

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ MOULOUD MAMMERI DE TIZI OUZOU

FACULTÉ DE GÉNIE ÉLECTRIQUE ET D'INFORMATIQUE

DÉPARTEMENT D'ÉLECTROTECHNIQUE



THÈSE DE DOCTORAT LMD EN ÉLECTROTECHNIQUE

Spécialité : Modélisation et conception des systèmes électromagnétiques

Présentée par

Mme MOHAND SAIDI Sabrina

Thème :

**Asservissement visuel coordonné d'un robot
mobile**

Devant le jury d'examen composé de :

P ^r . H. MOHELLEBI	Professeur	UMMTO	Président
P ^r . R. MELLAH	Professeur	UMMTO	Rapporteur
P ^r . M. DAOUI	Professeur	UMMTO	Examineur
P ^r . M. LAHDIR	Professeur	UMMTO	Examineur
P ^r . A. RAHOUI	M.C.A	ENSTP	Examineur

Soutenu le 03 juillet 2023

Remerciements

Ce travail n'aurait jamais abouti sans le soutien indéfectible de mon directeur de thèse, M^r. Rabah MELLAH, Professeur au Département d'Automatique de l'UMMTO. Je lui exprime mes profonds remerciements pour son soutien et son encadrement durant ces dures années de thèse. Je tiens également à le remercier pour ses remarques, ses orientations et ses conseils inestimables .

Mes sincères remerciements vont également à M^r. hassene MOHELLEBI , Professeur au département d' Électrotechnique de l'UMMTO pour avoir accepté de présider le jury de soutenance.

Mes vifs remerciements vont également à M^r. Mehammed DAOUI ,Professeur au département d'Informatique de l'UMMTO, à M^r. Mourad LAHDIR , Professeur au département d'Electronique de l'UMMTO et à M^r. Adel RAHOUI, M.C.A à l'ENSTP d'avoir accepté de participer dans le jury d'examen.

Je remercie également toute l'équipe du laboratoire de recherche L2CSP pour avoir mis à ma disposition, les moyens du laboratoire pour réaliser la partie expérimentale de mon travail de thèse.

Je tiens à remercier particulièrement M^r. Arezki FEKKIK , M.C.A à University Akli Mohand Oulhadj de Bouria, pour ses conseils. Ses remarques, ses orientations ont beaucoup aidé à faire avancer mon travail de recherche.

*Je dédie ce travail à ma chère mère
à mon époux
à ma fille
à toutes les personnes qui me sont chères.*

Résumé

Ce travail propose deux méthodes pour une navigation efficace et sûre d'un robot mobile à roues non holonome dans un environnement d'intérieur. La première méthode utilise un contrôleur PID flou pour permettre au robot de naviguer à la vitesse désirée, et la seconde méthode propose un système d'évitement d'obstacles basé sur la vision pour naviguer en toute sécurité au-delà des obstacles. Dans ce travail, un robot mobile à roues à entraînement différentiel est contrôlé en vitesse à l'aide d'un contrôleur PID flou, qui combine un contrôleur flou avec un PID conventionnel dans le but de réajuster les gains de ce dernier en temps réel. L'environnement peut contenir une gamme d'obstacles de forme et de taille arbitraires. Une caméra monoculaire est utilisée pour détecter les obstacles pendant le processus de navigation. Pour permettre à un robot de naviguer dans un espace d'intérieur tout en évitant les obstacles, une approche de traitement d'image basée sur la méthode de traitement par zone d'intérêt a été utilisée pour la détection d'obstacles et une méthode inspirée des algorithmes des bugs a été utilisée pour l'évitement d'obstacles. L'objectif de cette recherche est de développer et d'implémenter un régulateur de vitesse PID flou sur un robot réel, ainsi qu'un système visuel simple et efficace d'évitement d'obstacles. Le logiciel MATLAB est utilisé pour implémenter le système de contrôle. GUIDE (Graphical User Interface Development Environment) a permis la création d'interfaces utilisateur graphiques. Ces interfaces facilitent la manipulation et la visualisation du système en temps réel. Pour valider les approches proposées, des expériences ont été réalisées avec le robot mobile non holonome dr Robot i90.

Mots clés : Robots mobiles, Asservissement visuel coordonné, Commande en vitesse par PID floue, Évitement d'obstacles visuel, Traitement d'image par zone d'intérêt

Abstract

This work proposes methods for the efficient and safe navigation of a non-holonomic wheeled mobile robot in an indoor environment. It is essential for a mobile robot to navigate at a precise and well-defined speed depending on the activity and the environment. The first solution uses a fuzzy PID controller to allow the robot to navigate at the desired speed, and the second offers a vision-based obstacle avoidance system to safely navigate past obstacles. In this work, a differential wheeled mobile robot is controlled using a fuzzy PID controller, which combines fuzzy controller with a conventional PID in order to readjust the gains of the controller in real time. The environment can contain a range of obstacles of arbitrary shape and size. A monocular camera is used to detect obstacles during the navigation process. To allow a robot to move in an indoor space while avoiding obstacles, an image processing approach based on the area of interest processing method was used for obstacle detection and a method inspired by bug algorithms was used for obstacle avoidance. The objective of this research is to develop and implement a fuzzy PID speed controller on a real robot, as well as a simple and effective visual obstacle avoidance system. MATLAB is used to implement the control system. GUIDE (Graphical User Interface Development Environment) enabled the creation of graphical user interfaces. These interfaces facilitate manipulation and visualization of the system in real time. To validate the proposed approaches, experiments were carried out in an internal environment with the non-holonomic mobile robot dr Robot i90.

Keywords : Differential mobile robots , Visual obstacle avoidance , Fuzzy PID speed controller, Image processing approach based on the area of interest .

Table des matières

Résumé	i
Abstract	i
Table des matières	ii
Liste des figures	ix
Liste des tables	x
Liste des acronymes	xi
Introduction générale et état de l’art	1
Introduction	1
État de l’art	2
Structure de la thèse	5
1 Généralités sur la robotique mobile	7
Introduction	7
1.1 Robots et éthique	9
1.2 Classification des robots	10
1.2.1 Les robots de fabrication	11
1.2.2 Les robots de service	11
1.2.3 Les robots de terrain	11
1.2.4 Les robots humanoïdes	11
1.3 Robot mobile	12
1.3.1 Définition d’un robot mobile	12
1.3.1.1 Robots mobiles à pattes	13

1.3.1.2	Robots mobiles à roues	13
1.3.2	Les types de roues utilisées en robotique mobile :	14
1.3.2.1	Roues conventionnelles :	14
1.3.2.2	Roues spéciales :	15
1.3.3	Les classes des robots mobiles à roues	15
1.3.3.1	Classification selon le niveau d'autonomie	15
1.3.3.2	Classification selon le domaine d'application :	16
1.3.3.3	Classification basée sur le mode de locomotion	17
1.3.4	La manœuvrabilité d'un robot mobile à roues	20
1.4	La perception en robotique mobile	21
1.4.1	Caractéristiques des capteurs utilisés en robotique	23
1.4.2	Les différents capteurs utilisés en robotique	24
1.4.2.1	Capteurs de position et de vitesse :	24
1.4.2.2	Capteurs de distance	26
1.4.2.3	Quelques autres capteurs utilisés en robotiques	27
1.5	La perception visuelle	33
1.5.1	Classes des systèmes de vision	34
1.5.1.1	La stéréovision	34
1.5.1.2	La vision monoculaire	34
1.5.2	Technologies des caméras	34
1.5.2.1	Capteurs CCD	34
1.5.2.2	Capteurs CMOS	35
1.5.2.3	Autres capteurs particulier	35
2	Modélisation cinématique et dynamique d'un robot mobile	37
	Introduction	37
2.1	Modélisation cinématique d'un robot mobile	38
2.1.1	Robot à deux roues à entraînement différentiel	38
2.1.2	Définition des repères	38
2.1.3	Les matrices de passage homogènes ,le vecteur d'état et le vecteur de commande	39
2.1.4	Le torseur cinématique de la caméra	41
2.1.5	Modèle cinématique d'un robot mobile	41
2.1.6	Modélisation de la caméra : le modèle sténopé (pinhole)	42

