

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE 20 AOUT 1955 SKIKDA

Faculté de Technologie

Département : Pétrochimie



Mémoire

En vue de l'obtention du diplôme de Master.

Filière : Industries Pétrochimiques

Spécialité : Automatisation en industrie pétrochimique

Par : Touil Houssam Eddine

Thème:

Commande intelligent d'un robot mobile à quatre roues

Soutenu publiquement le : 03 / 07 / 2024 devant le jury composé de :

<i>Président</i>	<i>Dr. Bounezour Hichem</i>	<i>MCB</i>	<i>U-Skikda</i>
<i>Rapporteur</i>	<i>Pr. Zennir Youcef</i>	<i>Pr</i>	<i>U-Skikda</i>
<i>Examineur</i>	<i>Dr. Ganouche Abderrahmane</i>	<i>MCB</i>	<i>U-Skikda</i>

Promotion : 2023/2024

Remerciements

On exprimer notre profonde gratitude à ALLAH le tout-puissant et miséricordieux, Alhamdulillah qui nous a donné la force, le courage et la persévérance nécessaires pour mener à bien ce mémoire de fin d'étude en master LMD dans de bonnes conditions et dans les délais prévus.

A notre promoteur :

Pr. Zennir Youcef

Nous vous remercions d'avoir accepté de diriger notre travail. Nous vous témoignons notre profonde admiration, tant pour vos grandes qualités professionnelles que pour votre ouverture d'esprit. Vous avez bien voulu nous guider dans l'élaboration de ce travail, et nous avons été touchée par la confiance que vous nous avez témoignée. Veuillez trouver ici le témoignage de notre gratitude et de notre profond respect.

Au membre de jury: Dr. Bounezour Hichem et Dr. Ganouche Abderrahmane

Vous avez accepté avec gentillesse de bien vouloir juger notre travail. Nous vous sommes reconnaissants de l'enseignement que vous nous avez apporté. Veuillez trouver ici l'expression de nos remerciements les plus sincère pour votre présence et lecture attentive de notre mémoire, ainsi que, pour les remarques que vous m'adresseront lors de la soutenance afin d'améliorer notre travail.

On souhaite exprimer notre plus sincère gratitude à tous mes enseignants du parcours de formation Automatisation et nos collègues de l'Université de Skikda.

Dédicaces

En premier lieu alhamdulillah je remercie ALLAH alhamdulillah le tout puissant de m'avoir donné la volonté, la santé et le courage pour réaliser ce travail avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie ce modeste travail.

À mes parents, le Professeur TOUIL Hamouda et LATRACHE ounasa à celui qui m'a tout donné sans compter, à celui qui m'a soutenue toute ma vie, à celui que je dois ce que je suis et ce que je serai. Merci pour votre amour et votre soutien sans failles depuis le début. Une profonde gratitude pour votre présence, votre écoute, votre confiance en moi et tes conseils qui me sont si précieux, pour tous les efforts et les sacrifices que vous n'avez cessé de consentir pour mon instruction et mon bien être. C'est à vos côtés qu'a grandi ma vocation, vous restez pour moi un modèle de savoir et de générosité. En espérant ne jamais vous décevoir.

Je t'aime infiniment. De tout cœur, merci.

Mes sœurs (S, N, S, H, A) et mes confidentes. Nous nous sommes soutenues dans les bons et mauvais moments. Je suis confiant quant à l'avenir.

À, Mes frère (F, W, S, Y), je suis fière de vous. Merci pour votre soutien et vos encouragements constants.

À mes frères, KANDOULI Elheytem et TARFA wail Sur votre soutien sans faille et les efforts déployés pour m'aider, je suis fier de vous deux.

À mes frères à l'université, Pour tous nos bonheurs partagés et nos souvenirs passés.

À tous mes amis, mes amis de toujours. Notre amitié est très précieuse.

À toute ma famille, mes oncles, tantes, cousins et cousines.

حفظكم الله و رعاكم و سدد خطاكم.

TOUL HOUSSAM EDDINE.

ملخص: في هذا العمل، قمنا بدراسة التحكم في تتبع مسارات معقدة لروبوت متنقل (من نوع سيارة) باستخدام مختلف هندسيات التحكم. تم تقديم مراجعة شاملة حول الروبوتات المتنقلة ذات العجلات، تلتها نمذجة سينماتيكية لروبوت متنقل بعجلتين. تم تطوير هندسيات تحكم متعددة مثل المتحكم الضبابي، والمتحكم PID-floue، وضبط معلمات المتحكم PID استناداً إلى خوارزميات ميتا-هيورستيك (PSO و GJO). تم إجراء المحاكاة باستخدام ماتلاب سيميلينك مع مسارات مرجعية متنوعة. كانت النتائج التي تم الحصول عليها مرضية للغاية خاصة مع متحكم PID-PSO، وفي الأعمال المستقبلية، سيتم دراسة تنقل الروبوت في محيط بوجود عوائق ثابتة ومتحركة.

الكلمات المفتاحية: روبوت متنقل، متحكم ضبابي، متحكم PID-PSO، متحكم PID-PSO، متحكم PID-GJO، محاكاة

Résumé: Dans ce travail, nous avons étudié la commande du suivi des trajectoires complexes d'un robot mobile (type véhicule) avec différentes architectures de commande. Un état de l'art sur les robots mobiles à roues à été présenté, suivi par une modélisation cinématique d'un robot mobile à deux roues. Différentes architectures de contrôle sont développées comme contrôleur floue, PID-floue et ajustement des paramètres d'un contrôleur PID basé sur des algorithmes méta-heuristiques (PSO et GJO). Les simulations sont réalisées sous Matlab-Simulink avec différentes trajectoires de références. Les résultats obtenus sont très satisfaisants surtout avec le contrôleur PID-PSO, dans un travail futur étudier le cas de déplacement du robot avec la présence des obstacles fixe et mobile.

Mots clés : Robot mobile, contrôleur floue, PID-floue, PID-PSO, PID-GJO, simulation

Abstract: In this work, we studied the command of complex trajectory tracking for a mobile robot (like car) using different command architectures. A state-of-the-art on wheeled mobile robots was presented, followed by kinematic modeling of a two-wheeled mobile robot. Different control architectures were developed, such as fuzzy controller, Fuzzy-PID, and parameter adjustment of a PID controller based on meta-heuristic algorithms (PSO and GJO). Simulations were conducted using Matlab-Simulink with various reference trajectories. The results obtained were very satisfactory, especially with the PID-PSO controller. In future work, we will investigate the movement of the robot with the presence of both fixed and mobile obstacles.

Keywords: Mobile robot, fuzzy controller, fuzzy-PID, PID-PSO, PID-GJO, simulation

Table des matières

<i>Remerciements</i>	<i>I</i>
<i>Dédicaces</i>	<i>II</i>
<i>Résumé</i>	<i>III</i>
<i>Table des matières</i>	<i>IV</i>
<i>Table des figures</i>	<i>VIII</i>
<i>Liste des tableaux</i>	<i>XI</i>
<i>Acronymes</i>	<i>XII</i>
<i>Introduction générale</i>	<i>1</i>
<i>Chapitre I: État de l'art sur les robots mobiles</i>	<i>3</i>
I.1. Introduction	4
I.2. L'origine du mot robot	4
I.3. Dates marquantes de la robotique	5
I.4. Robot mobile	5
I.5. Classification des robots mobiles	6
I.5.1. Classification par degré d'autonomie	6
I.5.1.1. Robot télécommandé	6
I.5.1.2. Robots semi-autonomes	6
I.5.1.3. Robots autonomes.....	6
I.5.2. Classification par domaine d'application	7
I.5.2.1. Robots industriels et robots de service	7
I.5.2.2. Robots militaires	7
I.5.2.3. Les robots expérimentaux.....	7
I.5.3. Classification selon le type de locomotion	7
I.5.3.1. Les robots mobiles à chenilles.....	7
I.5.3.2. Les robots mobiles marcheurs	8
I.5.3.3. Les robots mobiles Aérienne	8
I.5.3.4. Les robots mobiles à base à d'eau	8
I.5.3.5. Les robots mobiles à roues	8
I.6. L'architecture des robots mobiles à roues	9
I.6.1. La motricité et l'énergie	9
I.6.2. Les organes de sécurité	9
I.6.3. Traitement des informations et gestion des tâches.....	10
I.7. Les types des robots mobiles à roues	10
I.7.1. Robots unicycle.....	10
I.7.2. Robots tricycle	11

I.7.3.	Robots voiture	11
I.7.4.	Robots mobiles omnidirectionnels.....	12
I.7.5.	Disposition des roues	13
I.8.	Les avantages et les inconvénients des différents types des robots à roues	16
I.9.	Caractéristiques des robots mobiles a roues.....	17
I.9.1.	La charge maximale transportable	17
I.9.2.	L'espace de travail	17
I.9.3.	Le positionnement absolu	18
I.9.4.	Vitesse de déplacement.....	18
I.10.	Conclusion	18
Chapitre II: Modélisation des robots mobiles à roues		19
II.1.	Introduction	20
II.2.	La forme générale de modélisation des robots mobiles	20
II.3.	Concepts et systèmes des directions.....	22
II.4.	Roulement sans glissement et contraintes non holonômes	23
II.4.1.	Roulement sans glissement (r.s.g.).....	23
II.4.2.	Contraintes non holonômes.....	25
II.5.	Hypothèses de modélisation.....	26
II.6.	Modèle cinématique et système de direction	26
II.6.1.	Robot mobile type unicycle	26
II.6.2.	Robot mobile type tricycle.....	28
II.6.3.	Robot mobile type voiture.....	29
II.7.	Modélisation de notre modèle	31
II.8.	Conclusion.....	34
Chapitre III: Architecture de Contrôle.....		35
III.1.	Introduction.....	36
III.2.	Définition et domaines d'applications de la logique floue	36
III.3.	Définitions et notions sur la théorie des ensembles flous	37
III.3.1.	Définition d'un ensemble floue	37
III.3.2.	Les variables linguistiques	37
III.3.3.	Univers du discours	38
III.3.4.	Fonctions d'appartenance.....	38
III.3.5.	Opérations sur les ensembles flous.....	39
III.4.	Description des contrôleurs flous.....	40
III.4.1.	Mise en forme des entrées.....	41
III.4.2.	Interface de fuzzification	42
III.4.3.	Base de connaissance	43

III.4.4.	Moteur d'inférence.....	43
III.4.4.1.	Méthode MAX-MIN (Mamdani)	43
III.4.4.2.	Méthode MAX-PROD	44
III.4.4.3.	Méthode SOMMEPROD	44
III.4.5.	Interface de défuzzification.....	44
III.4.5.1.	Défuzzification par centre de gravité	44
III.4.5.2.	Défuzzification par la valeur maximale	45
III.4.6.	Mise en forme des sorties.....	45
III.5.	Régulateur PID	45
III.5.1.	Définition.....	45
III.5.2.	Types de régulateurs.....	46
III.5.2.1.	Régulateur à action proportionnelle (P)	46
III.5.2.2.	Régulateur à action proportionnelle-intégrale (PI)	47
III.5.2.3.	Régulateur à action proportionnelle-intégrale (PD).....	47
III.5.2.4.	Structure du régulateur PID (proportionnelle intégrale dérivée)	48
III.5.3.	Utilisation	48
III.6.	Optimisation par la méthode de PSO (optimisation par essaim de particule)	49
III.6.1.	Problème d'optimisation	49
III.6.2.	Principe de fonctionnement.....	50
III.6.3.	Mécanisme de confinement.....	52
III.6.4.	Déroulement de l'algorithme	53
III.7.	Optimisation par la méthode de GJO (Optimisation du Chacal Doré.).....	54
III.7.1.	Optimisation du chacal doré (GJO).....	55
III.7.2.	Étape d'exploration	56
III.7.3.	Déroulement de l'algorithme	57
III.8.	Architecture de commande de notre robot mobile	58
III.8.1.	Structure de base de notre contrôleur floue.....	58
III.8.2.	Optimisation des paramètres du régulateur PID par PSO et GJO.....	60
III.9.	Conclusion	61
	Chapitre IV: Simulation et discussion.....	62
IV.1.	Introduction.....	63
IV.2.	Simulation et Discussion	63
IV.3.	La simulation avec le contrôleur floue	65
IV.3.1.	La simulation avec le trajectoire Tr1	66
IV.3.2.	La simulation avec trajectoires différentes (Tr2, Tr 3, Tr4, Tr5).....	69
IV.3.3.	La simulation avec la trajectoire Tr6.....	72
IV.3.4.	Discussion	73

Table des matières

IV.4.	La simulation avec le contrôleur PID optimisé par PSO.....	74
IV.4.1.	La simulation avec trajectoires différentes (Tr2, Tr 3, Tr4, Tr5).....	74
IV.4.2.	La simulation avec la trajectoire Tr7.....	76
IV.4.3.	Discussion	78
IV.5.	La simulation avec le contrôleur PID optimisé par GJO.....	78
IV.5.1.	La simulation avec la trajectoire $Tr1$	79
IV.5.2.	La simulation avec la trajectoire Tr7.....	80
IV.6.	La simulation avec PID-floue.....	82
IV.7.	La représentation en 2D du robot et discussion générale	84
IV.8.	Conclusion	86
	<i>Conclusion général</i>	87
	<i>Bibliographie</i>	89

Table des figures

Figure I.1 : Architecture d'un robot mobile à roues.	9
Figure I.2 : Robot de type unicycle.	10
Figure I.3 : Robot de type tricycle.	11
Figure I.4 : Robot de type voiture.	11
Figure I.5 : Google Self-Driving Car Project.	12
Figure I.6 : Robot de type omnidirectionnel.	12
Figure I.7 : Les types des roues.	13
Figure II.1 Le référentiel global et le référentiel local du robot.	21
Figure II.2 : Robot mobile de type voiture et son CIR.	22
Figure II.3 : Schéma de contact entre la roue et le sol.	23
Figure II.4 : Caractérisation du roulement sans glissement	24
Figure II.5 : Robot mobile à deux roues.	27
Figure II.6 : Schéma d'un robot de type unicycle..	27
Figure II.7 : Schéma d'un robot de type tricycle..	28
Figure II.8 : Schéma d'un robot de type voiture.	29
Figure II.9 : Modèle cinématique de véhicule..	32
Figure II.10 : schéma technique du robot et trajectoire désirée.	33
Figure III.1 : Variables linguistiques..	38
Figure III.2 : Formes des fonctions d'appartenance...	39
Figure III.3 : Les opérateurs logiques..	40
Figure III.4 : Structure du contrôleur floue.	41
Figure III.5 : Fuzzification d'une valeur exacte...	42
Figure III.6 : Asservissement par un régulateur...	45
Figure III.7 : Régulateur PID...	46
Figure III.8 : Schéma fonctionnel du régulateur P...	47
Figure III.9 : Schéma fonctionnel du régulateur PI...	47
Figure III.10 : Schéma fonctionnel du régulateur PD...	47
Figure III.11 : Vol en group d'oiseau...	49
Figure III.12 : Déplacement d'une particule...	50
Figure III.13 : Les étapes de L'algorithme PSO...	54
Figure III.14 : Les étapes du comportement de chasse du chacal doré...	55
Figure III.15 : Les étapes de l'algorithme GJO...	57

Figure III.16: Le modèle cinématique du robot sur Simulink ..	58
Figure III.17: La simulation avec contrôleur floue.	58
Figure III.18: Conception de notre logique floue... ..	59
Figure III.19: Les Fonctions d'appartenances pour les entrés... ..	59
Figure III.20: Les Fonctions d'appartenances pour les sorties... ..	59
Figure III.21: PID-Floue Contrôler sur Simulink... ..	60
Figure III.22: Approche d'optimisation du PID par l'algorithme PSO et GJO... ..	61
Figure IV.1 : Trajectoire de référence forme cercle (Tr1).	63
Figure IV.2: Trajectoire de référence forme triangle (Tr2).	63
Figure IV.3: Trajectoire de référence forme trapézoïdal (Tr3)... ..	64
Figure IV.4: Trajectoire de référence forme carré (Tr4).. ..	64
Figure IV.5: Trajectoire de référence forme trapézoïdal (Tr5)... ..	64
Figure IV.6: Trajectoire de référence forme infini (Tr6)... ..	65
Figure IV.7: Trajectoire de référence forme infini (Tr7)... ..	65
Figure IV.8: résultat de test 1.. ..	66
Figure IV.9: L'erreur de test 1... ..	66
Figure IV.10: Résultat de test 2.. ..	67
Figure IV.11: L'erreur de test 2... ..	67
Figure IV.12: Résultat de test 3... ..	67
Figure IV.13: L'erreur de test 3... ..	68
Figure IV.14: Résultat de test 4... ..	68
Figure IV.15: L'erreur de test 4... ..	68
Figure IV.16: Les fonction d'appartenance de différentes trajectoires (CF)... ..	70
Figure IV.17: Résultat de la trajectoire Tr2 et son erreur (CF).. ..	71
Figure IV.18: Résultat de la trajectoire Tr3 et son erreur (CF).. ..	71
Figure IV.19: Résultat de la trajectoire Tr4 et son erreur (CF).. ..	71
Figure IV.20: Résultat de la trajectoire Tr5 et son erreur (CF).. ..	72
Figure IV.21: Les fonction d'appartenance de trajectoires Tr6 (CF).. ..	72
Figure IV.22: résultat de la trajectoire Tr6 (CF)... ..	73
Figure IV.23: résultat de l'erreur... ..	73
Figure IV.24: Résultat de la trajectoire Tr1 et son erreur (PSO)... ..	75
Figure IV.25: Résultat de la trajectoire Tr2 et son erreur (PSO)... ..	75
Figure IV.26: Résultat de la trajectoire Tr3 et son erreur (PSO)... ..	75
Figure IV.27: Résultat de la trajectoire Tr4 et son erreur (PSO).	76

Figure IV.28: Résultat de la trajectoire Tr5 et son erreur (PSO).	76
Figure IV.29: résultat de test 3 sur la trajectoire Tr7 (PSO).	77
Figure IV.30: résultat de l'erreur Tr7 (PSO).	78
Figure IV.31: Résultat de test 2 sur la trajectoire Tr1 (GJO).	79
Figure IV.32: Résultat de l'erreur Tr1 (GJO).	80
Figure IV.33: résultat de test 3 sur la trajectoire Tr7 (GJO).	81
Figure IV.34: résultat de l'erreur Tr1(GJO).	81
Figure IV.35 : Trajectoire de référence Tr8.	82
Figure IV.36 : Structure de contrôleur flou (Flou-PID).	83
Figure IV.37: Les fonctions d'appartenances des paramètres K_p , K_i , K_d	83
Figure IV.38: Trajectoire de référence Tr8 et l'erreur.	84
Figure IV.39: La représentation en 2D du robot (PSO).	84
Figure IV.40: La représentation en 2D du robot (CL).	85
Figure IV.41: La représentation en 2D du robot (PID-floue).	85

Liste des tableaux

Tableau I.1 : Configurations des roues pour véhicules roulants.	14
Tableau I.2 : Les icônes pour chaque type de roue.	16
Tableau I.3 : les avantages et les inconvénients des différents types de robots à roues.	17
Tableau II.1 : les systèmes des directions.	30
Tableau II.2 : Paramètres utilisés dans la modélisation.	32
Tableaux III.1 : Les structures du régulateur PID.	48
Tableau IV.1: Les points de départ (CF).	66
Tableau IV.2 : Résultat des tests avec Tr1 (CF).	69
Tableau IV.3 : Résultat des tests de trajectoires différentes (CF).	70
Tableau IV.4 : Résultat de test de trajectoire Tr6 (CF).	72
Tableau IV.5: Paramètres d’algorithme PSO.	74
Tableau IV.6 : Gains l’erreur moyenne des tests avec trajectoires déférent (PSO-PID).	74
Tableau IV.7 : Les points de départs (PSO).	74
Tableau IV.8 : Gains des tests avec trajectoire Tr7 (PSO-PID).	77
Tableau IV.9 : Les valeurs de l’erreur Tr_7 (PSO-PID).	77
Tableau IV.10: Paramètres d’algorithme GJO.	78
Tableau IV.11: Gains des tests avec trajectoire de Tr_1 (GJO-PID).	79
Tableau IV.12: Les valeurs de l’erreur Tr_1 (GJO-PID).	79
Tableau IV.13 : Gains des tests avec trajectoire de Tr7 (GJO-PID).	80
Tableau IV.14 : Les valeurs de l’erreur Tr7 (GJO-PID).	80
Tableau IV.15 : Les règles de PID-floue.	82

Acronymes

R.U.R : Le robot universel de Rosum.

XVIIIe siècle : La période allant de l'an 1701 à l'an 1800.

L'ISO : Organisation internationale de normalisation.

AUAV : Les véhicules aériens autonomes sans pilote.

EPFL : École Polytechnique Fédérale de Lausanne.

CMU : L'université de Carnegie Mellon.

DMT: Distributed Measurement and Telecommunication.

CIR : Concepts et systèmes des directions.

R.S.G : Roulement Sans Glissement.

Norm : La distance entre le robot et la position désirée ainsi que l'orientation du robot.

MM : Méthode des Moments.

PSO : Optimisation par Essaim de Particules.

GJO : Optimisation du Chacal Doré.

PID : (Proportionnel, Intégral, Dérivé).

PD : (Proportionnel, Dérivé).

PI : (Proportionnel, Intégral).

P: Proportionnel.

Tr : Trajectoire de référence.

CF : Contrôleur Floue.

$e_{moyenne}$: L'erreur moyenne.

2D : Deux dimensions.

m : mètre.

Introduction générale

L'objet de la robotique est l'automatisation des systèmes mécaniques. En dotant le système de capacités de perception, d'action et de décision, dans le but est de lui permettre d'interagir rationnellement avec son environnement, et de façon autonome [1]. La robotique est un domaine de recherche qui se situe au carrefour de l'intelligence artificielle, de l'automatique, de l'informatique et de la perception par ordinateur; cette interdisciplinarité est à l'origine d'une certaine complexité [2]. Des applications dans des domaines aussi variés que l'industrie manufacturière, le spatial, l'automobile ou plus récemment les loisirs et le secteur médical [2], démontrent aujourd'hui l'intérêt économique et social de ces recherches.

La robotique mobile est généralement considérée comme un sous-domaine de la robotique et de l'ingénierie de l'information. Les robots mobiles peuvent se déplacer dans leur environnement et ne sont pas statiques dans un seul emplacement physique [3]. Les robots mobiles peuvent être « autonomes », Cela signifie qu'il est capable de naviguer dans un environnement non supervisé sans avoir besoin de routeurs physiques ou électromécaniques [4]. Alternativement, les robots mobiles peuvent s'appuyer sur des routeurs qui leur permettent de naviguer sur un chemin de déplacement prédéfini dans un espace relativement contrôlé. Les robots mobiles sont de plus en plus populaires dans les environnements commerciaux et industriels. Les hôpitaux utilisent des robots mobiles autonomes pour transporter des matériaux depuis de nombreuses années [5]. Les entrepôts ont installé des systèmes robotiques mobiles pour déplacer efficacement les matériaux des racks de stockage vers les zones de distribution. Les robots mobiles sont également un axe majeur de la recherche actuelle, et presque toutes les grandes universités ont un ou plusieurs laboratoires axés sur la recherche en robotique mobile. Les robots mobiles se retrouvent également dans les environnements industriels, militaires et de sécurité. Les robots ménagers sont des produits de consommation, y compris des robots récréatifs qui effectuent certaines tâches ménagères telles que le balayage ou le jardinage [6].

L'objectif de ce mémoire de fin d'étude est d'étudier le contrôle de la navigation autonome d'un robot mobile dans un environnement inconnu on utilise d'une part l'approche comportementale à base de la logique floue, et d'autre part le paradigme des algorithmes méta-heuristiques. Les techniques employées pour aborder ce problème sont basées sur les systèmes d'inférence flous et PID floue, et les algorithmes méta-heuristiques (PSO, GJO). Cette mémoire de fin d'étude est organisée de la manière suivante :

Le premier chapitre est consacré aux généralités des robots mobiles. Un aperçu général sur le domaine de la robotique mobile sera abordé pour examiner en bref les types des robots mobiles.

Dans le deuxième chapitre, on présente une brève présentation sur la modélisation de notre système (robot mobile type véhicule)

Le troisième chapitre est consacré à l'architecture de contrôle avec l'utilisation de la logique floue, PID-floue, contrôleur PID optimisé par l'algorithme PSO et GJO.

Quatrième chapitre : simulation et discussion sur les résultats obtenus

Nous terminerons par une conclusion générale récapitulant ce qui a été fait et expose les perspectives de ce travail.

Chapitre I:
État de l'art sur les robots mobiles

I.1. Introduction

Depuis les années 1960, la robotique mobile a connu une expansion remarquable grâce aux progrès techniques. Ces avancées ont permis la création de robots de plus en plus sophistiqués, équipés de nombreux capteurs, d'une puissance de calcul accrue et de moyens de communication performants. Ainsi, la mobilité autonome des robots est devenue un domaine de recherche très développé dans tous les pays industrialisés. Les applications sont diverses et étendues, incluant des robots mobiles à pattes, à roues, sous-marins et aériens. Ces applications couvrent un large éventail de fonctions telles que les services robotisés, la surveillance, la construction, le nettoyage, la manipulation de charges, l'automobile intelligente, les interventions d'urgence, l'exploration planétaire et sous-marine, ainsi que les applications militaires et satellitaires. Ce développement a ouvert un marché potentiel considérable pour la robotique, bien que cela nécessite de résoudre des défis technologiques fondamentaux en route vers la création de machines intelligentes [7].

Ce chapitre vise à fournir une introduction approfondie aux robots mobiles. Nous explorerons plusieurs points, et compte tenu que les robots mobiles à roues sont considérés comme la majorité des robots mobiles, nous nous risquerons à décrire en détail les robots mobiles à roues et à aborder les questions liées à leur mobilité autonome.

I.2. L'origine du mot robot

La robotique est une science qui étudie toutes les techniques qui permettent la conception et la réalisation de la robotique. C'est un domaine multidisciplinaire, avec des aspects liés à la mécanique et à l'informatique ainsi qu'à l'électronique [8].

Si le mot « robot » est devenu courant aujourd'hui, ses origines remontent à la pièce R.U.R., « Le robot universel de Rosum » de l'écrivain tchèque Karel Capek. Capek a utilisé ce terme pour la première fois en 1920 dans sa pièce pour décrire un objet artificiel capable de faire son travail tout seul. En conséquence, ils éliminent la main d'œuvre de la chaîne de production, réduisant ainsi le coût des marchandises vendues dans l'industrie textile. Elle a été choisie par Čapek sur suggestion de son frère Josef Čapek. En tchèque, le mot est dérivé de robota, qui signifie travail forcé ou pénibilité et est utilisé pour désigner ces machines comme on les appelle aujourd'hui. C'est un robot ou un humanoïde [9].

I.3. Dates marquantes de la robotique

L'histoire de la robotique est une phase importante et fascinante, pleine de créativité. De nombreuses années ont été témoins de moments déterminants qui ont contribué de manière significative au développement de ce domaine, de l'émergence des premières conceptions de robots aux avancées technologiques contemporaines. Nous constatons que l'histoire a façonné la robotique. À bien des égards, elle a contribué à changer notre monde aujourd'hui, et parmi les plus importantes de ces dates figurent :

- Avant le XVIIIe siècle : Invention de l'automate.
- 1738 Jacques Vaucanson création d'un canard en bronze articulé qui peut boire, manger et cancaner.
- En 1921 : Karel Capek a utilisé le terme 'robot' pour la première fois, RUR (Rossum's Universal Robots).
- En 1941: l'écrivain américain Isaac Asimov a employé le terme robotique pour la première fois dans sa nouvelle 'Liar!' et les trois lois de la robotique ont été introduites en 1942.
- Dans les années 1950, les transistors et les circuits intégrés ont permis la création de robots industriels, tels que l'UNIMATE de General Motors et la première ligne de production de robots de Nissan en 1972.
- Entre les années 1960 et 1980, des laboratoires d'intelligence artificielle ont été créés et les premiers robots mobiles ont été développés.
- La robotique mobile a connu un essor dans les années 1990 et 2000, avec la création de compétitions annuelles et l'envoi de robots mobiles sur Mars.

Depuis 2000, la robotique s'est démocratisée avec l'introduction d'Asimo et l'utilisation de drones dans le monde réel [10].

I.4. Robot mobile

Un robot mobile est un système complexe combinant mécanique, électronique et informatique pour interagir physiquement avec son environnement en vue d'atteindre des objectifs définis. Ce type de machine est polyvalent et peut ajuster ses actions en réponse à des variations dans ses conditions de fonctionnement. Équipé de capacités de perception, de prise de décision et d'exécution, il est capable d'accomplir diverses tâches de manière flexible, même dans des situations imprévues [11].

Le robot mobile est conçu pour se déplacer dans son espace de travail grâce à divers moyens. Selon son niveau d'autonomie et d'intelligence, il peut être équipé de capteurs et de capacités de raisonnement. Certains robots peuvent, avec peu d'intervention humaine, cartographier leur environnement et planifier des trajets même dans des environnements inconnus. Actuellement, les robots mobiles les plus avancés sont principalement destinés à des applications dans des environnements variables ou incertains, souvent remplis d'obstacles, nécessitant une grande adaptabilité [11].

I.5. Classification des robots mobiles

Il existe de nombreux types de robots mobiles et les robots peuvent être classés selon plusieurs critères, dont trois sont principalement [11]:

- Degré d'autonomie.
- Domaine d'application.
- Système de locomotion.

I.5.1. Classification par degré d'autonomie

Un robot autonome est un système équipé de capacités décisionnelles et de moyens pour acquérir et traiter des données, lui permettant d'accomplir un ensemble de tâches avec peu ou aucun contrôle humain, que ce soit dans un environnement familier ou inconnu. En fonction de ce critère, les robots mobiles peuvent être classés comme suit [11]:

I.5.1.1. Robot télécommandé

Robot téléopéré par un opérateur, qu'il soit machine ou humain, qui supervise et commande chaque tâche de base à effectuer, telles que avancer, reculer et tourner.

I.5.1.2. Robots semi-autonomes

Ce type de robot exécute un certain nombre de tâches de manière totalement autonome, mais peut être interrompu à tout moment pour recevoir de nouvelles commandes de l'opérateur.

I.5.1.3. Robots autonomes

Un robot est considéré comme totalement autonome s'il peut adapter son comportement à l'environnement dans lequel il évolue. L'autonomie est la capacité d'un système sans pilote à sentir, percevoir, analyser, communiquer, planifier, décider et agir pour atteindre les objectifs assignés au moment de sa conception.

I.5.2. Classification par domaine d'application

Le plus grand avantage des robots mobiles est que leurs champs d'application sont illimités:

I.5.2.1. Robots industriels et robots de service

Robots industriels : les robots industriels sont officiellement définis par l'ISO comme des commandes de manipulation polyvalentes programmables sur trois axes ou plus et pouvant être reprogrammées automatiquement. Les applications typiques sont les robots de soudage, les robots de peinture et les robots d'assemblage. Les robots industriels inspectent les produits rapidement et avec précision. Les robots industriels sont largement utilisés dans l'industrie automobile et leur conception nécessite de nombreuses connaissances et un très haut degré d'expertise.

La conception des robots industriels requiert de nombreuses connaissances et un très haut niveau d'ingénierie. Quant aux robots de service, ils sont conçus pour assister les personnes à mobilité réduite dans leur vie quotidienne, guider les aveugles et conduire des voitures autonomes.

I.5.2.2. Robots militaires

Les robots militaires, également connus sous le nom d'armes autonomes, sont des robots autonomes télécommandables conçus pour des applications militaires. Les drones sont une sous-classe des robots militaires. Les robots mobiles ont de nombreuses applications militaires. Ce secteur bénéficie de spécifications améliorées telles que la vitesse du véhicule, la capacité d'élimination des obstacles (la robustesse est essentielle pour ce type de robot) et le temps de réaction, ce qui en fait des robots très performants.

I.5.2.3. Les robots expérimentaux

Un certain nombre de laboratoires travaillent dans le domaine de la robotique pour valider les études théoriques sur la perception et la planification des mouvements. La robotique de laboratoire fait référence à l'utilisation de robots dans les laboratoires de biologie et de chimie. Par exemple, les entreprises pharmaceutiques utilisent des robots pour synthétiser de nouveaux produits chimiques et médicaments afin de tester leur valeur chimique.

