{x5}=11.6532$  ;  $P_{x6}=7.0979$  ;  $P_{x7}=-6.3543$  ;  
 $P_{x8}=8.0$  ;  $P_{x9}=9$  ;  $P_{x10}=9.0000$  ;  $P_{x11}=9.0000$  ;

$P_{y0}=0$  ;  $P_{y1}=0$  ;  $P_{y2}=0$  ;  $P_{y3}=0$  ;  $P_{y4}=-5$  ;  $P_{y5}=5.2719$  ;  $P_{y6}=2.5352$  ;  $P_{y7}=7.2342$  ;  $P_{y8}=7$  ;  
 $P_{y9}=7$  ;  $P_{y10}=7$  ;  $P_{y11}=7$  ;

Et les résultats sont illustrés ci-dessous par les figures suivantes :

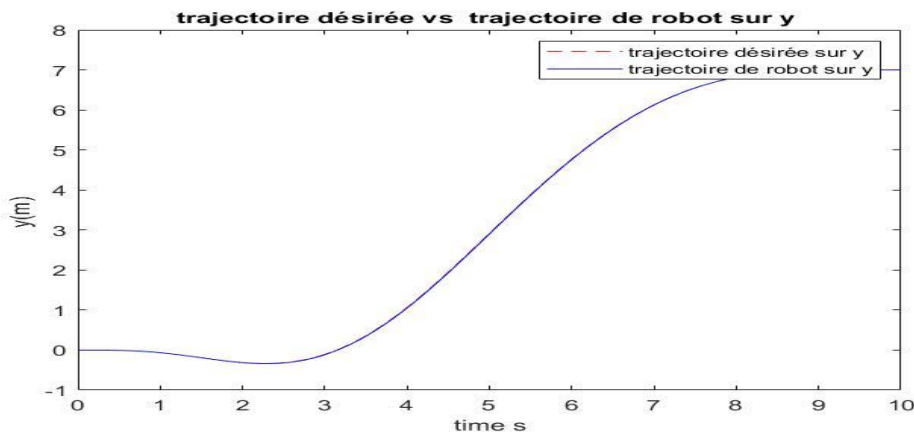


**Figure 52 : Trajectoire désirée et trajectoire de robot sur l'axe x**

La figure (52) montre deux trajectoires sur l'axe x : la trajectoire désirée générée par une courbe de Bézier et la trajectoire réelle du robot au fil du temps.

La trajectoire désirée est la trajectoire théorique que le robot doit suivre, générée par une courbe de Bézier. Cette courbe est définie par des points de contrôle qui permettent de spécifier la courbure et l'orientation de la trajectoire souhaitée.

La trajectoire réelle du robot représente son déplacement effectif dans l'environnement au fil du temps, le long de l'axe x. Cette trajectoire est le résultat de l'exécution du contrôle et du suivi de la trajectoire désirée par le robot

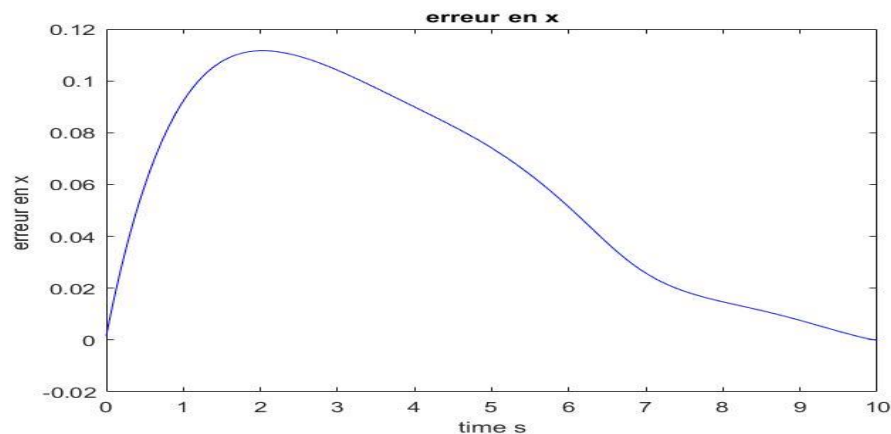


**Figure 53 : Trajectoire désirée et trajectoire de robot sur l'axe y**

La figure (53) montre deux trajectoires sur l'axe y : la trajectoire désirée générée par une courbe de Bézier et la trajectoire réelle du robot au fil du temps.

La trajectoire désirée est la trajectoire théorique que le robot doit suivre, générée par une courbe de Bézier. Cette courbe est définie par des points de contrôle qui permettent de spécifier la courbure et l'orientation de la trajectoire souhaitée.

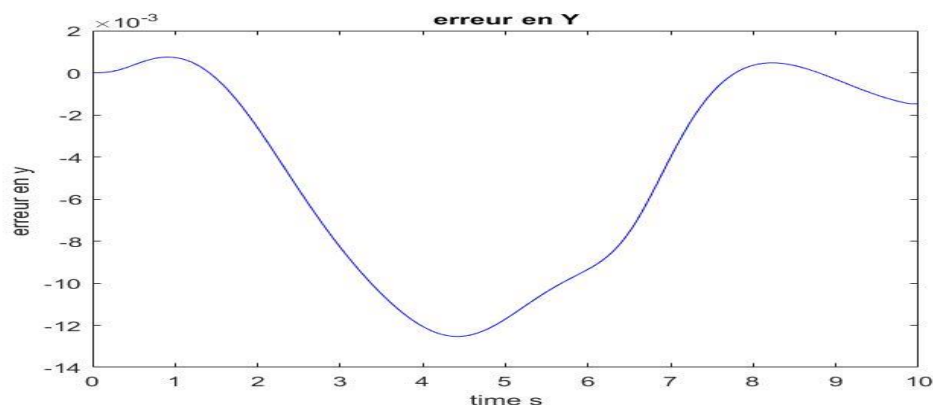
La trajectoire réelle du robot représente son déplacement effectif dans l'environnement au fil du temps, le long de l'axe y. Cette trajectoire est le résultat de l'exécution du contrôle et du suivi de la trajectoire désirée par le robot.



**Figure 54 : Erreur sur l'axe x**

La figure (54) représente l'erreur sur l'axe x entre la trajectoire désirée générée par une courbe de Bézier et la trajectoire réelle suivie par le robot au fil du temps.

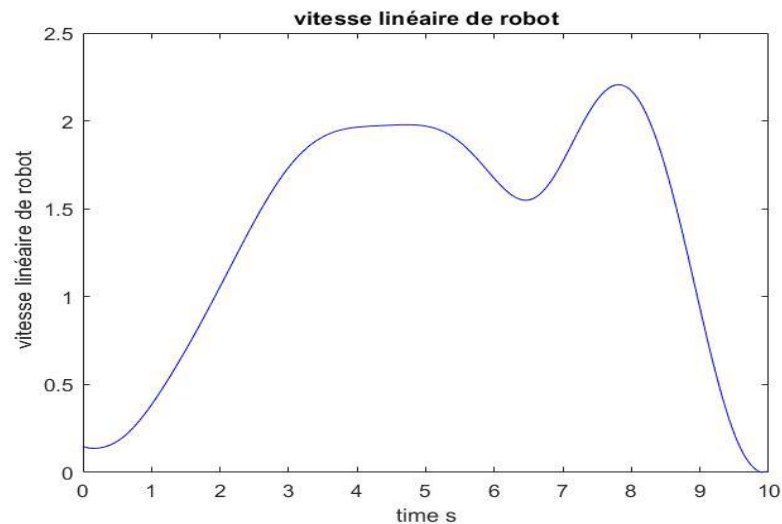
L'erreur sur l'axe x est la différence entre la position souhaitée du robot, telle que définie par la courbe de Bézier, et sa position réelle le long de l'axe x à chaque instant de temps. Cette erreur peut être positive ou négative, en fonction de la déviation du robot par rapport à la trajectoire désirée



**Figure 55 : Erreur sur l'axe y**

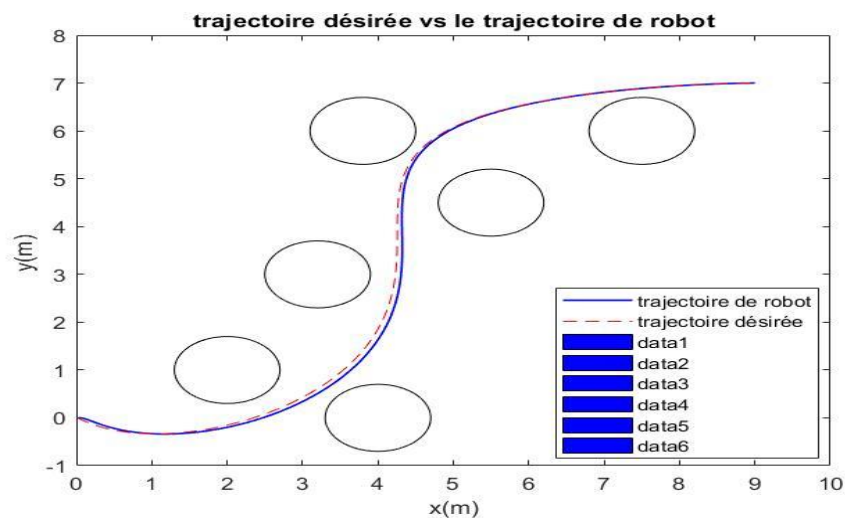
La figure (55) représente l'erreur sur l'axe y entre la trajectoire désirée générée par une courbe de Bézier et la trajectoire réelle suivie par le robot au fil du temps.

L'erreur sur l'axe y est la différence entre la position souhaitée du robot, telle que définie par la courbe de Bézier, et sa position réelle le long de l'axe y à chaque instant de temps. Cette erreur peut être positive ou négative, en fonction de la déviation du robot par rapport à la trajectoire désirée



**Figure 56 : Vitesse linéaire de robot**

La figure (56) représente la vitesse linéaire du robot au cours du temps. La vitesse linéaire est la vitesse à laquelle le robot se déplace dans une direction donnée sans prendre en compte sa direction. Elle est généralement mesurée en unités de longueur par unité de temps



**Figure 57 : Trajectoire désirée et trajectoire de robot**

La figure (57) représente la comparaison entre la trajectoire désirée générée par une courbe de Bézier et la trajectoire réelle du robot, en remarque que l'erreur est négligeable.