2.2	Modélisation dynamique du robot mobile	45
2.2.1	Modèle dynamique de Newton-Euler	46
2.2.2	Modèle dynamique de Lagrange	48
2.2.3	Modélisation dynamique de robots non holonomes	48
2.2.4	Le modèle dynamique d'un robot mobile à roue à entraînement dif- férentiel	49
2.2.4.1	Modèle dynamique d'un robot mobile à entraînement dif- férentiel par Newton Euler	50
2.2.4.2	Modèle dynamique d'un robot mobile à entraînement dif- férentiel par la méthode de Lagrange	50
2.3	L'holonomie :	52
2.3.1	Contraintes non holonomes	53
2.3.2	Définition de la contrainte non holonome	53
2.3.3	Définition de la contrainte Pfaffian	53
3	Commande par vision d'un robot mobile à roues	56
	Introduction	56
3.1	Les sous domaine de la vision en robotique	57
3.1.1	La détection visuelle	57
3.1.1.1	Étalonnage de la caméra	58
3.1.1.2	Acquisition d'image	58
3.1.1.3	L'illumination	59
3.1.1.4	Géométrie de l'image	60
3.1.2	Prétraitement	60
3.1.3	Segmentation d'images	61
3.1.4	Détection des contours	61
3.1.5	Description d'image	62
3.1.6	Reconnaissance d'images	62
3.1.7	Interprétation	63
3.2	Description de l'asservissement visuel d'un robot mobile	63
3.2.1	Définition de l'asservissement visuel	64
3.2.2	Historique sur les asservissements visuels	64
3.2.3	Classification des asservissements visuels	66
3.2.3.1	Classification en fonction de la position de la caméra	66

3.2.3.2	Classification en fonction du type de la commande	66
3.2.3.3	Classification en fonction du type de la mesure :	67
3.2.3.4	Classification en fonction De la rapidité	69
3.3	Planification de trajectoire d'un robot mobile avec évitement d'obstacles visuel	70
3.3.1	Les méthodes de planification de trajectoire globales	71
3.3.1.1	Méthode Roadmap	71
3.3.1.2	Méthode du champ potentiel	72
3.3.1.3	Méthode de décomposition cellulaire	73
3.3.2	Les méthodes de planification de trajectoire locales	74
3.3.2.1	Algorithmes des bugs	74
3.3.2.2	Histogramme de champ vectoriel	76
3.3.3	Les approches heuristiques de planification de trajectoire	77
3.3.3.1	Les réseaux neurones artificiels	77
3.3.3.2	La logique floue	78
3.3.3.3	Les algorithmes génétiques	78
3.3.3.4	La PSO	79
3.4	Commande par vision d'un robot mobile à roues	80
3.4.1	Les différents types de commande visuelle	80
3.4.1.1	Commande séquentielle	80
3.4.1.2	Commande cinématique	81
3.4.1.3	Commande dynamique	81
4	Résultats expérimentaux	83
	Introduction	83
4.1	Description et modélisation de la plate-forme utilisée dans la partie expérimentale	84
4.1.1	Description de dr robot i90	84
4.1.1.1	Capteurs et composants externes	84
4.1.1.2	Fonctionnement et connexion du système dr robot i90	85
4.1.1.3	Présentation du SDK WiRobot	86
4.1.2	L'interface de commande	87
4.1.3	La description d'un robot mobile de type uni-cycle	88
4.1.4	Modèle cinématique	90
4.2	Description des approches de contrôles proposés	91

4.2.1	Commande en vitesse d'un robot mobile par un régulateur PID . . .	91
4.2.2	Commande en vitesse d'un robot mobile par un régulateur PID flou	93
4.2.2.1	Suivi de trajectoire par robot mobile	96
4.2.3	Évitement d'obstacles par vision	98
4.2.3.1	Retour visuel	99
4.2.3.2	Détection d'obstacles	101
4.2.3.3	Processus d'évitement d'obstacles	101
4.3	Résultats expérimentaux	102
4.3.1	Première expérience	103
4.3.2	Deuxième expérience	107
4.3.3	Troisième expérience	108
	Conclusion générale et perspectives	114
	Bibliographie	116

Table des figures

1.1	Les différents mouvements possibles d'un robot mobile différentiel A : mouvement en ligne droite, B et C : mouvement circulaire.	18
1.2	Calcul du rayon de rotation instantanée d'un robot mobile différentiel.	18
1.3	Les différentes trajectoires possibles effectuées par un robot mobile tricycle.	19
1.4	Robot mobile omnidirectionnel. (A) cas à trois roues, (B) cas à quatre roues.	19
1.5	Les six modes de locomotion de base d'un WMR omnidirectionnel avec quatre roues Mecanum.	20
1.6	Tachymètre à courant continu [1]	25
1.7	Télémètre laser[1].	28
1.8	Schéma illustratif d'un gyroscope [2]	29
1.9	Capteur tactile monté sur un bras manipulateur [1]	32
2.1	Schéma représentant les différents repères associés à un robot mobile et à la caméra	39
2.2	Schématisation du modèle sténopé	43
2.3	Schémas explicatifs pour l'obtention du modèle sténopé	43
2.4	Projection d'un point 3D dans le plan image	44
2.5	Un corps solide B_i dans le repère des coordonnées inertielles $Oxyz$	47
3.1	schéma de la boucle d'asservissement visuel indirect.	66
3.2	Schéma de la boucle d'asservissement visuel direct	67
3.3	Schéma de la boucle de commande d'un asservissement 3D	68
3.4	schéma de la boucle de commande d'un asservissement 2D	68
3.5	Schéma de l'asservissement visuel 2D $\frac{1}{2}$	69
3.6	Les graphes de visibilité[3] et le diagramme de Voronoi[4]	73
3.7	Représentation du champs potentiel avec obstacle et objectif à atteindre par le robot[5]	74

3.8	Méthode de décomposition cellulaire[6].	75
3.9	Les méthodes de planification de trajectoire par les Algorithmes des bugs [7]	76
4.1	Photographie de dr robot i90	84
4.2	L'architecture de communication du système dr Robot i90.	85
4.3	Capteurs et composants externes de dr robot i90	86
4.4	Connexion sans fil du système dr robot i90	87
4.5	Architecture logicielle de WiRobot	88
4.6	L'environnement de création d'interface graphique de MATLAB	89
4.7	Robot mobile uni-cycle	89
4.8	Schéma de contrôle par PID	92
4.9	Organigramme représentant le Processus de conception du régulateur PID	93
4.10	Schéma du système de control par PID flou	95
4.11	Les fonctions d'appartenance de l'erreur de vitesse e et de la variation de l'erreur de vitesse Δe	95
4.12	Les fonctions d'appartenance des paramètres du régulateur PID k_p, k_i et k_d .	96
4.13	Systèmes de coordonnées du robot mobile et de la caméra monoculaire . .	100
4.14	Représentation de la zone d'intérêt dans l'image	100
4.15	Illustration de la stratégie d'évitement d'obstacles	102
4.16	Interface utilisateur de commande par PID	103
4.17	Contrôle de vitesse avec $K_p=0, K_d=0$ and $K_i=0$ for $v_{ref}=100$ impulsions/s	104
4.18	Contrôle de vitesse avec $K_p=0, K_d=0$ and $K_i=0$ for $v_{ref}=500$ impulsions/s.	104
4.19	Contrôle de vitesse avec $K_p=5, K_d=2$ and $K_i=10$ for $v_{ref}=100$ impulsions/s.	105
4.20	Contrôle de vitesse avec $K_p=5, K_d=2$ and $K_i=10$ for $v_{ref}=500$ impulsions/s.	106
4.21	Contrôle de vitesse avec $K_p=50, K_d=10$ and $K_i=100$ for $v_{ref}=100$ impul- sions/s.	106
4.22	Interface utilisateur créée pour le contrôle avec un PID flou.	108
4.23	La vitesse du robot en temps réel utilisant le contrôle PID flou pour la valeur désirée $v_{ref}=200$ impulsions/sec	109
4.24	La vitesse du robot en temps réel utilisant le contrôle PID flou pour la valeur désirée $v_{ref}=500$ impulsions/s.	110
4.25	Suivi de trajectoire du robot dr i90 à l'aide du contrôle PID (a). Suivi de trajectoire de dr robot i90 à l'aide d'un contrôle PID flou (b).	110
4.26	Interface utilisateur créée pour la navigation du robot i90 avec évitement d'obstacles	111