I.5.3. Classification selon le type de locomotion

I.5.3.1. Les robots mobiles à chenilles

Les systèmes de chenilles, similaires à ceux des chars d'assaut, sont adaptés aux environnements extérieurs et aux surfaces molles comme le sable et le gravier. Bien qu'elles n'augmentent pas la puissance, les chenilles réduisent le glissement et répartissent

uniformément le poids du robot. De plus, elles confèrent souvent au robot un aspect "agressif" [12].

I.5.3.2. Les robots mobiles marcheurs

Un nombre croissant de robots utilisent des membres articulés pour se déplacer, principalement des jambes adaptées aux terrains accidentés. Les robots amateurs sont souvent équipés de six pattes pour maintenir une stabilité statique constante, toujours en équilibre sur trois pattes. Cependant, stabiliser les robots avec moins de jambes s'avère plus difficile. Les configurations bipèdes (deux jambes), quadrupèdes (quatre jambes) et hexapèdes (six jambes) sont courantes, tandis que les chercheurs explorent également des modèles monopèdes (une jambe sauteuse) [12].

I.5.3.3. Les robots mobiles Aérienne

Les véhicules aériens autonomes sans pilote (AUAV) captivent l'intérêt des amateurs de robotique, malgré les risques associés pour les débutants. Les dirigeables et les drones à haute altitude sont envisagés pour des applications de communication, tandis que les passionnés utilisent principalement des avions télécommandés commerciaux. À titre d'exemple, les avions comme le Predator de l'armée américaine ont évolué vers une autonomie accrue au fil des ans, initialement conçus comme semi-autonomes [12].

I.5.3.4. Les robots mobiles à base à d'eau

De plus en plus d'amateurs, d'institutions et d'entreprises investissent dans le développement de véhicules sous-marins autonomes, bien que des défis subsistent pour les rendre attractifs pour toute la communauté des roboticiens. Malgré cela, plusieurs entreprises ont récemment lancé des robots nettoyeurs de piscine sur le marché. Les véhicules sous-marins utilisent divers moyens pour s'immerger dans l'eau, tels que le ballast (air comprimé ou compartiments immergés), les propulseurs, les ailerons de queue et même les ailes [12].

I.5.3.5. Les robots mobiles à roues

La mobilité par roues est la structure mécanique la plus utilisée en robotique mobile. Ce type de robot assure un déplacement aisé, 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 plus tard les classes principales de robots à roues [13].

I.6. L'architecture des robots mobiles à roues

La figure (I.1) montre l'architecture d'un robot mobile à roues qui se compose généralement de 4 éléments de base, c'est-à-dire [14]:

- La motricité et l'énergie.
- Structure mécanique.
- Les organes de sécurité.
- Le système de traitement des informations et gestion des tâches.

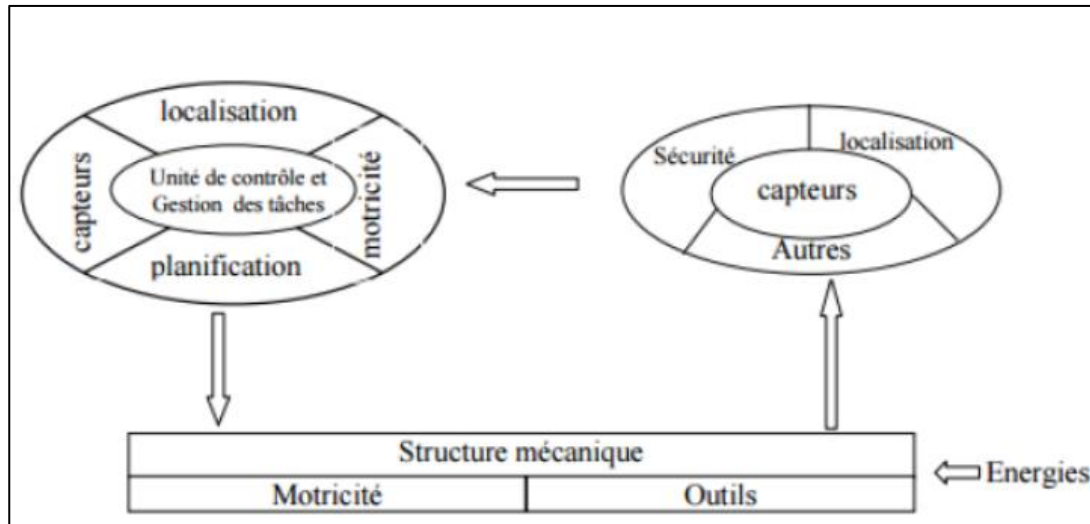


Figure I.1 : Architecture d'un robot mobile à roues [14].

I.6.1. La motricité et l'énergie

Les déplacements des robots à roues sont réalisés par des moteurs de types électrique. L'énergie électrique la plus fréquemment employée offre l'avantage d'une commande aisée. Par contre le transport et la génération présentent des difficultés, donc elle est utilisée batteries qui sont soit rechargées périodiquement de manière automatique ou manuelle, soit par un échange avec d'autres lorsqu'elles sont déchargées [14].

I.6.2. Les organes de sécurité

Les robots peuvent être utilisés à proximité du personnel, en fonction de la tâche confiée. C'est pourquoi il est obligatoire d'équiper l'orgue d'une garantie de sécurité. Le capteur est entièrement disponible pour détecter les obstacles sur une zone aussi large que possible autour du mobile. Deux types de capteurs sont utilisés: les capteurs proximétriques (ultra sonore, hyperfréquence, infrarouge) pour assurer la détection avant collision...Capteur de contact (contact électrique de pare-chocs, résistance variable, fibre optique) et pour détecter une

collision ou un choc avec l'environnement...). Ce sont des dispositifs redondants par rapport aux capteurs précédents [14].

I.6.3. Traitement des informations et gestion des tâches

L'ensemble de traitement de l'information et de gestion des tâches constitue un module d'information central qui établit des commandes qui permettent au corps mobile d'effectuer un déplacement et d'activer divers organes en fonction de la finalité. Nous abordons le problème de la génération d'un plan consistant à établir comment le robot se déplace par rapport à des connaissances a priori (statiques) ou acquises au cours de l'évolution (dynamiques). Plus de générations sont basées sur 3 concepts [14] :

- Stratégie de navigation
- Modélisation spatiale
- Évitement d'obstacles

I.7. Les types des robots mobiles à roues

I.7.1. Robots unicycle

Les robots de type unicycle utilisent des capteurs d'attitude (gyroscope) et de déplacement (accéléromètre) pour assurer leur stabilité et sont entraînés par une roue ou deux roues indépendantes. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. Ils sont des systèmes non autonomes et sont donc définis comme des robots qui se déplacent dans un plan bidimensionnel à une vitesse constante vers l'avant, mais pas dans une direction latérale instantanée. Les robots monocycles sont des systèmes non autonomes et ne peuvent donc pas se déplacer dans une direction perpendiculaire à leurs roues [15].

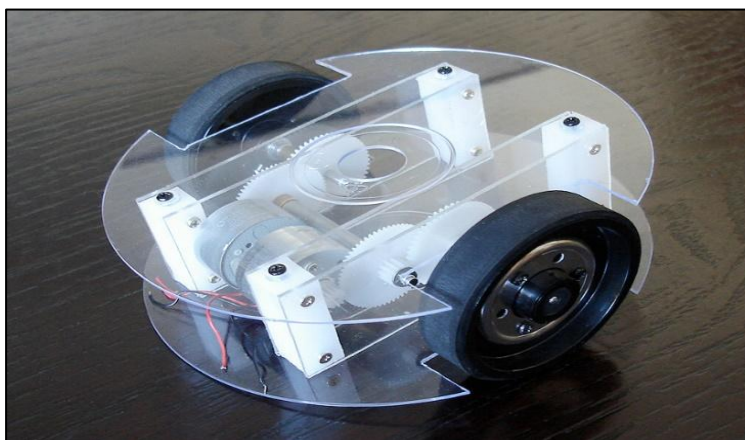


Figure I.2 : Robot de type unicycle [16].

I.7.2. Robots tricycle

Un robot de type tricycle se compose de deux roues fixes placées sur le même axe et d'une roue libre située au centre qui peut se diriger le long de l'axe longitudinal du robot. Le mouvement du robot est donné par la vitesse des roues fixes et son orientation est assurée par les roues libres. Il s'agit également d'un robot non autonome. Il ne peut pas se déplacer dans une direction perpendiculaire aux roues fixes [15].



Figure I.3: Robot de type tricycle [17].

I.7.3. Robots voiture

Les robots de type voiture sont similaires aux robots à trois roues, sauf qu'ils ont deux roues fixes sur le même essieu et deux roues centrales orientables sur le même essieu. Toutefois, le robot de type voiture est plus stable en raison des points d'appui supplémentaires. Toutes les autres caractéristiques du robot de type voiture sont les mêmes que celles du robot à trois roues [15].

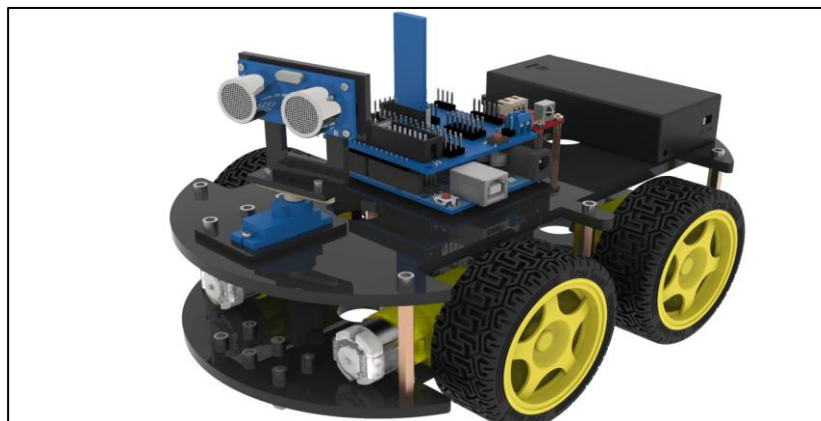


Figure I.4: Robot de type voiture [18].

L'un des plus grands projets dans ce domaine est le projet Google Self-Driving Car (**voir la figure I.5**), qui consiste en des véhicules entièrement autonomes équipés de capteurs ayant un champ de vision très large allant jusqu'à 800 mètres. Cela signifie qu'ils peuvent détecter toutes sortes d'objets (obstacles), qu'il s'agisse de véhicules, de piétons ou de sacs en plastique. Ces informations contribuent de manière significative à une navigation sûre et durable sans intervention humaine [15].

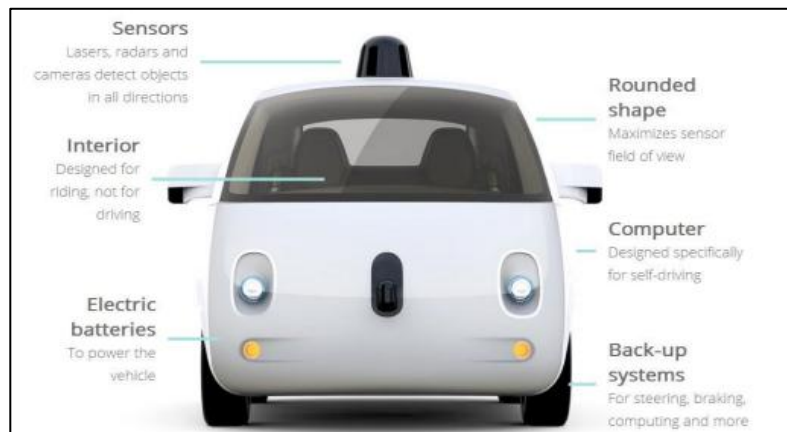


Figure I.5: Google Self-Driving Car Project [15].

I.7.4. Robots mobiles omnidirectionnels

Un robot mobile est dit omnidirectionnel s'il peut agir indépendamment à une vitesse de translation le long des axes x et y et à une vitesse de rotation autour de l'axe z . Il peut se déplacer librement dans toutes les directions. Il se compose généralement de trois roues orientables décentrées, disposées de manière à former un triangle équilatéral. Le principal avantage des robots omnidirectionnels est qu'ils peuvent se déplacer dans toutes les directions. Toutefois, cela se fait au prix d'une complexité mécanique considérablement accrue par rapport à d'autres types de robots mobiles [15].



Figure I.6: Robot de type omnidirectionnel [19].

I.7.5. Disposition des roues

La roue est de loin le mécanisme de locomotion le plus populaire dans la robotique mobile et dans les véhicules artificiels en général. Il peut atteindre de très bons rendements, comme le démontre la figure (I.7), et ce avec une mise en œuvre mécanique relativement simple. De plus, l'équilibre n'est généralement pas un problème de recherche dans la conception de robots à roues, car les robots à roues sont presque toujours conçus de manière à ce que toutes les roues soient à tout moment en contact avec le sol. Ainsi, trois roues suffisent pour garantir un équilibre stable, même si, comme nous le verrons ci-dessous, les robots à deux roues peuvent également être stables. Lorsque plus de trois roues sont utilisées, un système de suspension est nécessaire pour permettre à toutes les roues de maintenir le contact avec le sol lorsque le robot rencontre un terrain accidenté. Au lieu de se soucier de l'équilibre, la recherche sur les robots à roues a tendance à se concentrer sur les problèmes de traction et de stabilité, de maniabilité et de contrôle : les roues du robot peuvent-elles fournir [20].

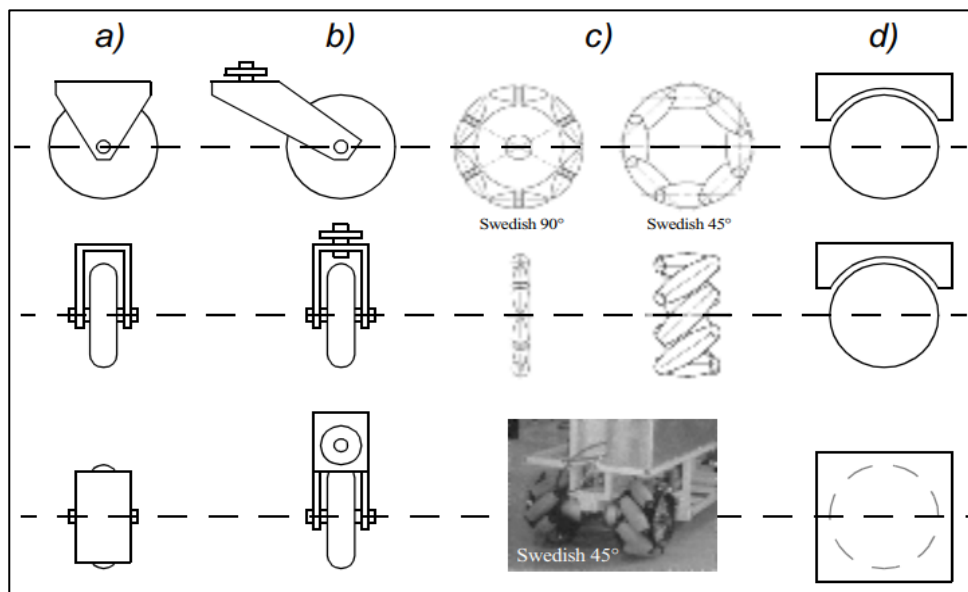



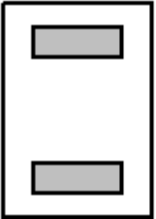
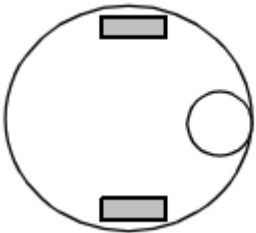
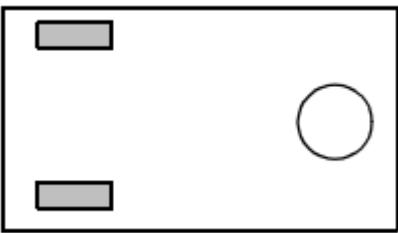
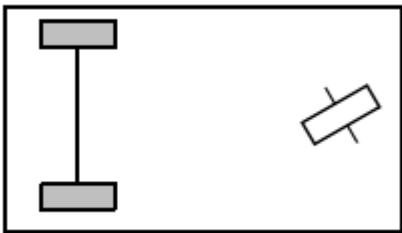
Figure I.7: Les types des roues [20].

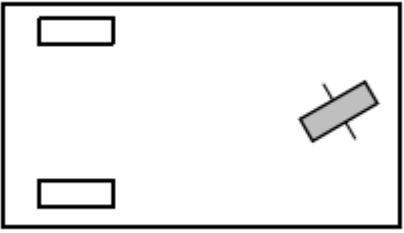
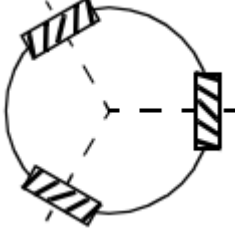
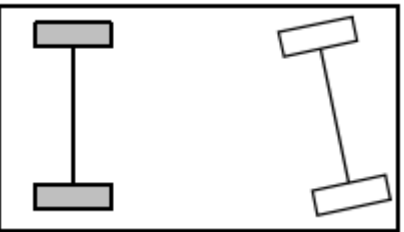
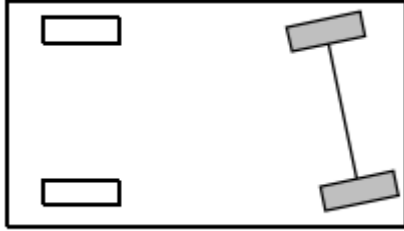
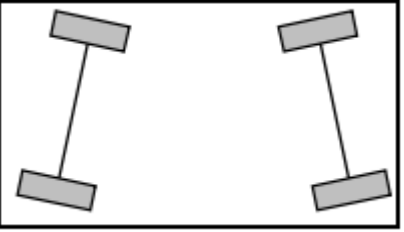
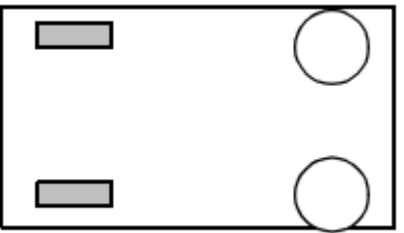
Les quatre types de roues de base [20]:

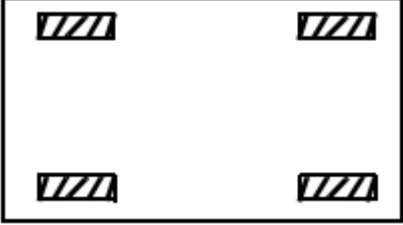
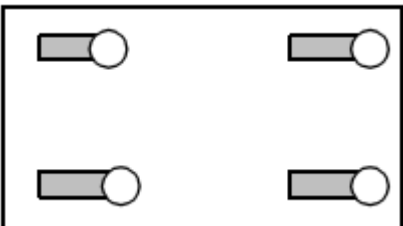
- A. **Roue standard** : deux degrés de liberté ; rotation autour de l'axe de roue (motorisé) et du point de contact.
- B. **Roue pivotante** : deux degrés de liberté ; rotation autour d'un joint de direction décalé.
- C. **Roue suédoise** : trois degrés de liberté ; rotation autour de l'axe de roue (motorisé), autour des rouleaux et autour du point de contact.
- D. **Boule ou roue sphérique** : réalisation techniquement difficile.

Configurations des roues pour les robots mobiles a roues [20] :

Tableau I.1 : Configurations des roues pour véhicules roulants.








# des roues	Arrangement	Description	Exemples typiques
2		Un volant à l'avant, une roue de traction à l'arrière	Vélo, moto
		Entraînement différentiel à deux roues avec centre de masse (COM) sous l'essieu	Robot personnel Cye
3		Entraînement différentiel centré sur deux roues avec un troisième point de contact.	Nomad Scout, smartRob EPFL
		Deux roues motrices indépendantes à l'arrière/avant, 1 roue omnidirectionnelle non motorisée à l'avant/arrière	De nombreux robots d'intérieur, dont les robots de l'EPFL Pygmalion et Alice
		Deux roues motrices reliées (différentielles) à l'arrière, 1 roue libre directrice à l'avant.	Mini-camions Piaggio

		<p>Deux roues libres à l'arrière, 1 roue motrice directrice à l'avant.</p>	<p>Neptune (Université Carnegie Mellon), Hero-1</p>
		<p>Trois roues suédoises ou sphériques motorisées disposées en triangle : un mouvement omnidirectionnel est possible.</p>	<p>Roue Stanford Tribolo EPFL, Kit Robot Pilote Palm (CMU)</p>
<p>4</p>		<p>Deux roues motorisées à l'arrière, 2 roues directrices à l'avant ; la direction doit être différente pour les 2 roues pour éviter de glisser/déraper.</p>	<p>Voiture à propulsion arrière</p>
		<p>Deux roues motorisées et directrices à l'avant, 2 roues libres à l'arrière ; la direction doit être différente pour les 2 roues pour éviter de glisser/déraper.</p>	<p>Voiture à traction avant</p>
		<p>Quatre roues directrices et motorisées</p>	<p>Quatre roues motrices, quatre roues directrices Hypérion (CMU)</p>
		<p>Deux roues motrices (différentielles) à l'arrière/avant, 2 roues omnidirectionnelles à l'avant/arrière</p>	<p>Charlie (DMT-EPFL)</p>

		Quatre roues omnidirectionnelles	Carnegie Mellon Uranus
		Quatre roulettes motorisées et directrices	Nomade XR4000

Les icônes pour chaque type de roue sont les suivantes :

Tableau I.2 : Les icônes pour chaque type de roue.

	roue omnidirectionnelle non motorisée (sphérique, roulette, suédoise).
	roue suédoise motorisée (roue Stanford).
	roue standard non motorisée.
	roue standard motorisée.
	roulette motorisée et directrice.
	roue standard directrice.
	roues connectées.

I.8. Les avantages et les inconvénients des différents types des robots à roues

Nous allons résumer les avantages et les inconvénients les plus importants des différents types de robots à roues dans le tableau ci-dessous.

Tableau I.3 : les avantages et les inconvénients des différents types de robots à roues [15].

Type du robot	Avantage	Inconvénient
Unicycle	<ul style="list-style-type: none"> • Rotation sur soi-même • Complexité mécanique faible 	<ul style="list-style-type: none"> • Non-holonyme
Tricycle	<ul style="list-style-type: none"> • Complexité mécanique modérée 	<ul style="list-style-type: none"> • Non-holonyme • Peu stable • Pas de rotation sur soi-même
Voiture	<ul style="list-style-type: none"> • Stable • Complexité mécanique modérée 	<ul style="list-style-type: none"> • Non-holonyme • Pas de rotation sur soi-même
Omnidirectionnel	<ul style="list-style-type: none"> • Holonyme • Stable • Rotation sur soi-même 	<ul style="list-style-type: none"> • Complexité mécanique importante

I.9. Caractéristiques des robots mobiles a roues

Les robots doivent être sélectionnés en fonction de l'usage auquel ils sont destinés.

Certains paramètres à prendre en compte sont énumérés ci-dessous [21]:

I.9.1. La charge maximale transportable

Charge maximale transportable de quelques kilos à plusieurs tonnes, déterminée par les conditions les plus défavorables (extension maximale). Un robot de livraison autonome dans un entrepôt peut transporter plusieurs boîtes à la fois avec une capacité maximale de quelques centaines de kilogrammes, assurant une conduite sûre et efficace. En revanche, un véhicule utilitaire autonome pour des applications industrielles peut transporter des charges beaucoup plus lourdes, allant jusqu'à plusieurs tonnes sur de longues distances, adapté à des environnements contrôlés comme les sites de fabrication ou les ports. Ainsi, la capacité maximale transportable d'un robot dépend de son usage spécifique et de sa conception.

I.9.2. L'espace de travail

Défini comme l'ensemble des points pouvant être atteints par l'appareil terminal. Toutes les opérations ne sont pas possibles en tous points du volume de travail. L'espace de travail

atteignable (volume), également appelé espace de travail maximal, est le volume d'espace que le robot peut atteindre dans au moins une direction. L'espace de travail dextre est le volume de l'espace que le robot peut atteindre dans toutes les directions possibles. Cet espace de travail est un sous-ensemble de l'espace de travail maximal.

I.9.3. Le positionnement absolu

Positionnement absolu correspond à l'erreur entre le point désiré (réel) et le point atteint, défini par la position et l'orientation dans l'espace cartésien. Il est calculé à partir du modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de la position et à la flexibilité du système de la machine.

I.9.4. Vitesse de déplacement

Vitesse maximale à l'extension maximale ou à l'accélération maximale. Les autres caractéristiques sont les suivantes :

- Masse du robot.
- Coût du robot.

I.10. Conclusion

Ce chapitre a été consacré à un état de l'art sur la robotique mobile d'une façon générale et la robotique mobile à roues en particulier. Nous avons présenté brièvement les différents domaines d'interventions de ce type de robot avec les avantages et les inconvénients. Une modélisation du comportement du robot mobile à roues sera présentée dans le chapitre suivant.

Chapitre II:
Modélisation des robots mobiles
a roues

II.1. Introduction

Les modèles cinématiques des robots mobiles à roues constituent un domaine important de la robotique moderne. Ces modèles visent à décrire comment le mouvement des roues du robot est traduit en déplacements globaux dans l'environnement. De la théorie de base aux applications avancées, ces modèles fournissent une compréhension précise du comportement dynamique des robots à roues et facilitent le développement d'algorithmes de navigation, de contrôle et de planification de trajectoire [22].

Ce chapitre est concentré à la modélisation des robots mobiles à roues, une étape importante dans le processus de conception et d'optimisation. La modélisation représente mathématiquement différents aspects du robot, du système de propulsion à la dynamique du robot, y compris son interaction avec le sol et les conditions de conduite. En comprenant ces aspects, les ingénieurs et les chercheurs peuvent simuler le comportement des robots mobiles à roues dans différents scénarios, évaluer leurs performances et optimiser leur conception pour répondre à des exigences spécifiques en termes d'efficacité énergétique, de sécurité et de confort.

Par conséquent, dans ce chapitre nous modéliserons le robot mobile type véhicule après avoir abordé les bases de la modélisation cinématique. Nous explorerons la forme générale de la modélisation des robots mobiles à roues et examinons les différents composants et systèmes impliqués. Nous décrivons également les hypothèses qui sous-tendent cette modélisation, les aspects relatifs à la cinématique du robot pour faire contrôler de la trajectoire.

II.2. La forme générale de modélisation des robots mobiles

Modéliser un robot consiste à établir un modèle mathématique en le considérant comme un système mécanique articulé, actionné et contrôlé. Outre sa fonction générale d'aide à la conception, la modélisation a de nombreuses autres utilisations, telles que la prévision du comportement, l'adaptation des actionneurs, la planification des tâches, l'établissement des lois de commande et l'intégration des robots dans des simulations informatiques. D'une manière générale, la modélisation précède la simulation, mais les deux activités ne peuvent être clairement séparées [23].

Les types de locomotion définissent deux types de contraintes :

- Les contraintes cinématiques concernent la géométrie des mouvements possibles du robot et sa position.
- Les contraintes dynamiques sont liées aux effets du mouvement (accélération limitée, vitesse limitée, présence de forces d'inertie ou de frottement).

En fonction de sa cinématique, le robot est :

- Holonome s'il peut se déplacer instantanément dans toutes les directions.
- Non-holonome si le mouvement donné au robot est une courbe à courbure limitée [13].

Le système de référence du robot est tridimensionnel, prenant en compte sa position et son orientation sur un plan, et est représenté par les coordonnées $\{X_R, Y_R, \theta\}$. Les axes $\{X_I, Y_I\}$ définissent le système de référence inertiel global et l'origine est notée O. La différence angulaire entre le système de référence global et le système de référence du robot est représentée par l'angle θ . Enfin, les coordonnées (x, y) sont utilisées pour localiser le point P sur le châssis du robot dans le système de référence global [23].

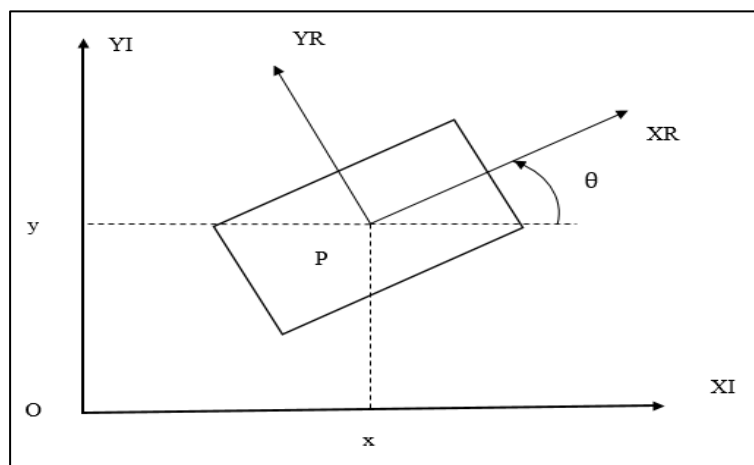


Figure II.1: Le référentiel global et le référentiel local du robot [23].

Une matrice de rotation cartésienne est utilisée pour transformer le mouvement du cadre de référence global $\{X_I, Y_I\}$ en mouvement du cadre de référence local du robot $\{X_R, Y_R\}$. Cela permet des transformations de coordonnées précises et exactes et garantit que les mouvements du robot peuvent être positionnés de manière fiable.

$$R(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{II.1})$$

Une matrice de rotation cartésienne est utilisée pour convertir la vitesse du robot dans le cadre de référence global en composantes de son mouvement le long de ses axes locaux {XR, YR}. Cette matrice transforme la vitesse en coordonnées locales du robot, ce qui permet de mieux comprendre la direction du mouvement du robot [23].

$$\mathcal{E}'_r = R(\theta). \mathcal{E}'_i = R(\theta). [x' \ y' \ \theta']^T \quad (\text{II.2})$$

II.3. Concepts et systèmes des directions

Le châssis d'un robot peut être équipé a priori avec quatre types des roues principales, ces quatre types de roues sont mentionné dans le premier chapitre. Il est très important de choisir le bon type des roues pour éviter le glissement sur le sol et de répondre aux exigences spécifiques de direction et de stabilité des robots. Pour cela, toutes les roues doivent avoir un point de vitesse nulle autour duquel le robot orbite, ce qui appelle centre de rotation instantané(CIR) [24].

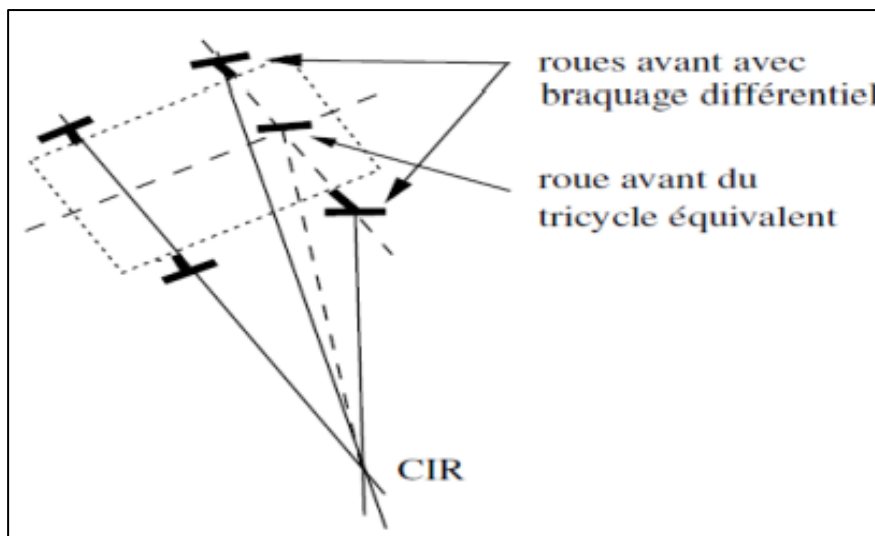


Figure II.2: Robot mobile de type voiture et son CIR [24].

Pour garantir un seul centre de rotation instantané (CIR), les axes des roues du robot doivent se croiser en un seul point. Dans le cas des robots de type voiture, un système de direction différentielle (système Ackermann) est utilisé pour orienter les roues avant de manière à ce que le CIR soit unique. De plus, les vitesses sont différentes et interdépendantes en raison des différents rayons de courbure des roues. L'équivalence entre les voitures et les tricycles peut être établie en introduisant une roue virtuelle positionnée au centre de l'essieu avant de la voiture, de sorte que le CIR reste inchangé [24].