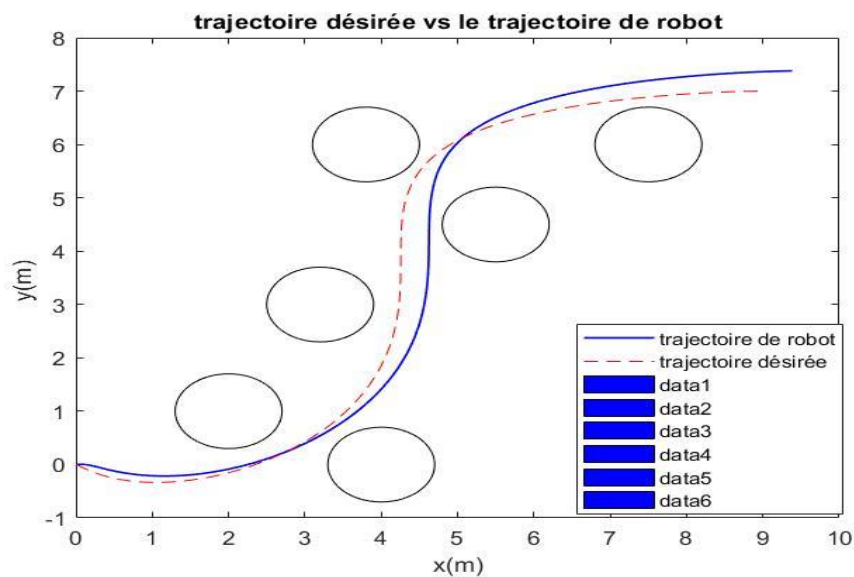
#### IV. Interprétation générale

En analysant les résultats de la simulation, nous pouvons conclure que le robot a suivi les trajectoires avec succès en utilisant la commande par platitude.

Ces résultats démontrent l'efficacité de la commande à platitude pour le contrôle précis du robot mobile dans des conditions idéales, sans perturbations. Cela ouvre des perspectives prometteuses pour l'application de cette approche dans des scénarios réels.

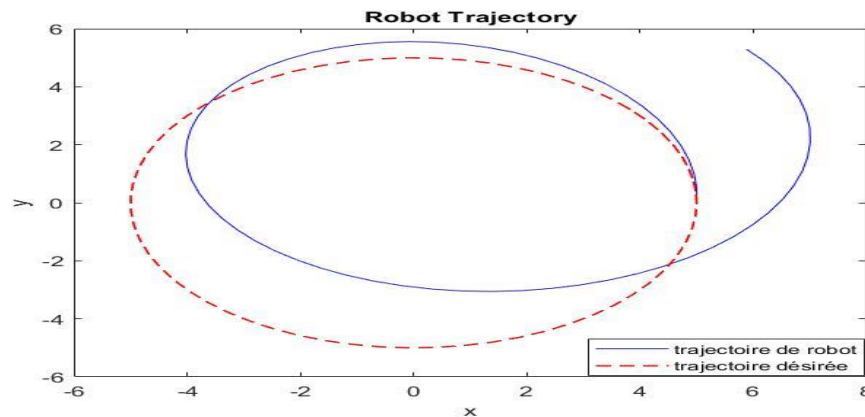
La simulation en présence d'incertitudes est une étape importante pour évaluer la robustesse et les performances du système de contrôle du robot mobile face aux variations et aux perturbations.

#### V. Simulation de deux trajectoires en présence de perturbation



**Figure 58 : Trajectoire désirée et trajectoire de robot avec perturbation**

La figure (58) représente la trajectoire prévue pour un robot définie par courbe de Bézier, mais en présence de perturbations, le robot s'éloigne de son chemin souhaité. Cela montre que les perturbations peuvent avoir un impact négatif sur la capacité du robot à suivre sa trajectoire prévue, ce qui peut entraîner des erreurs dans son mouvement.



*Figure 59 : Trajectoire de robot2D avec perturbation*

De même pour La figure (59) représente la trajectoire circulaire prévue pour un robot, mais en présence de perturbations, le robot s'éloigne de son chemin souhaité.

L'interprétation des figures (58) et (59) semble indiquer que, lorsque le système est soumis à des perturbations, la trajectoire du robot s'éloigne de la trajectoire définie dans le cas de la commande en platitude.

Cela peut suggérer que la commande en platitude seule n'est pas suffisamment robuste pour maintenir le robot sur la trajectoire désirée lorsque des perturbations externes sont présentes. Les perturbations peuvent provenir de facteurs tels que des variations imprévues dans l'environnement, des erreurs de mesure, des conditions météorologiques changeantes, etc.

## VI. Conclusion

En conclusion, ce chapitre a présenté deux concepts clés : la commande par platitude et les courbes de Bézier. La commande par platitude simplifie la commande des systèmes complexes en transformant les variables d'état en variables de sortie directement contrôlables. Les courbes de Bézier sont utilisées pour représenter des trajectoires lisses grâce à des points de contrôle et des polynômes de Bernstein.

Dans le prochain chapitre, nous aborderons la commande robuste basée sur la Sliding Mode Control (SMC) pour améliorer les performances du système dans des conditions réelles. Cette approche permettra de garantir la stabilité et les performances du système face à des perturbations et des variations des conditions de fonctionnement.

# Chapitre 4 : Commande non-linéaire robuste

## I. Introduction

Dans ce chapitre, nous aborderons le concept général de la commande par mode glissant. Cette approche fait partie d'une classe plus large appelée commande à structure variable. La commande par mode glissant a été largement étudiée et développée depuis son introduction.

L'objectif de la commande par mode glissant est d'utiliser une commande discontinue pour amener le système à évoluer vers une surface appelée surface de glissement dans un temps fini, et de maintenir le comportement du système sur cette surface, correspondant aux dynamiques souhaitées.

## II. Définition de commande par mode glissant

Un système à structure variable est un système dont la configuration ou la structure peut changer pendant son fonctionnement. Cette caractéristique est rendue possible par l'utilisation d'une logique de commutation qui permet au système de passer d'une structure à une autre à tout moment. Cette flexibilité de commutation entre différentes structures confère au système de nouvelles propriétés qui peuvent ne pas être présentes dans chaque structure individuelle [9].

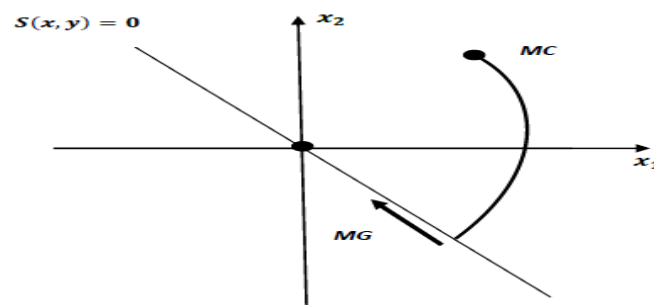
Chaque structure dans un système à structure variable peut avoir ses propres caractéristiques, dynamiques et modes de fonctionnement. La commutation entre ces structures permet au système d'exploiter différentes propriétés et fonctionnalités en fonction des besoins ou des conditions rencontrées.

La commande par mode de glissement est une méthode de commande qui vise à amener la trajectoire d'état d'un système vers une surface spécifique, et ensuite, en utilisant une loi de commutation appropriée, à maintenir cette trajectoire à proximité de la surface de glissement. Les mouvements qui se produisent dans cette région sont communément appelés mouvements de glissement.

Dans cette approche, la surface de glissement est choisie de manière à ce que les propriétés dynamiques souhaitées soient atteintes lorsque le système évolue sur cette surface. La loi de commutation détermine comment le système doit passer d'un mode de glissement à un autre en fonction de la position de la trajectoire d'état par rapport à la surface de glissement.

La figure (60) décrit les différentes parties de la trajectoire dans le plan de phase

- Le mode de convergence (MC) : est le mode durant lequel la variable à régler se déplace à partir de n'importe quel point initial dans le plan de phase  $(x_1, x_2)$ , est tend vers la surface de commutation  $S(x_1, x_2) = 0$ . Ce mode est caractérisé par une loi de commande discontinue et le critère de convergence.
- Le mode de glissement (MG) : est le mode durant lequel la variable d'état 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  $S(x_1, x_2) = 0$ .
- Le mode de régime permanent (MRP) : ce mode est ajouté pour l'étude de la réponse du système autour de son point d'équilibre, il est caractérisé par la qualité et les performances de la commande.



*Figure 60 : Différents modes pour la trajectoire dans le plan de phase*

### III. Conception de la commande par mode glissant

La commande par mode glissant, à des avantages importants et multiples, tels que la haute précision, la bonne stabilité, la simplicité de conception, la robustesse, etc. Ceci lui permet d'être particulièrement adaptée pour les systèmes ayant un modèle imprécis.

Dans ce cas, la structure d'un contrôleur comporte deux parties : une partie continue représentant la dynamique du système durant le mode glissant et une autre discontinue représentant la dynamique du système durant le mode de convergence. Ce dernier est important dans la commande non linéaire, elle a pour rôle d'éliminer ou réduire les effets d'imprécisions et des perturbations sur le modèle.

Pour illustrer la méthode de conception de la commande par mode glissant d'ordre un, nous considérons le cas d'un système non linéaire [9].

$$\dot{x}(t) = f(x) + g(x) \cdot U \quad (\text{IV.1})$$

Avec  $x \in R^n$ : est le vecteur d'état.

$u \in R^m$ : est le vecteur de commande.

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

- Choix de la surface de glissement.
- L'établissement des conditions d'existence et de convergence.
- Détermination de la loi de commande.

### III.1. Choix de la surface de glissement

La surface de glissement ou la fonction de commutation (switching function) représente le comportement dynamique désiré du système en boucle fermée. Elle peut être choisie linéaire ou non linéaire.

L'une des formes que la surface peut prendre est donnée par [9] :

$$S(x) = \left( \frac{\partial}{\partial t} + \lambda \right)^{r-1} e(x) \quad (\text{IV.2})$$

$\lambda$  : Constante positive.

$r$  : est le degré relatif, il représente le nombre de fois qu'il faut dériver la surface pour faire apparaître la commande.

$e(x) = x - xd$ : est l'écart entre la variable à régler et sa référence.

$x$  : variable d'état.

$xd$  : valeur désirée.

Une autre forme de la surface est la suivante [9] :

$$S(x) = \sum_{i=1}^n c_i (x_i - x_i d) \quad (\text{IV. 3})$$

La dynamique de convergence est assurée par les choix des coefficients  $c_i$ .

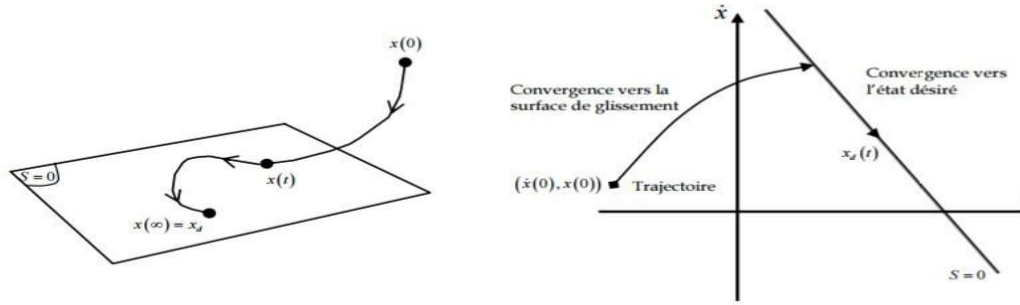
L'objectif de la commande est de maintenir la surface à zéro ( $S(x)=0$ ), dont l'unique solution est ( $e(x) = 0$ ) pour un choix convenable des paramètres.