4.27	Les étapes de traitement d'image utilisée pour la détection d'obstacle . . .	111
4.28	Identification du centre de gravité de l'obstacle détecté et de sa circonférence	112
4.29	Série de photographies montrant le dr robot i90 évitant un obstacle	113

Liste des tableaux

4.1	Table des règles floues utilisées dans le calcul de $k_p / k_i / k_d$	96
-----	---	----

Liste des acronymes

1D	one-dimensional (à une dimension)
2D	two-dimensional (à deux dimensions)
3D	three-dimensional (à deux dimensions)
AC	Alternative Current
AGV	Automatic guided vehicle
ANFIS	Adaptive neuro fuzzy inference system
ANN	Artificial neural network
API	application programming interface
ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
B	Big
CMOS	Complementary Metal Oxide Semi-conductor
CCD	Charge Coupled Device
COG	Centre Of Gravity
CPU	Central Processing Unit
DC	Direct Current
DDL	Degré De Liberté
DDOF	Differential degrees of freedom
DOF	degrees of freedom
DSP	Digital Signal Processor
FL	Fuzzy logic
FOV	Field Of View
GA	Genetic Algorithm
GPS	Global Positioning System
GUIDE	Graphical User Interface Development Environment
ICC	Instantaneous Center of Curvature
IRM	imagerie par résonance magnétique
LAN	Local Area Network
LED	Light-Emitting Diode

LIDAR	Light Direction And Ranging Sensors
LMR	legged Mobile Robot
M	Medium
Mbps	Megabits per second
NB	Negative Big
NM	Negative Medium
NS	Negative Small
PB	Positive Big
PC	Personal Computer
PD	Proportionnel-Dérivé
PID	Proportionnel-intégral-dérivé
PM	Positive Medium
PS	Positive Small
PSO	Particle swarm optimisation
PVDF	Fluorure de polyvinylidene
PWM	Pulse Width Modulation
RAM	Random Access Memory (mémoire vive)
rpm	revolutions per minute
S	Small
SDK	Software Development Kit
SONAR	Sound Navigation And Ranging
UAV	Unmanned Aerial Vehicle (drone)
VFH	Vector Field Histogram
WMR	Wheeled Mobile Robot
Z	Zero

Introduction générale et état de l'art

Introduction

La robotique mobile représente un champ d'application intéressant pour des disciplines aussi diverses que l'intelligence artificielle, l'informatique, l'automatique, l'électronique, l'instrumentation et la mécanique. Ceci conduit à l'évolution des robots mobiles qui se traduit par le passage du stade de robot mobile asservissement au stade de navigation pour accomplir diverses tâches dans un environnement mal connu et parsemé d'obstacles grâce à des moyens de perception visuel et des algorithmes performants utilisant l'intelligence artificielle. Malgré le fait que les premiers tests avec des robots mobiles remontent à la fin des années 1960, le sujet n'a attiré l'attention qu'au cours les années 1990. Une quantité importante de travaux de recherche a été publiée. Les robots mobiles ont fait leurs preuves sur terre, dans les airs, et en mer. Ils sont maintenant utilisés dans les industries, la sécurité, les premiers secours, l'assistance personnelle et l'exploration [8, 9, 10, 11, 12, 13, 14]. La robotique mobile est clairement à la pointe de l'innovation technologique, avec des robots compagnons, des robots d'assistance de malades ou encore des systèmes de transport robotisés. La majorité des recherches sont consacrées à la conception et au développement de mouvements de robots [15, 16], la planification de trajectoires[17, 18], la construction de cartes[19], l'évitement d'obstacles [20], le suivi d'objets [21] et le contrôle de vitesse [22, 23]. Beaucoup de recherches se sont concentrées sur l'utilisation de modèles dynamiques pour construire et mettre en œuvre le contrôle de robots mobiles [23, 24, 25]. D'autres n'ont pas vu la nécessité d'utiliser le modèle dynamique ou n'ont pas accès à des informations spécifiques essentielles à son utilisation, d'où le modèle cinématique est largement utilisé [15, 26, 27]. Le robot mobile différentiel devient de plus en plus utile et largement utilisé

dans la vie quotidienne. En raison de sa facilité de construction et de ses simples caractéristiques cinématiques, ce type de robot est très populaire. En conséquence, il y a eu un intérêt pour la recherche dans le domaine des robots mobiles différentiels [15, 28]. La translation latérale étant impossible, les robots mobiles non holonomes n'ont que deux degrés de liberté sur un plan[15, 18].

La perception en robotique est très importante est indispensable. L'utilisation de capteurs visuels permet la perception de l'environnement sans contact physique. L'asservissement visuel est une méthode de commande qui utilise un ou plusieurs capteurs de vision (caméras) pour contrôler le mouvement d'un robot. Plus précisément, les données de commande des moteurs du robot sont produites en traitant les données extraites de l'image. En robotique mobile, la tâche du contrôleur de vision est de contrôler la pose du robot par rapport à un certain nombre de cibles ou certains points de repère. La stabilité du suivi peut être assurée seulement si les délais de détection sont suffisamment petits et/ou si le modèle dynamique du robot est suffisamment précis. Au fil des années, de nombreuses techniques de commande par vision ont été développées. Une riche littérature a été orientée vers la commande de systèmes non holonomes afin de traiter divers problèmes difficiles associés à la commande visuelle basée sur la position et l'image.