II.4. Roulement sans glissement et contraintes non holonomes

II.4.1. Roulement sans glissement (r.s.g.)

La locomotion sur roues utilise le frottement entre les roues et le sol à l'endroit où elles entrent en contact l'une avec l'autre. Pour ce faire, la nature du contact (régularité, matériau de contact) influence fortement les caractéristiques du mouvement de la roue par rapport au sol. Dans de bonnes conditions, la roue roule sans glisser sur le sol (r.s.g.), c'est-à-dire que la vitesse relative de la roue par rapport au sol au point de contact est nulle. Théoriquement, pour vérifier cette condition, les hypothèses suivantes doivent être remplies :

- Le contact entre la roue et le sol est ponctuel.
- La roue est invariante de rayon r .
- Le sol est parfaitement plat.

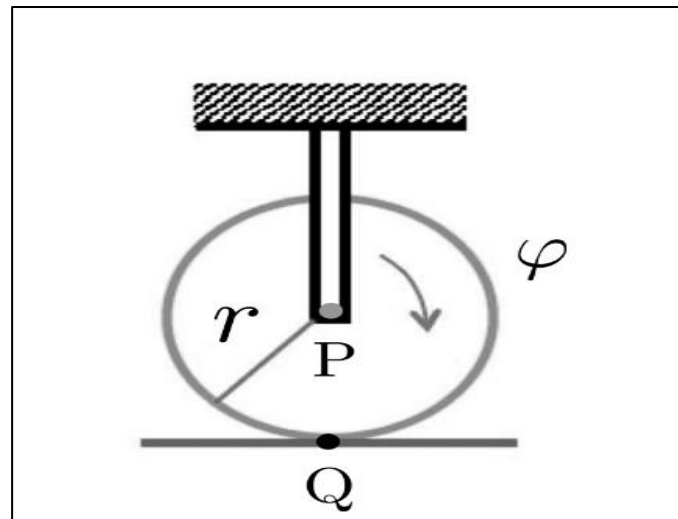


Figure II.3 : Schéma de contact entre la roue et le sol [25].

En pratique, le contact a lieu en surface, provoquant apparemment un léger glissement. De même, il est raisonnable de dire qu'une roue pleine est indéformable, mais pour une roue sans pneus, cette hypothèse est presque toujours fautive. Néanmoins, nous supposons toujours qu'il existe une force centrifuge et que le sol est parfaitement plat [25].

La condition d'une roue fixe [25]:

- La composante de la vitesse dans la direction de mouvement de la roue est égale à la vitesse de roulement (à savoir,)
- La composante de la vitesse dans la direction perpendiculaire au mouvement de la roue est zéro

Conditions de r.s.g d'une roue fixe [25]:

- La composante de vitesse dans la direction du mouvement de la roue est égale à la vitesse de roulement (c'est-à-dire, $r\dot{\varphi}$).
- La composante de vitesse v_n dans la direction perpendiculaire au mouvement de la roue est nulle .

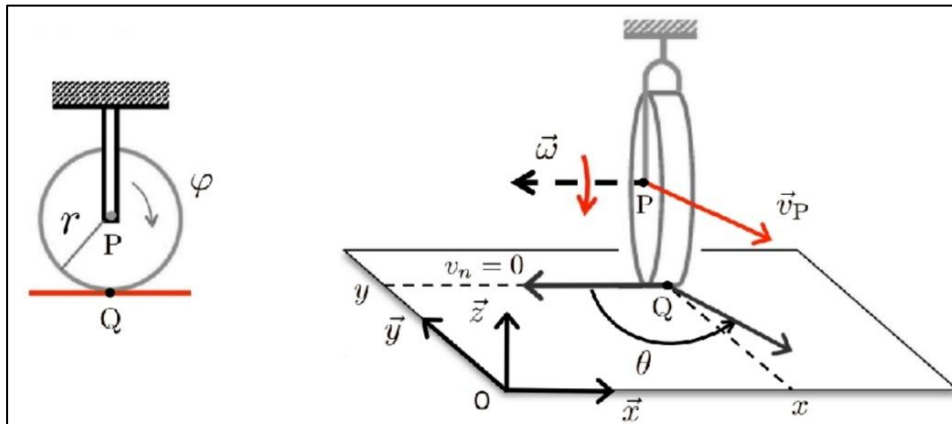


Figure II.4 : Caractérisation du roulement sans glissement [25].

$$\vec{v}_Q = \vec{v}_P + \vec{\omega} \times \overline{PQ} = \vec{0} \quad (\text{II.3})$$

$P = (x, y, r)$: centre de la roue

$Q = (x, y, 0)$: le point de contact de la roue avec le sol

$\vec{\omega}$: vecteur vitesse de rotation de la roue

φ : angle de rotation propre de la roue

θ : orientation du plan de la roue

En utilisant l'équation précédente pour les points P et Q, on obtient :

$$\dot{x} \vec{x} + \dot{y} \vec{y} + (\dot{\theta} \vec{x} + \dot{\varphi} (\vec{x} \sin(\theta) - \vec{y} \cos(\theta))) \times (-r \vec{z}) = \vec{0} \quad (\text{II.4})$$

Le calcul du produit vectoriel le montre :

$$(\dot{x} + r \dot{\varphi} \cos(\theta)) \vec{x} + (\dot{y} + r \dot{\varphi} \sin(\theta)) \vec{y} = \vec{0} \quad (\text{II.5})$$

Il en résulte le système de contraintes scalaires suivant :

$$\dot{x} + r\dot{\phi}\cos(\theta) = 0 \quad (\text{II.6})$$

$$\dot{y} + r\dot{\phi}\sin(\theta) = 0 \quad (\text{II.7})$$

Ces contraintes peuvent être transformées pour indiquer la direction perpendiculaire à la roue et la composante de vitesse dans le plan de la roue [25]:

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{y} & r\dot{\phi}\sin(\theta) \\ \dot{x} & r\dot{\phi}\cos(\theta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{II.8})$$

II.4.2. Contraintes non holonômes

Un robot non holonome est celui qui ne peut pas intégrer les équations de contraintes cinématiques dans une contrainte de position. Les contraintes non intégrables sont également appelées " non holonômes ". Ces contraintes réduisent le nombre de mouvements instantanés qu'un robot mobile peut effectuer et affectent donc la forme de la trajectoire qu'il peut suivre. Les robots non holonomes ont moins de paramètres de contrôle que de degrés de liberté. Le concept d'holonomie en robotique est généralement appliqué aux robots mobiles [26].

Les robots non holonomes n'ont que deux degrés de liberté dans le plan :

- **La translation** : avancer ou reculer.
- **Rotation** : tourner à droite ou à gauche.

Il est difficile de ramener un robot non holonome à sa position initiale sans manipulation.

Contrainte de non-glissement est représentée par :

$$-\dot{x}\cos(\theta) + \dot{y}\cos(\theta) = 0 \quad (\text{II.9})$$

Contrainte de roulement pur est représentée par:

$$\dot{x}\cos(\theta) + \dot{y}\sin(\theta) = -r\dot{\phi} \quad (\text{II.10})$$

(II.9) et (II.10) traduisent le fait que \vec{v}_p soit dans le plan de la roue et ait pour module $r\dot{\phi}$: ils s'appellent contraintes non holonômes [25].

II.5. Hypothèses de modélisation

Nous décrivons et discutons les hypothèses de modélisation. Nous avons divisé les données en deux groupes :

Portent sur la structure générale des robots mobiles et sur leur environnement :

- Le robot mobile est traité comme un robot rigide opérant dans un plan plat pour simplifier la modélisation mathématique et l'analyse.
- Les roues conventionnelles sont supposées rigides et de rayon constant pour simplifier les équations de mouvement, éliminant ainsi les variables imprévisibles introduites par les déformations des roues.

Portent sur le contact roues-sol :

- Chaque contact roue-sol est simplifié à un seul point pour faciliter l'interaction entre le robot et son environnement, en se concentrant sur la cinématique sans plonger dans les complexités des interactions roue-sol.
- On suppose que les roues roulent sans glisser sur le sol pour simplifier le contrôle et la prédiction du mouvement du robot, permettant l'utilisation d'équations cinématiques simples pour prédire sa position et son orientation.

Ces hypothèses nous permettent d'accéder à la modélisation mathématique du robot et d'accéder à des modèles cinétiques haute résolution [23].

II.6. Modèle cinématique et système de direction

La cinématique consiste à étudier le mouvement des points, des corps et des systèmes corporels (tels que les robots) sans tenir compte des forces agissant sur ces systèmes. Nous discuterons de certaines configurations des robots mobiles à roues courantes et de leurs modèles cinématiques respectifs utilisés dans les robots mobiles qui utilisent des moteurs pour les propulser.

II.6.1. Robot mobile type unicycle

Le système de direction d'un robot à deux roues peut être réalisé de différentes manières, mais voici la méthode courante :

- *À direction différentielle* : Ce système utilise deux roues motrices placées de chaque côté du robot. Pour tourner, la vitesse de rotation de chaque roue est ajustée de manière opposée : une roue tourne plus vite que l'autre pour créer une différence de vitesse qui

fait tourner le robot. Cela permet au robot de tourner sur place en faisant tourner les roues dans des directions opposées [27].

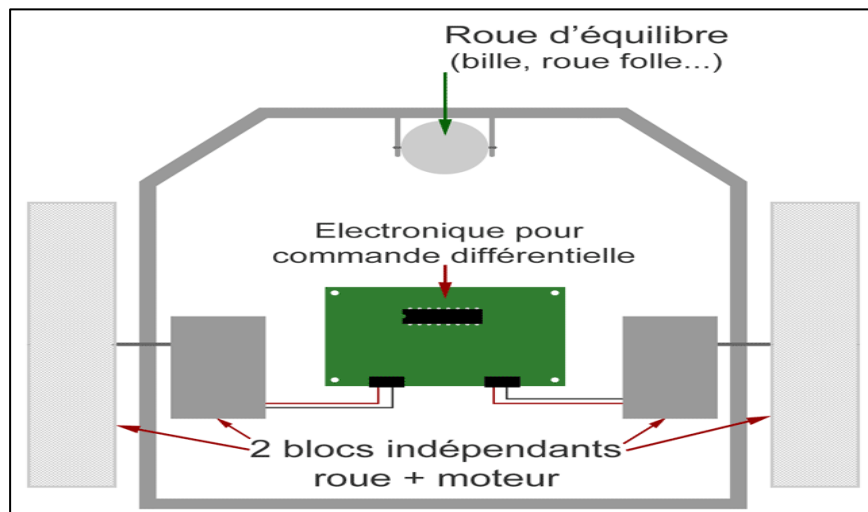


Figure II.5 : Robot mobile à deux roues [27].

Le modèle cinématique d'un robot de type unicycle est souvent utilisé pour représenter la dynamique de robots mobiles à une roue ou de véhicules simplifiés.

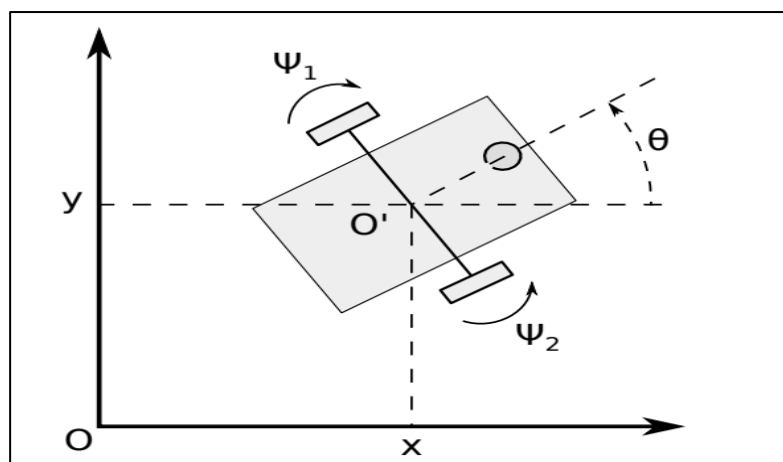


Figure II.6 : Schéma d'un robot de type unicycle [28].

Le vecteur de configuration est :

$$q = [x, y, \theta]^T \quad (\text{II.11})$$

Voici les équations de base du modèle cinématique pour un robot unicycle [28]:

$$\dot{x} = u_1 \cdot \cos(\theta) \quad (\text{II.12})$$

$$\dot{y} = u_1 \cdot \sin(\theta) \quad (\text{II.13})$$

$$\dot{\theta} = u_2 \quad (\text{II.14})$$

Avec : $u_1 = \frac{r}{2}(\Psi_1 + \Psi_2)$; $u_2 = \frac{r}{2L}(\Psi_2 - \Psi_1)$.

Si $\Psi_1 = \Psi_2$ le robot se déplace en ligne droite.

Si $\Psi_1 = -\Psi_2$ le robot tourne sur lui-même.

- (x, y) sont les coordonnées du robot dans le plan.
- (θ) est l'orientation du robot par rapport à l'axe horizontal.
- (u_1) est la vitesse linéaire du robot.
- (u_2) est la vitesse angulaire de rotation du robot.
- (r) le rayon des roues.
- (L) la longueur de l'axe.
- (Ψ_1) Vitesse angulaire de la roue gauche
- (Ψ_2) Vitesse angulaire de la roue droite.

Ce modèle suppose que le robot se déplace dans un plan avec une seule roue de contact qui peut se déplacer vers l'avant ou vers l'arrière et tourner sur place. Il est utile pour la planification de trajectoire et le contrôle de robots mobiles simples.

II.6.2. Robot mobile type tricycle

Le modèle cinématique d'un robot de type tricycle peut être décrit par les équations suivantes. Ce modèle prend en compte une roue avant orientable et deux roues arrière fixes.

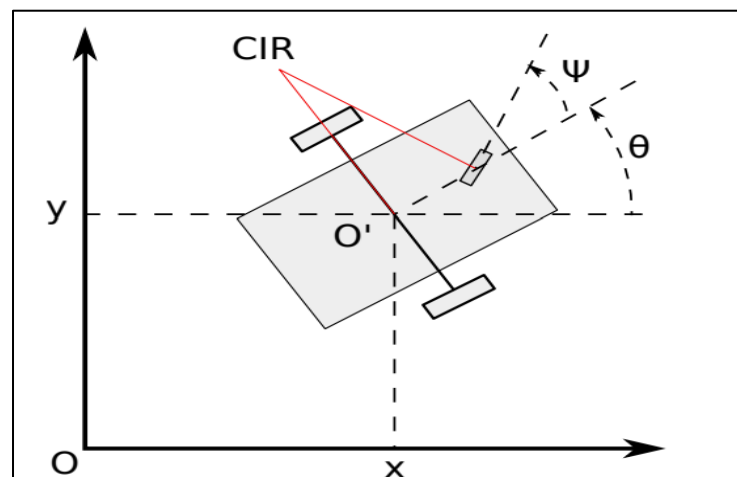


Figure II.7: Schéma d'un robot de type tricycle [28].

Le vecteur de configuration est :

$$q = [x, y, \theta, \Psi]^T \quad (\text{II.15})$$

Voici les équations qui décrivent le mouvement du robot [28]:

$$\dot{x} = u_1 \cdot \cos(\theta) \quad (\text{II.16})$$

$$\dot{y} = u_1 \cdot \sin(\theta) \quad (\text{II.17})$$

$$\dot{\theta} = \frac{u_1}{D} \tan(\Psi) \quad (\text{II.18})$$

$$\dot{\Psi} = u_2 \quad (\text{II.19})$$

- (x, y) sont les coordonnées du robot dans le plan.
- (θ) est l'orientation du robot.
- (u_1) est la vitesse linéaire du robot.
- (D) est la distance entre l'axe des roues arrière et la roue avant.
- (Ψ) la vitesse de rotation du robot.

Le modèle cinématique direct permet de calculer la position et l'orientation du robot dans le plan à partir des vitesses de translation et de rotation de la roue avant. C'est un outil essentiel pour la planification de trajectoire et le contrôle de robots mobiles.

II.6.3. Robot mobile type voiture

Le modèle cinématique d'un robot de type voiture, aussi connu sous le nom de modèle de bicycle ou d'Ackermann, est utilisé pour décrire le mouvement d'un véhicule à quatre roues dont les roues avant sont orientables et les roues arrière sont fixes.

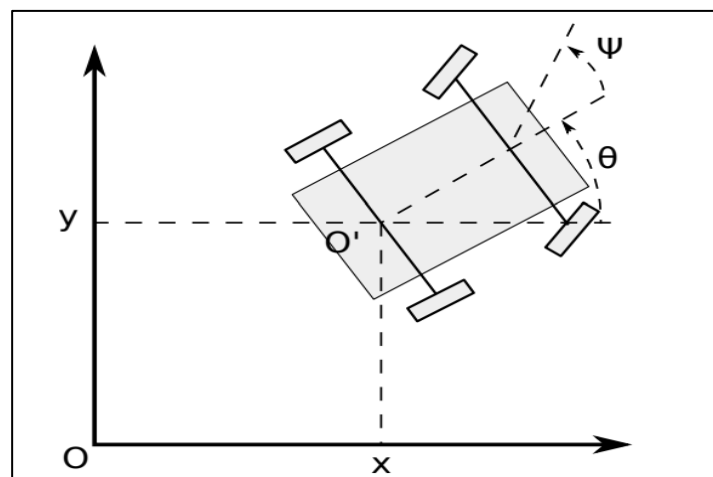


Figure II.8: Schéma d'un robot de type voiture [28].

Voici les équations de base de ce modèle [28]:

$$\dot{x} = u_1 \cdot \cos(\theta) \quad (\text{II.20})$$

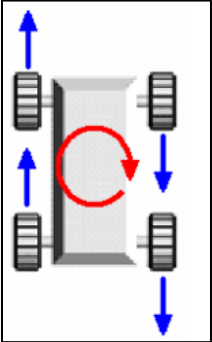
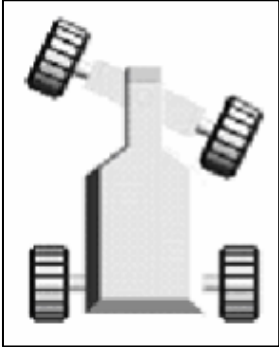
$$\dot{y} = u_1 \cdot \sin(\theta) \quad (\text{II.21})$$

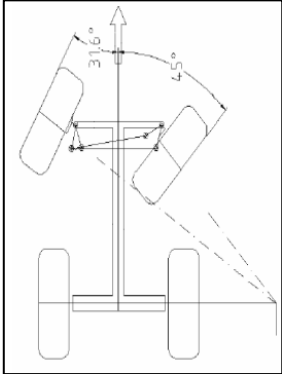
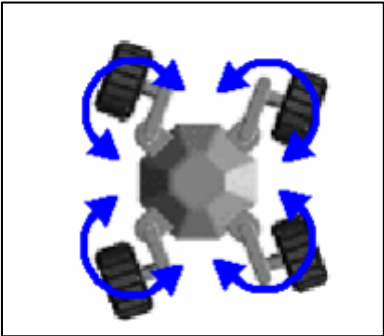
$$\dot{\theta} = \frac{u_1}{D} \tan(\Psi) \quad (\text{II.22})$$

$$\dot{\Psi} = u_2 \quad (\text{II.23})$$

Les systèmes de direction potentiels pour les véhicules robotiques est résumé dans cette tableaux [29]:

Tableau II.1 : les systèmes des directions.

les systèmes	Description	Schéma
Direction par glissement	<ul style="list-style-type: none"> - Utilise des ensembles de roues ou de chenilles parallèles de chaque côté du véhicule. - Les virages sont générés en ajustant les vitesses de rotation des roues de chaque côté. - Offre une grande manœuvrabilité et est compacte. - Convient aux terrains accidentés ou avec des surfaces lâches ou molles. - Nécessite beaucoup de puissance pour les virages et peut entraîner une usure prématurée des pneus ou des chenilles. 	
Direction articulée	<ul style="list-style-type: none"> - Utilise une articulation sur une partie du châssis ou de l'essieu pour tourner. - Plus efficace que la direction par glissement mais a un rayon de braquage plus large, réduisant la manœuvrabilité. - La conception mécanique est simple, mais nécessite un différentiel pour compenser les distances parcourues par les roues intérieures et extérieures. 	
Direction d'Ackerman	<ul style="list-style-type: none"> - Méthode courante utilisée dans les automobiles. - Les pneus tournent à des angles différents de chaque côté du véhicule pour réduire les forces sur les pneus. 	

	<ul style="list-style-type: none"> - Chaque véhicule doit être conçu de manière unique en raison des variations des angles de pneus selon le châssis. - La manœuvrabilité est réduite en raison du large rayon de braquage. 	
Direction indépendante	<ul style="list-style-type: none"> - Chaque roue est dirigée de manière indépendante. - Offre une flexibilité maximale de mouvement et une grande manœuvrabilité, surtout dans des espaces confinés. - Nécessite un contrôle et une précision des angles de direction plus complexes. - Permet des demi-tours en tournant toutes les roues perpendiculairement aux lignes radiales, et la capacité à "crab steer" (diriger en crabe). - Requier quatre moteurs pour la direction et l'entraînement des roues sur un véhicule à quatre roues. 	

II.7. Modélisation de notre modèle

Le modèle cinématique de véhicule est donné par [30]:

$$\dot{x} = v_{moy} \cdot \cos(\Psi + 2\delta_r) \quad (\text{II.24})$$

$$\dot{y} = v_{moy} \cdot \sin(\Psi + 2\delta_r) \quad (\text{II.25})$$

$$\dot{\theta} = v_{moy} \cdot \sin(2\delta_f) / L \cdot \cos(2\delta_f) \quad (\text{II.26})$$

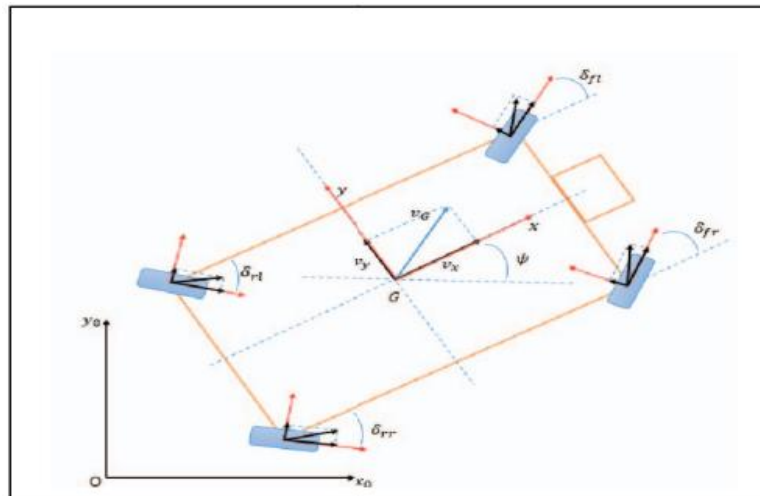


Figure II.9 : Modèle cinématique de véhicule [30].

Tableau II.2 : Paramètres utilisés dans la modélisation.

Ψ	Angle de lacet du véhicule
v_G	Le vecteur de vitesse au centre de gravité
v_x, v_y	La vitesse longitudinale et latérale
δ_f, δ_r	Angle de direction de la roue

Avec : Ψ : Angle du véhicule; δ_f, δ_r Angle de direction ; v_{moy} : vitesse moyenne.

Le modèle cinématique donné avec un système de direction double prend en compte trois degrés de liberté :

1. La translation longitudinale
2. La translation Latérale
3. La rotation autour de la course.

Dans cette modélisation a été utilisé un mode de direction simple de véhicule, nous considérons le cas d'un robot se déplaçant dans un plan 2D avec un référentiel global $[X, Y, \theta]$. Dans ce référentiel cartésien, la véhicule est représenté par une position donnée par les coordonnées (x, y) et une angle de lacet θ sur le plan, comme le montre la Figure (II.9) [30].

Le modèle cinématique d'un tel robot est défini par les équations suivantes [30]:

$$\dot{x} = v_1 \cdot \cos(\theta) \quad (\text{II.27})$$

$$\dot{y} = v_1 \cdot \sin(\theta) \quad (\text{II.28})$$

$$\dot{\theta} = v_2 \quad (\text{II.29})$$

Où : v_1 : vitesse de translation ; v_2 : vitesse de rotation.

Pour maintenir le robot mobile sur notre trajectoire désirée, il est nécessaire de concevoir un régulateur qui permettra de suivre des trajectoires arbitraires $(x_r(t), y_r(t))$. La conception du contrôleur que nous utiliserons doit recevoir les valeurs de distance et la position du robot par rapport au trajectoire comme indiqué dans la Figure (II.10), à la sortie du contrôleur, puis nous obtenons les deux paramètres, la vitesse linéaire et l'angle de direction qui seront nécessaires pour que le robot reste toujours sur la trajectoire désirée [30].

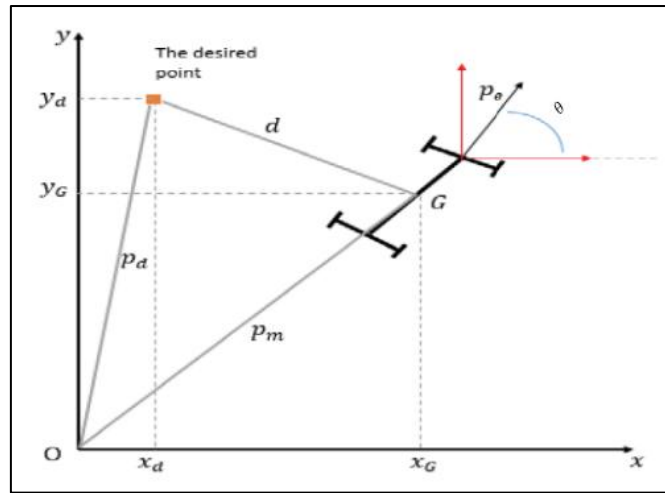


Figure II.10 : Schéma technique du robot et trajectoire désirée [30].

Les vecteurs \vec{p}_m , \vec{p}_d , \vec{p}_e et d représentent : la position, la position désirée, l'angle d'orientation et la distance entre l'état actuel et la position désirée. Ces vecteurs sont donnés comme suit :

$$\vec{p}_m = x\vec{i} + y\vec{j} \quad (\text{II.30})$$

$$\vec{p}_d = x_d(t)\vec{i} + y_d(t)\vec{j} \quad (\text{II.31})$$

$$\vec{p}_e = \cos(\theta)\vec{i} + \sin(\theta)\vec{j} \quad (\text{II.32})$$

$$d = \vec{p}_d - \vec{p}_m = (x_d - x_G)\vec{i} + (y_d - y_G)\vec{j} \quad (\text{II.33})$$

La norm représente la distance entre le robot et la position désirée ainsi que Vec_{prod} l'orientation du véhicule.

$$norm = \sqrt{(x_d - x_G)^2 + (y_d - y_G)^2} \quad (\text{II.34})$$

$$Vec_{prod} = \frac{\vec{p}_e \times \vec{d}}{\|\vec{d}\|} \quad (\text{II.35})$$

Selon les équations (II.32) et (II.33)

$$\vec{p}_e \times \vec{d} = (y_d - y_G) \cdot \cos(\theta) - (x_d - x_G) \cdot \sin(\theta) \quad (\text{II.36})$$

II.8. Conclusion

La modélisation de différents robots mobiles est présentée brièvement et plus en détail la modélisation du robot mobile type véhicule. Ce modèle sera utilisé dans le chapitre suivante afin de développer l'architecture de contrôle dans le but de suivi des trajectoires proches de la réalité.

Chapitre III:
Architecture de Contrôle

III.1. Introduction

Ce chapitre examine l'utilisation des contrôleurs flous et diverses méthodes d'optimisation dans la conception et l'amélioration des régulateurs PID pour les systèmes cinématiques et dynamiques. Il débute par la théorie des ensembles flous avec la définition des ensembles, des variables linguistiques, des fonctions d'appartenance et des opérations sur ces ensembles, établissant ainsi les fondements nécessaires à la compréhension des contrôleurs flous abordés plus tard.

En parallèle, nous abordons les régulateurs PID, omniprésents dans l'industrie de contrôle automatique, en définissant leurs types principaux et leur structure. Nous examinons également leurs applications courantes et les scénarios où chaque type est le plus efficace.

Enfin, nous explorons deux méthodes d'optimisation modernes, PSO (Optimisation par Essaim de Particules) et GJO (Optimisation du Chacal Doré), et discutons de leur application spécifique à l'optimisation des paramètres des régulateurs PID. Ces algorithmes sont analysés pour leur efficacité dans la recherche de solutions optimales dans des espaces de paramètres complexes, enrichissant ainsi notre compréhension des techniques avancées d'optimisation dans le domaine du contrôle.

III.2. Définition et domaines d'applications de la logique floue

La définition de Zadeh : La logique floue est définie comme un ensemble de principes mathématiques pour la représentation des connaissances basée sur les degrés d'appartenance plutôt que sur l'appartenance brute de la logique binaire classique. Et La logique floue peut être définie comme :

- La première est que la logique floue est utilisée pour traiter les connaissances inadéquates et imprécises. Elle permet de représenter et de traiter les connaissances de manière à pouvoir manipuler des systèmes complexes.
- La seconde est une extension de la logique classique, visant à raisonner sur des connaissances incomplètes.

Parmi les nombreux domaines d'application de la logique floue, le contrôle flou est le plus actif au monde. L'éventail des applications de la logique floue est large. Il s'agit notamment de:

- En automatique, pour faire de la commande et de la régulation floue, etc.
- En robotique, pour faire de la planification de trajectoire, etc.

- En traitement du signal, pour faire de la fusion de données, de la classification, de la reconnaissance de forme ou de la recherche d'information, etc.
- En traitement d'image, pour atténuer le bruit d'une image, pour faire de l'interpolation, etc [31].

III.3. Définitions et notions sur la théorie des ensembles flous

III.3.1. Définition d'un ensemble flou

Un ensemble net A sur un cadre de référence X (aussi appelé univers de discours) est une collection d'éléments où chaque élément $x \in X$ appartient soit entièrement (100%) à l'ensemble A , soit n'appartient pas du tout à A . Cela peut être décrit de manière précise par la fonction caractéristique $k_A(x)$, définie comme :

$$k_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{sinon} \end{cases} \quad (\text{III.1})$$

La fonction clarifiant ainsi l'appartenance ou la non-appartenance de manière binaire pour chaque élément.

Un ensemble flou est introduit pour refléter la réalité où les objets physiques ne sont souvent pas clairement définis dans des catégories distinctes. Mathématiquement, il est défini sur un univers de discours X par une fonction d'appartenance $\mu_A(x)$ qui varie de 0 à 1.

Cette fonction indique le degré auquel chaque élément $x \in X$ appartient à l'ensemble flou A . Ainsi, un élément donné peut être complètement ou partiellement membre de A . Un ensemble flou peut être représenté comme un ensemble de paires ordonnées [31].

$$A = \{(x, \mu_A(x); x \in X)\} \quad (\text{III.2})$$

III.3.2. Les variables linguistiques

Les variables linguistiques, utilisées pour représenter des connaissances imprécises ou vagues où la valeur précise est souvent inconnue, comprennent des mots ou des phrases dans un langage naturel ou artificiel. Elles décrivent les états ou les situations par des expressions floues. Chaque variable linguistique est définie par son nom (x), son ensemble de valeurs linguistiques ($T(x)$), et son univers de discours (U), avec des fonctions d'appartenance qui associent chaque élément de U à chaque terme linguistique. Par exemple, une variable linguistique "vitesse" pourrait inclure des termes comme "petite", "moyenne", et "grande", comme illustré dans la figure suivante [32]:

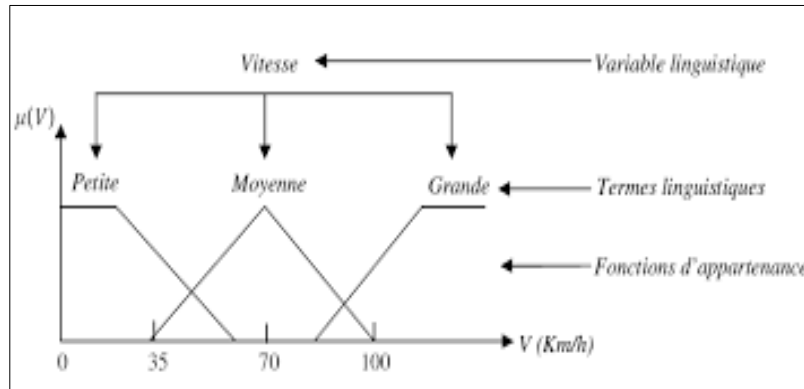


Figure III.1: Variables linguistiques [32].