### III.2. Conditions de convergence et d'existence

Les conditions de convergence et d'existence sont des critères importants pour garantir que les dynamiques d'un système convergent vers la surface de glissement et y restent malgré les perturbations. Ces conditions assurent la convergence et la stabilité du système (figure 61).



Il existe deux conditions générales pour assurer ce mode de convergence [9].



**Figure 61 : Convergence du système glissant.**

### III.2.1. La fonction discrète de commutation

C'est la première condition de convergence, elle est proposée par Emilyanov et Utkin. Il s'agit de donner à la surface une dynamique convergente vers zéro. Elle est donnée par [9].

$$\dot{s}(x) > 0 \quad \text{si} \quad s(x) < 0 \quad (\text{IV.4})$$

$$\dot{s}(x) < 0 \quad \text{si} \quad s(x) > 0$$

Cette condition peut être formulée comme suit :

$$\dot{s}(x)s(x) < 0 \quad (\text{IV.5})$$

### III.2.2. La fonction de Lyapunov

L'idée consiste à choisir une fonction scalaire  $S(x)$  appropriée pour assurer l'attraction de la variable à contrôler vers sa valeur de référence, et à concevoir la commande  $u$  en conséquence. Une approche courante pour analyser la stabilité d'un système est d'utiliser une fonction de Lyapunov [9].

$$v(x) = \frac{1}{2}s^2(x) \quad (\text{IV.6})$$

La dérivée de cette fonction est :

$$\dot{v}(x) = \dot{s}(x)s(x) \quad (\text{IV.7})$$

Pour que la fonction  $\dot{v}(x)$  puisse décroître, il suffit d'assurer que sa dérivée soit strictement négative. Ceci n'est vérifié que si la condition suivante est vérifiée :

$$\dot{v}(x) = \dot{s}(x)s(x) < 0 \quad (\text{IV.8})$$

L'équation (IV.8) explique que le carré de la distance entre un point donné du plan de phase et la surface de glissement exprimé par  $s^2(x)$  diminue tout le temps, contraignant

la trajectoire du système à se diriger vers la surface à partir des deux côtés de cette surface. Cette condition suppose un régime glissant idéal où la fréquence de commutation est infinie.

### III.2.3. Calcul de la commande

La commande nécessaire pour attirer la trajectoire d'état vers la surface de glissement et ensuite vers son point d'équilibre tout en maintenant les conditions d'existence du mode de glissement peuvent être obtenues en utilisant un contrôleur par mode de glissement.

Dans un contrôleur par mode de glissement, il est souvent nécessaire d'utiliser une commande discontinue pour parvenir à un régime de glissement. La surface de glissement doit être attractive des deux côtés, ce qui signifie que la trajectoire d'état doit converger vers la surface de glissement depuis n'importe quelle direction. Cependant, l'utilisation d'une commande discontinue ne signifie pas nécessairement qu'il n'y a pas de partie continue ajoutée.

En réalité, l'ajout d'une partie continue permet de réduire autant que nécessaire l'amplitude de la partie discontinue. Cela permet de minimiser les effets indésirables d'une commande discontinue, tout en maintenant les propriétés d'attractivité de la surface de glissement.

Lorsqu'une perturbation est présente, la partie discontinue du contrôleur a principalement pour but de vérifier les conditions d'attractivité de la surface de glissement. Dans ce cas, la structure du contrôleur par mode de glissement se compose de deux parties : une partie de linéarisation ( $u_{eq}$ ) et une partie stabilisante ( $u_n$ ) [9].

On écrit : 
$$u = u_{eq} + u_n \quad (IV.9)$$

Tel que :

$u_{eq}$  : La commande proposée par Utkin, elle sert à maintenir la variable à contrôler sur la surface de glissement ( $x$ )=0, la commande équivalente est déduite en considérant que la dérivée de la surface est nulle  $\dot{s}(x) = 0$ . Elle peut être interprétée comme étant un retour d'état particulier jouant le rôle d'un signal de commande appliqué sur le système à commander, elle peut être comme une valeur moyenne que prend la commande lors de la commutation rapide entre les valeurs  $u_{max}$  et  $u_{min}$

$u_n$  : est déterminée pour vérifier la condition de convergence.

Pour mettre en évidence le calcul de la commande, nous considérons un système défini dans l'espace d'état par l'équation (IV.1). Il s'agit de trouver l'expression de la commande  $u$ .

$$\dot{s}(x) = \frac{\partial s}{\partial t} = \frac{\partial s}{\partial x} \frac{\partial x}{\partial t} \quad (\text{IV.10})$$

Il en résulte :

$$\dot{s}(x) = \frac{\partial s}{\partial x} (f(x) + g(x)u_G) + \frac{\partial s}{\partial x} g(x)u_n \quad (\text{IV.11})$$

Durant le mode de glissement et le régime permanent, la surface de glissement atteint zéro, ce qui signifie que sa dérivée est également nulle. Par conséquent, la partie discontinue de la commande devient nulle à ce stade. Ainsi, nous pouvons déduire l'expression de la commande équivalente dans cette situation.

$$u_{eq} = - \left( \frac{\partial s}{\partial x} g(x) \right)^{-1} \frac{\partial s}{\partial x} f(x) \quad (\text{IV.12})$$

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

$$\frac{\partial s}{\partial x} g(x) u_n \neq 0$$

Durant le mode de convergence, et en remplaçant la commande équivalente par son expression dans (IV.11) nous trouvons la nouvelle expression de la dérivée de la surface :

$$\dot{s}(x) = \frac{\partial s}{\partial x} g(x) u_n \quad (\text{IV.13})$$

Et la condition d'attractivité exprimée par (IV.6) devient :

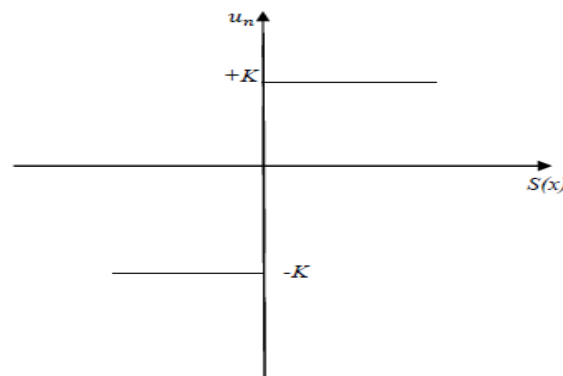
$$\frac{\partial s}{\partial x} g(x) u_n S(x) < 0 \quad (\text{IV.14})$$

Afin de satisfaire cette condition le signe de  $u_n$  doit être opposé à celui de  $\frac{\partial s}{\partial x} g(x) S(x)$

La forme la plus simple que peut prendre la commande discontinue est celle d'un relais [9]. (Figure 62).

$$u_n = K \text{sing } s(x) \quad (\text{IV.15})$$

Le signe de K doit être différent de celui de  $\frac{\partial s}{\partial x} g(x)$ .



*Figure 62 : Représentation de la commande discontinue*

### III.3. Phénomène de broutement

La commande discontinue  $u_n$ , en raison de sa fréquence de commutation finie, peut générer des oscillations à haute fréquence dans un voisinage de la surface de glissement, ce phénomène est connu sous le nom de réticence ou chattering en anglais. C'est le principal inconvénient de la commande par mode glissant d'ordre un. Les oscillations à haute fréquence peuvent exciter des dynamiques non modélisées dans le système, ce qui peut conduire à l'instabilité du système.

Il est important de noter que le chattering peut entraîner des effets indésirables tels que des vibrations, des bruits et des contraintes mécaniques élevées. Ces oscillations à haute fréquence peuvent également affecter les performances du système de contrôle et la précision de suivi de la trajectoire.

Afin de minimiser les effets du chattering et d'améliorer la stabilité du système, différentes techniques peuvent être utilisées, telles que la fonction tanh, la fonction de saturation, la fonction arctan. Ces méthodes visent à réduire les oscillations à haute fréquence et à améliorer la convergence du système vers la surface de glissement de manière plus lisse et précise.

- La fonction de saturation

Elle consiste à remplacer la fonction  $\text{sig}(s)$  par la droite de pente  $1/\varepsilon$  ; à l'intérieur d'une bande de largeur  $2\varepsilon$  située de part et d'autre de la surface de glissement, la discontinuité étant conservée à l'extérieur de cette bande. Son expression est donnée par [10]. (Figure 64)

$$\text{sat}(s, \varepsilon) = \begin{cases} \frac{s}{\varepsilon} & \text{si } |s| \leq \varepsilon \\ \text{sing}(s) & \text{si } |s| > \varepsilon \end{cases}$$

- La fonction pseudo-signe [15]

$$v(s, \varepsilon) = \frac{s}{|s| + \varepsilon}$$

- La fonction avec tangente [15]

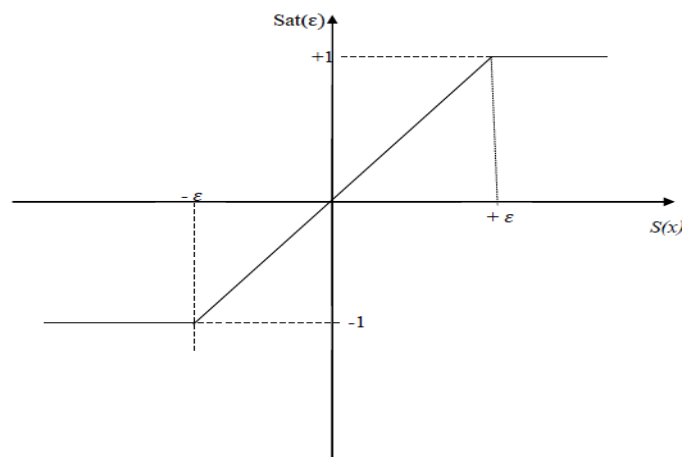
$$v(s, \varepsilon) = \frac{2}{\pi} \arctan\left(\frac{s}{\varepsilon}\right)$$