État de l'art

Le contrôle avec PID est une technique utilisée en ingénierie pour réduire la variabilité des processus en modifiant certaines variables quantifiables du système avec rétroaction et compensation. Depuis sa conception en 1910 et la présentation des règles de ses paramètres par Ziegler-Nichols en 1942, le contrôle PID a considérablement gagné en popularité. Le contrôleur PID est encore utilisé dans presque tous les domaines. Beaucoup de temps et d'efforts ont été consacrés à la détermination des meilleurs paramètres du régulateur PID pour divers modèles de processus. Les auteurs de [29, 30] ont mené une étude comparative des approches de réglage du contrôleur PID. Le réglage idéal des paramètres du PID reste une tâche difficile. La majorité des recherches portent sur l'utilisation de méthodes intelligentes pour ajuster les paramètres de ce contrôleur, comme les réseaux de neurones

[31, 32] et la logique floue [33] ou les deux techniques en même temps[34]. Le contrôleur PID flou combine la facilité d'utilisation du contrôleur PID conventionnel avec les avantages du contrôleur flou à travers sa robustesse et sa stratégie voisine de celle du raisonnement humain. Il est utilisé dans une variété de domaines, par exemple la détection des résidus de pesticides [35] et le contrôle de système pour la purification de l'eau [36]. Il est aussi fréquemment utilisé pour piloter des robots mobiles différentiels, notamment en contrôle de vitesse. Un contrôleur de vitesse PID flou à double boucle a été proposé par [37]. Les contrôleurs de vitesse ANFIS, flous et PID pour les robots mobiles à roues sont développés et comparés dans [38]. [39] présente un contrôleur hybride PID flou basé sur le modèle dynamique pour atteindre la position cible en tenant compte du suivi de trajectoire. La référence[40] présente deux techniques de contrôle de robots mobiles non-holonomes. La première est basée sur un contrôleur PD flou, et la seconde est basée sur un contrôleur PID intelligent dont les paramètres sont optimisés par un réseau de neurones. Dans [41], il est suggéré qu'un robot à roues non holonome suit un chemin donné en utilisant une combinaison de technique de retour arrière (back stepping) et une technique PID floue adaptative. Les auteurs de [42] ont présenté un contrôleur PID flou d'ordre fractionnel pour le suivi de trajectoire. Ammar et al. ont proposé dans [13] un PID d'ordre fractionnaire pour contrôler le suivi de trajectoire du robot mobile Pioneer-3. [43] ont discuté de l'efficacité du contrôle PID et du contrôle flou dans le suivi de mur pour les robots mobiles à entraînement différentiel. Dans [44] ils ont proposé un contrôleur flou PD+I optimal pour la trajectoire du robot uni-cycle et pour optimiser et obtenir les valeurs idéales des paramètres du contrôleur, l'approche PSO (Particle Swarm Optimization) est utilisée. Tous les travaux précédents sur le contrôle des robots mobiles à roues ont été validés en simulation, mais le comportement d'un robot mobile dans un environnement réel n'est pas toujours le même qu'en simulation, et il existe peu d'études validées avec des données expérimentales. Les auteurs de [45] ont présenté une méthode d'autoréglage basée sur la méthodologie de colonies de fourmis et le contrôle flou pour le changement des paramètres du contrôleur en temps réel pour les robots mobiles. Dans [46] a été développé un contrôleur de vitesse à base de la logique floue pour un robot mobile à roues qui peut suivre des points prédéterminés

et réagir rapidement pour atteindre une vitesse prédéfinie. L'entrée du système est l'erreur de vitesse et la variation de l'erreur de vitesse du robot, sa sortie est la modulation de largeur d'impulsion (PWM) de 8 bits traduite en formes floues. Des recherches ultérieures se sont concentrées sur des robots dotés de systèmes de détection supplémentaires pour construire des systèmes de guidage autonomes afin de manœuvrer un robot mobile à roues en toute sécurité dans son environnement [47]. Guzel et Bicker [48] ont décrit des méthodes d'évitement d'obstacles basées sur la vision. La détection d'objets est un élément essentiel pour l'évitement d'obstacles [49]. Néanmoins la difficulté de calcul des algorithmes d'évitement et le coût des capteurs sont les facteurs les plus pertinents pour les applications en temps réel. Les télémètres lasers, les capteurs à ultrasons et des systèmes de stéréovision sont souvent utilisés pour identifier les obstacles. En revanche, tous ces capteurs présentent des inconvénients. Les capteurs à ultrasons ont un problème de résolution angulaire. Le prix d'un télémètre laser et un système de stéréovision est assez important. De nombreux robots mobiles utilisent des caméras car ce sont des capteurs qui fournissent des informations supplémentaires sur leur environnement. La méthode de contrôle basée sur les informations fournies par une ou plusieurs caméras est appelée asservissement visuel. Dans [50] Pasteau et al. ont réalisé un contrôle de fauteuil roulant électrique basé sur la vision. Tsalatsanis et al. [51] ont présenté la stéréovision pour le suivi d'objets en temps réel et l'évitement des collisions. Le guidage par stéréovision pour robot est également utilisé par [52]. Benn et Lauria [53] ont proposé un système à vision monoculaire pour diriger un robot mobile dans un environnement dynamique changeant. La vision est utilisée pour suivre une cible en mouvement par [54]. Nadour et al. [55] ont proposé un système visuel de navigation et d'évitement d'obstacles d'un robot mobile. Par ailleurs il existe deux techniques d'évitement d'obstacles basées sur la vision. La première est basée sur le principe du flux optique [55, 56], qui n'est pas très efficace et souffre lors des changements de la lumière et des discontinuités de mouvement créées par les objets en mouvement. La deuxième consiste à identifier les changements de l'apparence des pixels dans l'image [57].

Notre travail a pour objectif principal de réaliser en temps réel la navigation visuelle autonome d'un robot mobile dans un environnement statique d'intérieur, en utilisant des

informations visuelle pour assurer une navigation sans collision par une stratégie de contrôle de vitesse qui exploite un contrôleur PID flou auto ajustable adapté au contrôle dynamique du robot mobile. Dans ce travail, nous avons essayé d'améliorer les performances de vitesse du robot i90 par un régulateur PID. Le PID parfait n'existe pas. Les résultats obtenus de l'expérience ont montré l'efficacité de cette stratégie de contrôle[58]. Mais l'application d'un régulateur flou pour le choix de ces paramètres peut être une meilleure solution et nous avons développé et implémenté un contrôleur de vitesse qui utilise l'auto-ajustement en temps réel des paramètres du PID via un régulateur flou à deux entrées; l'erreur de vitesse et la variation de l'erreur de vitesse, et trois sorties : k_p , k_i et k_d qui représentent les paramètres du régulateur PID. Les paramètres du régulateur changeront à chaque pas d'échantillonnage afin de s'adapter au changement de vitesse détecté par l'encodeur quadratique. Dans cette thèse nous avons proposé également une nouvelle technique de navigation avec évitement d'obstacles visuel d'un robot mobile, se déplaçant dans un espace d'intérieur. La simplicité de cette technique est ce qui la rend meilleure . Cette approche est basée sur le traitement d'image et détecte les obstacles en utilisant la méthode de soustraction de fond. Lorsqu'un obstacle est repéré, le robot mobile s'arrête pour extraire de l'image les informations nécessaires au calcul du mouvement d'évitement d'obstacle puis l'évitement se fait grâce à une méthode inspirée des algorithmes des Bugs. Le principal avantage de la méthode proposée est qu'elle ne nécessite pas de déterminer la distance du robot à l'obstacle. Les deux principaux avantages de cette stratégie sont sa simplicité et sa facilité de la mettre en œuvre en temps réel,les résultats obtenus ont fait l'objet d'une publication [59].

Structure de la thèse

La robotique mobile est une science qui vise à donner à une machine la capacité de se déplacer dans un environnement sans assistance ni intervention humaine. L'utilisation de robots autonomes ou partiellement autonomes devient de plus en plus indispensable pour la vie humaine. Ainsi nous avons organisé notre thèse comme suit : Le chapitre 1 donne les généralités sur les robots et en particulier les robots mobiles, la perception en

robotique mobile ainsi que la perception visuelle avec plus de détails. Le chapitre 2 présente les algorithmes de calcul pour l'obtention des modèles cinématique et dynamique du robot mobile à deux roues motrices à entraînement différentiel, ainsi que le modèle de la caméra et les différents outils permettant le passage du repère du monde réel vers l'image à savoir les matrices de rotation et le torseur cinématique de la caméra. La fin de ce chapitre est consacrée à des notions générales sur l'holonomie. Dans le chapitre 3, l'asservissement visuel d'un robot mobile est entrepris, il est composé essentiellement de trois parties. La première partie fait l'objet d'une description de l'asservissement visuel d'un robot mobile ainsi que sa classification selon différents critères. La deuxième partie aborde les différentes méthodes de planification de trajectoire avec évitement d'obstacles par vision. La troisième partie est consacrée aux stratégies de commande par vision appropriées aux robots mobiles à savoir la commande séquentielle, la commande cinématique et la commande dynamique. Le chapitre 4 présente l'implémentation en temps réel des stratégies de commande proposées. Après une description succincte du mode de fonctionnement du robot mobile dr robot i90 que nous avons utilisé pour valider les stratégies de commande proposées. Les résultats expérimentaux obtenus sont présentés. Nous terminons notre thèse par une conclusion générale et quelques perspectives.