III.3.3. Univers du discours

L'univers de discours, noté U , est l'ensemble qui contient toutes les valeurs possibles pour une variable linguistique donnée. Dans le cadre de l'application des ensembles flous, la première étape consiste à définir cet univers de discours pour chaque variable linguistique. Il représente le cadre de référence contenant tous les éléments pertinents au contexte donné. Par exemple, pour la variable "température", l'univers de discours pourrait inclure des termes tels que "chaud", "tiède" et "froid", chacun ayant une fonction d'appartenance associée [32].

III.3.4. Fonctions d'appartenance

Un ensemble flou est défini par sa fonction d'appartenance, équivalente à la fonction caractéristique en logique classique. Par exemple, pour l'ensemble des personnes de taille moyenne en logique classique, on définirait cet ensemble comme celles dont la taille est entre 1,60 m et 1,80 m, avec une fonction caractéristique qui vaut 1 dans cet intervalle et 0 en dehors. En revanche, un ensemble flou des personnes de taille moyenne est caractérisé par une fonction d'appartenance qui peut prendre n'importe quelle valeur dans l'intervalle $[0, 1]$. Cette fonction indique le degré d'appartenance de chaque taille possible à l'ensemble des tailles moyennes, pouvant varier de 0 à 1. Les formes courantes de fonctions d'appartenance incluent le trapézoïde, le triangle, la gaussienne et la sigmoïde [32].

- La fonction trapézoïde:

Elle est définie par quatre paramètres $\{a, b, c, d\}$ qui déterminent les coordonnées des quatre sommets.

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (\text{III.3})$$

- **La forme triangulaire :**

Elle est définie par trois paramètres {a, b, c,} qui déterminent les coordonnées des trois sommets.

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{c-x}{c-d}\right), 0\right) \quad (\text{III.4})$$

- **La forme triangulaire :**

Elle est définie par deux paramètres { σ , a.}

$$\mu(x) = \exp\left(-\left(\frac{x-a}{\sigma}\right)^2\right) \quad (\text{III.5})$$

- **La forme triangulaire :**

Elle est définie par deux paramètres {a, c}.

$$\mu(x) = \frac{1}{1+\exp(-a(x-c))} \quad (\text{III.6})$$

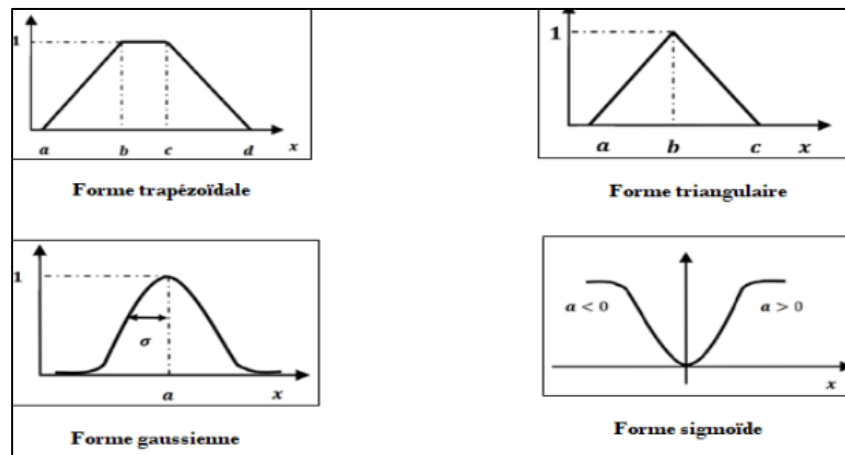


Figure III.2: Formes des fonctions d'appartenance [32].

III.3.5. Opérations sur les ensembles flous

Il est important de savoir construire entre les différents prédicats et les fonctions auxquelles ils appartiennent. L'opération sur les ensembles flous est une extension de l'opération connue sur les ensembles classiques [32].

- **Egalité:**

Soit deux ensembles flous A et B dans un univers X, on dit que A et B sont égaux ($A=B$), si leurs fonctions d'appartenance ont la même valeur en tout point $x \in X$.

$$\mu_A(x) = \mu_B(x), \forall x \in X \quad (\text{III.7})$$

- **Inclusion :**

Soit deux ensembles flous A et B dans un univers X, on dit que A est inclus dans B noté $A \subseteq B$ si leurs fonctions d'appartenance sont telles que :

$$\forall x \in X; \mu_A(x) \leq \mu_B(x) \quad (\text{III.8})$$

- **Intersection (ET) :**

L'intersection (ET) de deux ensembles flous A et B de X est l'ensemble flou C de X qui contient tous les éléments $x \in X$ appartenant à la fois à A et B. L'intersection de deux ensembles flous A et B de X est l'ensemble flou C.

$$\forall x \in X; \mu_C(x) = \min(\mu_A(x), \mu_B(x)) \quad (\text{III.9})$$

- **Union (OU) :**

L'union (OU) de deux ensembles flous A et B de est l'ensemble flou D de X qui contient tous les éléments $x \in X$ appartenant à A ou bien à B. L'union de deux ensembles flous A et B ($A \cup B$) de X est l'ensemble flou tel que.

$$\forall x \in X; \mu_D(x) = \max(\mu_A(x), \mu_B(x)) \quad (\text{III.10})$$

- **Le complément (NON) :**

Soit un ensemble flou A de X, son complément \bar{A} est l'ensemble flou contenant tous les x qui n'appartenant pas à A. Le complément \bar{A} est défini comme suit :

$$\forall x \in X; \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (\text{III.11})$$

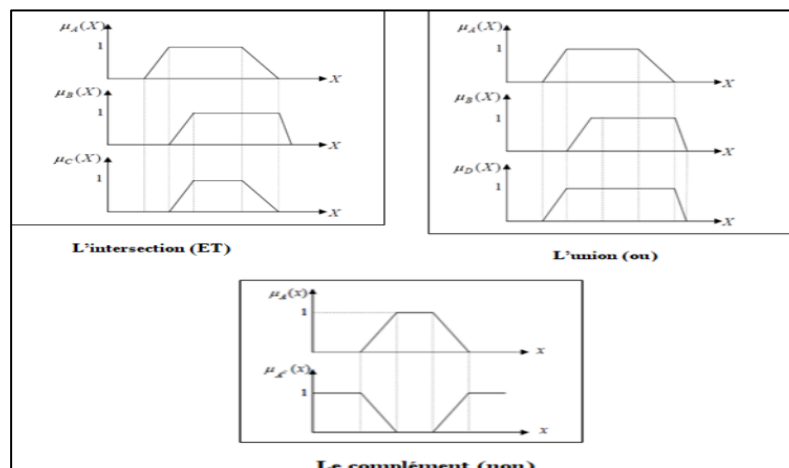


Figure III.3: Les opérateurs logiques [32].

III.4. Description des contrôleurs flous

Les contrôleurs flous sont généralement utilisés dans les structures de contrôle des processus en boucle fermée.

Les variables caractéristiques et les valeurs de consigne du système à contrôler définissent les variables d'entrée du contrôleur flou. Les variables caractéristiques sont généralement des variables de sortie du processus. Les variables de sortie du contrôleur flou sont des commandes appliquées au processus. Ainsi, un système flou se compose de quatre étapes de base, comme le montre la figure à savoir [31]:

- Interface de fuzzification
- Base de connaissance
- Moteur d'inférence
- Interface de défuzzification

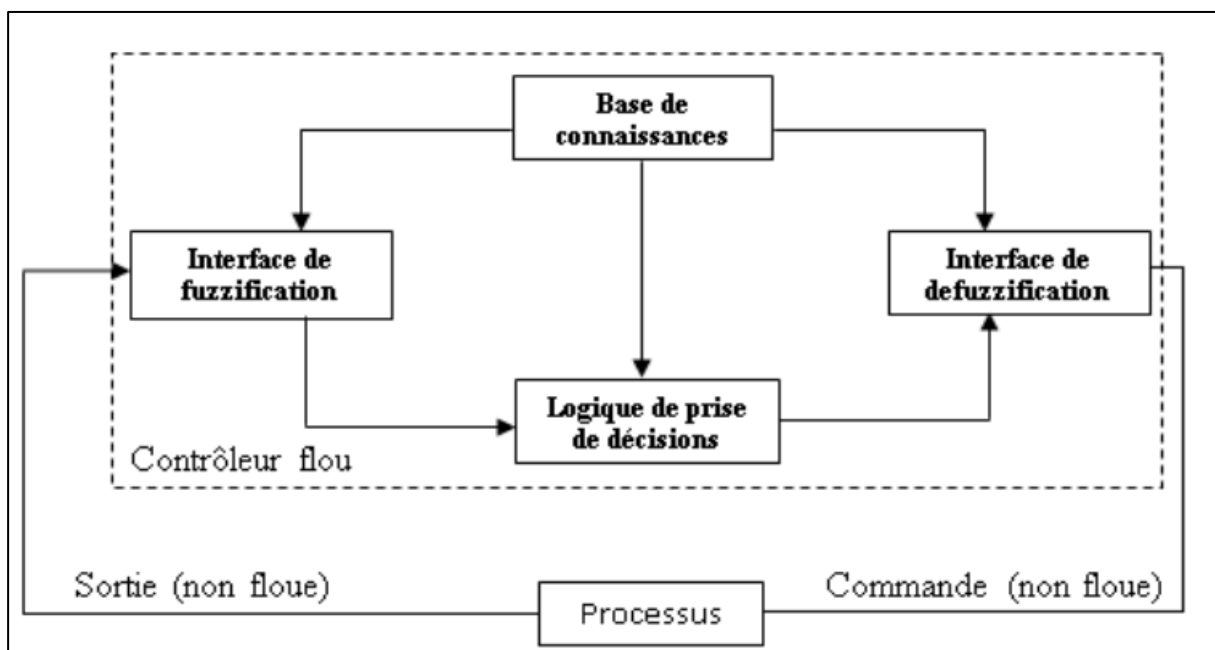


Figure III.4: Structure du contrôleur flou [31].

Les contrôleurs flous sont décrits par un ensemble de règles de type **SI (condition) ALORS (action)** qui permettent de traduire les stratégies de contrôle linguistiques des experts humains en stratégies de contrôle automatiques adaptées au monde réel [31].

III.4.1. Mise en forme des entrées

Cette première étape permet de traiter les variables d'entrée. En particulier, certaines quantités sont combinées pour calculer l'erreur (par exemple, la différence entre la sortie et la valeur fixée) et la dérivée de l'erreur. Très souvent, la dynamique des variables est normalisée par rapport à l'univers du discours.

III.4.2. Interface de fuzzification

La fuzzification est le premier processus à considérer dans la structure d'un système flou. Il s'agit de donner le degré d'appartenance à la valeur d'entrée réelle en fonction de l'ensemble flou. Dans un système flou, l'entrée et la sortie du système doivent être floues. Dans un système flou, le flou variable est une étape importante du processus de mise en œuvre [31].

L'étape floue consiste en :

- Établir des variables linguistiques
- Établir les quantificateurs flou (nombre des valeurs linguistiques)
- Chaque quantificateur flou : attribue une signification numérique à la fonction d'appartenance.

Le choix de la forme des fonctions d'appartenance (triangulaires, trapézoïdales, exponentielles, gaussiennes, ...) est arbitraire. La forme triangulaire facilite la programmation et explique pourquoi ils sont le plus souvent utilisés. Quant au nombre de fonctions d'appartenance, elles sont généralement impaires car elles sont réparties autour de zéro. Deux modes de fuzzification sont possibles :

- *Fuzzification par singleton :*

Cette approche est utilisée lorsque la valeur fournie par le capteur est supposée exacte. Le singleton flou de la valeur $x_0 \in U$ est la partie floue numérique où la fonction d'appartenance est définie comme :

$$\begin{cases} \mu_{F_x}(x) = 1 \text{ si } x = x_0 \\ \mu_{F_x}(x) = 0 \text{ si } x \neq x_0 \end{cases} \quad (\text{III.12})$$

Cette stratégie est largement utilisée dans les applications de contrôle flou car elle est facile à mettre en œuvre.

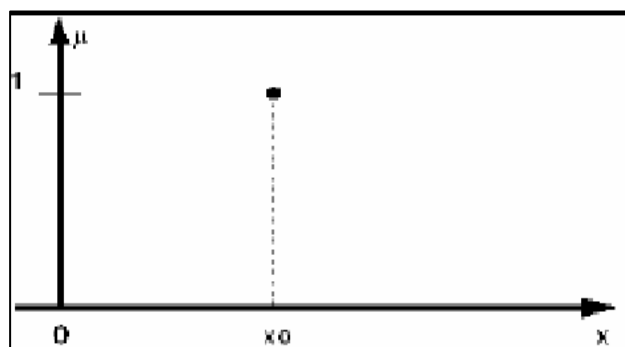


Figure III.5: Fuzzification d'une valeur exacte [31].

- **Fuzzification non singleton :**

Cette approche tient compte du fait que le capteur est sujet à des erreurs de mesure et à une exposition au bruit. Dans ce cas, la fonction d'appartenance de F_x dépend de l'erreur ou du bruit. Elle est généralement définie par une fonction trigonométrique ou trapézoïdale [31].

III.4.3. Base de connaissance

Ce bloc comprend la définition des fonctions d'appartenance et des règles d'inférence pour les variables d'entrée et de sortie. Celles-ci constituent le cœur du contrôleur et caractérisent souvent la connaissance ambiguë et qualitative du système contrôlé. Elles sont exprimées sous la forme d'un ensemble de propositions conditionnelles linguistiques de la forme (SI-ALORS) et d'associations de formes (condition_action) qui peuvent être facilement interprétées par des implications floues. En général, les contrôleurs flous utilisent des règles de la forme :

Si x_1 est A_1 et x_2 est A_2 et...et x_n est A_n alors u_1 est B_1 et u_2 est B_2 et... u_m est B_m ou x_1, x_2, x_3, \dots et x_n sont les variables physiques caractéristiques du processus, u_1, u_2, u_3, u_m sont les variables de commande du système, $A_1, A_2, A_3, \dots, x_n$ et $B_1, B_2, B_3, \dots, B_n$ sont les valeurs linguistiques [31].

III.4.4. Moteur d'inférence

Il s'agit d'un mécanisme de décision. Sur la base des faits observés dans la base de règles floues, des décisions peuvent être prises à l'aide d'un raisonnement approximatif. Dans le raisonnement des contrôleurs à logique floue, les opérateurs ET et OU sont utilisés. L'opérateur ET s'applique aux variables d'une règle, tandis que l'opérateur OU relie différentes règles. Il existe plusieurs façons de mettre en œuvre ces opérateurs dans l'inférence et ils s'appliquent aux fonctions d'appartenance [31]:

- Méthode d'inférence MAX-MIN (Mamdani)
- Méthode d'inférence MAX-PROD (Mamdani)
- Méthode d'inférence SUM-PROD.

III.4.4.1. Méthode MAX-MIN (Mamdani)

La méthode d'inférence "max-min" est réalisée au niveau de la condition par « OU » par la formation de l'opérateur maximum et « ET » par la formation du minimum, et le sens "ALORS" est réalisé par la formation du minimum [31].

III.4.4.2. Méthode MAX-PROD

Cette méthode est généralement réalisée au niveau de la condition, l'opérateur « OU » par la formation de la valeur maximale et l'opérateur « ET » par la formation de la valeur minimale. D'autre part, la conclusion de chaque règle introduite par « ALORS », cette dernière est réalisée cette fois par la formation de produits. L'opérateur « OU », qui relie différentes règles, est à nouveau réalisé par la formation de la valeur maximale [31].

III.4.4.3. Méthode SOMMEPROD

Contrairement aux méthodes d'inférence précédentes, la méthode d'inférence SOMMEPROD réalise l'opérateur « OU » en formant la somme (moyenne) et l'opérateur « ET » en formant le produit au niveau de la condition. L'opérateur « ALORS », qui relie le coefficient d'appartenance de la condition à la fonction d'appartenance de la variable de sortie, est réalisé cette fois par la formation du produit. L'opérateur « OU » qui relie différentes règles, se réalise à nouveau par la formation de sommes [31].

III.4.5. Interface de défuzzification

C'est à ce stade que les différentes commandes générées par le moteur d'inférence sont combinées en une seule commande de sortie et que cette variable linguistique qualitative de sortie est convertie en données numériques quantitatives, ce qui signifie qu'il faut trouver la valeur quantitative optimale en fonction de la fonction d'appartenance de la variable linguistique. Parmi les sept méthodes de défuzzification qui peuvent être appliquées aux systèmes flous, il appartient au concepteur de choisir celle qu'il souhaite utiliser [31].

III.4.5.1. Défuzzification par centre de gravité

Cette méthode, génère le centre de gravité de la surface engendrée par la fonction d'appartenance de l'ensemble flou agrégé. L'abscisse du centre de gravité peut être déterminée à l'aide de la relation suivante [31]:

$$\text{Cas discret : } u^* = \frac{\sum_{i=1}^n u_i \cdot \mu_u(u_i)}{\sum_{i=1}^n \mu_u(u_i)} \quad (\text{III.13})$$

n : Le nombre de niveaux de quantification de la sortie du contrôleur flou.

$$\text{Cas continu : } u^* = \frac{\int_u u \cdot \mu_u(u) du}{\int_u \mu_u(u) du} \quad (\text{III.14})$$

III.4.5.2. Défuzzification par la valeur maximale

Si plus d'un ensemble a la même hauteur maximale, cette méthode est effectuée, qui prendra l'axe horizontal correspondant à la moyenne de l'axe horizontal avec la valeur maximale de la fonction d'appartenance comme axe vertical. Mathématiquement, MM est énoncé comme suit [31]:

$$x_{MM} = \frac{\int_S x dx}{\int_u dx} = \frac{\sum_{i=0}^n x_i}{N}, \text{ avec } \begin{cases} S = x \in U \text{ tel que } \mu(x) = \text{SUP}_{x \in U} \mu(x) \\ N \text{ le nombre de point appartenat a } S \end{cases} \quad (\text{III.15})$$

III.4.6. Mise en forme des sorties

Le dernier bloc de ce contrôleur flou sert à traiter chaque commande appliquée afin de définir la dynamique de la commande (pour le raisonnement, son monde discursif est souvent normalisé entre [-1,1] pour obtenir des effets spécifiques (proportionnel, intégral, ...etc.) [31].

III.5. Régulateur PID

III.5.1. Définition

Le régulateur PID, également connu sous le nom de correcteur PID (proportionnel, intégral, dérivé), est un dispositif de contrôle utilisé dans les systèmes industriels pour assurer un asservissement en boucle fermée. Il se compose d'un comparateur qui détecte l'écart (ou erreur) entre la mesure et la consigne, ainsi que d'un correcteur dont l'algorithme permet de réguler la mesure du processus selon les spécifications requises [33].

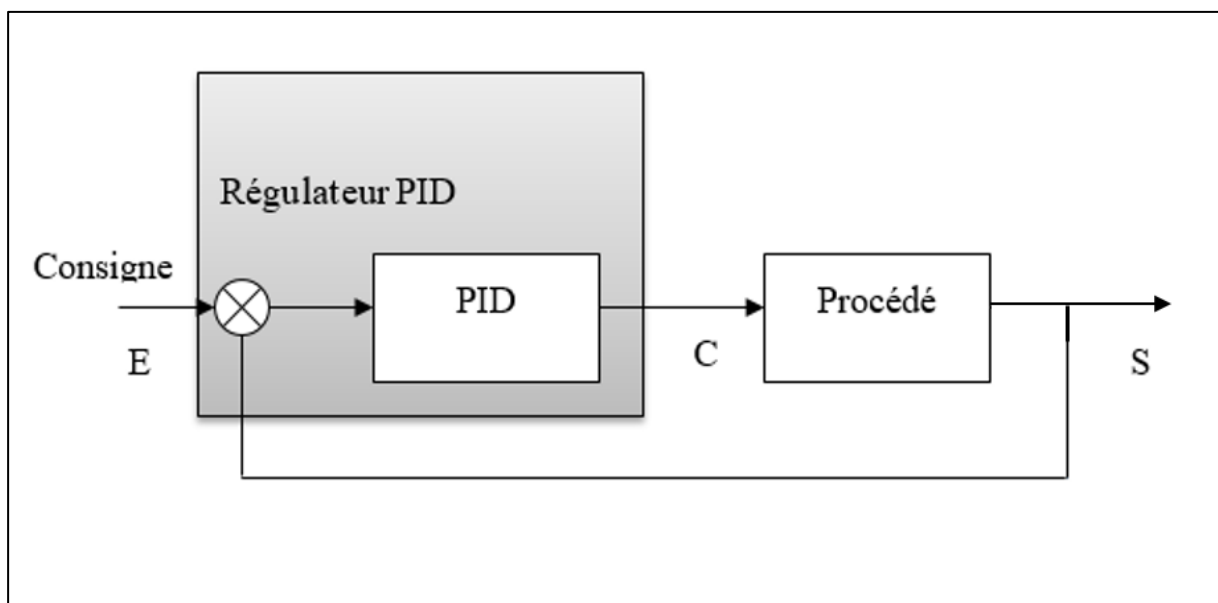


Figure III.6: Asservissement par un régulateur.

Les signaux S (mesure), E (consigne ou valeur d'entrée), et ε (écart ou erreur $e(t)$) sont exprimés dans la même unité. Le signal C généré par le régulateur représente le signal de commande. Pour rappel, l'erreur statique correspond à la différence entre la consigne d'entrée et la valeur mesurée.

Un régulateur PID est constitué par la combinaison de trois actions et remplit principalement les trois fonctions suivantes [33] :

- Il génère un signal de commande en ajustant l'évolution du signal de sortie par rapport à la consigne.
- Il élimine l'erreur statique grâce au terme intégral.
- Il prévoit les variations de la sortie en utilisant le terme dérivé.

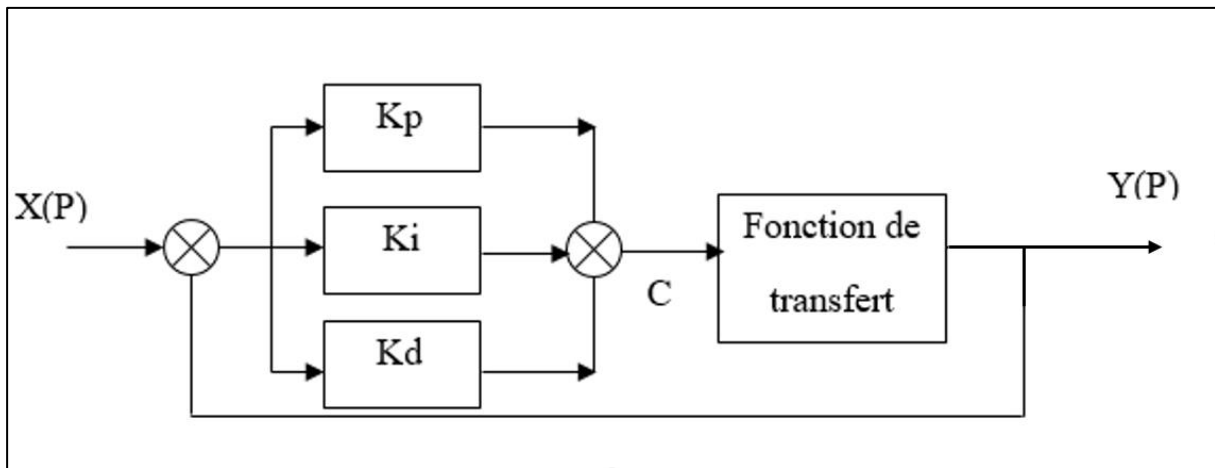


Figure III.7: Régulateur PID.

III.5.2. Types de régulateurs

III.5.2.1. Régulateur à action proportionnelle (P)

Le régulateur à action proportionnelle, également appelé régulateur P, agit de manière simple en générant une commande kp proportionnelle à l'erreur $e(t)$. Cette erreur est amplifiée virtuellement par un gain constant qui doit être déterminé en fonction des caractéristiques spécifiques du système [34].

- La fonction de transfert du régulateur P est :

$$C(p) = \frac{U(p)}{E(p)} = k_p \quad (\text{III.16})$$

- La loi de commande du régulateur P :

$$C(p) = k_p \times e(p) \quad (\text{III.17})$$

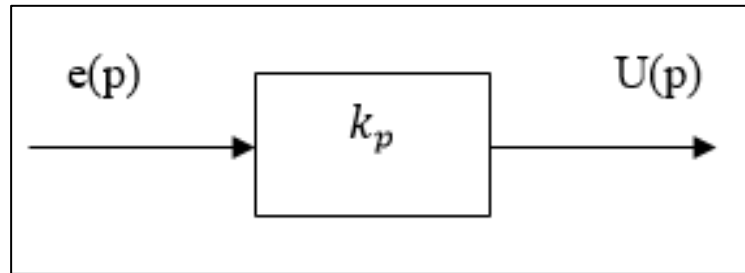


Figure III.8: Schéma fonctionnel du régulateur P.

III.5.2.2. Régulateur à action proportionnelle-intégrale (PI)

La fonction de transfert du régulateur PI est :

$$C(p) = k_p \frac{1+T_i p}{T_i p} \quad (\text{III.18})$$

L'équation temporelle est :

$$u(t) = k_p T(e(t) + \frac{1}{t_i} \int_0^t e(t)) \quad (\text{III.19})$$

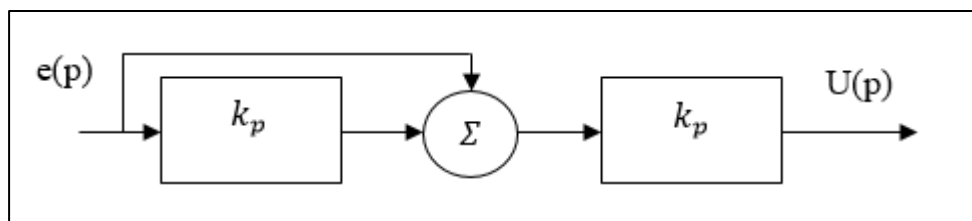


Figure III.9: Schéma fonctionnel du régulateur PI.

III.5.2.3. Régulateur à action proportionnelle-dérivée (PD)

La fonction de transfert du régulateur PD est :

$$C(p) = k_p(1 + T_d p) \quad (\text{III.20})$$

L'équation temporelle est :

$$u(t) = k_p(e(t) + T_d \frac{Te}{dt}) \quad (\text{III.21})$$

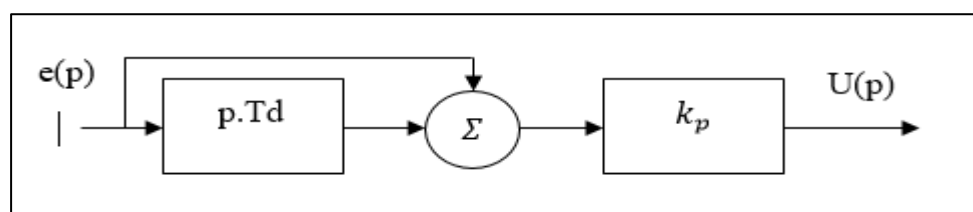


Figure III.10: Schéma fonctionnel du régulateur PD.

III.5.2.4. Structure du régulateur PID (proportionnelle intégrale dérivée)

Le tableau III.1 résumé les structures du régulateur PID :

Tableau III.1: Les structures du régulateur PID.

Structure	Symbole	Equation
En Série		$u(t) = k_p e(t) \left[\left(1 + \frac{T_d}{T_i} \right) + \frac{1}{T_i} \int_0^t e(t) \times dt + T_d \times \frac{de}{dt} \right]$
En Parallèle		$u(t) = k_p e(t) + \frac{1}{T_i} \int_0^t e(t) \times dt + T_d \times \frac{de}{dt}$
Mixte		$u(t) = k_p e(t) + k_p \left[\frac{1}{T_i} \int_0^t e(t) \times dt + T_d \times \frac{de}{dt} \right]$

III.5.3. Utilisation

Les contrôleurs proportionnels-intégraux-dérivés (PID) sont utilisés dans la plupart des applications de contrôle de processus automatique dans l'industrie aujourd'hui pour réguler la vitesse, la température, la pression, le niveau et de nombreuses autres variables de processus industriels. Des exemples de contrôleurs PID :

- Calibreur de température.
- Outils pour fils et câbles de température.
- Étiquettes de température, laques et marqueurs.
- Maintenir la vitesse de manière précise et efficace dans une application industrielle.

III.6. Optimisation par la méthode de PSO (optimisation par essaim de particule)

Le Particle Swarm Optimization (PSO) est un algorithme évolutionnaire qui utilise une population de solutions candidates pour trouver une solution optimale à un problème. Il a été proposé en 1995 par Russel Eberhart et James Kennedy, inspiré du comportement social des animaux en essaim. Dans PSO, un essaim de particules représente une population d'agents simples appelés particules. Chaque particule a une position (vecteur solution) et une vitesse, et possède une mémoire pour enregistrer sa meilleure performance et celle de ses informatrices (particules voisines). Chaque particule est associée à un voisinage d'informatrices [35].



Figure III.11: Vol en group d'oiseau [36].

III.6.1. Problème d'optimisation

Un problème d'optimisation est défini comme la recherche de la valeur minimale ou maximale (optimale) d'une fonction donnée. Les variables de cette fonction sont souvent obligées d'évoluer dans certaines parties de l'espace de recherche. Par conséquent, des problèmes d'optimisation surviennent sous contraintes.

Mathématiquement, on cherche à minimiser f sur E , c'est-à-dire, on cherche le point optimal ($x^* \in E$) tel que :

$$f(x^*) = \min f(x) \quad (\text{III.22})$$

III.6.2. Principe de fonctionnement

Dans cet algorithme, des particules, représentant des solutions candidates dans la population, cohabitent et évoluent ensemble en partageant leurs connaissances avec les particules voisines. Après avoir exploré tout l'espace de recherche, chaque particule utilise son vecteur vitesse pour itérer vers une nouvelle solution. Chaque particule ajuste sa vitesse en se basant sur son propre historique de mouvement (sa meilleure position précédente) ainsi que sur les expériences des particules voisines (les meilleures solutions trouvées dans la population).

Le déplacement des particules est influencé par trois composantes principales [37]:

- La composante inertielle : Cela indique que les particules tendent à suivre leur direction de déplacement actuelle.
- La composante cognitive : Chaque particule a tendance à se baser sur son expérience personnelle, en se déplaçant vers la meilleure position qu'elle a déjà atteinte.
- La composante sociale : Les particules ont tendance à se déplacer vers les meilleures positions découvertes collectivement par le groupe, en se basant sur l'expérience partagée avec leurs congénères.

Dans un espace de recherche de dimension D , la particule i de l'essaim est modélisée par son vecteur position $\vec{x}_{ij} = (x_{i1} x_{i2} \dots x_{iD})^T$ et par son vecteur vitesse $\vec{v}_{ij} = (v_{i1} v_{i2} \dots v_{iD})^T$. La qualité de sa position est déterminée par la valeur de la fonction objective en ce point. Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on note $\vec{p}_{ij} = (p_{i1} p_{i2} \dots p_{iD})^T$. La meilleure position atteinte par ses particules voisines est notée $\vec{g}_{ij} = (g_1 g_2 \dots g_D)^T$. Le principe de la méthode d'essaim de particule est résumé par la figure (III.12) :

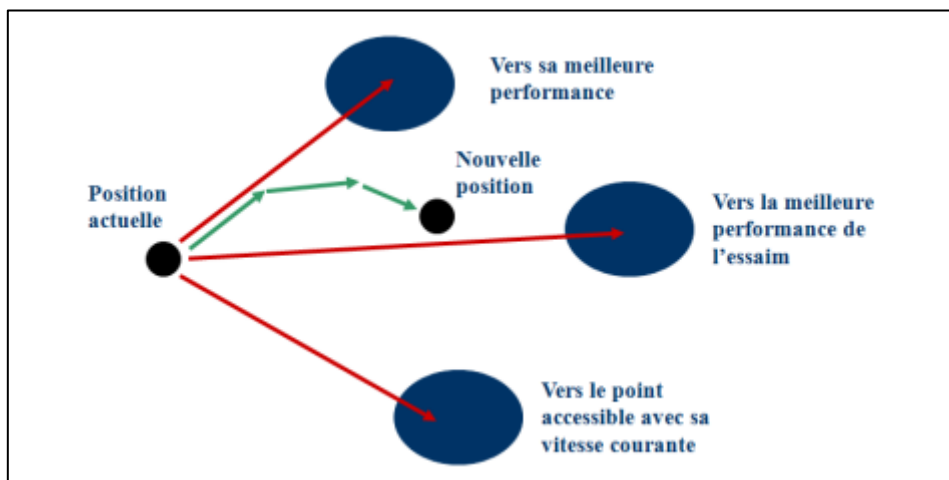


Figure III.12: Déplacement d'une particule [37].