- La fonction tangente hyperbolique [15]

$$v(s, \varepsilon) = \tanh\left(\frac{s}{\varepsilon}\right)$$

- La fonction tangente sigmoïde

$$v(s) = \frac{1}{1 + e^{-2s}}$$



*Figure 63 : Représentation de la fonction Sat*

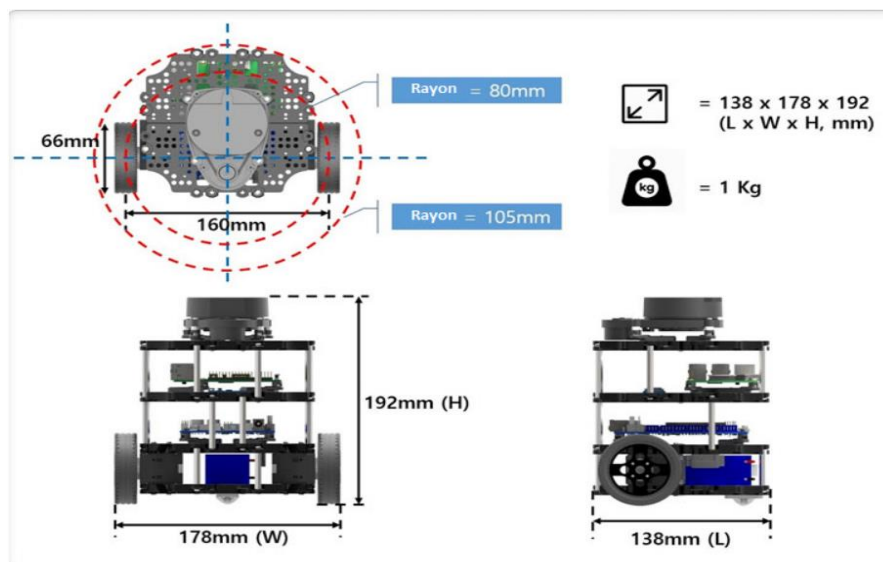
#### IV. Application de la commande non-linéaire robuste sur Turtlebot3

Le TurtleBot3 Burger est un robot mobile compact et abordable conçu à des fins éducatives, de recherche et de loisirs. Il fait partie de la série TurtleBot3 et offre une solution rentable tout en conservant les fonctionnalités essentielles de la plateforme TurtleBot3. Le modèle Burger comprend un ordinateur mono-carte, des capteurs et des actionneurs. La plateforme TurtleBot est d'une grande valeur dans le milieu académique pour tester des algorithmes de contrôle. Elle offre une option abordable et accessible aux chercheurs, étudiants et amateurs pour explorer et valider des algorithmes de contrôle sans avoir besoin de systèmes robotiques coûteux. L'intégration du TurtleBot avec le système d'exploitation Robot (ROS) améliore encore ses capacités en fournissant une

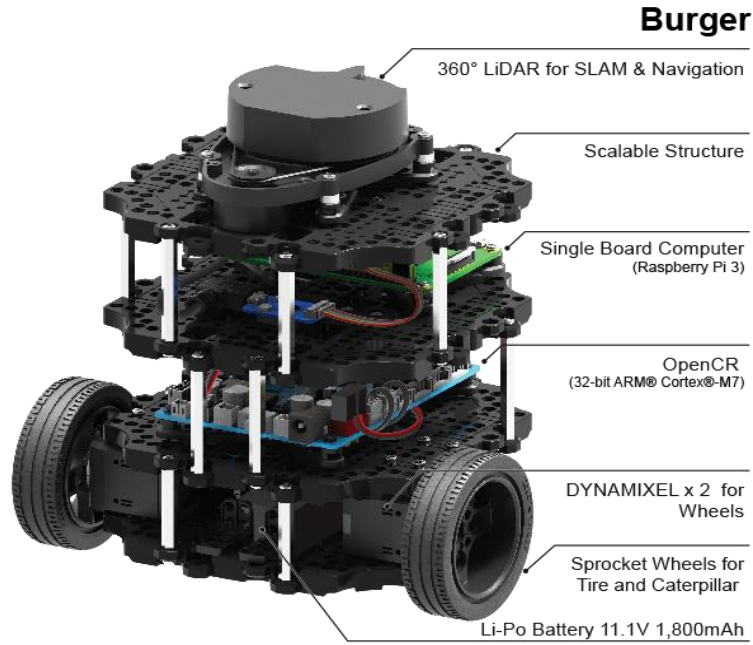
suite complète d'outils et de bibliothèques spécialement conçus pour développer et tester des algorithmes de contrôle. Cette intégration simplifie le processus de développement et permet aux utilisateurs de tirer parti de la fonctionnalité robuste de ROS et de son écosystème associé. Les composants détaillés sont illustrés dans la figure

En adaptant les composants matériels, les chercheurs peuvent configurer le TurtleBot selon leurs besoins expérimentaux spécifiques. En incorporant des capteurs, des actionneurs ou d'autres composants matériels supplémentaires, ils peuvent créer différentes configurations pour tester des algorithmes de contrôle dans divers scénarios. Cette flexibilité améliore la polyvalence de la plateforme et permet d'explorer un large éventail de stratégies de contrôle.

Le TurtleBot est équipé d'une gamme complète de capteurs, comprenant un capteur LiDAR à 360 degrés, une unité de mesure inertielle (IMU) et des codeurs de roue. Ces capteurs fournissent une perception précise de l'environnement et des informations en retour, offrant ainsi aux chercheurs des données riches et précises. Ces données constituent une ressource précieuse pour le développement et l'amélioration des algorithmes de contrôle qui exploitent efficacement les informations sensorielles disponibles sur la plateforme.



*Figure 64 : Le dimensionnement du TurtleBot*



*Figure 65 : Les composants principaux du TurtleBot3*

#### IV.1. Le modèle cinématique avec glissement

Pour tenir compte des conditions réelles, nous prenons en compte des incertitudes telles que le patinage et les perturbations environnementales externes lors de la description du modèle cinématique du WMR (robot mobile à roues). Par conséquent, le modèle est défini différemment, comme indiqué ci-dessous

$$\begin{aligned}\dot{x} &= v \cos(\theta) + v_t \cos(\theta) + v_s \sin(\theta) + p_x \\ \dot{y} &= v \sin(\theta) + v_t \sin(\theta) - v_s \cos(\theta) + p_y \\ \dot{\theta} &= \omega + \omega_s\end{aligned}\tag{IV.16}$$

Les variables  $p_x$  et  $p_y$  représentent les perturbations environnementales externes, indiquant les influences potentielles des conditions environnantes. D'autre part,  $v_t$  et  $v_s$  représentent les vitesses de glissement, où  $v_t$  désigne la vitesse de glissement dans la direction avant, et  $v_s$  représente la vitesse de glissement normale à celle-ci. De plus,  $\omega_s$  désigne la vitesse de glissement angulaire. Selon [11], on suppose que le phénomène de patinage peut être défini et borné de la manière suivante :

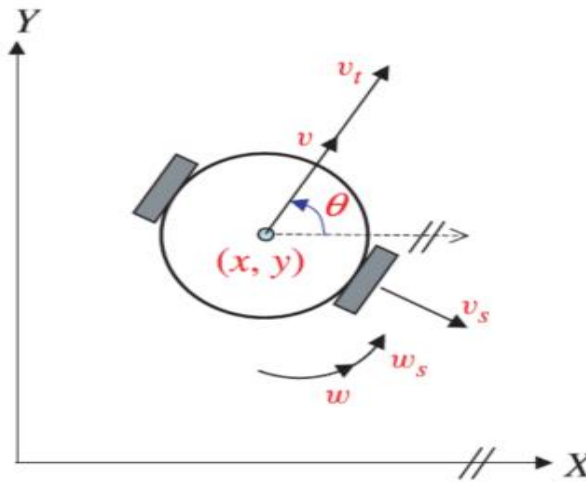
$$v_t(t) = v_s(t) = \omega_s(t) = k_1 v(t)\tag{IV.17}$$

$$\|v_t\| \leq \varepsilon_1 \|v\|, \|v_s\| \leq \varepsilon_2 \|v\|, \|\omega_s\| \leq \varepsilon_3\tag{IV.18}$$

où  $\kappa_1$ ,  $\varepsilon_1$ ,  $\varepsilon_2$  et  $\varepsilon_3$  sont des constantes positives.

En supposant que  $f_{xd}$  et  $f_{yd}$  sont les trajectoires de référence pour  $f_1$  et  $f_2$  respectivement, nous pouvons définir la dynamique de l'erreur comme  $e_i = f_i - f_{id}$  pour  $i = 1, 2$ . Afin d'obtenir la convergence de l'erreur de suivi  $e_i$  vers zéro en présence d'incertitudes, nous utilisons une approche de contrôle par mode glissant qui repose sur les principes de la loi de planéité. En incorporant cette stratégie de contrôle, nous visons à assurer des performances de suivi robustes et précises même en présence d'incertitudes du système.

La conception du contrôle par mode glissant implique deux étapes essentielles : le choix de la surface de glissement et le développement de la loi de contrôle. Ces étapes jouent un rôle crucial dans l'établissement d'une stratégie de contrôle par mode glissant efficace et stable. La sélection de la surface de glissement détermine le comportement souhaité du système et ses propriétés de convergence, tandis que la conception de la loi de contrôle vise à générer des signaux de contrôle qui guident le système vers la surface de glissement souhaitée et garantissent son maintien sur cette surface. Dans le contexte de l'exemple de suivi pour le robot mobile à roues, nous utilisons la variable de glissement  $\sigma r = [s_x, s_y]^T$  pour représenter le suivi.



**Figure 66 : Robot mobile à deux roues soumis à des incertitudes**

Pour définir la surface de glissement, nous prenons en compte le comportement de suivi souhaité et l'exprimons comme suit, en tenant compte des exigences spécifiques du système :

$$s_x = \dot{e}_1 + \beta e_1 \quad (\text{IV.19})$$



$$s_y = \dot{e}_2 + \beta_2 e_2 \quad (\text{IV.20})$$

Où les gains  $\beta_1$  et  $\beta_2$  peuvent être sélectionnés à l'aide de techniques de placement de pôles pour assurer la convergence asymptotique des erreurs de suivi  $e_1 = f_1 - f_{xd}$  et  $e_2 = f_2 - f_{yd}$  vers zéro. Dans cet exemple de suivi, la variable de glissement  $\sigma_r = [s_x, s_y]^T$  est choisie comme l'erreur de suivi. Par conséquent, la surface de glissement pour le peut être définie comme suit :

$$\dot{e}_1 + \beta_1 e_1 = 0 \quad (\text{IV.21})$$

$$\dot{e}_2 + \beta_2 e_2 = 0 \quad (\text{IV.22})$$

Comme suggéré par [12], pour garantir que la surface de glissement  $\sigma_r = 0$  est attractive, nous pouvons imposer la dynamique de  $\sigma_r$  comme suit :

$$\dot{\sigma}_r = -k_i \text{sgn}(\sigma_r) \quad (\text{IV.23})$$

Où la fonction de signe standard est représentée par  $\text{sgn}$  et  $k_i$  ( $i = 1, 2$ ) est une constante. Une approche pour prouver la stabilité de la dynamique de l'erreur consiste à analyser la fonction de Lyapunov suivante :

$$v_s = \frac{1}{2} \dot{\sigma}_r \sigma_r \quad (\text{IV.24})$$

La dérivée de  $V_s$  est définie comme suit :

$$\dot{v}_s = \sigma_r \dot{\sigma}_r \quad (\text{IV.25})$$

Nous pouvons conclure que  $V_s$  est une fonction positive et que sa dérivée  $V_s$  est négative ou nulle. Par conséquent, le système présente une stabilité asymptotique de Lyapunov. En utilisant les équations (IV.19), (IV.20) et (IV.23), nous obtenons :

$$-k_1 \text{sign}(s_x) = \ddot{e}_1 + \beta_1 \dot{e}_1 \quad (\text{IV.26})$$

$$-k_2 \text{sign}(s_y) = \ddot{e}_2 + \beta_2 \dot{e}_2 \quad (\text{IV.27})$$

En conséquence, en utilisant les équations (IV.26) et (IV.27), nous pouvons obtenir :