Chapitre 1

Généralités sur la robotique mobile

Introduction

Le terme robot est apparu pour la première fois dans une pièce de science-fiction tchèque en 1920 "Rossum's Universal Robot " de Karel Čapek. Le terme en tchèque signifie le travail dur ou la corvée. Dans la pièce, les robots sont présentés comme des serviteurs soumis et compétents qui se rebellent contre leurs créateurs. En science et technologie il n'y a pas de définition globale et unique d'un robot. Lorsque Joseph Engelberger, le père de la robotique moderne, a été invité à définir un robot, il a dit : «Je ne peux pas définir un robot, mais j'en connais un quand je l'en vois » Ronald Arkin dit : « Un robot intelligent est une machine capable d'extraire des informations de son environnement et de les utiliser pour se déplacer en toute sécurité dans un sens et de manière intentionnelle »[1]. Rodney Brooks dit : « Pour moi, un robot est quelque chose qui a un effet physique sur le monde, mais il le fait en fonction de la façon dont il perçoit le monde et de la façon dont le monde change autour de lui ». La définition retenue dans la norme européenne EN775/1992 est la suivante « Le robot industriel manipulateur est un robot programmable à commande automatique. Machine de manipulation polyvalente à plusieurs degrés de liberté, qui peut être fixe ou mobile pour une utilisation dans des applications d'automatisation industrielle » [1]. En résumé, un robot est désigné dans la littérature comme une machine qui effectue un lien intelligent entre la perception et l'action. La conception de robots mobiles intelligents utilise les méthodologies et technologies de contrôle intelligent, cognitif et comportemental[1]. La série

de robots d'Isaac Asimov, comprenant de nombreux livres et nouvelles écrites entre 1950 et 1985, a exploré les questions d'interaction et de moralité entre les humains et les robots. Le milieu du vingtième siècle a également vu l'avènement du domaine de la cybernétique - un terme peu commun aujourd'hui mais alors une science passionnante aux frontières de la compréhension de la vie et de la création de machines intelligentes. Le premier brevet pour ce que nous considérerions aujourd'hui comme un robot a été déposé en 1954 par George C. Devol et publié en 1961. L'appareil comprenait un bras mécanique avec une pince et la séquence de mouvements était codée sous forme de motifs magnétiques stockés sur un tambour rotatif. La première entreprise de robotique était "Unimation", fondée par Devol et Joseph Engelberger en 1956. Plusieurs millions de robots de type bras manipulateurs ont été construits pour accomplir des tâches tels que le soudage, la peinture, l'assemblage électronique, le conditionnement et la palettisation. Cela a permis d'augmentation de la productivité et l'amélioration de la qualité du produit. Aujourd'hui, de nombreux produits que nous achetons ont été assemblés ou manipulés par un robot. Ces robots de première génération sont fixes. En revanche, les robots mobiles peuvent se déplacer en utilisant diverses formes de mobilité. 1970 est l'année du démarrage de la robotique mobile. Le robot mobile Shakey a été développé à Stanford Research Institute (aujourd'hui connu sous le nom de SRI Technology), contrôlé par des algorithmes intelligents qui observent via des capteurs et réagissent à leurs propres actions. Le robot mobile à roues Shakey est équipé d'une logique embarquée, d'une caméra, d'un télémètre et d'un détecteur de bosses[1]. Les télé-robots sont des machines semblables à des robots qui sont contrôlés à distance par un opérateur humain. Le plus ancien était un bateau radiocommandé présenté par Nikola Tesla en 1898 appelé un télé-automate. Selon la définition ci-dessus, ce ne sont pas des robots, mais ils étaient un précurseur important des robots et sont toujours importants aujourd'hui pour de nombreuses tâches que les gens ne peuvent pas exécuter mais qui sont trop complexes pour qu'une machine puisse les exécuter seule. Par exemple, les robots sous-marins qui ont inspecté l'épave du Titanic étaient des véhicules techniquement télécommandés. Un robot chirurgical moderne est également télécommandé, le mouvement des outils est contrôlé à distance par le chirurgien et cela permet d'utiliser des incisions beaucoup plus petites que

l'ancienne approche où le chirurgien travaille à l'intérieur du corps avec ses mains [60] . Les différents rovers martiens naviguent de manière autonome sur la surface de Mars mais des opérateurs humains fournissent les objectifs de haut niveau. C'est-à-dire que les opérateurs disent au robot où aller et le robot lui-même détermine les détails de l'itinéraire. La prise de décision locale sur Mars est essentielle étant donné que le délai de communication est de plusieurs minutes. Certains robots sont hybrides et la tâche de commande est partagée ou échangée avec un opérateur humain. La fonction de contrôle est transmise dans les deux sens entre l'opérateur humain et l'ordinateur. Par exemple, un pilote d'avion peut passer le contrôle à un pilote automatique et reprendre le contrôle. Dans une voiture de tourisme autonome l'ordinateur garde la voiture en toute sécurité sur la voie pendant que le conducteur humain contrôle simplement la vitesse.

1.1 Robots et éthique

Un certain nombre de questions éthiques découlent de l'avènement de la robotique. Peut-être la plus grande préoccupation pour le grand public, ce sont « Des robots qui prennent des emplois à des gens ». C'est une question complexe mais nous ne pouvons pas ignorer le fait que de nombreux emplois actuellement occupés par des personnes seront, à l'avenir, futur, pris par des robots. De toute évidence, il existe des emplois dangereux que les gens devraient ne pas faire, par exemple manipuler des substances dangereuses ou travailler dans des environnements dangereux. Il existe de nombreux emplois peu qualifiés où la main-d'œuvre humaine est de plus en plus difficile à trouver, par exemple dans des emplois comme la cueillette de fruits. Dans de nombreux pays développés, les gens ne sont plus attirés par des travaux d'extérieurs pénibles dans des endroits éloignés. Quelles sont les alternatives si les gens ne veulent pas faire le travail ? Dans des domaines comme la fabrication, en particulier la fabrication automobile, l'adoption de l'automatisation robotique a été essentielle pour augmenter la productivité qui a permis à cette industrie d'être économiquement viable dans des pays comme l'Europe, le Japon et les États-Unis. Sans robots, ces industries ne pourraient pas exister ; ils n'emploieraient personne, ne paieraient