Le vecteur vitesse est calculé à partir de l'équation (III.23) suivante [37]:

$$v_{ij} = wv_{ij}(k-1) + c_1r_1(p_{ij}(k-1) - x_{ij}(k-1)) + c_2r_2(p_{ij}(k-1) - x_{ij}(k-1)) \quad (\text{III.23})$$

La position à l'itération k de la particule i est alors définie par l'équation (III.24) :

$$x_{ij}(k) = x_{ij}(k-1) + v_{ij} \quad (\text{III.24})$$

$$i = 1,2,3 \dots N_p. \quad J = 1,2,3 \dots N_D. \quad K = 1,2,3 \dots K_{max}.$$

Avec:

N_p : le nombre de particules de l'essaim.

N_d : le nombre de variables du problème (c.-à-d. dimension d'une particule).

k_{max} : le nombre maximal d'itération.

$v_{ij}(k)$ est la vitesse de la $j^{ième}$ composante de la $i^{ième}$ particule de l'essaim, à la $k^{ième}$ itération.

p_{ij} : La $j^{ième}$ composante de la meilleure position occupée par la $i^{ième}$ particule de l'essaim enregistrée dans les itérations précédentes (local best).

g_i : la $j^{ième}$ composante de la meilleure position occupée par la $i^{ième}$ particule globale de l'essaim (global best).

$x_{ij}(k)$: la $j^{ième}$ coordonnée de la position actuelle de la particule, à la $i^{ième}$ itération.

ω : est en général une constante appelée coefficient d'inertie, c_1 et c_2 sont deux constantes appelées coefficients d'accélération, r_1 et r_2 sont deux nombres aléatoires tirés uniformément dans $[0,1]$ à chaque itération et pour chaque dimension.

$w.V_{ij}(k-1)$: Correspond à la composante physique du déplacement. Le paramètre w contrôle l'influence de la direction de déplacement sur le déplacement futur. Il est à noter que, dans certaines applications, le paramètre w peut-être variable.

$c_1r_1(p_{ij\ best} - x_{ij}(k-1))$: correspond à la composante cognitive du déplacement où c_1 contrôle le comportement cognitif de la particule.

$c_2 r_2 (p_{ij \text{ best}} - x_{ij}(k - 1))$: Correspond à la composante sociale du déplacement, où contrôle l'aptitude sociale de la particule.

Les gammes appropriées de valeur pour c_1 et c_2 , sont de 1 à 2, mais 2 est le plus approprié dans beaucoup de cas [38].

Le coefficient d'inertie peut être variable selon [39]:

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{k_{max}} \right) \times k \quad (\text{III.25})$$

Avec :

k_{max} : est le nombre maximum d'itérations et k est le nombre d'itérations en cours.

w_{min} : est le coefficient minimum d'inertie et w_{max} est le coefficient maximum d'inertie.

Le coefficient d'inertie w est essentiel dans le processus de recherche, en équilibrant la recherche locale et globale.

w : Un choix approprié de ce paramètre améliore l'efficacité de la méthode pour atteindre une solution globale.

Réduire linéairement la valeur de w de 0.9 à 0.4 tout au long du processus de recherche a été démontré comme conduisant à de meilleurs résultats.

III.6.3. Mécanisme de confinement

Compte tenu de la position actuelle et de la vitesse actuelle de la particule, il arrive parfois que cette dernière ait tendance à quitter l'espace de recherche lors de sa déplacement. In dans ce cas, l'algorithme inclut un mécanisme de confinement appelé confinement par intervalles pour gérer le déplacement des particules, l'amenant à un nouveau point qui appartient également à l'espace de recherche. In de manière générale, ce mécanisme consiste à ramener les particules au point admissible le plus proche.

$$\begin{cases} \text{si } x_{ij} > x_{jmax} & \text{alors } x_{ij} = x_{jmax} \\ \text{si } x_{ij} < x_{jmin} & \text{alors } x_{ij} = x_{jmin} \\ & V_{ij} = 0 \end{cases} \quad (\text{III.26})$$

Où : x_{jmax} et x_{jmin} sont les valeurs limites du paramètre x_{ij} .

III.6.4. Déroulement de l'algorithme

Pour illustrer le principe de l'algorithme PSO dans le contexte de la résolution d'un problème d'optimisation, prenons en considération le problème de minimisation suivant :

$$\begin{cases} \text{Min } f(x) \\ x \in D \end{cases} \quad (\text{III.27})$$

La fonction fitness associée est :

$$\text{fitness}(x_j) = f(x_j) \quad (\text{III.28})$$

L'algorithme PSO se déroule suivant les étapes suivantes [40]:

Étape 1: initialiser une population de particules et de vitesses uniformément réparties dans l'espace de recherche D et définir un critère d'arrêt.

Étape 2: évaluer la fonction objective, pour chaque particule.

Évaluez la fonction objective pour chaque particule.

Étape 3: mettre à jour la meilleure position P_{best} pour chaque particule et la meilleure position globale g_{best} dans la population.

Étape 4: mettre à jour la position et la vitesse utilisant les équations (III.23) et (III.24)

Étape 5: si une particule, déplacée par l'algorithme vers la limite ou à l'extérieur de l'espace de recherche, le mécanisme de confinement des particules exprimé par les équations (III.25) intervient et ramène la particule à l'intérieur de l'espace de recherche.

Étape 6: vérifier le critère d'arrêt. Si le critère d'arrêt n'est pas satisfait, aller vers l'étape 2; sinon, le programme se termine, et la solution optimale est produite.

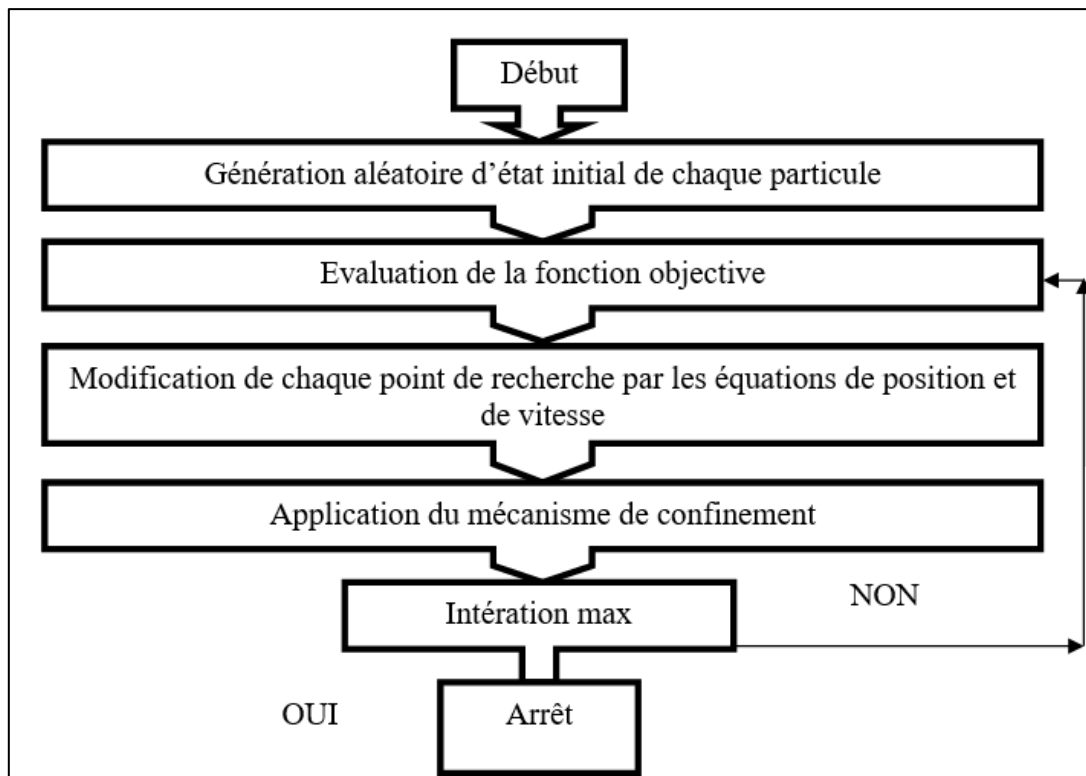


Figure III.13: Les étapes de L'algorithme PSO.

III.7. Optimisation par la méthode de GJO (Optimisation du Chacal Doré.)

L'optimisation par le chacal doré (GJO), proposée par Nitish Chopra et Mohsin Ansari en 2022, reproduit le comportement de chasse du chacal doré dans la nature. Ce prédateur terrestre de taille modérée, membre de la famille des canidés, habite en Afrique du Nord, en Afrique de l'Est, en Europe, en Asie du Sud-Est, au Moyen-Orient et en Asie centrale. Sa petite taille et ses longues pattes lui permettent de parcourir de longues distances pour chasser sa proie. Les chacals dorés chassent généralement en couple, mâles et femelles. Leur comportement de chasse, illustré à la Figure (III.14), se divise en trois étapes [41]:

- Exploration de la proie.
- Encerclement et perturbation de la proie jusqu'à ce qu'elle cesse de bouger.
- Attaque de la cible avec un bond.



Figure III.14: Les étapes du comportement de chasse du chacal doré [41].

III.7.1. Optimisation du chacal doré (GJO)

Nous montrerons les étapes de l'algorithme d'optimisation du Chacal Doré. Le premier processus dans le GJO est de construire la population de N solutions en utilisant la formule suivante [41]:

$$X_i = Ib + r \times (Ub - Ib), i = 1,2,3 \dots, N \quad (\text{III.29})$$

où Ib et Ub sont les limites des paramètres à l'intérieur de l'espace de recherche. $r \in [0,1]$ désigne la valeur aléatoire. Ensuite, pour chaque $X_i, i = 1,2, \dots, N$ la valeur de fitness est déterminée, et le meilleur parmi eux est choisi (c'est-à-dire la solution mâle Xb) ainsi que le deuxième meilleur (c'est-à-dire la femelle chacal). L'étape suivante consiste à construire la matrice des proies en utilisant l'équation (III.30) [41]:

$$Prey = \begin{bmatrix} X_{11} & X_{N1} & \dots & X_{1d} \\ X_{21} & X_{22} & \dots & X_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ X_{N1} & X_{N2} & \dots & X_{Nd} \end{bmatrix} \quad (\text{III.30})$$

III.7.2. Étape d'exploration

Les chacals peuvent détecter et suivre leur proie, mais parfois celle-ci parvient à échapper à la capture. Par conséquent, les chacals attendent et chassent une nouvelle proie. Un chacal mâle X_M prend généralement la tête lors des chasses, tandis que le chacal femelle X_{FM} suit le chacal mâle. La capacité d'évasion de la proie est réduite lorsqu'elle est harcelée par les chacals, et les couples de chacals encerclent ensuite la proie découverte lors de la phase précédente. Après l'encercllement, ils sautent sur leur proie et la dévorent. Ce comportement de chasse, avec les chacals mâle et femelle, est exprimé mathématiquement comme suit [41]:

$$X_1(t) = X_M(t) - E \times |X_M(t) - r_l \times \text{Prey}(t)| \quad (\text{III.31})$$

$$X_2(t) = X_{FM}(t) - E \times |X_{FM}(t) - r_l \times \text{Prey}(t)| \quad (\text{III.32})$$

Où X_M et X_{FM} sont mis à jour en X_1 et X_2 , respectivement à la $t_{\text{ème}}$ itération. E représente l'énergie de la proie qui est mise à jour comme suit :

$$E = E_0 \times E_1 \quad (\text{III.33})$$

Dans l'équation (11), E_0 et E_1 représentent respectivement la valeur initiale et décroissante de l'énergie. Les valeurs de E_0 et E_1 sont mises à jour à l'aide des équations (III.34) et (III.35) :

$$E_0 = 2 \times r - 1 \quad (\text{III.34})$$

$$E_1 = c_1 \times (1 - (t/T)) \quad (\text{III.35})$$

Dans l'équation (13), T fait référence au nombre maximum de générations. Pendant ce temps, r_l désigne une valeur aléatoire produite selon la distribution de Levy en utilisant l'équation :

$$r_l = 0.05 \times \text{Levy} \quad (\text{III.36})$$

Ensuite, les positions des chacals sont mises à jour selon l'équation (III.37) :

$$X_1(t) = \frac{X_1(t)+X_2(t)}{2} \times Levy \quad (III.37)$$

$$Levy = \frac{s|u \times \sigma|}{|v|^{\frac{1}{\beta}}} \quad (III.38)$$

$$\sigma = \frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \quad (III.39)$$

Où u et v font référence à des nombres aléatoires ; de plus, $\beta = 1,5$ et $s = 0,01$ sont des constantes.

III.7.3. Déroulement de l'algorithme

Dans GJO, la transition entre les phases d'exploration et d'exploitation est accomplie en utilisant l'énergie d'évasion de la proie. Si la valeur de $|E| > 1$, les solutions commencent leur phase d'exploration ; sinon, elles passent à la phase d'exploitation. Les étapes de GJO sont illustrées dans la Figure (III.14).

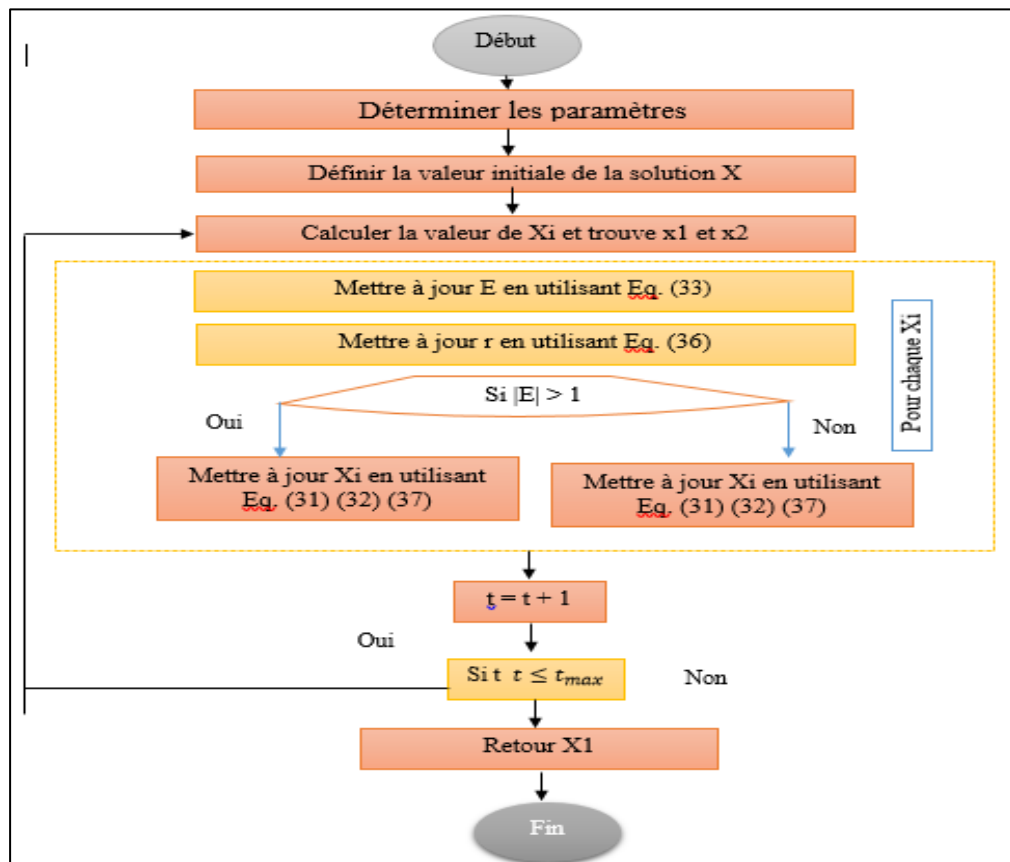


Figure III.15: Les étapes de l'algorithme GJO.

III.8. Architecture de commande de notre robot mobile

On utilise le modèle cinématique qui définit par les équations (II.27), (II.28) et (II.29) pour modéliser sur Simulink Matlab, on utilise deux saturations pour la protection des composants du robot. La simulation de ce modèle est :

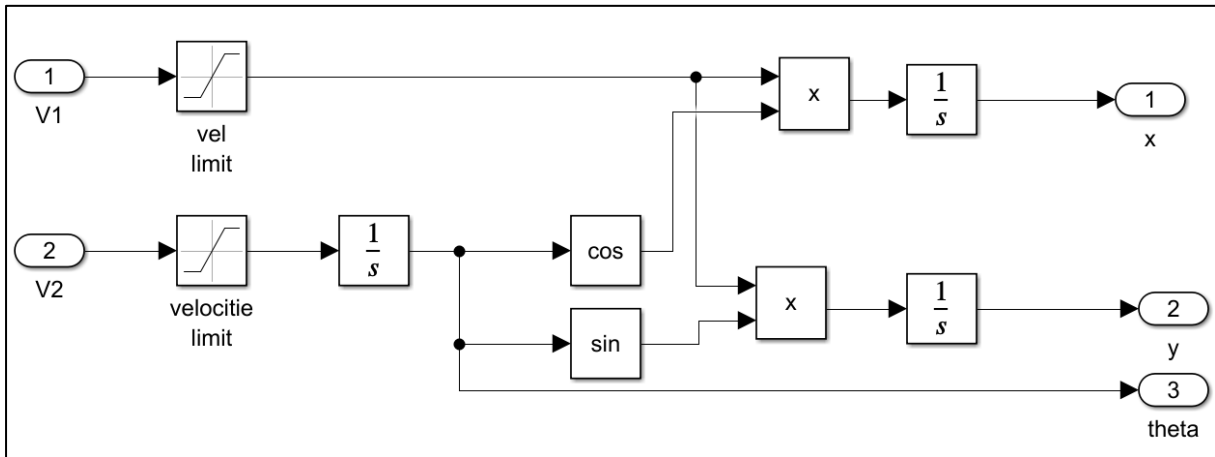


Figure III.16: Le modèle cinématique du robot sur Simulink.

La simulation de notre robot est représentée dans la figure (III.17) avec logique floue Controller :

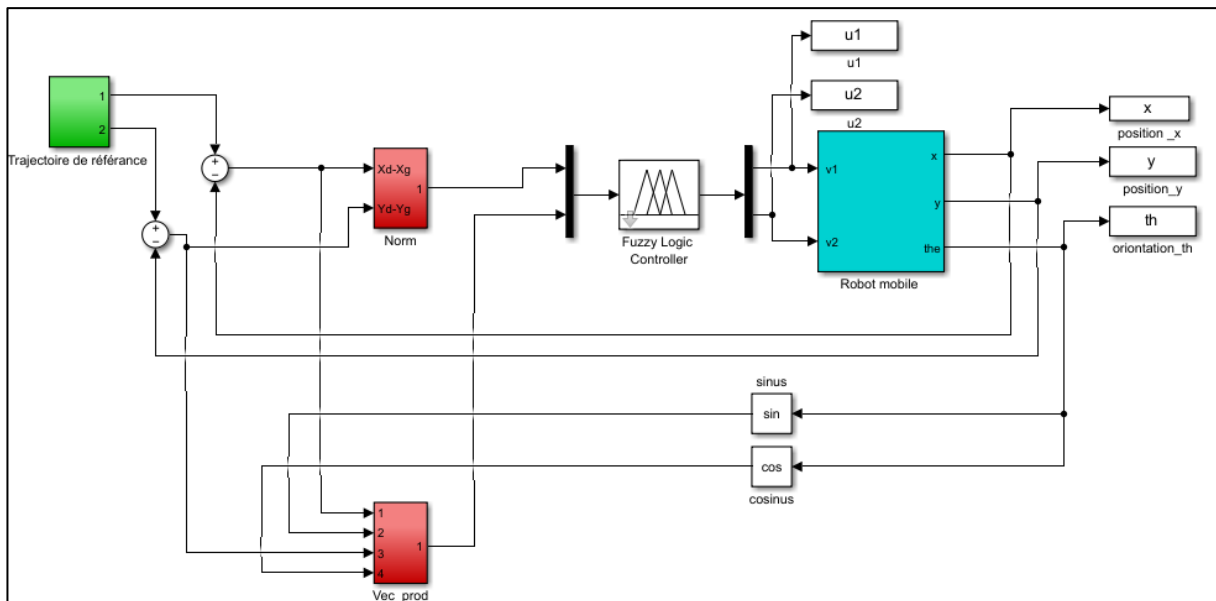


Figure III.17: La simulation avec contrôleur floue.

III.8.1. Structure de base de notre contrôleur floue

La structure de base de notre contrôleur floue est composée de trois blocs le fuzzification, inférence et défuzzification. Dans notre robot mobile nous avons choisi un contrôleur floue de deux entrées et deux sorties, les entrées de notre contrôleur sont : la distance entre le robot et le position souhaité (norm), et l'orientation du robot (Vecprod). Et les sorties de notre contrôleur sont : la vitesse du robot (u1) et la vitesse de rotation (u2).

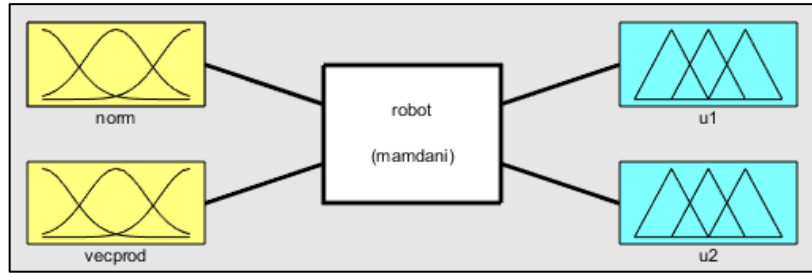


Figure III.18: Conception de notre logique floue.

Le contrôleur floue utilisé est de type MAMDANI avec quatre variables linguistique erreur, l'orientation du robot. Vitesse du robot, angle du robot. Pour chaque variable linguistique a choisi les cinq valeurs linguistiques suivantes :

N : négative ; **Z** : nulle ; **P** : positive ; **Pm** : positif moyenne ; **Pg** : positif grand.

- *Fuzzification et inférence*

Les valeurs fournies par les capteurs du robot sont converties en sous-ensembles flous qui caractérisent les variables linguistiques associées. Après plusieurs tests de fonction d'appartenance ont utilisé trois formes des fonctions d'appartenances :

- Triangulaire
- Trapézoïdal
- Gaussien

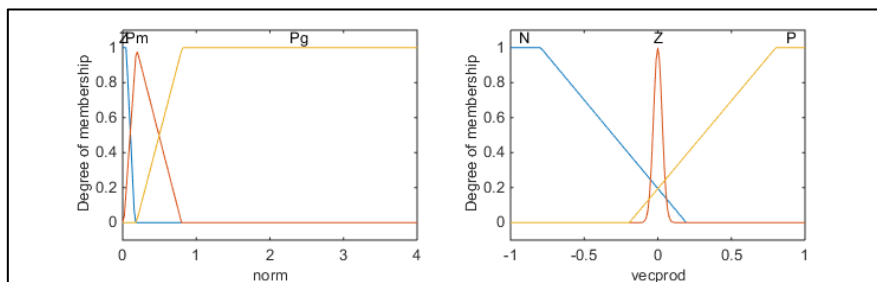


Figure III.19: Les Fonctions d'appartenances pour les entrées.

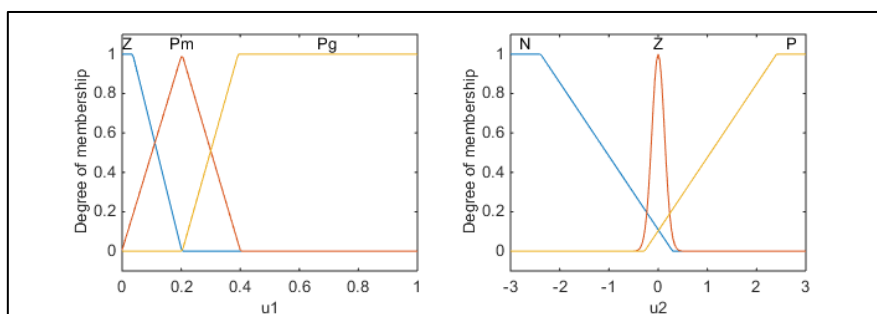


Figure III.20: Les Fonctions d'appartenances pour les sorties.

Nous avons utilisé 6 règles pour l'étape d'inférence pour représentant le lien entre les différentes variables entrées-sorties :

1. Si (vecprod est P) alors (u1 est Pm) et (u2 is P)
2. Si (vecprod est N) alors (u1 est Pm) et (u2 is N)
3. Si (norm est Pg) alors (u1 est Pg)
4. Si (norm est Pm) alors (u1 est Pg)
5. Si (norm est Z) alors (u1 est Z)
6. Si (vecprod est Z) alors (u2 est Z)

- **Défuzzification**

La méthode la plus couramment utilisée pour la défuzzification consiste à déterminer le centre de gravité de la fonction d'appartenance résultante. Cette méthode est la plus utilisée en raison de sa simplicité de calcul et de sa rapidité.

Pour le Contrôleur PID-floue représenté par la figure (III.23) :

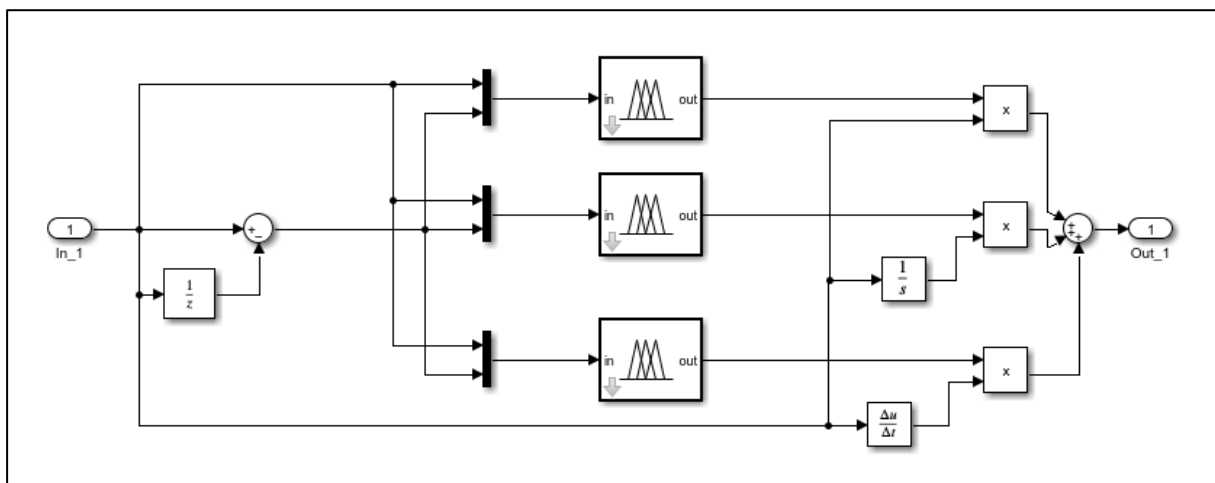


Figure III.21: PID-floue contrôler sur Simulink.

III.8.2. Optimisation des paramètres du régulateur PID par PSO et GJO

Le manque de méthodes exactes pour concevoir précisément les régulateurs PID a conduit les chercheurs à adopter les algorithmes évolutionnaires, en raison de leur capacité à explorer globalement des environnements complexes. Dans cette étude, nous appliquons les essais particuliers (PSO) et Chacal Doré (GJO) pour optimiser les paramètres d'un régulateur PID utilisé dans le contrôle de divers systèmes cinématique. La figure (III.15) présente le schéma de cette approche d'optimisation, où l'on observe la boucle de régulation classique agrémentée d'une couche supérieure basée sur PSO et le même chose avec GJO. Ce dernier joue un rôle

importance en fournissant les paramètres optimaux du régulateur PID afin d'améliorer les performances de la boucle de régulation.

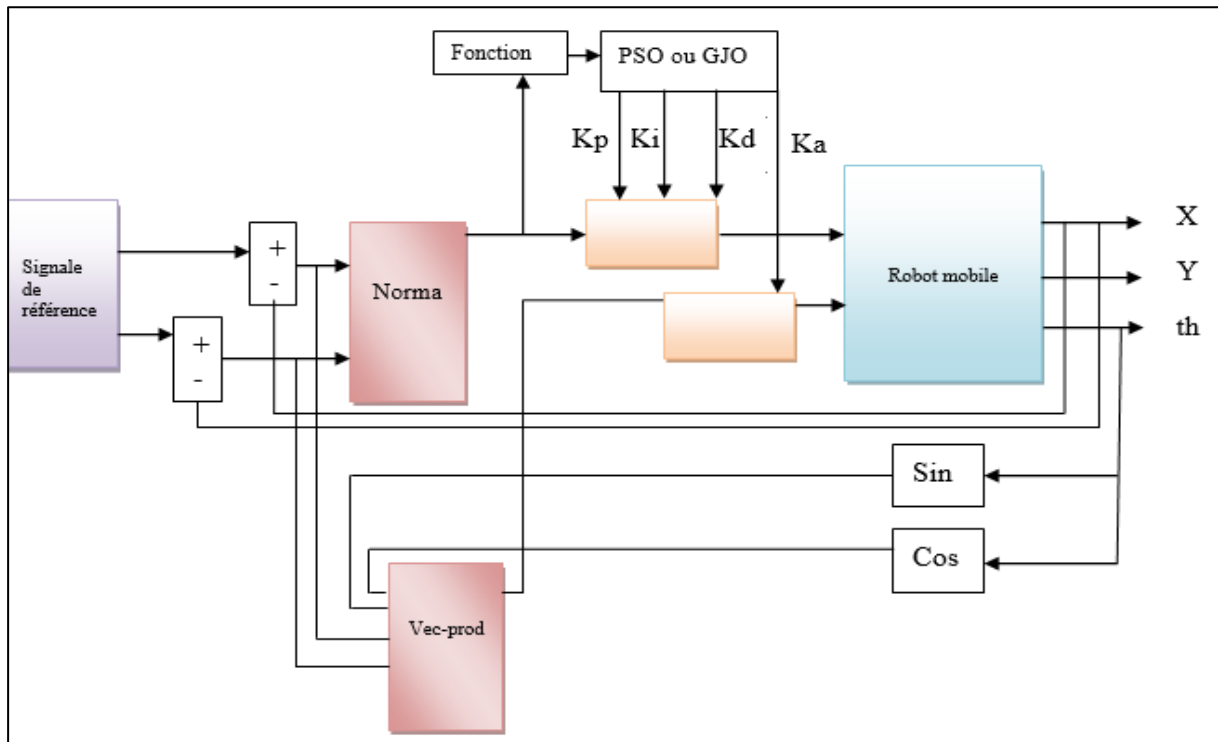


Figure III.22: Approche d'optimisation du PID par l'algorithme PSO et GJO.

III.9. Conclusion

Dans ce chapitre, nous avons présenté brièvement la conception et les principes principaux de différents contrôleurs qui sera utilisé ensuite, comme le contrôleur flou, contrôleur PID-flou et l'optimisation des régulateurs PID à l'aide de techniques d'optimisation méta-heuristiques avancées comme PSO et GJO.

Dans le chapitre suivant, nous appliquerons ces contrôleurs sur notre simulation du robot mobile pour le contrôler sur différentes trajectoires de références et nous discutons les résultats obtenus.

Chapitre IV:
Simulation et discussion

IV.1. Introduction

Dans ce chapitre, nous simulons et utilisons le modèle cinématique du robot mobile sur Matlab et Simulink avec différents types de contrôleurs (Floue, PSO-PID, GJO-PID, Floue-PID) afin d'étudier le contrôle du suivi de la trajectoire, on utilise des trajectoires de références proches de la réalité.