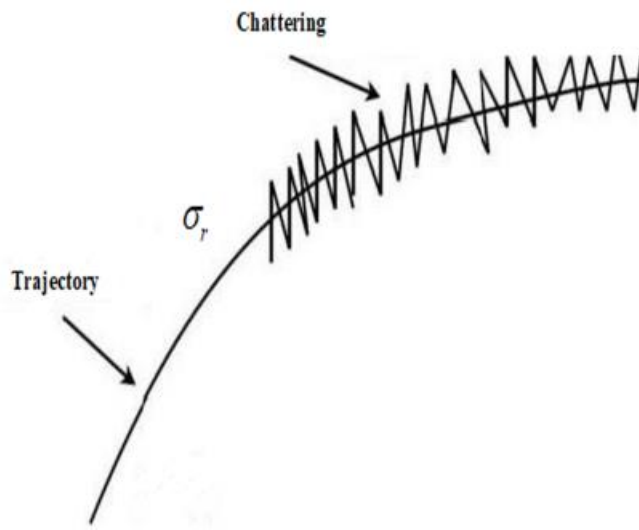
$$\ddot{f}_1 = \ddot{f}_{xd} - \beta_1 \dot{e}_1 - k_1 \text{sign}(s_x) \quad (\text{IV.28})$$

$$\ddot{f}_2 = \ddot{f}_{yd} - \beta_2 \dot{e}_2 - k_2 \text{sign}(s_y) \quad (\text{IV.29})$$

En substituant  $\ddot{f}_1$  et  $\ddot{f}_2$  par leurs nouvelles expressions définies par les équations (IV.26) et (IV.27) dans le contrôle défini par [13], le contrôleur de platitude mode glissant (flatness sliding tracking controller) appliqué à l'est défini comme suit :

$$\begin{bmatrix} u_{fsmcx} \\ u_{fsmcy} \end{bmatrix} = B_{rob}^{-1} \begin{bmatrix} \ddot{f}_{xd} - \beta_1 \dot{e}_1 - k_1 \text{sign}(s_x) \\ \ddot{f}_{yd} - \beta_2 \dot{e}_2 - k_2 \text{sign}(s_y) \end{bmatrix} \quad (\text{IV.30})$$

Le FSTC défini par les équations (IV.30) contient un terme de contrôle discontinu en raison de la fonction  $sg(\sigma)$ . Bien que le choix de valeurs suffisamment grandes pour  $k_1$  et  $k_2$  puisse permettre d'atteindre la convergence vers la variable de glissement en un temps limité et fournir une robustesse contre les perturbations, cela entraîne également le Phénomène de broutement, comme illustré à la Figure (67). Ainsi, pour éviter ce problème,

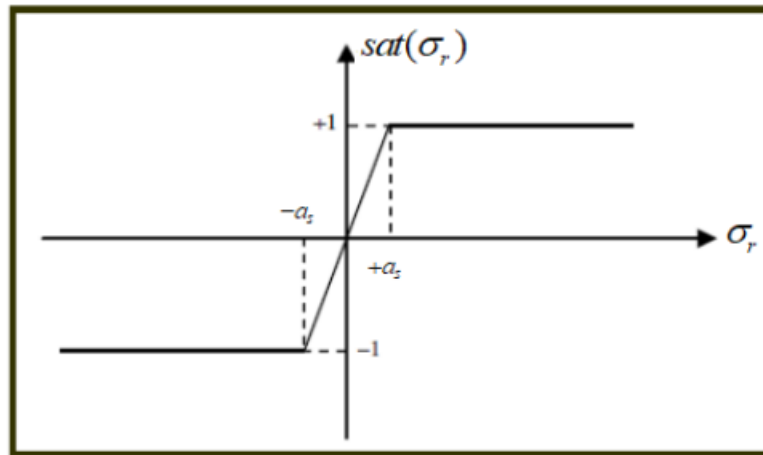


**Figure 67 : Le Phénomène de broutement**

La fonction  $sg(\sigma)$  peut être remplacé par la fonction  $Sat$  (Figure 68) définie comme suit :

$$sat(\sigma) = \begin{cases} \frac{\sigma}{a_s} & \text{si } |s| \leq a_s \\ \text{sing}(\sigma) & \text{si } |s| > a_s \end{cases}$$

Avec  $a_s$  étant la largeur du seuil de la fonction de saturation. L'augmentation de la valeur de la variable  $a_s$  conduit à une plus grande déviation de l'approximation par rapport à la fonction  $sgn$ , ce qui permet de réduire davantage le Phénomène de broutement mais au détriment de la préservation de la robustesse.



*Figure 68 : Fonction de saturation*

## IV.2. Simulation de la commande non-linéaire robuste sur ros et MATLAB

Le système ROS est un Robotiques Operating System, le premier projet collaboratif robotique à grande Échelle qui fournit un ensemble d'outils informatique permettant de gagner du temps dans la mise au point D'un robot, ou d'un système robotique.

### IV.2.1. Structure de ROS

- Les paquets :

La notion de paquet est très importante pour la compréhension de ROS. Les paquets sont essentiels à ROS car ce sont principalement eux qui font l'organisation de celui-ci. Le middleware est organisé sous forme de paquets qui peuvent contenir de nombreux éléments. On peut y retrouver, par exemple, un processus d'exécution (noeud), une librairie, des données, des fichiers de configuration. Dans ROS, le paquet est la plus petite unité de construction qui puisse être créée.

Les paquets ont tendance à tous suivre la même structure. Voici une liste des fichiers qui sont souvent présents dans les paquets :

- include/package\_name : en-têtes d'inclusion C ++ ;
- msg/ : dossier contenant des messages ;
- src/package\_name/ : fichiers sources ;
- srv/ : dossier contenant des services ;
- scripts/ : scripts exécutables ;

- CMakeLists.txt : fichier de construction ;
- package.xml.

Chaque paquet exprime un besoin. Par exemple, s'il faut calculer la trajectoire d'un robot, un paquet sera créé pour ce besoin.

- Le master :

Le master de ROS permet aux nœuds de s'enregistrer. Son rôle est de permettre à ceux-ci de se localiser. Grâce au master de ROS, les nœuds sont en capacité de communiquer entre eux. Sans le master, les nœuds ne sont pas capables de se trouver les uns les autres et donc ne peuvent pas communiquer entre eux.

Le master contient également le « paramètre server ». Celui-ci contient des données utilisées comme paramètres.

- Nœud :

Un nœud est un processus qui permet d'effectuer un calcul. Chaque nœud a un but. Par exemple, il y a un nœud qui permet de déplacer un robot jusqu'à un point, un deuxième nœud permettant l'arrêt du robot.

Un système de contrôle du robot comprendra plusieurs nœuds. Il est donc nécessaire que ceux-ci communiquent entre eux. Tous les nœuds en cours d'exécution doivent avoir un nom unique.

#### IV.2.2. Quelques commandes utiles

Il existe de nombreuses commandes dans ROS, nous avons recensé celles qui nous paraissent les plus importantes lors de l'utilisation de ROS.

Voici quelques commandes liées à la notion de paquet :

- catkin\_create\_pkg : permet de créer un nouveau paquet ;
- rospack : permet de donner les informations concernant un paquet ;
- catkin\_make : permet de compiler un ensemble de paquets ;
- rosdep : permet de créer un système de dépendances d'un paquet ;
- rqt : permet de visualiser les dépendances d'un paquet sous forme de graphique.

Nous pouvons également trouver des commandes liées au master :

- roscore : permet de démarrer ROS et de créer le master.

Lorsque nous lançons cette commande, nous pouvons voir sur la dernière ligne résout. Rosout est ce qui permet d'avoir une interface avec le « cœur » de ROS.

Tout comme pour les paquets, les nœuds ont des commandes qui leur sont associées :

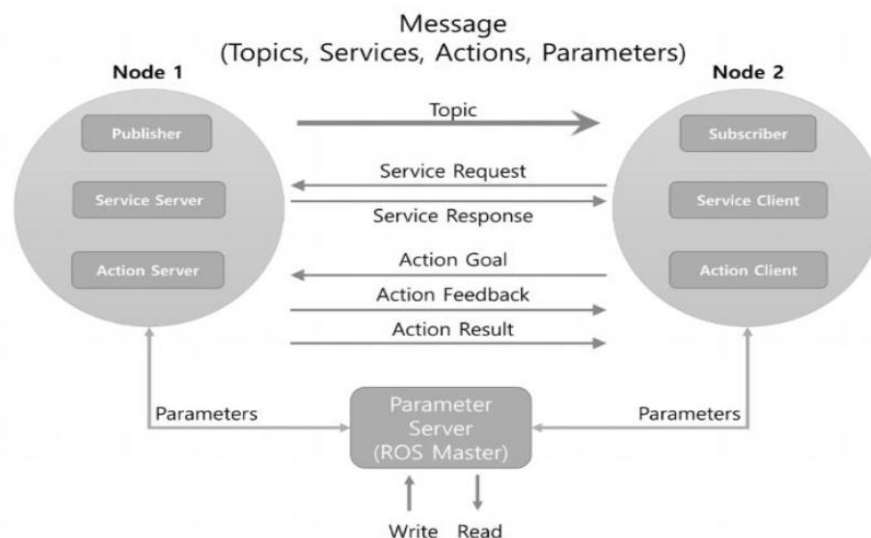
- Rosnode info : retourne les informations concernant le nœud ;
- Rosnode kill : permet d'arrêter un nœud ;
- Rosnode list : permet d'afficher la liste des nœuds actifs ;
- Rosnode ping : permet de tester la connexion entre les nœuds.

D'autres commandes utiles :

- rostopic list : retourne la liste des topics actifs ;
- rosrn [package\_name][node\_name] : permet de lancer un fichier exécutable.

Comme expliqué précédemment, les nœuds sont en lien direct avec le master. Nous allons maintenant regarder comment ils font pour communiquer entre eux.