pas d'impôts et ne consommeraient pas de produits et les services d'autres secteurs de l'économie. L'industrie automatisée pourrait employer moins de gens, mais elle apporte quand même une contribution importante à la société. Plutôt que de prendre les emplois, nous pourrions dire que la robotique et l'automatisation ont contribué à maintenir la fabrication industrielle viable dans les pays à coût de main-d'œuvre élevé. Comment concilier le bien de la société avec le bien de l'individu ? Outre les emplois, il y a d'autres problèmes. Pensez aux voitures autonomes. Nous acceptons étonnamment les voitures à conduite manuelle même si elles tuent plus d'un million de personnes chaque année, mais beaucoup sont mal à l'aise avec l'idée de voitures autonomes, même si ils réduiront considérablement ces pertes de vie. Nous nous inquiétons de savoir qui blâmer si un robot voiture fait une erreur alors que le carnage causé par les conducteurs humains continue. Similaire des inquiétudes qui sont soulevées lorsqu'on parle de soins et de chirurgie robotiques. Les chirurgiens humains ne sont pas parfaits, mais les robots sont apparemment tenus à un compte beaucoup plus élevé. Là on parle beaucoup d'utiliser des robots pour s'occuper des personnes âgées, mais cela nuit-il de leur qualité de vie en supprimant le contact humain, la conversation et la compagnie ? Doit-on utiliser des robots pour s'occuper de nos enfants, voire leur apprendre ? Qu'est-ce que on pense à des armées de robots combattant et tuant des êtres humains ? Les voitures robotisées, pourraient apporter des avantages économiques à notre société, mais est-cela est la bonne chose à faire ? Est-ce une direction que nous voulons pour notre société ? Encore une fois, comment équilibrer le bien de la société avec le bien de l'individu ? Ce sont des questions éthiques profondes qui ne peuvent et ne doivent pas être tranchées par roboticiens seuls. Mais les roboticiens ne doivent pas non plus les ignorer. Ceci est une discussion pour l'ensemble de la société et les roboticiens doivent d'être des acteurs actifs de ce débat.

1.2 Classification des robots

Une taxonomie est basée sur la fonction que le robot exécute et classe les robots en robots de fabrication, robots de service, robots de terrain et robots humanoïdes.

1.2.1 Les robots de fabrication

Les robots de fabrication sont les descendants technologiques de la première génération de robots. Un robot de fabrication est généralement un manipulateur de type bras sur une base fixe qui effectue des tâches répétitives au sein d'une cellule de travail locale dans une usine. Les pièces sont présentées au robot d'une manière ordonnée qui maximise l'avantage de haute rapidité et précision. Les robots à grande vitesse sont dangereux et la sécurité est obtenue en excluant les personnes des lieux de travail , en plaçant généralement le robot à l'intérieur d'une cage.

1.2.2 Les robots de service

Ces robots fournissent des services à des personnes telles que le nettoyage, les soins, les services médicaux, la rééducation ou le transport.

1.2.3 Les robots de terrain

Ce type de robots travaille à l'extérieur sur des tâches telles que la surveillance de l'environnement, l'agriculture, l'exploitation minière, la construction et la foresterie.

1.2.4 Les robots humanoïdes

Les robots humanoïdes ont la forme physique d'un être humain.

Les robots de terrain et de service sont des robots mobiles et confrontés à des défis spécifiques et importants. Le premier défi est que le robot doit fonctionner et se déplacer dans un environnement complexe, encombré et changeant. Un robot de livraison dans un hôpital doit fonctionner malgré une foule de personnes et une configuration des chariots variable dans le temps. Un rover martien doit naviguer dans les rochers et les petits cratères bien qu'il n'ait pas de carte locale précise. Les voitures robotisées ou autonomes doivent suivre les routes, éviter les obstacles et respecter les signaux de circulation et les règles de la route. Le deuxième défi pour ces types de robots est qu'ils doivent fonctionner en toute

sécurité en présence de personnes. Le robot de livraison de l'hôpital fonctionne parmi les personnes, la voiture robotisée contient des personnes et un dispositif chirurgical robotisé fonctionne à l'intérieur des gens.

1.3 Robot mobile

Nous désignons le plus souvent par le terme robot mobile, les robots mobiles à roues, les autres robots mobiles sont en effet désignés par leur types de locomotion ; robots marcheurs, robots aériens ou sous-marins. Les robots mobiles ont soulevé un grand nombre de sujets difficiles comme la localisation, la navigation, la construction de cartes de l'environnement ou la cartographie. On trouve les robots mobiles dans presque tous les domaines contrairement aux robots manipulateurs généralisés dans l'industrie. Les robots manipulateurs sont prévus à travailler de manière répétitive dans des espaces connus contrairement aux robots mobiles qui sont destinés à évoluer dans des environnements peu ou non structurés.

1.3.1 Définition d'un robot mobile

Un robot mobile a la particularité de se déplacer librement dans un espace de travail prédéfini pour atteindre un objectif souhaité. Contrairement à la majorité des robots industriels qui ne peuvent se déplacer que dans un espace de travail spécifique. La capacité de mobilité des robots mobiles les rend adaptés à un large répertoire d'applications dans des environnements structurés et non structurés. Les robots mobiles comprennent également les véhicules aériens sans pilote UAV (unmanned aerial vehicles) et les véhicules sous-marins autonomes AUV (autonomous underwater vehicles) . L'une des fonctions les plus importantes d'un robot mobile est de se déplacer dans son environnement. Le robot empruntera un chemin pour atteindre sa destination tout en faisant face à des défis tels que l'évitement des obstacles. Les premières applications commerciales des robots mobiles sont apparues dans les années 1980 lors de l'automatisation des véhicules guidés AGV (automated guided vehicles) qui ont été développés pour le transport du matériel dans des usines. Les robots mobiles peuvent être des Systèmes robotiques de terrain tels que les ca-

mions dans les mines, les véhicules de transport de conteneurs dans les ports d'expédition et les tracteurs autonomes pour l'agriculture. Ils peuvent aussi être des véhicules aériens sans pilote UAV , des véhicules autonomes sous-marins AUV ou des bateaux robotisés ASV (autonomous surface vehicle) . Parmi les robots mobiles de terrain on distingue les robots mobiles à roues WMR (wheeled mobile robot) et robots mobiles à pattes LMR (legged mobile robot). Les WMR sont très populaires parce qu'ils sont appropriés pour des applications typiques avec une faible complexité mécanique et relativement une basse consommation d'énergie. Les robots à pattes conviennent à des tâches non standard comme les escaliers. Les robots mobiles comprennent également les manipulateurs mobiles ; robots à roues ou à pattes équipé d'un ou plusieurs manipulateurs pour effectuer diverses tâches [61].

1.3.1.1 Robots mobiles à pattes

La roue est une invention humaine, mais la jambe est un élément biologique. Pour être utile dans la réalisation des tâches, un robot doit être statiquement stable. Un robot à pattes est dit dynamiquement stable s'il ne tombe pas, malgré le fait qu'il n'est pas statiquement stable. Les robots à pattes se distinguent en deux grandes catégories : Robots à deux pattes (bipèdes) et les robots à plusieurs pattes. La locomotion bipède du robot nécessite une interaction complexe entre sa conception mécanique et les caractéristiques du système de contrôle. Les robots à plusieurs pattes sont destinés à se déplacer sur des terrains extrêmement difficiles, tels que des zones montagneuses, des fossés, des endroits endommagés par un tremblement de terre, etc.

1.3.1.2 Robots mobiles à roues

La locomotion à roues est l'une des grandes innovations de l'humanité. La roue a été inventée vers 3000 ans av. J.-C. et la charrette à deux roues vers 2000 ans av. J.-C. Aujourd'hui les véhicules à quatre roues sont omniprésents et le nombre total d'automobile de la planète dépasse un milliard. L'efficacité des voitures, et notre familiarité avec elles, en font un choix pour les plates-formes de robots qui se déplacent sur le sol. Nous savons,

d'après notre expérience quotidienne avec les voitures, qu'il y a des limites à la façon dont elles se déplacent. Il n'est pas possible de conduire de côté, mais avec un peu de pratique, nous pouvons apprendre à suivre un chemin qui fait que le véhicule se trouve d'un côté de sa position initiale – cela est un parking parallèle. Une voiture ne peut pas non plus tourner sur place, mais nous pouvons suivre un chemin qui fait que le véhicule est à la même position mais pivoté de 180° – un trois points tourner. La nécessité d'effectuer de telles manœuvres est la marque distinctive d'un système non holonomes – un concept important. En dépit ces petites limitations la voiture est le moyen le plus simple et le plus efficace pour se déplacer. Les véhicules à direction différentielle qui sont mécaniquement plus simples que les voitures et n'ont pas de roues directrices. Il s'agit d'une configuration courante pour les petits robots mobiles et aussi pour les machines plus grandes comme les bulldozers. La manœuvrabilité d'un robot mobile à roues dépend du type de roues et de l'entraînement utilisé.