IV.2. Simulation et Discussion

Toutes les simulations sont effectuées sous Matlab-Simulink avec un modèle cinématique du robot et comme trajectoires de références les trajectoires suivantes :

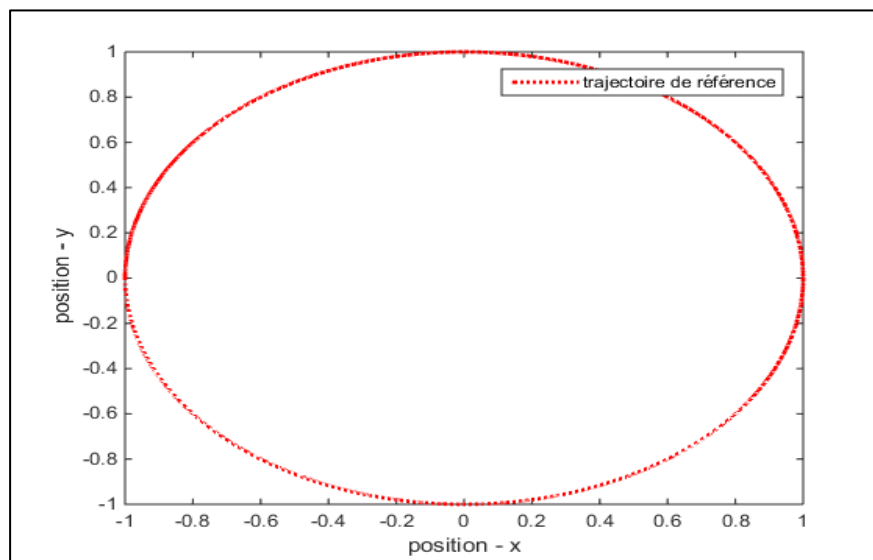


Figure IV.1 : Trajectoire de référence forme cercle (Tr1).

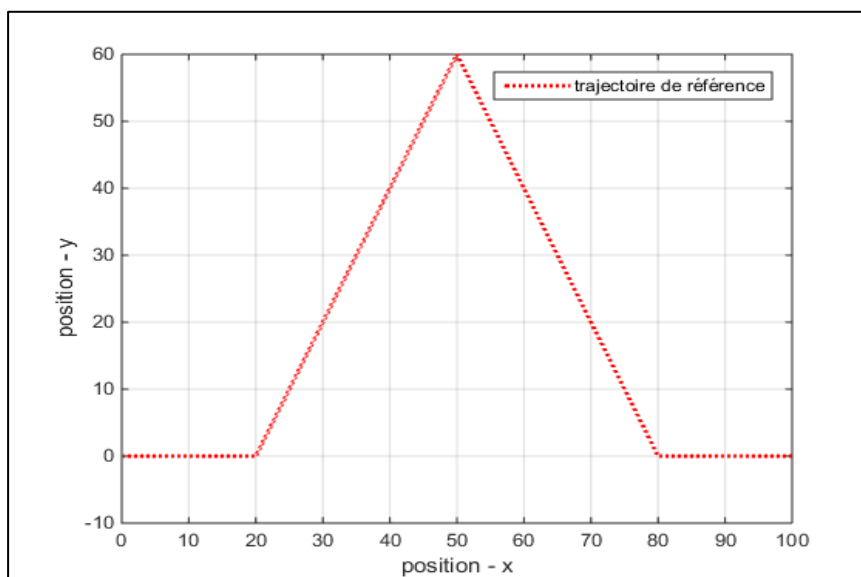


Figure IV.2: Trajectoire de référence forme triangle (Tr2).

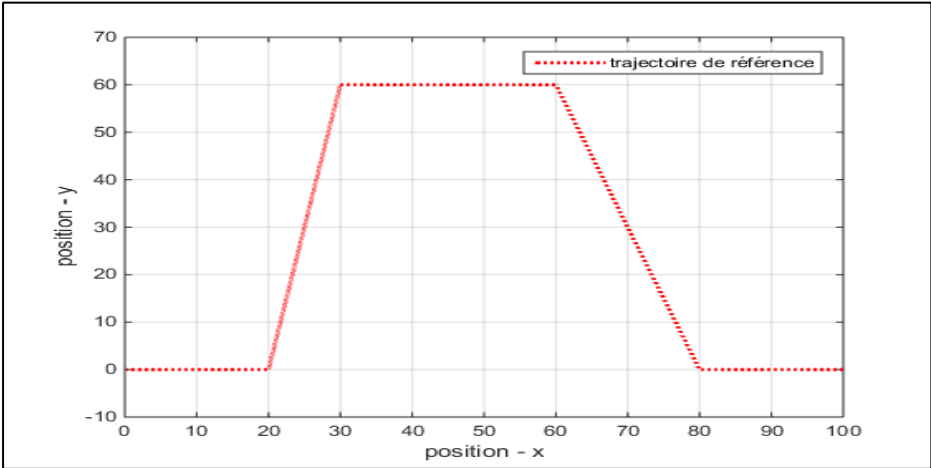


Figure IV.3: Trajectory of reference form trapezoidal (Tr3).

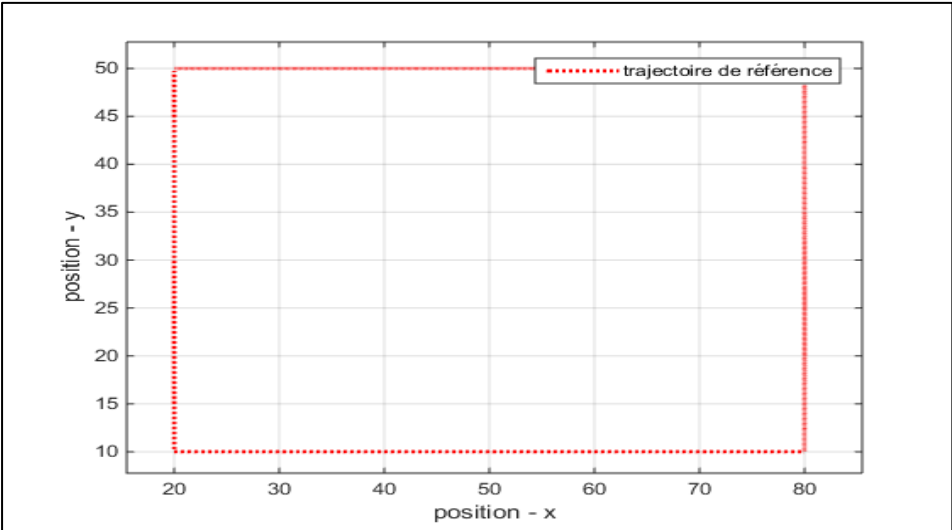


Figure IV.4: Trajectory of reference form square (Tr4).

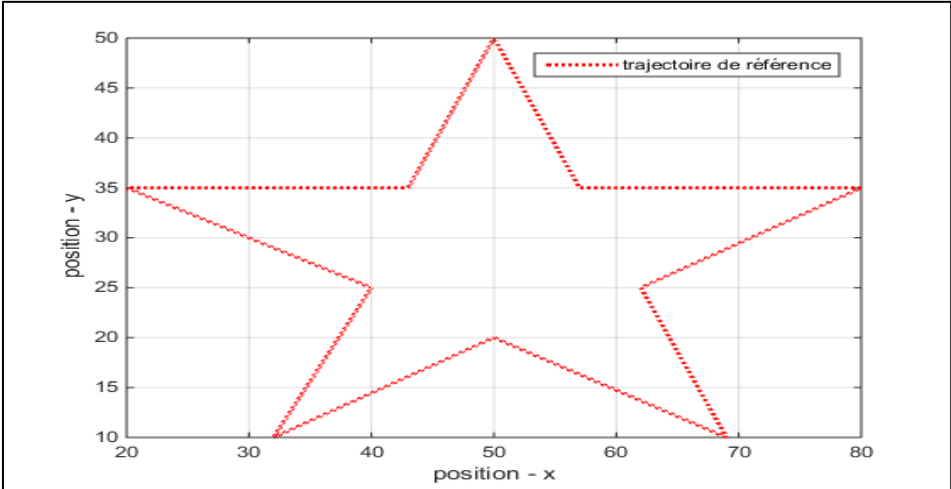


Figure IV.5: Trajectory of reference form trapezoidal (Tr5).

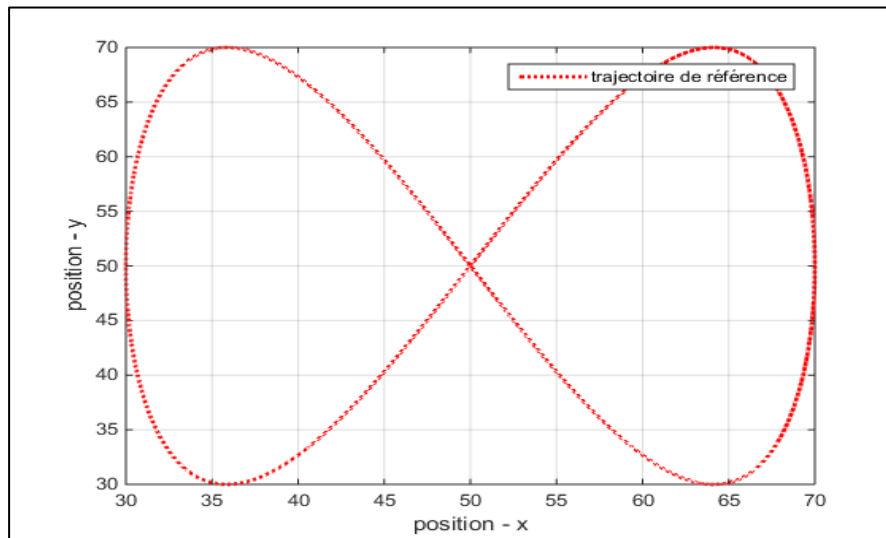


Figure IV.6: Trajectoire de référence forme infini (Tr6).

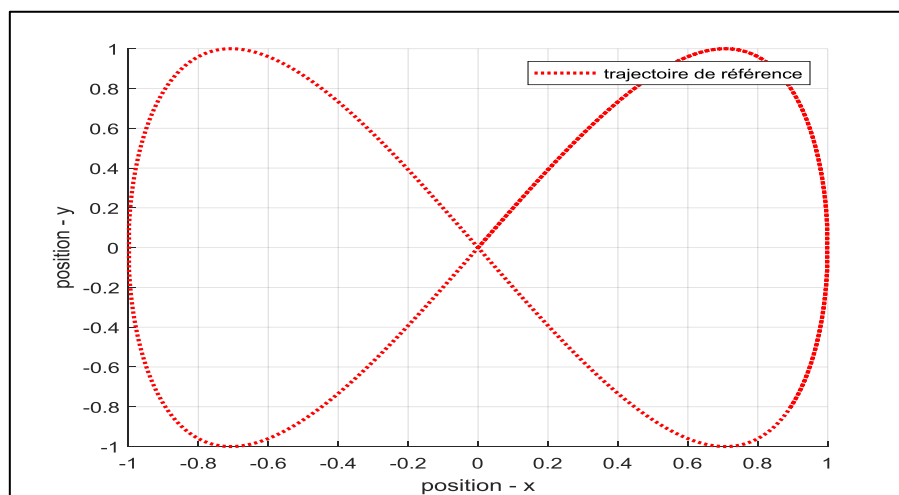


Figure IV.7: Trajectoire de référence forme infini (Tr7).

La simulation est organisée comme suite :

- Contrôle du suivi de la trajectoire avec un contrôleur floue (en anglais Fuzzy Controller).
- Contrôle du suivi de la trajectoire avec un contrôleur PID-floue (en anglais Fuzzy-PID).
- Contrôle du suivi de la trajectoire avec un contrôleur PID optimisé par PSO.
- Contrôle du suivi de la trajectoire avec un contrôleur PID optimisé par GJO.

IV.3. La simulation avec le contrôleur floue

La simulation est réalisée en utilisant un modèle cinématique d'un robot mobile type véhicule et la boîte à outils de logique floue du Matlab-Simulink.

IV.3.1. La simulation avec le trajectoire Tr1

Les différentes configurations de trajectoire de références utilisés sont illustrées dans le tableau suivant :

Tableau IV.1: Les points de départ (CF).

Tableau de test	Tableau de test		
	Position X	Position Y	Angel θ
Test 1	-0.5	0.5	-1
Test 2	0.5	0.5	0.5
Test 3	0.0	0.5	0.0
Test 4	-0.5	0.0	-1.0

Les résultats obtenus sont :

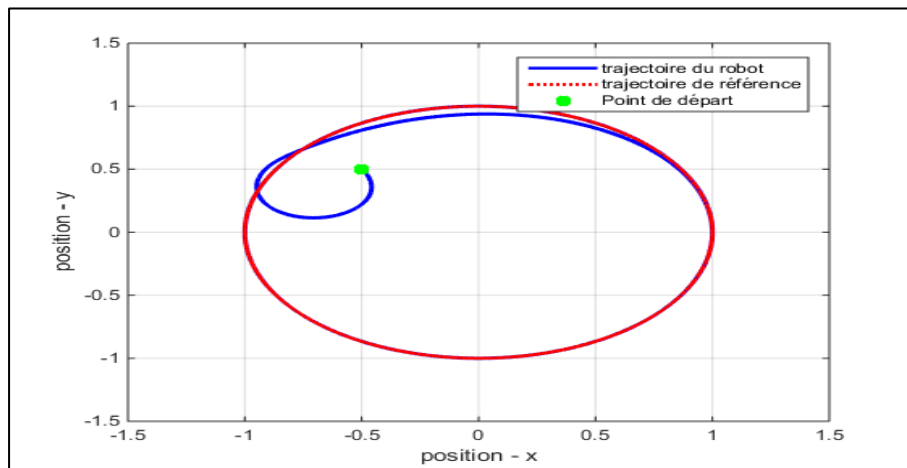


Figure IV.8: résultat de test 1.

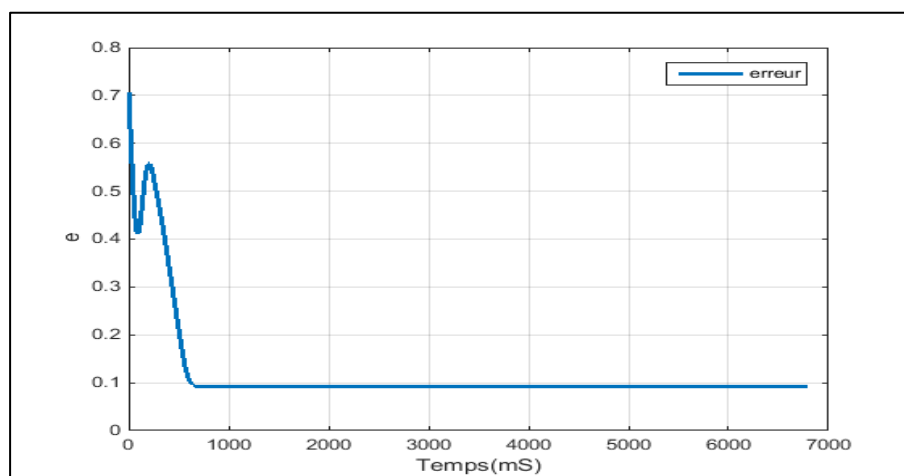


Figure IV.9: L'erreur de test 1.

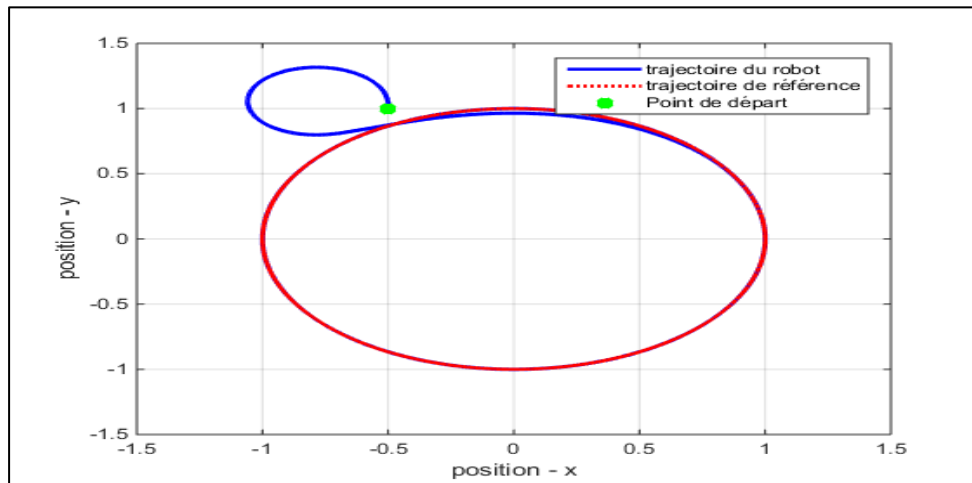


Figure IV.10: Résultat de test 2.

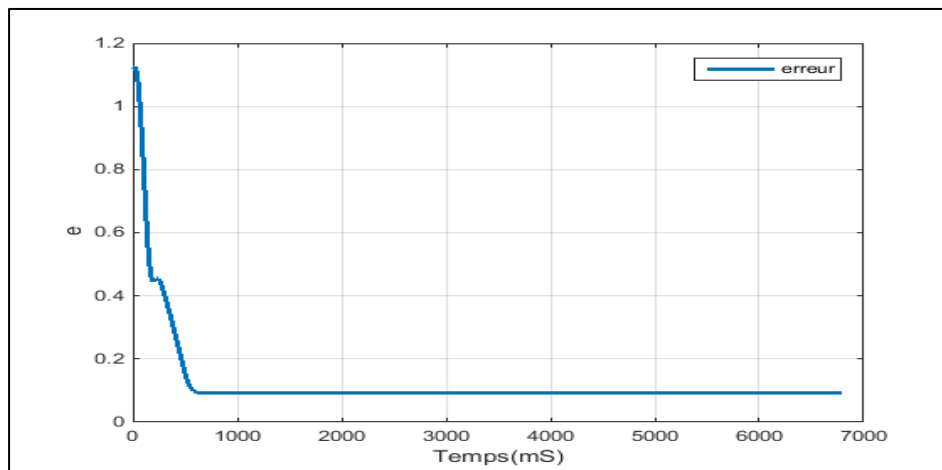


Figure IV.11: L'erreur de test 2

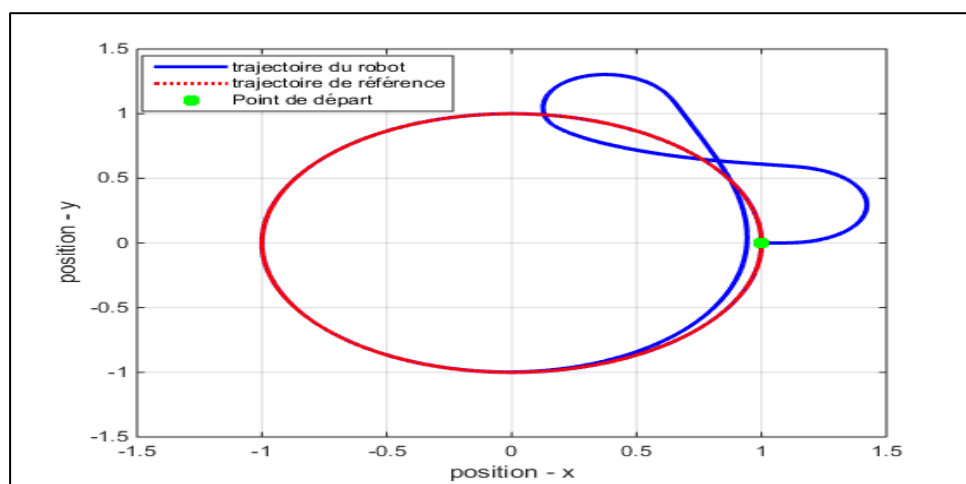


Figure IV.12: Résultat de test 3.

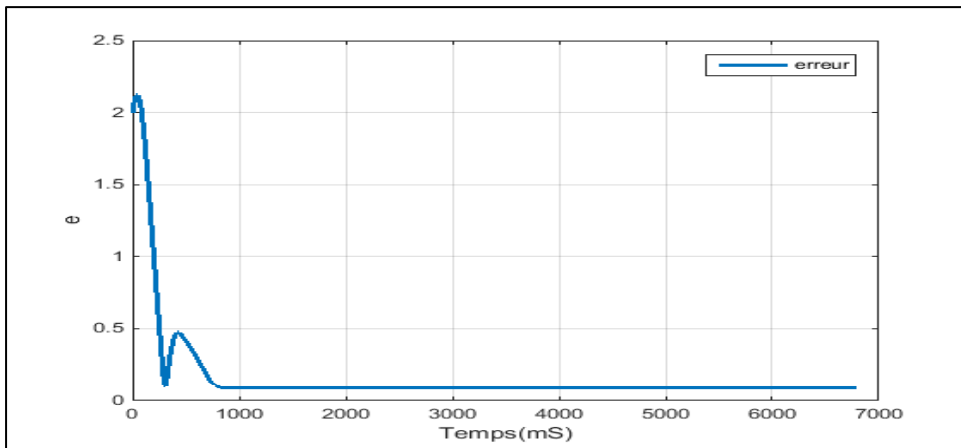


Figure IV.13: L'erreur de test 3.

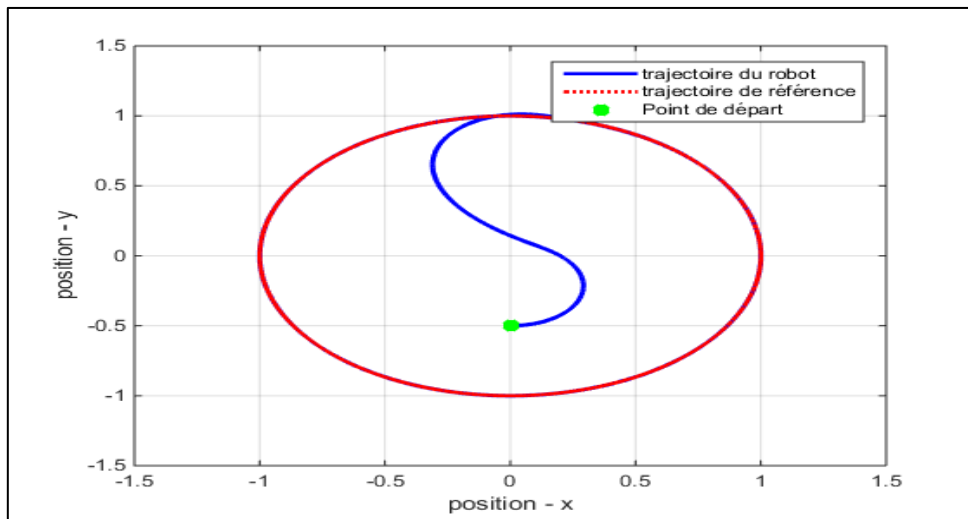


Figure IV.14: Résultat de test 4.

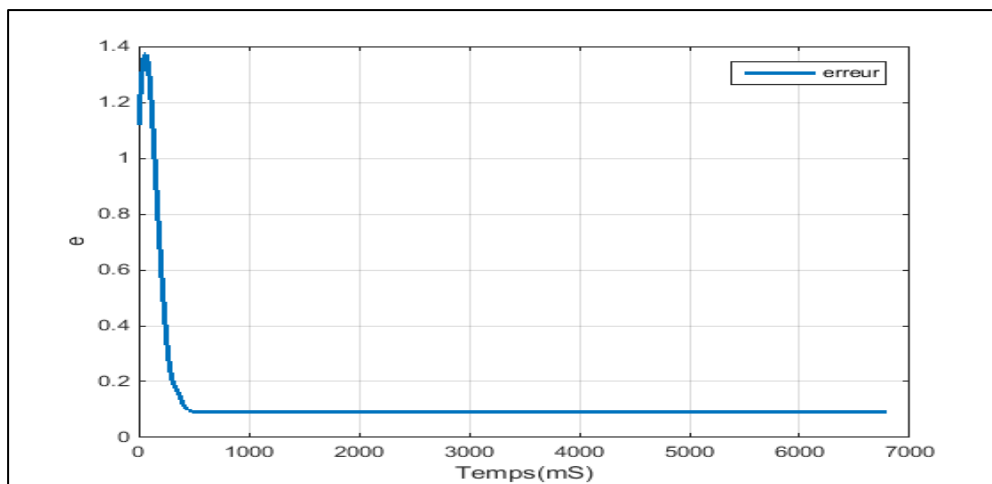


Figure IV.15: L'erreur de test 4.

Tableau IV.2 : Résultat des tests avec Tr1 (CF).

Tableau de résultat	Erreur moyen (cm)
Test 1	0.1203
Test 2	0.1256
Test 3	0.1667
Test 4	0.1301

D'après les résultats obtenus nous remarquons que :

Le contrôleur flou à ajusté parfaitement la trajectoire du robot mobile par rapport la trajectoire de référence dans toutes les tests.

Il est clair qu'il existe des différences significations dans la moyenne d'erreur entre les différents tests. Par exemple, le Test 3 donne une erreur moyenne nettement plus élevée par rapport aux autres tests et le test 1 donné une erreur moyenne nettement plus basse par rapport aux autres tests. Cela indique que les conditions initiales peuvent avoir un impact considérable sur la précision du système. Le meilleur résultat obtenu était en test 1 avec le pointe de départ $[-0.5 \ 0.5 \ -1]$.

Ces résultats offrent une vision claire de la manière dont le contrôleur répond aux conditions initiales, et nous pousse à chercher d'amélioration les performances de notre architecture de contrôle par rapport les bornes des fonctions d'appartenances.

IV.3.2. La simulation avec trajectoires différentes (Tr2, Tr 3, Tr4, Tr5)

Nous allons maintenant réaliser plusieurs tests avec plusieurs trajectoires différentes (tous les tests avec $\theta = 0$). Pour cela, il est nécessaire de modifier les fonctions d'appartenances du contrôleur floue, pour les trajectoires Tr_2, Tr_3, Tr_4, Tr_5 nous l'utiliserons les fonctions d'appartenances suivantes :

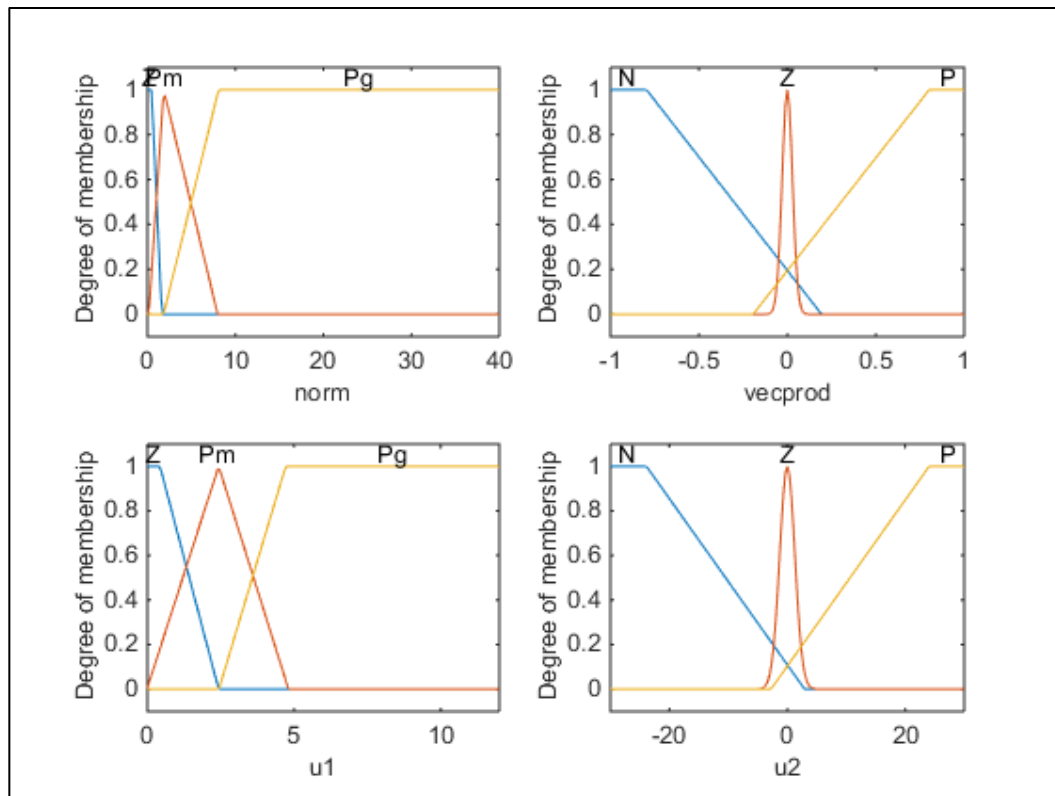


Figure IV.16: Les fonction d'appartenance de trajectoires différentes (CF).

Tableau IV.3 : Résultat des tests de trajectoires différentes (CF).

Trajectoire	$e_{moyenne}$
Tr_2	0.4350
Tr_3	0.4179
Tr_4	0.4113
Tr_5	0.4284

À partir des résultats du tableau, nous pouvons observer que les valeurs moyennes d'erreur sont presque constantes entre les quatre tests des trajectoires (Tr_2, Tr_3, Tr_4, Tr_5), tandis que nous obtenons la plus faible valeur d'erreur dans le test de la trajectoire de Tr_4 , qui représente la meilleure performance obtenue dans ces tests.

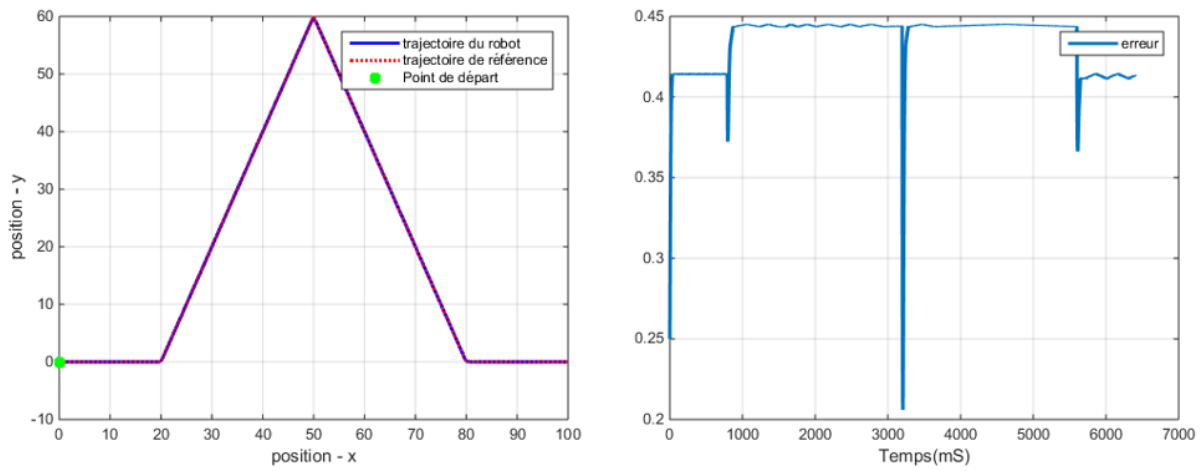


Figure IV.17: Résultat de la trajectoire Tr2 et son erreur (CF).

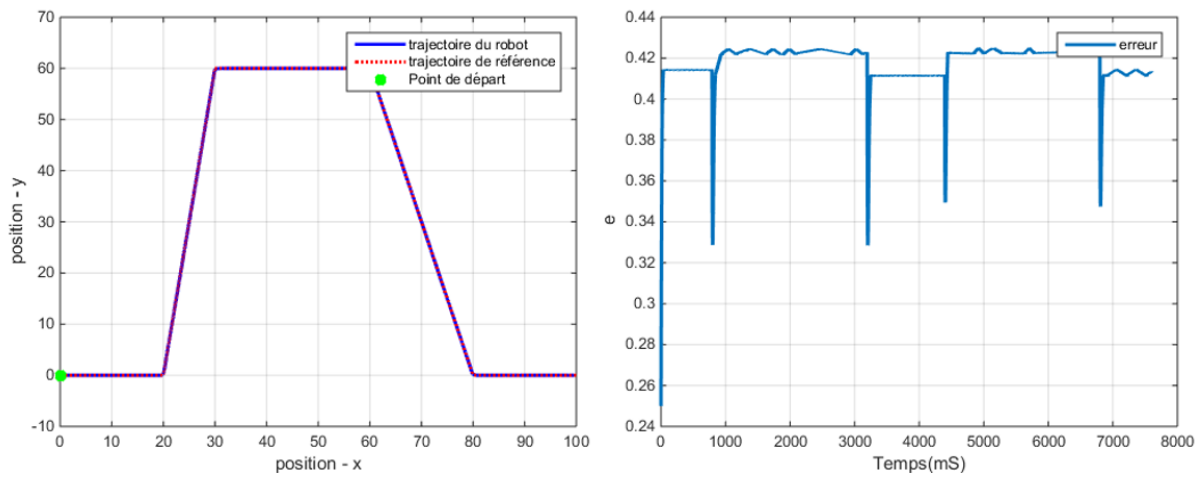


Figure IV.18: Résultat de la trajectoire Tr3 et son erreur (CF).