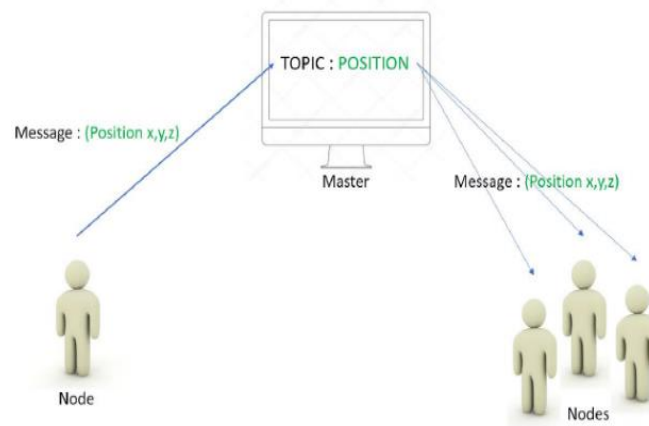
#### IV.2.3. Communication interne dans ROS



**Figure 69 : Différents types de messages dans ROS**

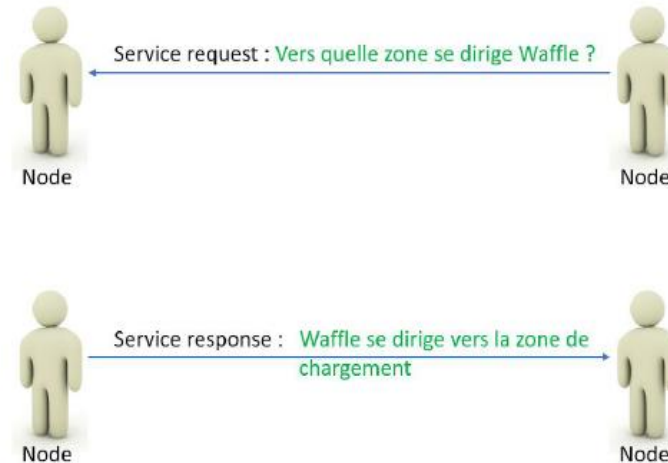
Les nœuds communiquent entre eux avec des messages. Il existe trois types de messages qui ont chacun leur particularité comme montre la figure (70).

- Topic : (littéralement comme un sujet de conversation). Le Publisher Node enregistre d'abord le *topic* au *master*. Les autres *nœuds* ont juste à s'y abonner. Ils recevront les informations du Publisher par contre ils ne peuvent pas demander d'informations (unidirectionnelles). On utilise ce type de message quand les nœuds s'échangent des données tout le temps, figure (71).



**Figure 70 : Exemple du type de message Topic**

- Service : pour utiliser ce type de messages, il faut un service server et un service client. Le serveur répond aux demandes ; le client transmet les demandes. Une fois la réponse transmise, le lien est coupé entre les deux nodes. Ici, il s'agit d'un lien bidirectionnel. Il est avantageux d'utiliser ce type de message lorsque l'on souhaite connaître l'état du système au moment de la demande, figure (72).



**Figure 71 : Exemple du type de message Service**

- Action : ils fonctionnent comme le service. Le serveur exécute l'action que le client a demandée. Ils sont utilisés lorsque le temps de réponse est long et/ou quand on souhaite avoir un feedback figure (73).

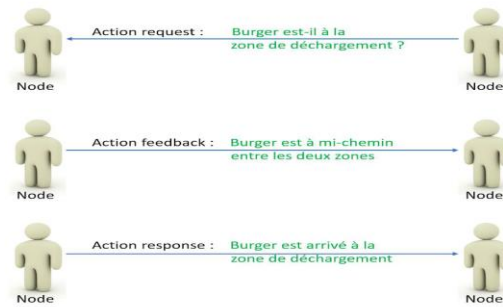


Figure 72 : Exemple du type de message Action

#### IV.2.4. Simulation de trajectoire circulaire avec perturbation

Après avoir évoqué brièvement Rose et toutes les commandes nécessaires pour effectuer une simulation et observer les résultats réels sur TurtleBot, nous allons maintenant procéder à cette simulation, la figure (73) décrit le diagramme flux de processus

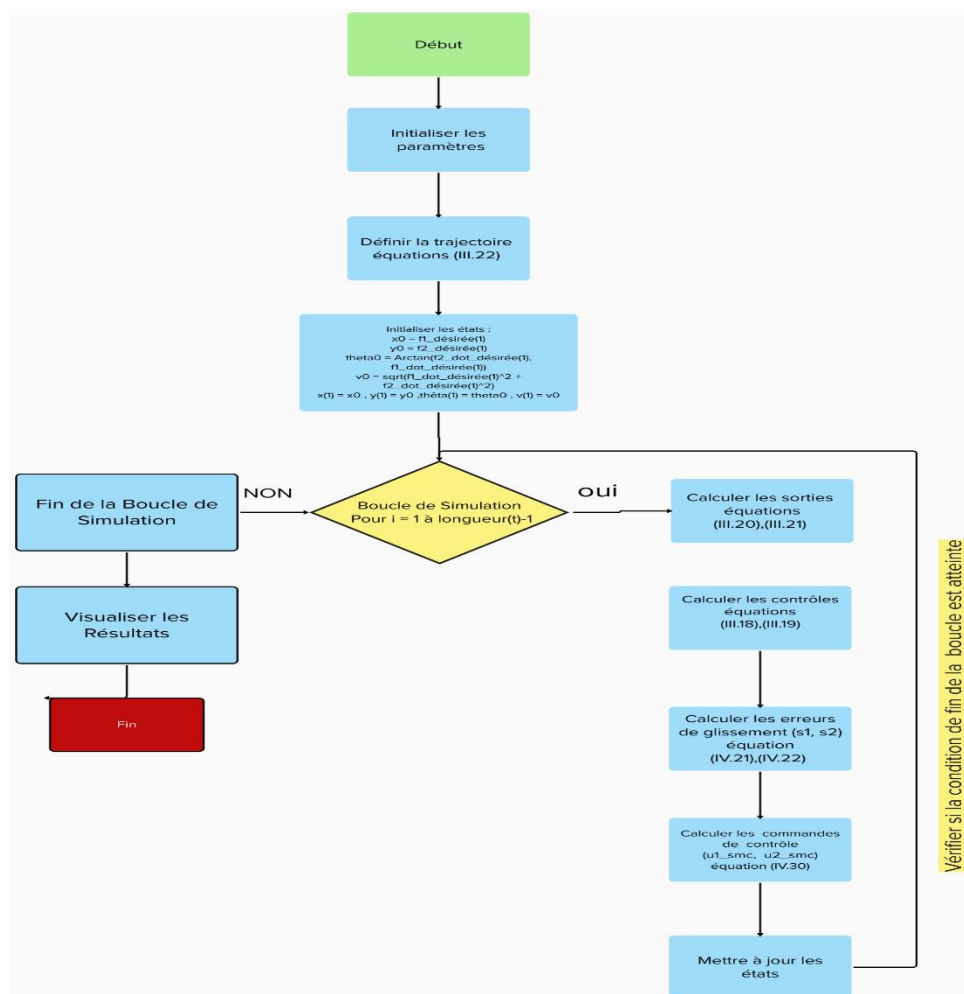
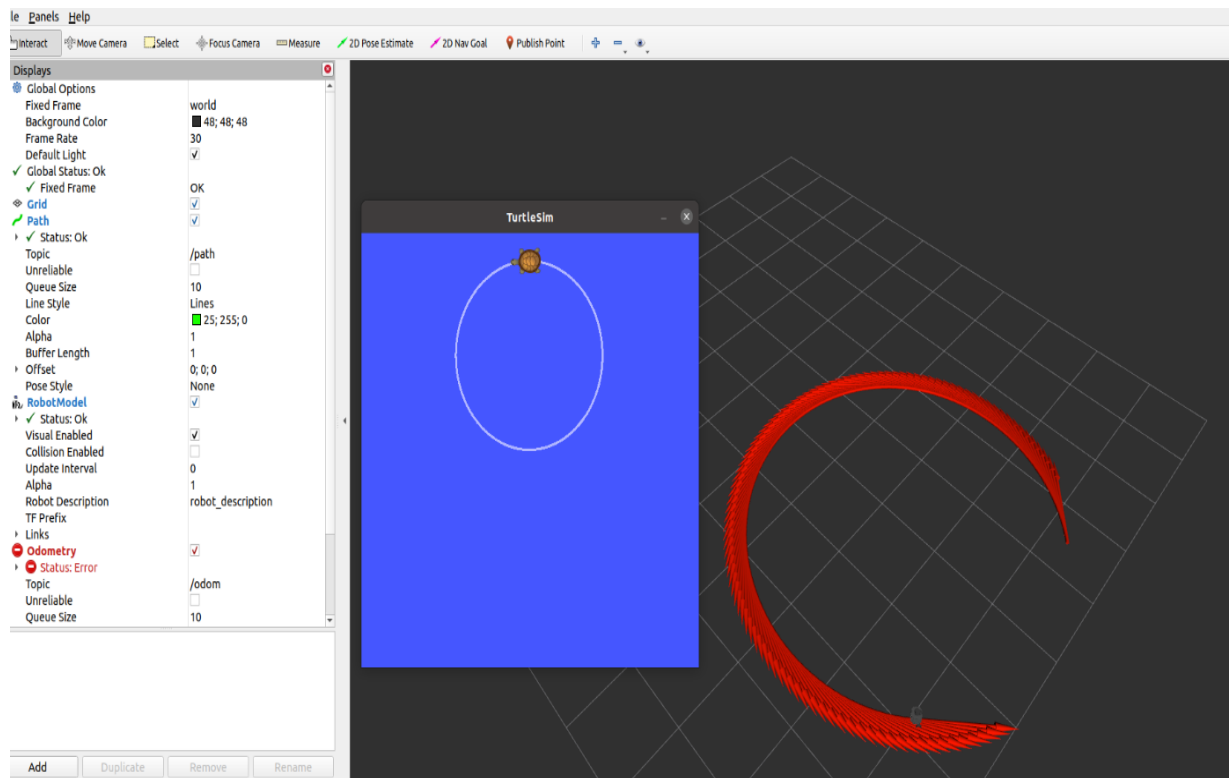


Figure 73 : diagramme de flux processus commande par platitude basé sur mode glissant