1.3.2 Les types de roues utilisées en robotique mobile :

Les roues utilisées en robotiques mobiles peuvent être des roues conventionnelles ou spéciales [62].

1.3.2.1 Roues conventionnelles :

Ce type de roues se distingue en roues motrices fixes, roues folles et roues directrices.

— Les roues motrices

Les roues motrices sont entraînées par des moteurs montés sur des positions fixes du véhicule. Leurs axes de rotation sont placés à une direction fixe par rapport au repère de coordonnées de la plate-forme.

— Les roues folles ou libres

Les roues libres ne sont pas motorisées mais elles peuvent aussi tourner librement autour d'un axe perpendiculaire à leur axe de rotation.

— **Les roues directrices**

La roue directrice possède un moteur d'entraînement pour leur rotation et peuvent être dirigées autour d'un axe perpendiculaire à leur axe de rotation.

1.3.2.2 Roues spéciales :

Ces roues sont conçues de manière à être motrices et passives, permettant ainsi une plus grande manœuvrabilité dans des environnements encombrés. Nous avons trois principaux types de roues spéciales :

— **Les roues universelles**

La roue universelle contient des petits rouleaux autour de son diamètre extérieur, monté perpendiculairement à l'axe de rotation de la roue. De cette façon, la roue peut rouler dans la direction parallèle à l'axe de la roue en plus de la rotation normale de la roue.

— **Les roues Mecanums**

La roue Mecanum est similaire à la roue universelle sauf que les rouleaux sont montés à un angle α différent de 90 [63].

— **Les roues sphériques**

La roue sphérique n'impose aucune contrainte directe sur le mouvement, c'est-à-dire, l'axe de rotation de la roue peut avoir n'importe quelle direction arbitraire. En raison de sa construction difficile, la roue sphérique est très rarement utilisée dans la pratique.

1.3.3 Les classes des robots mobiles à roues

La classification des robots se fait selon plusieurs critères, trois principalement [64] :

- Le niveau d'autonomie.
- Les domaines d'application.
- Le système de locomotion.

1.3.3.1 Classification selon le niveau d'autonomie

Un robot autonome est une machine ayant la capacité de collecter et de traiter des données et de prendre des décisions lui permettant d'effectuer un nombre d'activités dans

un environnement connu ou inconnu. Les robots mobiles peuvent être classés dans les catégories suivantes en fonction de leurs niveaux d'autonomie :

— **Robots télécommandés :**

Ce sont des robots guidés par un opérateur (machine ou humain) qui leur dicte des tâches à exécuter.

— **Robots semi-autonomes :**

Ces robots peuvent effectuer une variété de tâches de manière totalement autonome, mais ils peuvent exécuter des commandes d'un opérateur.

— **Robots autonomes :**

Un robot est dit entièrement autonome s'il peut adapter son comportement à l'environnement dans lequel il évolue. L'autonomie fait référence à la capacité d'un système autonome à percevoir, analyser, communiquer, planifier, faire des choix et agir afin d'atteindre les objectifs qui lui sont fixés au cours du développement.

1.3.3.2 Classification selon le domaine d'application :

L'un des principaux avantages des robots mobiles est leur vaste champ d'application :

— **Les robots industriels et de service :**

Les robots industriels sont fréquemment utilisés dans l'industrie automobile et leur conception nécessite des connaissances approfondies et un haut degré d'expertise en ingénierie. Les robots de service sont conçus pour assister les personnes handicapées dans leur vie quotidienne, ainsi que pour guider les aveugles et faire fonctionner des voitures autonomes.

— **Robots militaires :**

Un robot militaire est un robot autonome télécommandé développé pour usage militaire. Les drones sont un type de robot militaire qui entre dans la catégorie des véhicules aériens sans pilote. Ce type de robots sont très performant avec leur grande vitesse, leurs capacités à franchir les obstacles et leurs temps de réponse.

— **Robots de recherche :**

De nombreux laboratoires sont engagés dans le domaine de la robotique afin de valider

des travaux théoriques sur la perception et la planification du mouvement. L'utilisation de robots dans les laboratoires est connue sous le nom de robotique de laboratoire.

1.3.3.3 Classification basée sur le mode de locomotion

Le système de locomotion est l'un des aspects les plus importants. La structure mécanique la plus courante en robotique mobile est la roue. Ce type de robot permet un mouvement facile mais nécessite une surface plane. Les robots à roues sont classés en plusieurs catégories en fonction de l'emplacement et du nombre de roues dont ils disposent. Nous allons passer en revue les nombreux types de robots à roues.

— Robot mobile à entraînement différentiel

Son entraînement est composé de deux roues motrices fixes indépendantes montées sur les côtés gauche et droit de la plate-forme du robot. Une ou deux roulettes passives sont utilisées pour assurer la stabilité du robot. L'entraînement différentiel est l'entraînement mécanique le plus simple. Si les roues tournent à la même vitesse, le robot avance tout droit ou recule. Si une roue tourne plus vite que l'autre, le robot suit une trajectoire circulaire. Si les deux roues tournent à la même vitesse dans des directions opposées, le robot tourne autour de lui-même, la figure (1.1) montre les différents mouvements.

Le centre de courbure instantané (ICC) du robot se situe au point de croisement de tous les axes des roues. ICC est le centre du cercle de rayon R dépend des vitesses des deux roues motrices ; v_l est la vitesse de la roue gauche et v_r est la vitesse de la roue droite comme illustré par la figure (1.2). Le rayon R est déterminé par la relation

$$\frac{v_l - v_r}{2a} = \frac{v_r}{R - a} \quad (1.1)$$

donc

$$R = \frac{v_l + v_r}{v_l - v_r} \quad (1.2)$$

avec $v_l \neq v_r$

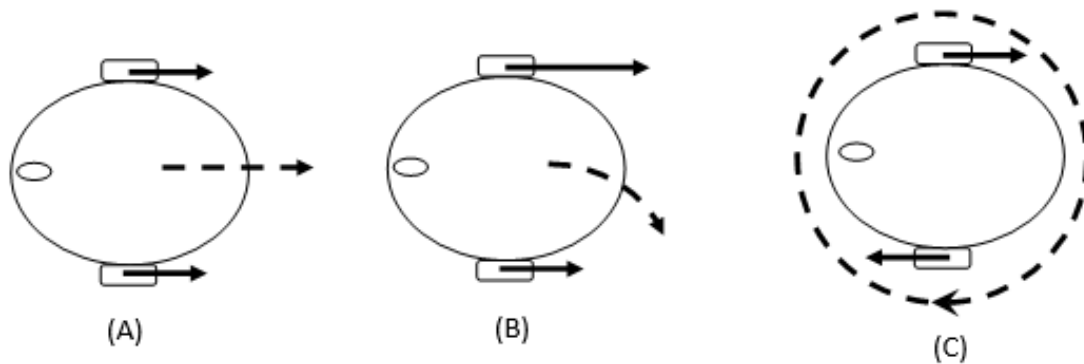


FIGURE 1.1 – Les différents mouvements possibles d'un robot mobile différentiel A : mouvement en ligne droite, B et C : mouvement circulaire.