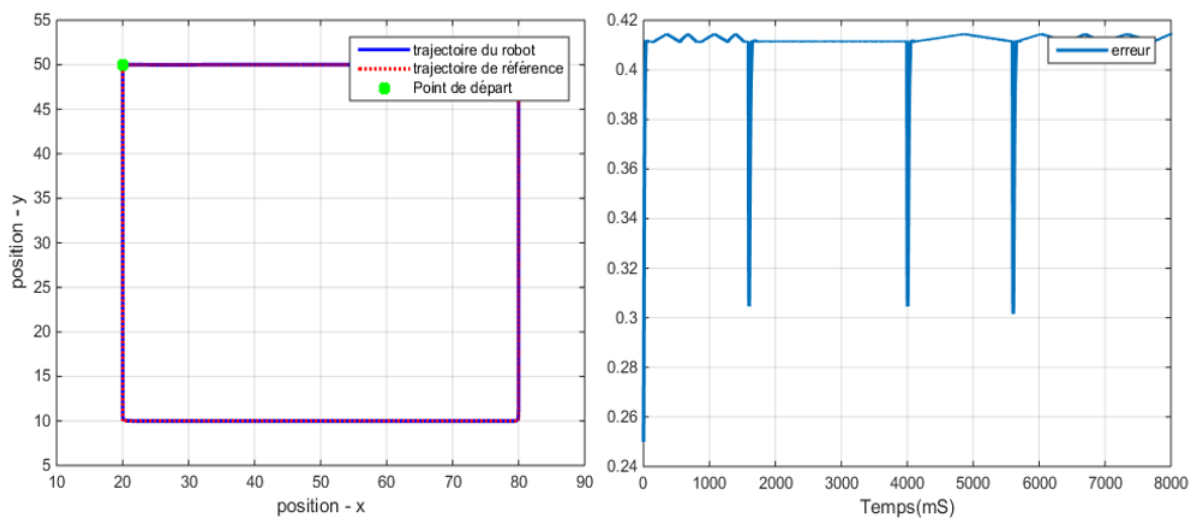


Figure IV.19: Résultat de la trajectoire Tr4 et son erreur (CF).

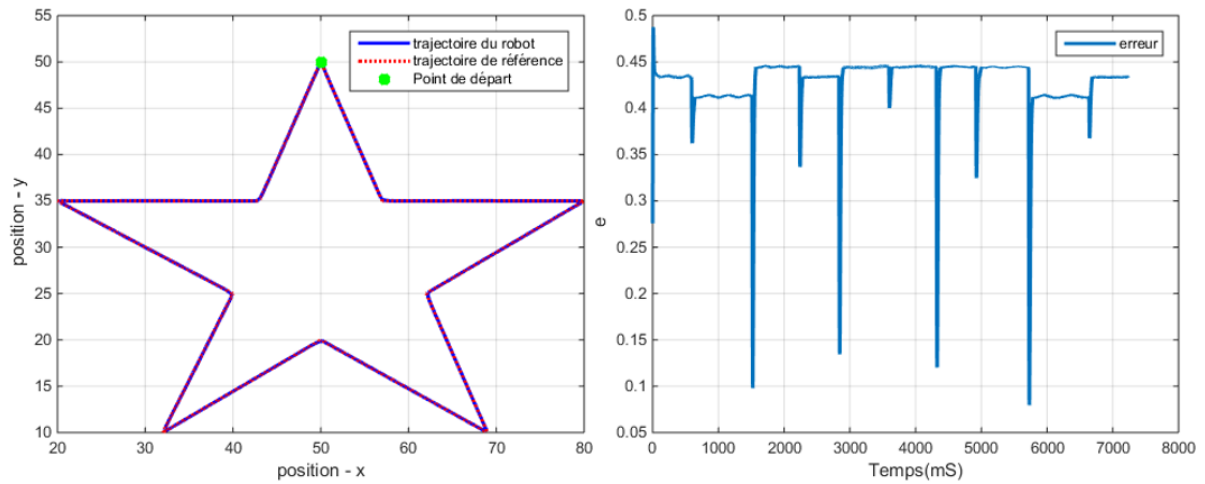


Figure IV.20: Résultat de la trajectoire Tr5 et son erreur (CF).

IV.3.3. La simulation avec la trajectoire Tr6

Pour la trajectoire Tr_6 nous l'utiliserons le point de départ [50 50 1] et les fonctions d'appartenances suivantes :

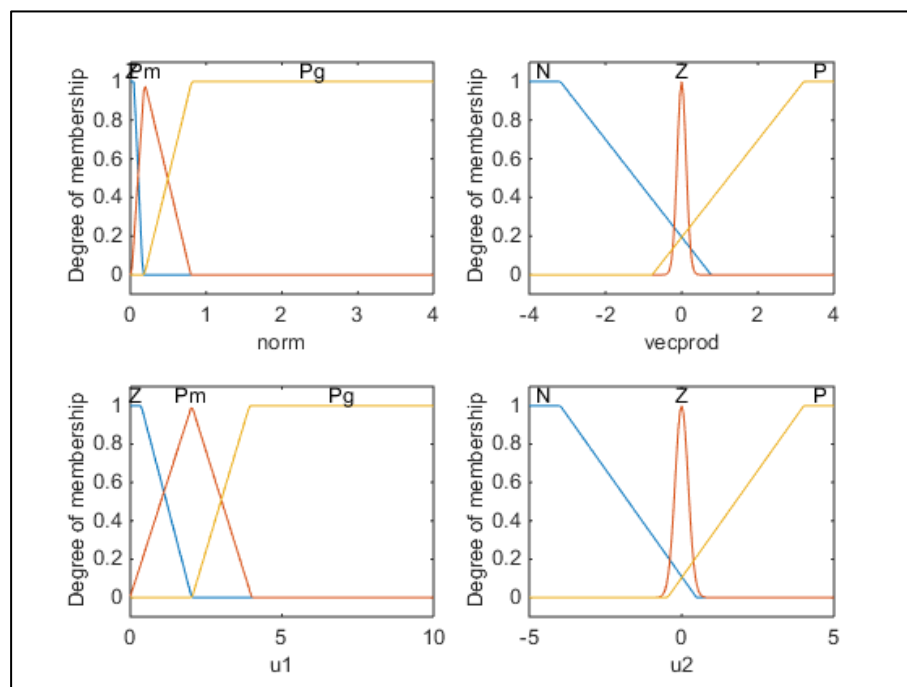


Figure IV.21: Les fonction d'appartenance de trajectoires Tr6 (CF).

Tableau IV.4 : Résultat de test de trajectoire Tr6 (CF).

Trajectoire	$e_{moyenne}$
Tr_6	0.0712

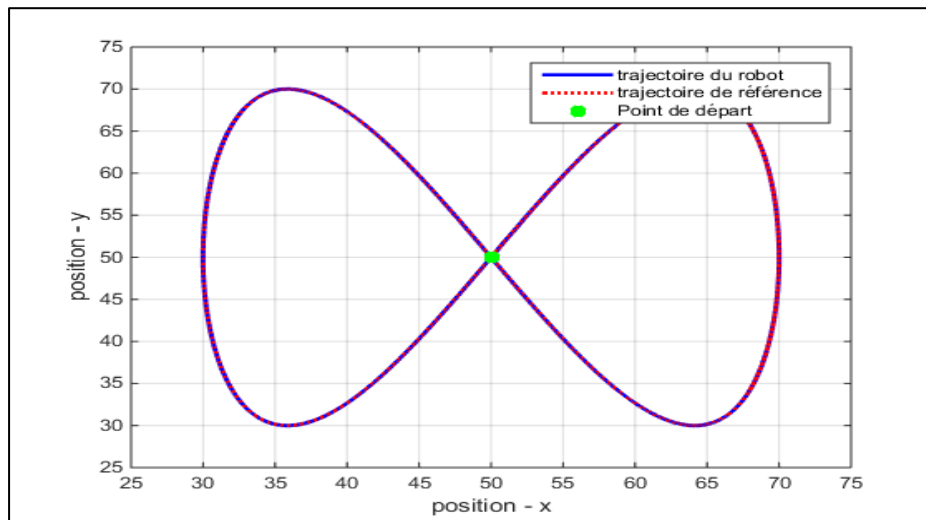


Figure IV.22: résultat de la trajectoire Tr_6 (CF).

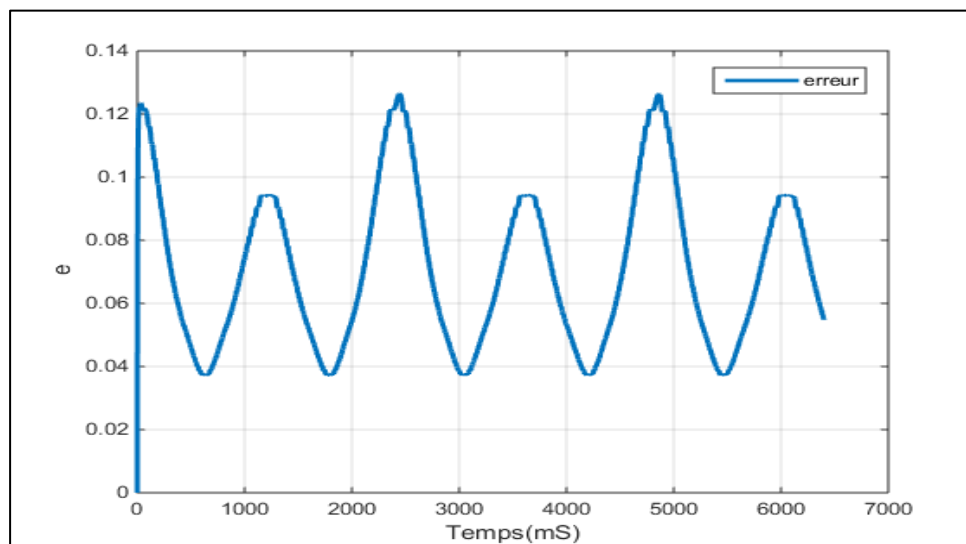


Figure IV.23: résultat de l'erreur.

Nous remarquons que l'erreur moyenne de ce test est très faible par rapport aux tests précédents.

IV.3.4. Discussion

Les résultats de tous les tests démontrent une capacité à suivre parfaitement les trajectoires de références. Le test de la trajectoire infini (Tr_6) se démarque comme le meilleur en raison de sa précision supérieure par rapport aux autres tests où l'erreur moyenne est égale 0.0712. Les résultats montrent aussi que le contrôleur flou permet au robot mobile de s'adapter efficacement aux différentes formes de trajectoires, en minimisant l'erreur moyenne de position. Cela suggère que le contrôleur flou est capable de moduler les commandes du robot mobile

pour répondre aux exigences spécifiques de chaque trajectoire, bien que des améliorations pourraient encore être envisagées pour réduire davantage l'erreur comme par exemple l'ajustement des bornes des fonctions d'appartenances.

IV.4. La simulation avec le contrôleur PID optimisé par PSO

Nous utiliserons la technique d'optimisation méta-heuristique (PSO) discutée dans le chapitre III pour ajuster les gains des contrôleurs PID.

Les paramètres utilisés pour l'algorithme sont illustrés dans le tableau (IV.5):

Tableau IV.5: Paramètres d'algorithme PSO.

Paramétré PSO	La valeur
W	0.9
C1=C2	2.05
Dim	3

Pour contrôler la vitesse de rotation du robot on utilise un correcteur proportionnel $K_a=100$ [30].

IV.4.1. La simulation avec trajectoires différentes (Tr2, Tr 3, Tr4, Tr5)

Pour les trajectoires Tr 3, Tr4, Tr5 on utilise la trajectoire Tr_2 , pour faire les tests. Après les simulations, les résultats obtenus sont illustrés dans le tableau suivant :

Tableau IV.6 : Gains et l'erreur moyenne des tests avec trajectoires déférent (PSO-PID).

Tr	N	interation	Kp	Ki	Kd	$e_{moyenne}$
Tr1	30	30	7.4420	1.4291	0.6080	0.0122
Tr2	20	20	9.877	0.6316	0.979	0.0978
Tr3	30	30	25	1.0087	0.6079	0.0317
Tr4	10	10	17.5304	0	0.0144	0.1424
Tr5	10	10	17.5304	0	0.0144	0.1553

Tableau IV.7 : Les points de départ (PSO).

Tr	Position X	Position Y	Angel θ
Tr1	-1	0	0
Tr2 et Tr3	0	0	0
Tr4	20	50	0
Tr5	50	50	0

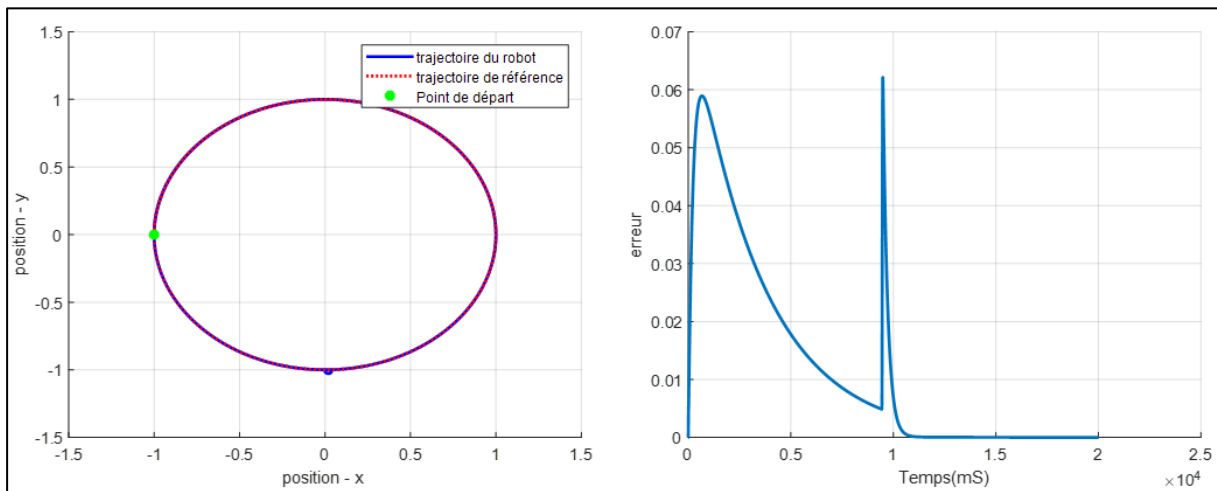


Figure IV.24: Résultat de la trajectoire Tr1 et son erreur (PSO).

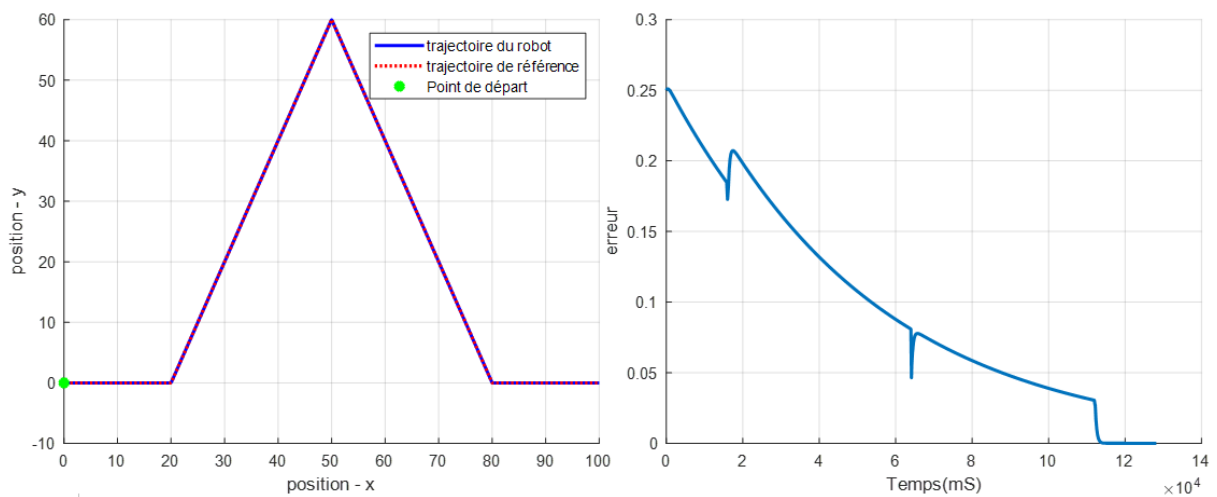


Figure IV.25: Résultat de la trajectoire Tr2 et son erreur (PSO).

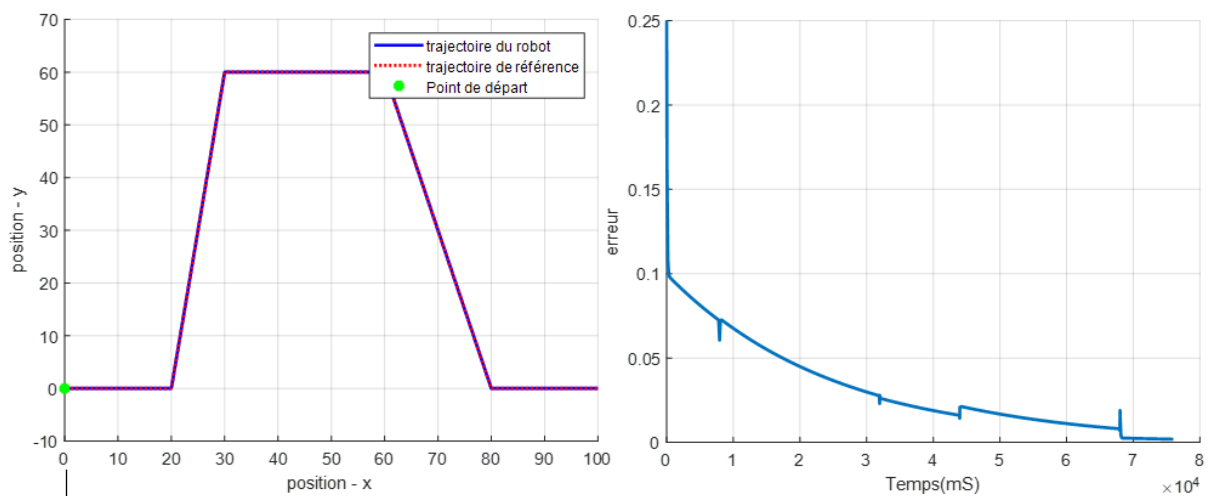


Figure IV.26: Résultat de la trajectoire Tr3 et son erreur (PSO).

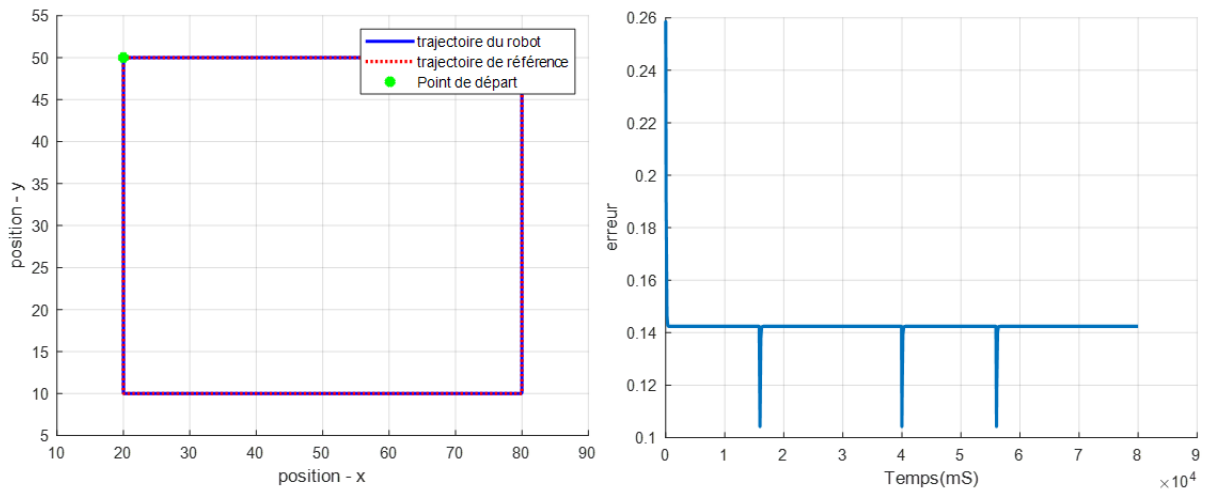


Figure IV.27: Résultat de la trajectoire Tr4 et son erreur (PSO).

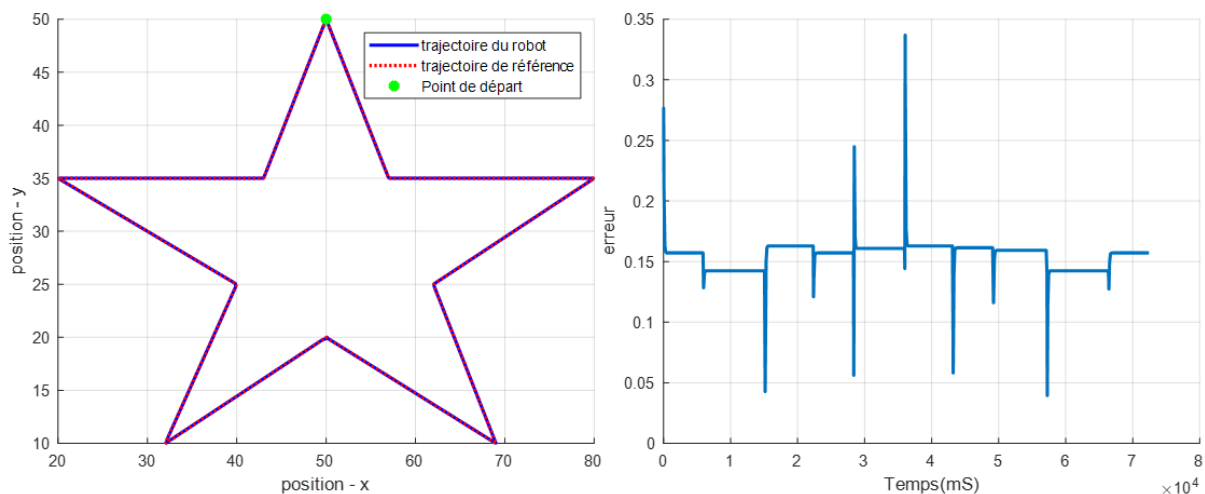


Figure IV.28: Résultat de la trajectoire Tr5 et son erreur (PSO).

Nous observons d’après les résultats obtenus que le suivi du trajectoire est meilleure avec les trajectoires de référence Tr1, Tr2 et Tr3 et l’optimale avec la trajectoire cercle (Tr1) où l’erreur est égale 0.0122.

IV.4.2. La simulation avec la trajectoire Tr7

Pour le robot suivre la trajectoire de Tr_7 , nous ajouterons une dimension supplémentaire aux paramètres de PSO (Dim=4)

Quatre tests différents ont été effectués avec PSO-PID sur la trajectoire Tr_7 . Il nous est donné les valeurs suivantes :

Tableau IV.8 : Gains des tests avec trajectoire Tr7 (PSO-PID).

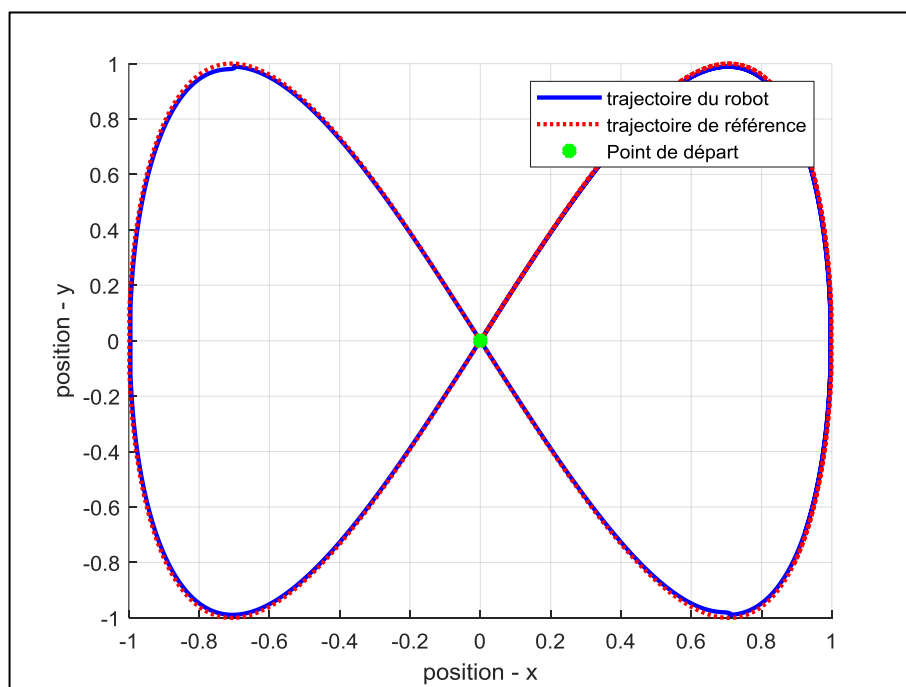
Test	N	Iteration	K_p	K_i	K_d	K_a
1	10	10	0.8882	0	0	2.3753
2	20	20	1.0398	0	0	3.6088
3	30	30	0.7388	0	0	18.3262

On teste les valeurs de K_p , K_i , K_d , K_a sur la trajectoire Tr_7 , ce qui nous donne les valeurs moyennes d'erreur suivante :

Tableau IV.9 : Les valeurs de l'erreur Tr_7 (PSO-PID).

Test	$e_{moyenne}$
1	0.2180
2	0.1858
3	0.1206

Nous avons choisi le meilleur résultat obtenu, qui est le test 3 avec les valeurs de $K_p = 0.7388$, $K_i = 0$, $K_d = 0$ et $K_a = 18.3262$ et le point de départ $[0 \ 0 \ 1]$ pour tracer la trajectoire du robot et la courbe d'erreur :

**Figure IV.29:** Résultat de test 3 sur la trajectoire Tr7 (PSO).

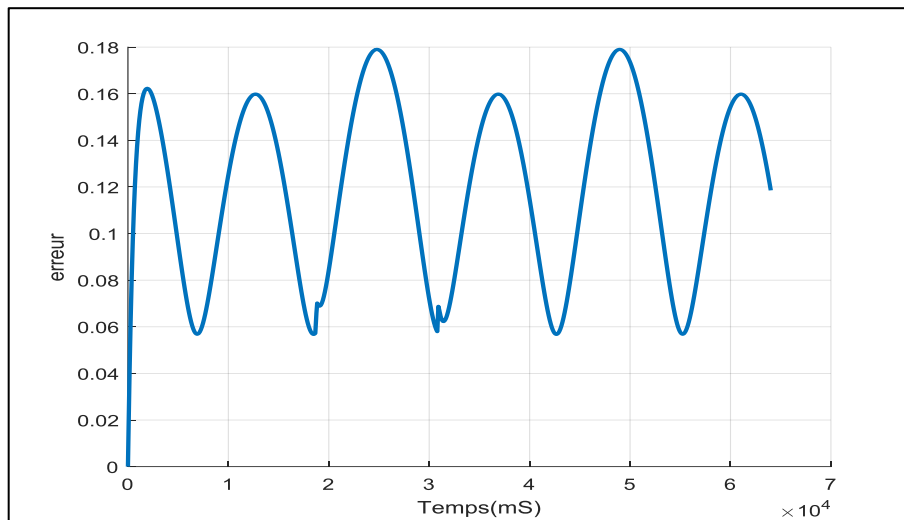


Figure IV.30: Résultat de l'erreur Tr_7 (PSO).

On voit que l'augmentation du nombre de particules et d'itérations semble avoir amélioré globalement les performances du système, avec des erreurs plus faibles observées dans les tests plus élevés (Test 3) par rapport aux tests inférieurs (Test 1) et le robot suivi la trajectoire le test 3 avec une grande précision.

IV.4.3. Discussion

Les résultats de tous les tests démontrent une capacité à suivre parfaitement les trajectoires de référence. Le test de la trajectoire Tr_1 se démarque comme le meilleur paramètre en raison de sa précision supérieure par rapport aux autres tests. Ces améliorations sont représentées par la configuration optimale pour atteindre les performances désirées.

Les résultats montrent que le PSO-PID permet au robot de s'adapter efficacement aux différentes formes de trajectoires, en minimisant l'erreur moyenne. Cela suggère que le PSO-PID est capable de moduler les commandes du robot pour répondre aux exigences spécifiques de chaque trajectoire, bien que des améliorations pourraient encore être envisagées pour réduire davantage l'erreur pour certaines trajectoires.

IV.5. La simulation avec le contrôleur PID optimisé par GJO

Nous utiliserons dans la simulation la technique d'optimisation (GJO) avec les trajectoires Tr_1 et Tr_7 . Les paramètres utilisés pour l'algorithme sont donnés par le tableau (IV.10):

Tableau IV.10: Paramètres d'algorithme GJO.

Paramètre GJO	La valeur
I_b	[0, 0, 0, 0]
U_b	[25, 25, 25, 25]
Dim	4

IV.5.1. La simulation avec la trajectoire Tr_1

Après différentes du contrôleur GJO-PID avec la trajectoire Tr_1 . Les paramètres optimaux du contrôleur obtenus sont dans le tableau suivant :

Tableau IV.11: Gains des tests avec trajectoire de Tr_1 (GJO-PID).

Test	N	Itération	K_p	K_i	K_d	K_a
1	10	10	0.7388	0.0375	0	8.6391
2	20	20	1.2760	0.0062	0	25

Et les résultats du suivi de la trajectoire sont illustrés dans le tableau suivant :

Tableau IV.12: Les valeurs de l'erreur Tr_1 (GJO-PID).

Test	$e_{moyenne}$
1	0.5511
2	0.3652

Nous avons choisi le meilleur résultat obtenu, qui est le test 2 avec les valeurs de $K_p = 1.2760$, $K_i = 0.0062$, $K_d = 0$ et $K_a = 25$ avec le point de départ $[-1 \ 0 \ 0]$ pour tracer la trajectoire du robot et la courbe d'erreur. Le résultat est comme suivante :

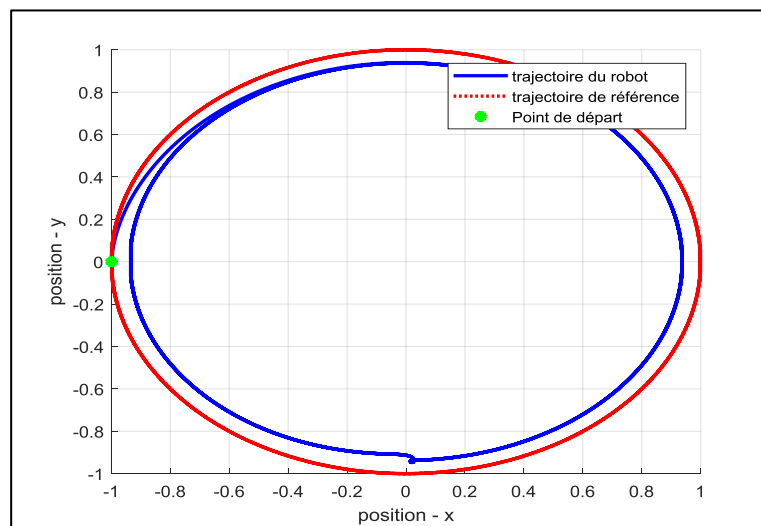


Figure IV.31: Résultat de test 2 sur la trajectoire Tr_1 (GJO).