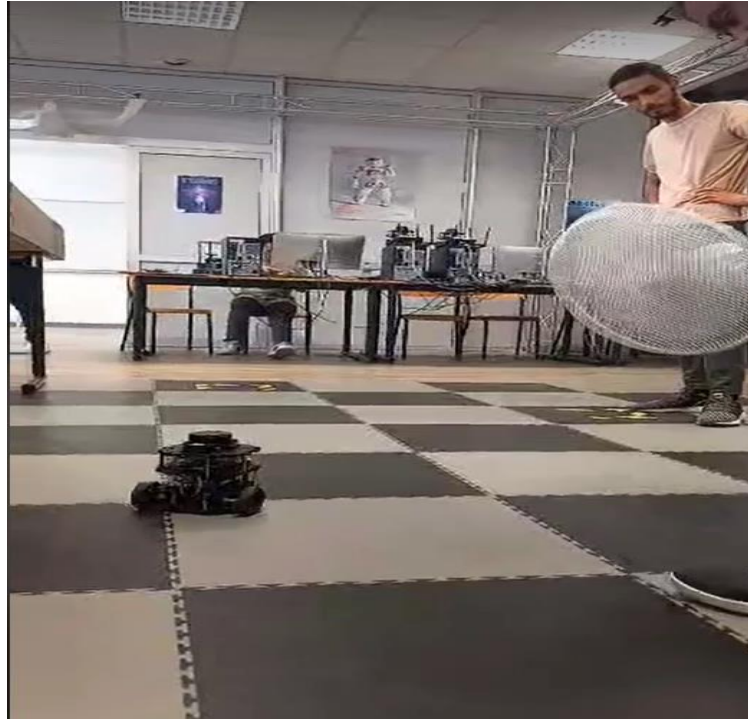


*Figure 74 : Simulation de la trajectoire circulaire sur ros*



*Figure 75 : Simulation réelle avec perturbation de vent*

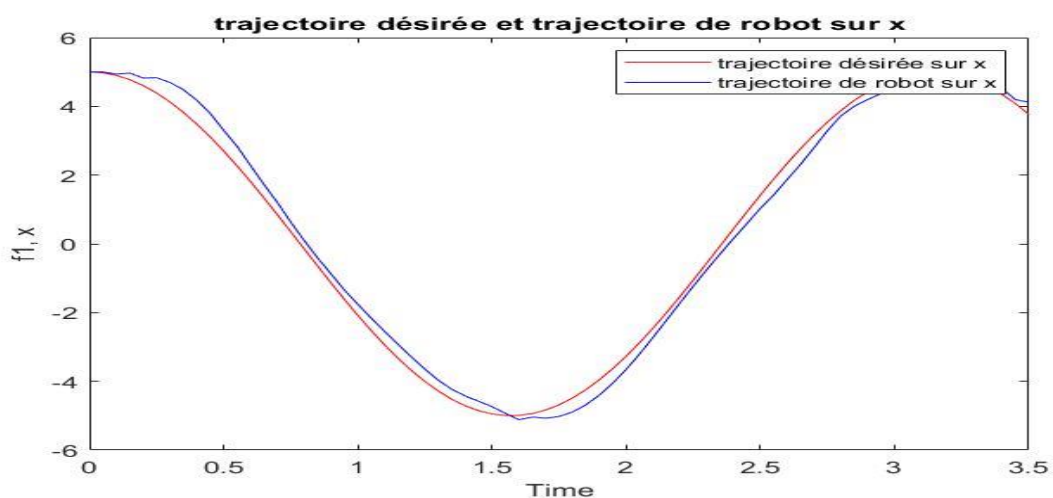




**Figure 76 : Simulation réelle avec perturbation de vent**

Les figures (74 ,75 et 76) représentent des simulations réelles et des simulations sur ROS (Robot Operating System) d'un robot suivant une trajectoire circulaire. On observe que le robot, lorsqu'il est contrôlé par une commande en planitude basée sur le mode glissant, réussit à devenir plus stable et à suivre la trajectoire circulaire de manière plus précise, même en présence de perturbation.

On peut voir aussi la simulation sur Matlab pour une trajectoire circulaire les figures suivantes représentent une simulation de robot qui suit une trajectoire circulaire avec perturbation variable.



**Figure 77 : Trajectoire sur l'axe x**

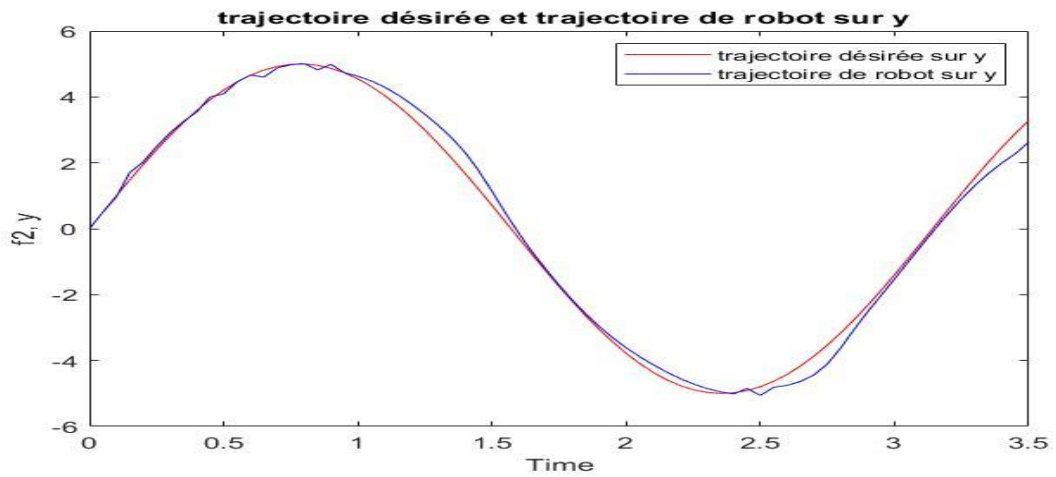


Figure 78 : Trajectoire sur l'axe y

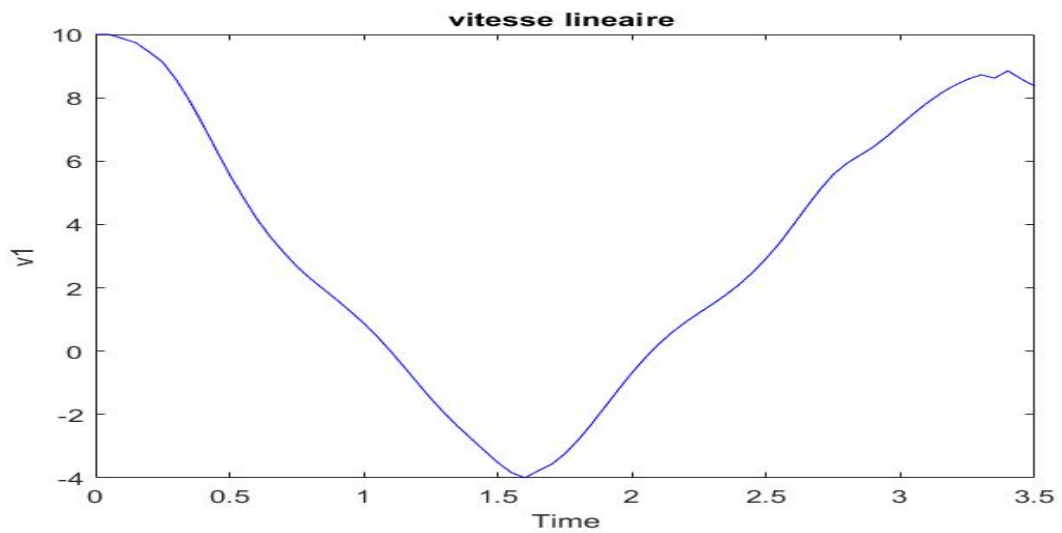


Figure 79 : Vitesse linéaire de robot

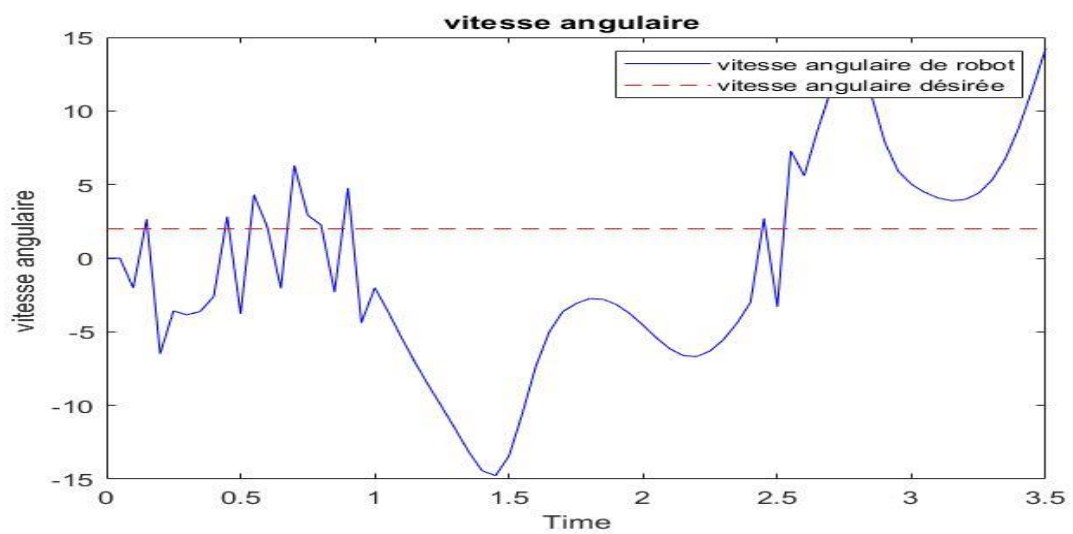
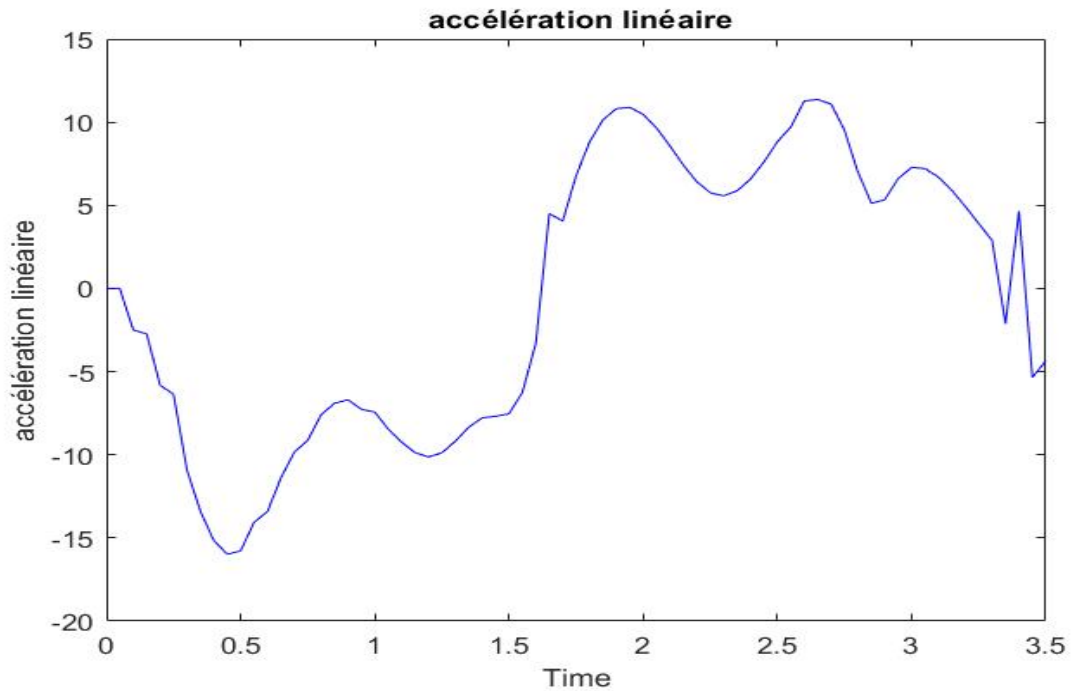
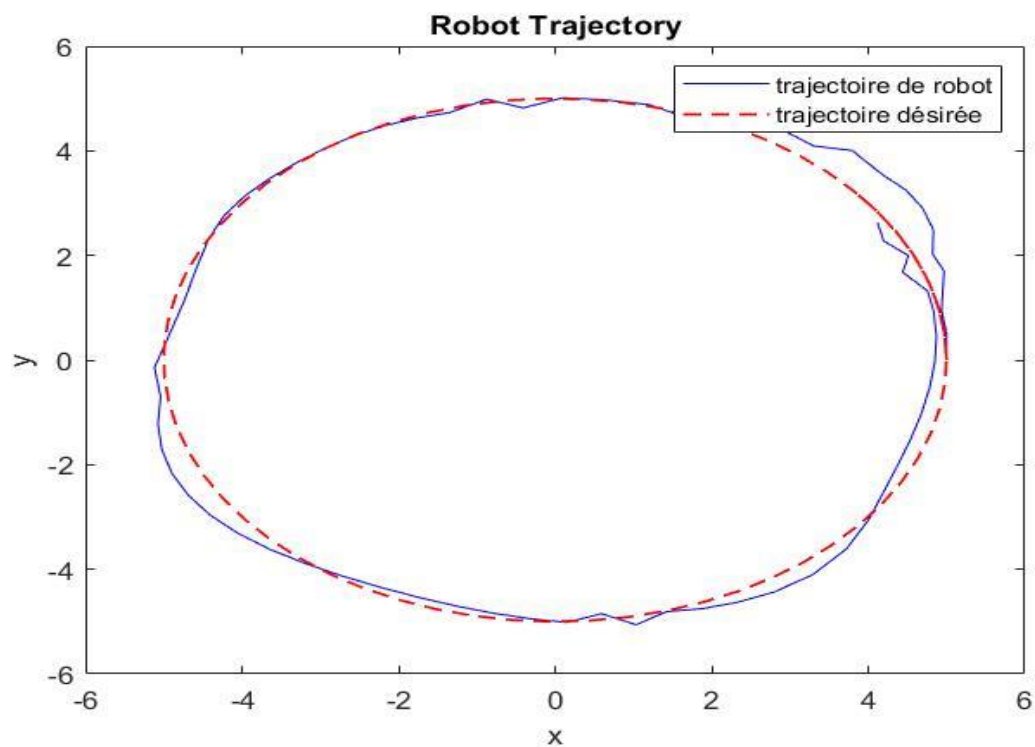


Figure 80 : Vitesse angulaire de robot



*Figure 81 : Accélération de robot*



*Figure 82 : Trajectoire de robot et trajectoire désirée*

Les figures semblent être que, même en présence de perturbations dans un système, la commande basée sur la "platitude avec mode glissant" reste une méthode efficace pour suivre une trajectoire désirée dans des conditions réelles.

## V. Conclusion

En conclusion de ce chapitre, nous avons abordé les généralités sur la commande par mode glissant, une technique de commande robuste et efficace pour stabiliser et contrôler les systèmes non linéaires et incertains. Nous avons examiné les principes fondamentaux de la commande par mode glissant, en mettant l'accent sur la création d'une surface de glissement sur laquelle le système est contraint de se déplacer.

En utilisant les concepts de la commande par mode glissant, nous avons développé une méthode de contrôle basée sur la platitude, qui permet de générer des trajectoires de référence pour les systèmes. Cette approche offre des avantages en termes de suivi précis des trajectoires et de robustesse face aux perturbations.

Enfin, nous avons introduit le ROS (Robot Operating System) et présenté une simulation pratique de contrôle basé sur la commande par mode glissant dans un environnement ROS. Nous avons utilisé une trajectoire circulaire comme référence et démontré comment la commande par mode glissant peut être appliquée pour guider un robot sur cette trajectoire avec précision.

## Conclusion générale

Dans le cadre de ce projet de fin d'études, le développement d'un contrôleur de poursuite non linéaire pour un robot mobile est une problématique complexe mais très pertinente. Cette approche vise à permettre au robot de suivre une trajectoire spécifique de manière précise, même dans un environnement dynamique et non linéaire.

En conclusion, ce rapport a présenté le développement d'un contrôleur de poursuite non linéaire pour un robot mobile. Nous avons débuté en fournissant une présentation détaillée du laboratoire ImViA, de son organisation et de ses différentes équipes de recherche. Cette introduction nous a permis de situer notre projet dans le contexte des activités de recherche menées au sein du laboratoire.

Ensuite, nous avons abordé la problématique de contrôle non linéaire et robuste des robots mobiles. Nous avons examiné les avancées récentes dans le domaine, en mettant en évidence les différents types de robots mobiles ainsi que leurs applications. Nous nous sommes concentrés sur les méthodes de commande non linéaire, robuste et optimale, en soulignant leur capacité à gérer des systèmes dynamiques complexes.

Dans le troisième chapitre, nous nous sommes intéressés à la commande non linéaire par platitude, une approche prometteuse pour la poursuite précise d'une trajectoire prédéfinie. Nous avons exploré les concepts fondamentaux de la platitude et nous avons réalisé des simulations pour évaluer les performances de cette méthode de commande.

Le quatrième chapitre s'est concentré sur la commande non linéaire robuste, en particulier la commande par mode glissant. Nous avons étudié les principes de cette approche et nous avons également utilisé la plateforme ROS et le robot mobile TurtleBot pour réaliser des simulations réalistes de notre contrôleur.

En conclusion générale, ce projet de fin d'études nous a permis d'approfondir nos connaissances dans le domaine du contrôle non linéaire des robots mobiles. Nous avons exploré différentes approches de commande, en mettant en évidence leurs avantages et leurs limites. Les résultats de nos simulations ont démontré l'efficacité de nos contrôleurs dans des scénarios spécifiques.

En guise de perspectives futures, il serait intéressant d'élargir nos investigations en explorant d'autres méthodes de commande non linéaire et robuste, ainsi que d'explorer l'intégration de techniques d'intelligence artificielle dans le contrôle des robots mobiles.

De plus, il serait bénéfique de mettre en œuvre notre contrôleur sur un véritable robot mobile pour valider expérimentalement ses performances.

En somme, ce projet nous a permis d'approfondir notre compréhension du contrôle non linéaire des robots mobiles et de contribuer à l'avancement de ce domaine passionnant. Nous espérons que ce rapport servira de base solide pour des recherches futures et pour l'application pratique de ces concepts dans des domaines tels que la robotique autonome, la logistique, la surveillance et bien d'autres.

## Référence bibliographique

- [1] SLIMANE Nouredine, « système de localisation pour robots mobiles », thèse de docteur d'état de l'université de Batna, novembre 2005.
- [2] Ami Khalil, Djenane Mohamed amine, « pilotage automatique d'un robot mobile par l'intelligence artificielle, Université Mohamed Larbi Ben M'hidi - Oum El bouaghi Faculté des Sciences et des Sciences Appliquées Département de génie électrique master académique électronique de système embarques, octobre 2020.
- [3] H. Tédjini, M. Moudjahed, Y. Meslem and M. Tadjine, « Commande non linéaire adaptative par backstepping et observateur d'une machine synchrone à aimant permanent alimentée par un onduleur à trois niveaux, » 4th International Conférence on Computer Integrated Manufacturions CIP'2007, Novembre 2007.
- [4] A. R. Benaskeur, « Aspects de l'application du backstepping adaptatif à la commande décentralisée des systèmes non linéaires, » Thèse de doctorat (Ph.D), Université Laval, Québec, 2000.
- [5] Melle. Khesrani Samia, « commande de robot par platitude », mémoire Présenté à la faculté de Technologie département d'électronique, 2014
- [6] Salah Eddine Ghamri, « Commande Neuronale d'un Robot Mobile Non-Holonome avec glissement des roues », Université de Batna 2 Faculté de Technologie Département d'Électronique Mémoire.
- [7] ZIDANI Ghania, Commande Robuste d'un Robot Mobile à Roues, Université de Batna -2-Faculté de Technologie Département d'Électronique, DOCTORAT en SCIENCES ,2009.
- [8] Amine Abadi, « Commande robuste par platitude application : robot mobile », école nationale d'ingénieurs de Sousse, Mémoire de Mastère de recherche,2013.
- [9] SAIBI Hayat, KASSOURI Dalila, « Commande par mode glissant d'ordre fractionnaire d'un réseau électrique », Faculté de Génie Electrique et Informatique département d'Automatique, Master Académique en Automatique,2018
- [10] I. Podlubny, «Fractional-order system and p- $\mu$ -controllers», IEEE Trans Autom Control 44(1):208–14, 1999.

[11] Abadi Amine, « Robust Tracking Control of Wheeled Mobile Robot Subject to Uncertainties » IMVIA Laboratory, University of university Bourgogne

Franche-Comté ,2023.

[12] Mauledoux M, Mejia-Ruda E, Aviles Sanchez O, Dutra MS, Rojas Arias A. « Design of Sliding Mode Based Differential Flatness Control of Leg-wheel Hybrid Robot. » In: . 835. Trans Tech Publ.; 2016.

[13] Khalesi R, Yousefi M, Pishkenari HN, Vossoughi G, « Robust independent and simultaneous position control of multiple magnetic microrobots by sliding mode controller ». Mechatronics 2022.

[14] A. Ramdani, « Commande Prédictive Des Systèmes Dynamiques : Etude Comparative Avec Les Régulateurs Classiques », Mémoire De Magister, Université M'hamed Bougara-Boumerdes, 2013

[15] S. Olaru, « Contribution A L'étude De La Commande Prédictive Sous Contraintes Par Approche Géométrique », Thèse De Doctorat, Université Paris Xi Orsay, France, 2005.

[16] O. Bourebia, « Commande Prédictive Floue Des Systèmes Non Linéaires », Thèse De Doctorat, Université Mentouri De Constantine, 2011.

[17] A. Bezzini, «Commande Prédictive Non Linéaire En Utilisant Les Systèmes Neuro-Flous Et Les Algorithmes Génétiques», Mémoire De Magister, Université Mohamed Khider De Biskra, 2013.