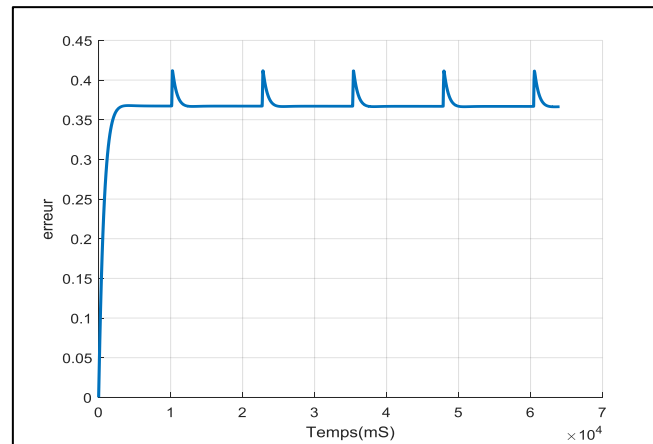


Figure IV.32: Résultat de l'erreur Tr1 (GJO).

L'augmentation du nombre d'agents de recherche et d'itérations semble avoir amélioré globalement les performances du système, avec des erreurs plus faibles observées dans les tests plus élevés (Test 2) par rapport aux tests inférieurs (Test 1). Cependant, le test 3 présente une erreur significativement plus élevée.

IV.5.2. La simulation avec la trajectoire Tr7

Différents tests ont été réalisés avec GJO-PID pour suivre la trajectoire Tr_7 avec le point de départ [0 0 1]. Voici les valeurs obtenues :

Tableau IV.13 : Gains des tests avec trajectoire de Tr7 (GJO-PID).

Test	N	Iteration	K_p	K_i	K_d	K_a
1	10	10	0.9844	0	0	2.87
2	20	20	1.8008	0	0.0189	25

On teste les valeurs de K_p , K_i , K_d sur la trajectoire Tr_7 , ce qui nous donne les valeurs moyennes d'erreur suivante :

Tableau IV.14 : Les valeurs de l'erreur Tr7 (GJO-PID).

Test	$e_{moyenne}$
1	0.1970
2	0.1078

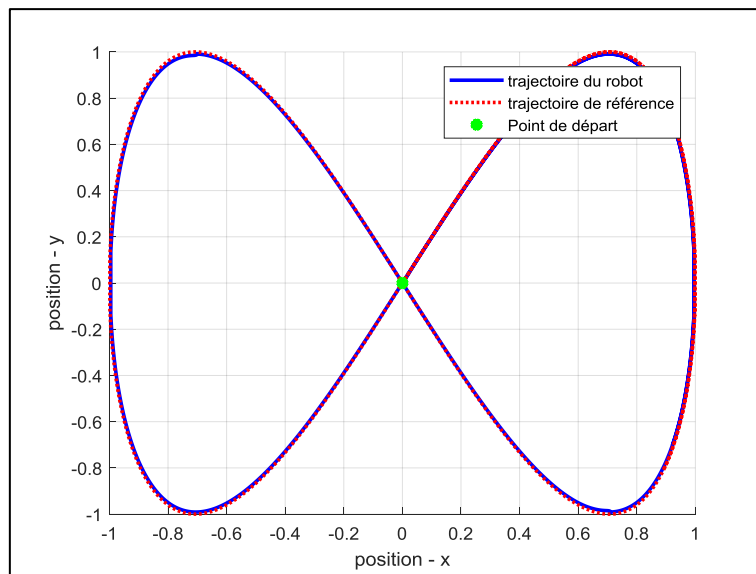


Figure IV.33: résultat de test 3 sur la trajectoire Tr_7 (GJO).

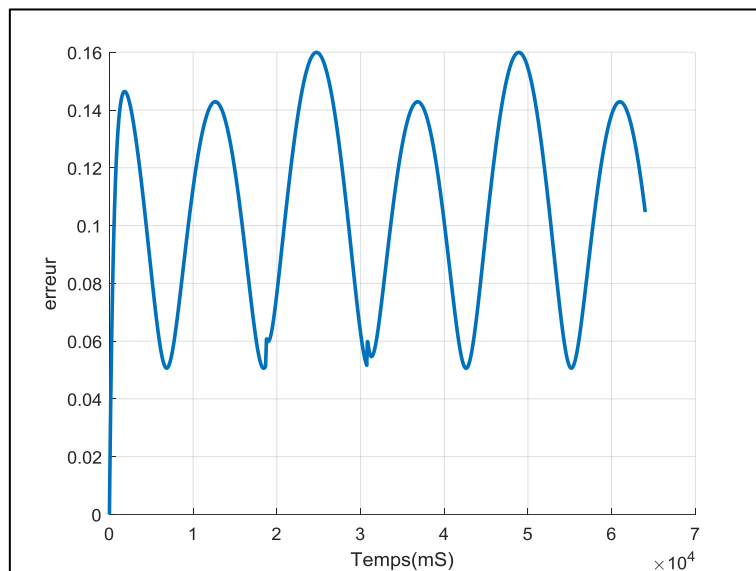


Figure IV.34: résultat de l'erreur Tr_7 (GJO).

L'augmentation du nombre d'agents de recherche et d'itérations semble avoir amélioré globalement les performances du système et Le test de la trajectoire Tr_7 se démarque comme le meilleur paramètre en raison de sa précision supérieure par rapport aux test de Tr_1 et le test de Tr_7 avec PSO-PID

IV.6. La simulation avec PID-floue

Dans cette partie, nous utilisons une trajectoire différente des trajectoires précédentes afin de tester la capacité du contrôleur flou à contrôler le robot pour suivre une trajectoire comme réelle :

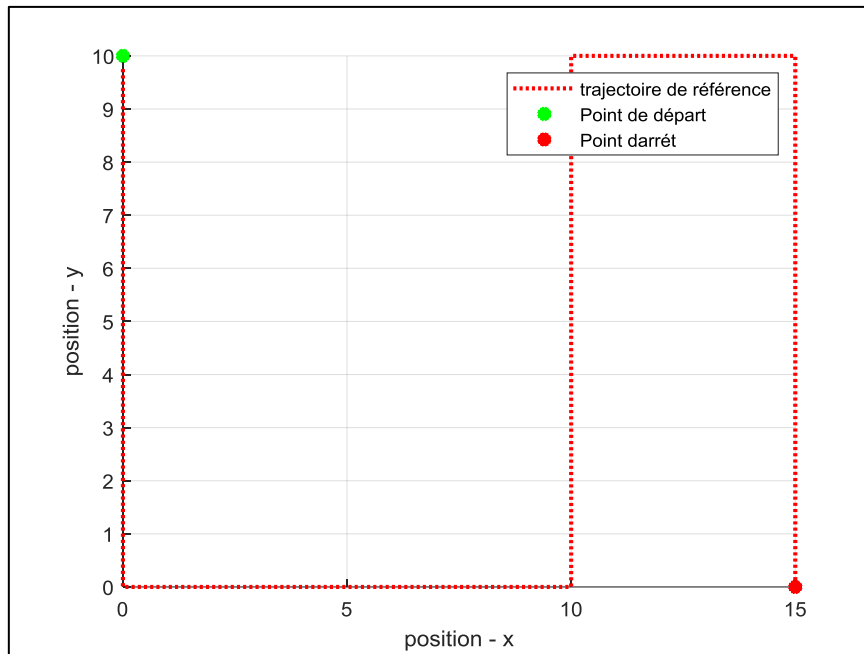


Figure IV.35 : Trajectoire de référence Tr8.

Le contrôleur flou utilisé est de type MAMDANI avec trois variables linguistiques : erreur, delta erreur et la valeur du gain. Le schéma fonctionnel du contrôleur PID-flou est visible sur la figure (III.23). Dans cette structure, 3 blocs de contrôle de logique floue sont utilisés, chacun d'eux a la même entrée, à savoir erreur et delta erreur et la sortie est la valeur du gain. Puisqu'il y a cinq degrés d'appartenance floue pour chaque entrée, cela signifie qu'il y a 25 règles floues utilisées.

Les règles utilisées sont résumées sur le tableau suivante [42]:

Tableau IV.15 : Les règles de PID-floue.

e\ud	NB	NS	Z	PS	PB
NB	DB	DB	DB	DS	NC
NS	DB	DB	DS	NC	IS
Z	DB	DS	NC	IS	IB
PS	DS	NC	IS	IB	IB
PB	NC	IS	IB	IB	IB

DB : Diminution Grande ; **DS** : Diminution Petite ; **NC** : Pas de Changement ; **IS** : Augmentation Petite ; **IB** : Augmentation Grande ;

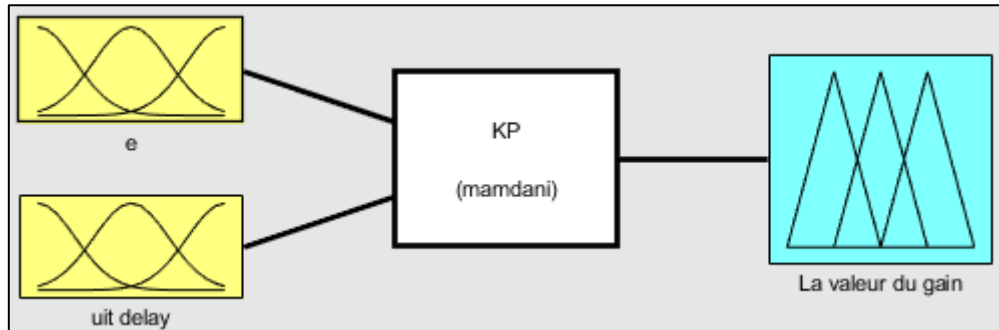


Figure IV.36 : Structure de contrôleur floue (PID-floue).

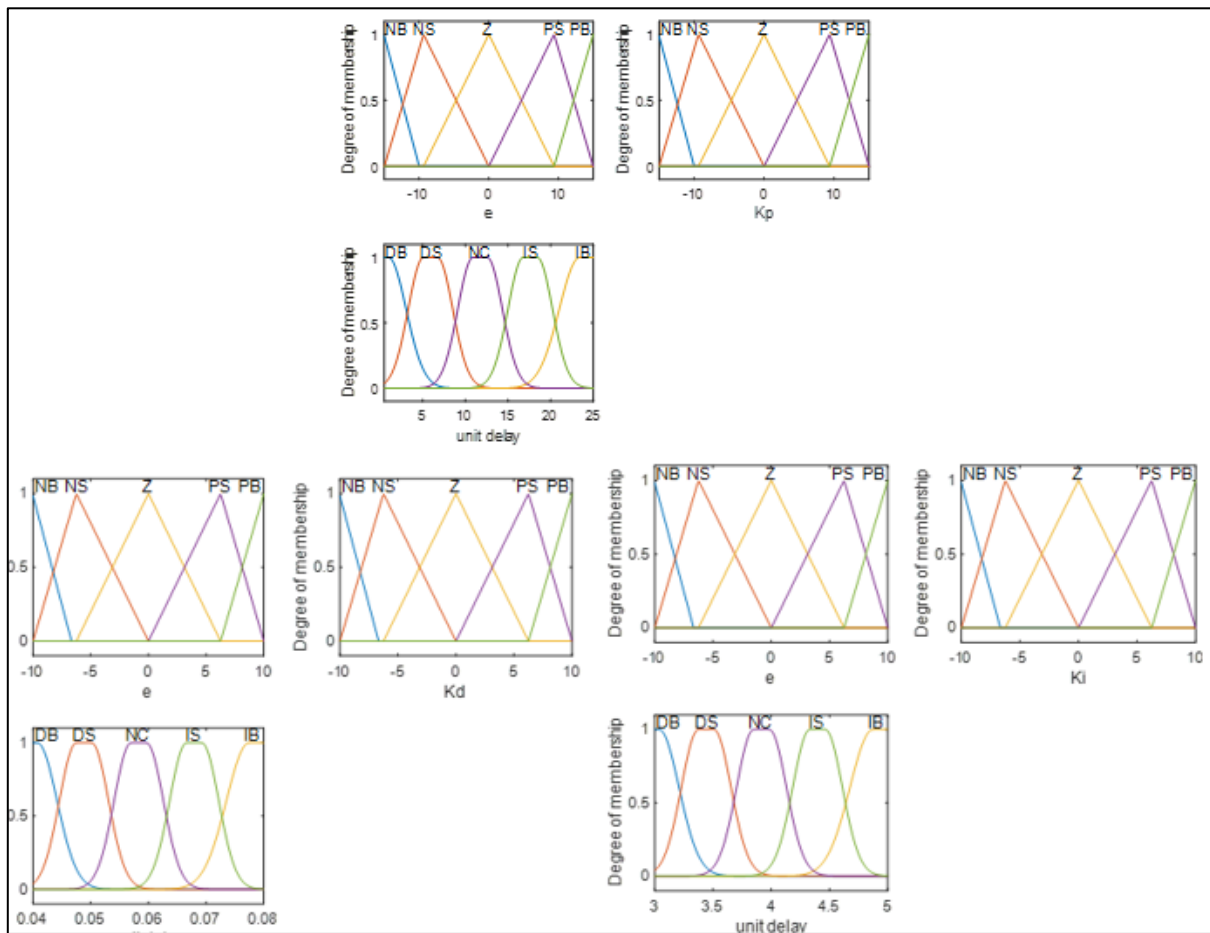


Figure IV.37: Les fonctions d'appartenance des paramètres K_p , K_i , K_d .

Nous allons effectuer la simulation avec le point de départ $[0 \ 10 \ 0]$ et le résultat est présentés dans la figure suivante :

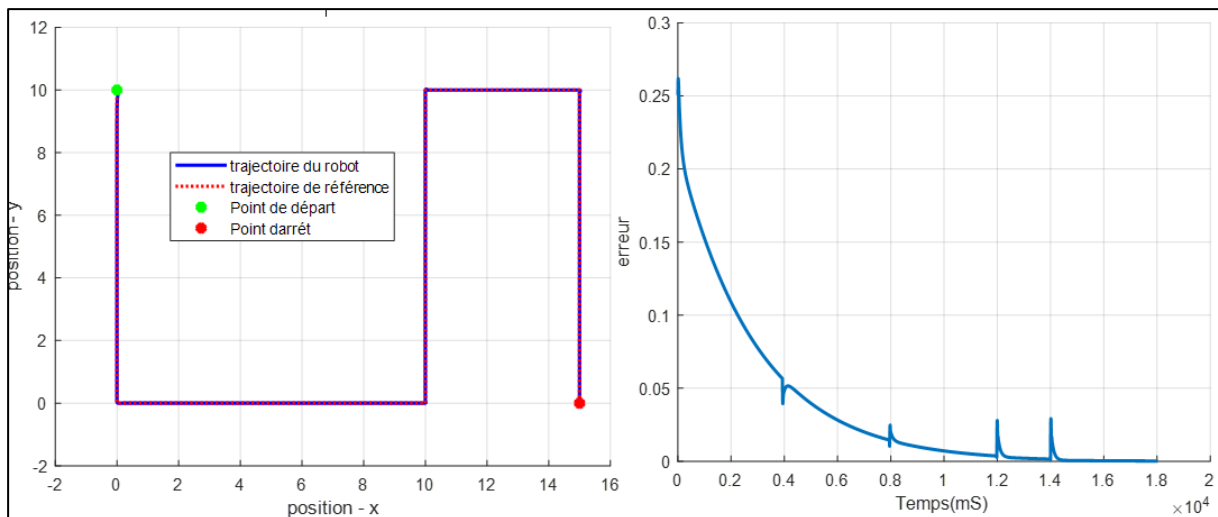


Figure IV.38: Trajectoire de référence Tr8 et son erreur.

On remarque que le contrôleur PID-floue a permis de contrôler le robot afin qu'il suive la trajectoire avec une précision très élevée, avec une erreur moyenne égale 0.0353. Ce résultat montre la précision du contrôleur PID-floue pour contrôler le robot.

IV.7. La représentation en 2D du robot et discussion générale

Nous avons représenté les meilleurs résultats obtenus pour les contrôleurs PID-PSO, floue et PID-floue et ils sont présentés ci-dessous :

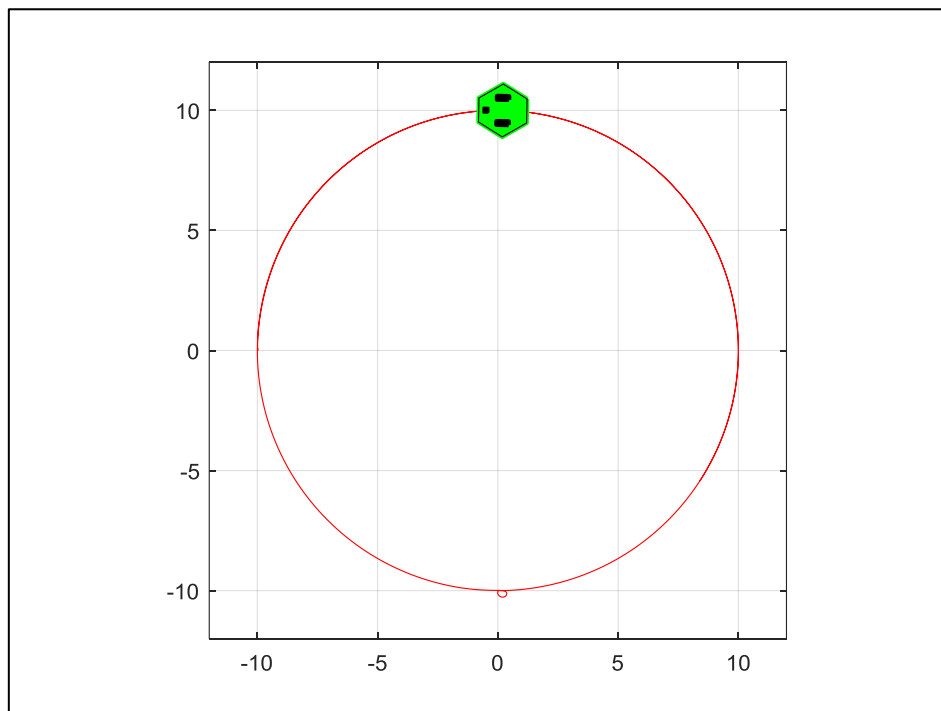


Figure IV.39: La représentation en 2D du robot (PSO).

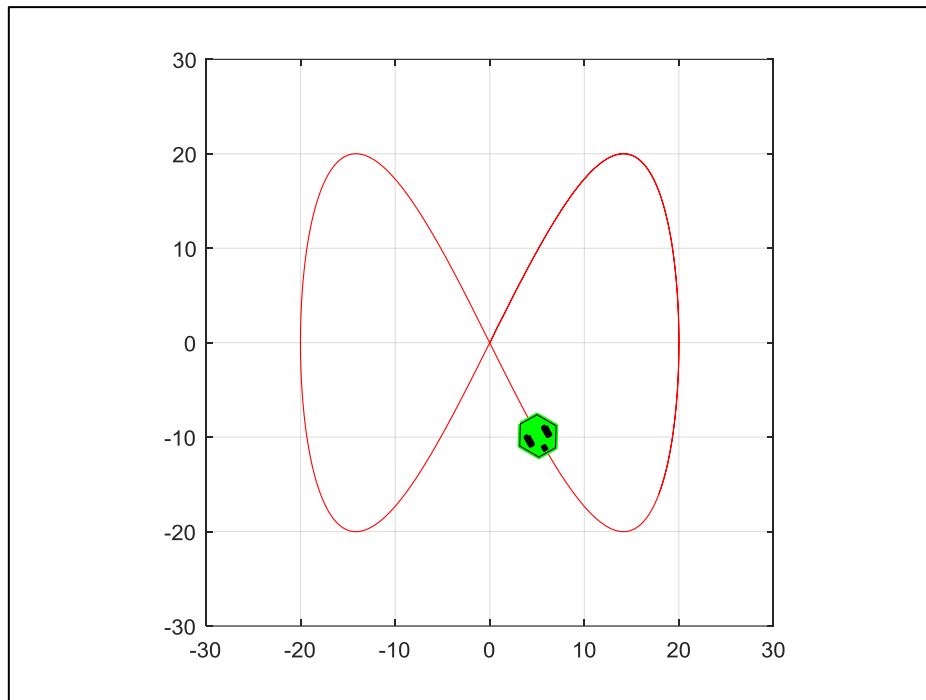


Figure IV.40: La représentation en 2D du robot (FL).

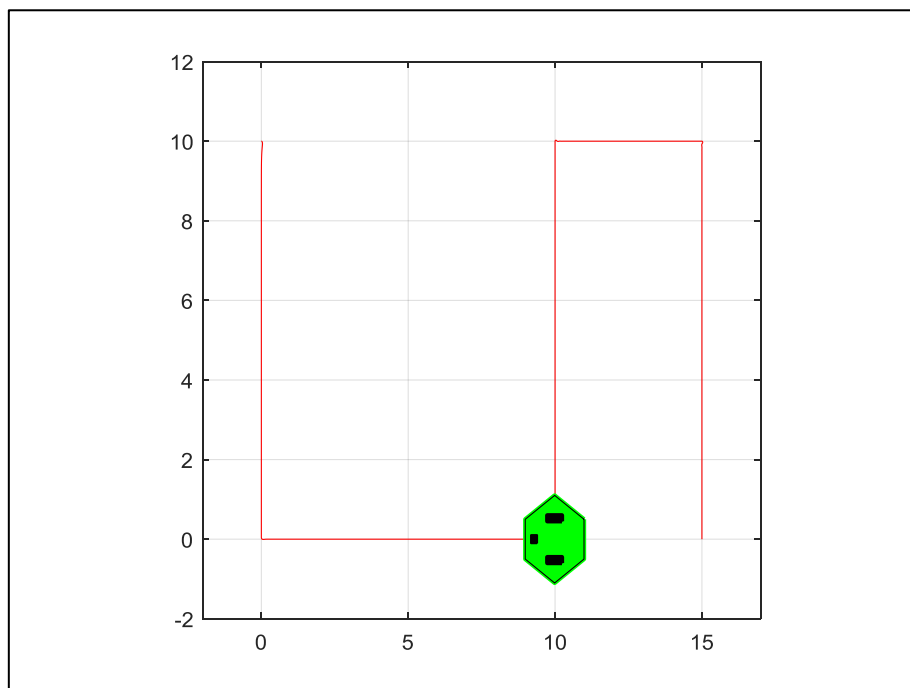


Figure IV.41 : La représentation en 2D du robot (PID-floue).

À partir des résultats précédents, il a été constaté que chaque contrôleur a la capacité de contrôler une trajectoire spécifique avec plus de précision que les autres. Le contrôleur flou excellait dans les trajectoires Tr6 (infini), tout en donnant de bons résultats dans le reste trajectoires. Alors que l'algorithme PSO excellait dans les trajectoires Tr2 (triangulaires),

Tr3(trapézoïdale), Tr4 (carré) et Tr5 (étoile) aux dépens du contrôleur flou, et donnait le meilleur résultat dans la trajectoire Tr1 (circulaire), Aussi, grâce à des expérimentations sur les trajectoires Tr1 et Tr7 utilisant l'algorithme GJO, il a été constaté que ce dernier fournit de meilleurs résultats que l'algorithme PSO sur la trajectoire Tr7 (infini). Quant au contrôleur flou, il a obtenu des résultats étonnants sur la trajectoire Tr8, ce qui indique que ce contrôleur est capable d'obtenir des résultats positifs sur les restes trajectoire.

IV.8. Conclusion

Dans ce chapitre nous avons simulé le contrôle du suivi de la trajectoire avec différentes architectures de contrôle basée sur un contrôleur flou, un contrôleur PID-flou, PSO-PID et un contrôleur GJO-PID. Différentes trajectoires de références ont été utilisées afin de valider l'architecture de contrôle.

Les résultats obtenus montrent clairement l'efficacité des différents contrôleurs et en particulier le contrôleur PSO-PID où l'erreur de suivi est nettement petit quel que soit la trajectoire de référence par rapport aux autres contrôleurs. Néanmoins il reste possible à améliorer les performances du contrôleur flou et PID-flou en choisissons bien les bornes des fonctions d'appartenances.

Conclusion général

Dans ce travail, nous avons développé un modèle cinématique d'un robot mobile type véhicule. Nous avons appliqué différentes architectures de contrôle basées sur les types de contrôleurs suivants :

- Floue
- PSO-PID
- GJO-PID
- PID-floue

Avec l'utilisation de différentes trajectoires proches de la réalité, les résultats obtenus sont très satisfaisants, surtout dans le cas où le contrôleur flou avec PID a donné des résultats satisfaisants sur toutes les trajectoires, en particulier le meilleur résultat dans la trajectoire Tr7. L'algorithme PSO avec un contrôleur PID a également donné des bons résultats sur les trajectoires Tr2, Tr3 et Tr5. Il convient également de mentionner que le contrôleur Floue-PID a fourni d'excellents résultats avec la trajectoire Tr8.

Grâce à un effort continu et intense, nous avons atteint notre objectif avec succès. Ce succès témoigne de l'engagement et de la persévérance que nous avons investi dans ce travail. Nous espérons sincèrement que ce travail modeste servira de référence et d'inspiration pour les futures promotions. Il offre une perspective concrète sur l'importance des méthodes de commande dans le domaine des robots mobiles, soulignant leur rôle vital dans la navigation précise et efficace de ces machines intelligentes. En réalisant ce travail, il pourrait être amélioré en utilisant d'autres trajectoires contenant des obstacles fixe ou mobiles.

Bibliographie

- [1] MOSTEFAOUI, Kadda. Apport de la simulation à événements discrets DEVS Dans la navigation réactive. 2012. Thèse de doctorat. Université Ibn Khaldoun-Tiaret-.
- [2] KHELFI, FAYCAL. Système de contrôle flou pour la navigation autonome d'un groupe de robots mobile Thymio II.
- [3] Stephen Bouquin « L'automatisation, une arme de destruction massive de L'emploi ». Marché des robots ménagers Insights. 2024.
- [4] FLEURY, Sara. Architecture de contrôle distribuée pour robot mobile autonome: principes, conception et applications. 1996. Thèse de doctorat. Université Paul Sabatier-Toulouse III.
- [5] hospitalière. DSIH. extrait de : <https://www.dsih.fr/article/5335/uniha-ouvre-l-acces-aux-robots-mobiles-autonomes-veritable-revolution-pour-la-logistique-hospitaliere.html>.
- [6] BC News Afrique. « Intelligence artificielle : les robots effectueront 39 % des tâches domestiques d'ici 2033, selon les experts. ». extrait de :
- [7] L.Fatima, « Evitement d'obstacle pour un robot mobile à 3 roues omnidirectionnelle ». 2019
- [8] BOUTARFIF, Darine. Réalisation et Etude de la maintenance de la partie puissance d'un Robot mobile minier. 2022. Thèse de doctorat
- [9] Santrot.F, Les 7 robots qui ont transformé l'histoire de la robotique. article en ligne de site fnac,
- [10] Laetitia Matignon, « Introduction à la robotique Licence 1`ere année, Diapositives de cours » Université de Caen, France,2012.
- [11] Ahcene Hamoudi, Karim Berkani, « Conception et réalisation d'un robot mobile autonome » Mémoire de Fin d'Etudes de Master Academique Université Mouloud MOULOUD de tizi-ouzou,2016.
- [12] cbenson, What Types of Mobile Robots, D'article en ligne de site robotshop. extrait de: <https://community.robotshop.com/tutorials/show/basics-what-types-of-mobile-robots-are-there>.

- [13] Chaher, BALI, « Réalisation d'un robot mobile avec évitement d'obstacle et trajectoire programmée. »
- [14] Sofiane MEZIOUD. Hamza Medani « Conception et réalisation d'un robot mobile à base de carte arduino » Mémoire de Fin d'Etude De Master Professionnel, Université Mouloud Mammeri De Tizi-Ouzo,2017.
- [15] Ahcene Hamoudi, Karim Berkani, « Conception et réalisation d'un robot mobile autonome » Mémoire de Fin d'Etudes de Master Academique Université Mouloud MOULOUD de tizi-ouzou,2016
- [16] C, J. Mon premier robot ! (2010) - PoBot. <https://pobot.org/Mon-premier-robot-2010.html>
- [17] RLX COMPONENTS « Electronic Components Distributor. (s. d.). Tricycle Bot (SE-110070028) easy-to-assemble, Grove compatible DIY Robot platform. ».
- [18] TP voiture – robot, « Elegoo Smart Robot Car - éducol STI. ».
- [19] Gadgetify, « Nexus Omni Wheel Robot Kit, », extrait de : <https://www.gadgetify.com/nexus-omni-wheel-robot-kit/>.
- [20] Roland Ilah, R. Siegwart Nourbakhsh « Introduction to Autonomous Mobile Robots » 2004.
- [21] Mahmoudi Ghilas. Amari Idir « Commande à distance d'un robot mobile à base d'une Raspberry Pi » Mémoire de Fin d'Etudes de Master Academique,Université Mouloud Mammeri De Tizi-Ouzou.2018.
- [22] SRAIRI, Fawzi. Platitude et planification de trajectoires en robotique. 2012. Thèse de doctorat. Université de Batna 2.
- [23] Benoît Thuilot, « Contribution à la modélisation et à la commande de robots mobiles à roues » L'École Des Minse De Paris,2013.
- [24] Bernard Byle, « Robotique mobile », University of Strasbourg, Article, 2016
- [25] Fabio morbridi, « Perception avancée et robotique mobile », UPJV, Département EEA.

- [26] robot non holonome.(1999). https://vitrinelinguistique.oqlf.gouv.qc.ca/fiche_gdt/fiche/8874238/robot-no_holonome#:~:text=Les%20contraintes%20non%20int%C3%A9grables%20sont,chemin%20qu'il%20peut%20suivre.
- [27] Lejeune, « Programme de commande d'un entraînement différentiel pour robots mobiles », © 2004 2024.
- [28] Julien Alexandre dit Sandretto, « Cinématique et commande » Department U2IS ENSTA ParisTech ROB316-2019-2020.
- [29] Saurabh Malhotra « Simulation of Steering Systems for Robotic Vehicles Florida State University Libraries », 2006
- [30] sami allou, youcef zennir,aissa belmeguenai, « Fuzzy logic controller for autonomous vehicle path tracking », 2017.
- [31] Mémoire fin d'étude master 2018/2019 commande floue des machines électriques Université 8 Mai 1945 Guelma.
- [32] Mémoire fin d'étude master 2016 Etude Comparative En Simulation Entre Un Régulateur PID Classique Et Un Régulateur PID Intelligent Université Mohamed Boudiaf Msila.
- [33] UFMC1/ISTA, «Auto-5 Asservissement et Régulation Chapitre 7 : Les correcteurs PID »,
- [34] Abdelhadi AOUAICHIA , Othmane BOOUHAKKOUFA « Synthèse sur l'avancement théorique et l'application des méthodes d'intelligence artificielle appliquées sur les régulateurs outomatiques », Ecole supérieure en sciences appliquées Tlemcen.
- [35] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," Chemometrics and Intelligent Laboratory Systems,
- [36] Parry, M. « Big Garden Birdwatch counts over 3m birds in UK parks and gardens ».
- [37] Y. Cooren, « Perfectionnement d'un algorithme adaptatif d'optimisation par essaim Particulare – applications en génie médical et en électronique » Thèse de Doctorat de L'Université Paris 12, Val de Marne, 2008.

- [38] R. Eberhart, Y. Shi, "Comparing inertial weights and constriction factor in particle swarm optimization," Proceeding of the International Congress on Evaluation of Computation, San Diego, California, IEEE Service Center, Piscataway.
- [39] M. Padma Lalitha, V. C. Veera Reddy, V. Usha, "Optimal DG placement for minimum real power loss in radial distribution systems using PSO," Journal of Theoretical and Applied Information Technology.
- [40] R Qi, B Hu, P.H Cournede « PSOTS: A Particle Swarm Optimization Toolbox in Scilab» 2009.
- [41] International Journal of Computational Intelligence Systems «An Improved Golden Jackal Optimization Algorithm Using Opposition- Based Learning for Global Optimization and Engineering Problems ».
- [42] H.Maghfiroh, F.Adriyano,A.Raelan « Fuzzy-PID in BLDC Motor Speed Control Using MATLAB/Simulink» Journal of Robotics and Control (JRC) Volume 3 Issue, 1, January 2022.