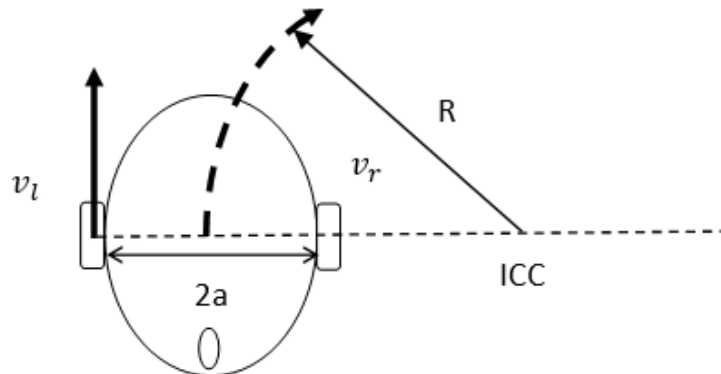


FIGURE 1.2 – Calcul du rayon de rotation instantanée d'un robot mobile différentiel.

Lorsque :

- $v_l = v_r$; $R = \infty$ (le robot suit une ligne droite)
- $v_l = -v_r$; $R = 0$ (le robot tourne sur lui-même)

$2a$ est égale la distance entre les deux roues motrices du robot mobile.

— **Robot mobile tricycle :**

Ce type de robot a une seule roue motrice et orientable. Pour maintenir sa stabilité, deux roues libres (non motorisées) à l'arrière sont utilisées, afin d'avoir toujours le contact à trois points requis. La vitesse linéaire et la vitesse angulaire de la roue motrice sont entièrement découplées. Pour rouler en ligne droite, la roue motrice est positionnée en position médiane et entraînée à la vitesse souhaitée. Lorsque la roue avant est inclinée, le véhicule suit une

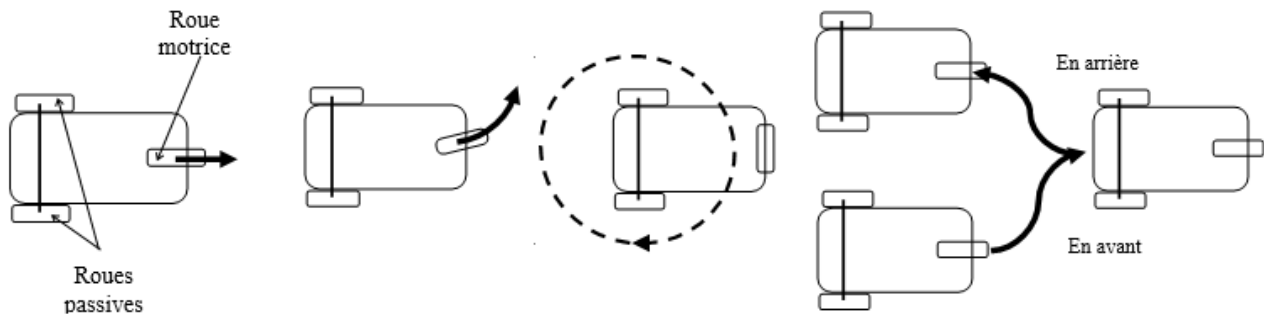


FIGURE 1.3 – Les différentes trajectoires possibles effectuées par un robot mobile tricycle.

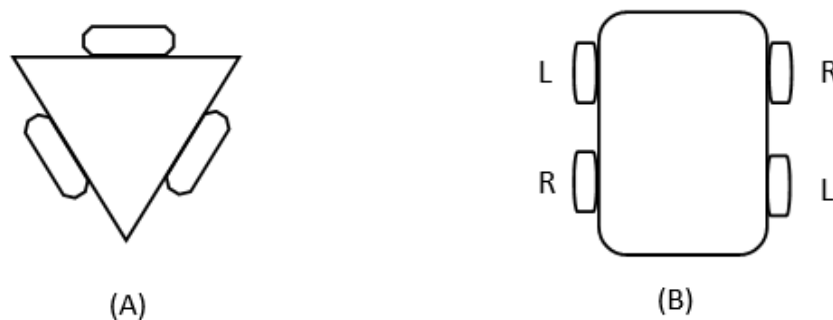


FIGURE 1.4 – Robot mobile omnidirectionnel. (A) cas à trois roues, (B) cas à quatre roues.

trajectoire circulaire. Si la roue avant est positionnée à 90° , le robot tournera en suivant une trajectoire circulaire dont le centre se situe au milieu des roues arrière et pas au centre géométrique du robot. Cela signifie que ce robot ne peut pas tourner sur place. Les robots mobiles non holonomes (comme les véhicules à entraînement différentiel ou tricycle) ne peuvent pas effectuer de stationnement parallèle directement mais par un certain nombre de manœuvres avec des mouvements vers l'avant et vers l'arrière comme le montre la figure (1.3).

— **Robot mobile omnidirectionnel :**

Ce type de robot peut avoir trois, quatre ou plus de roues comme illustré sur la figure (1.4). Les robots mobiles à trois roues utilisent les roues universelles qui ont un angle de roulement de 90° . Les robots mobiles omnidirectionnels à quatre roues utilisent des roues Mecanums.

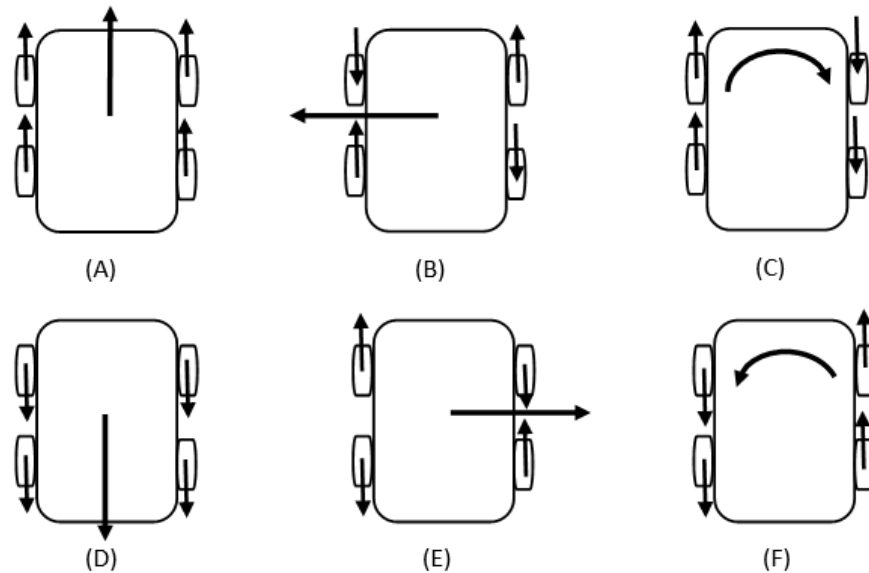


FIGURE 1.5 – Les six modes de locomotion de base d’un WMR omnidirectionnel avec quatre roues Mecanum.

La figure (1.5) montre six mouvements de base d’un robot omnidirectionnel à quatre roues, à savoir (A) mouvement vers l’avant, (B) glissement vers la gauche, (C) rotation dans le sens des aiguilles d’une montre (sur place), (D) mouvement vers l’arrière, (E) glissement vers la droite et (F) rotation dans le sens inverse des aiguilles d’une montre.

1.3.4 La manœuvrabilité d’un robot mobile à roues

La manœuvrabilité M_{wmr} d’un robot mobile à roues est définie comme suit :

$$M_{wmr} = D_m + D_s \quad (1.3)$$

Où D_m est le degré de mobilité et D_s le degré d’orientabilité (steerability).

— Degré de mobilité :

Le degré de mobilité D_m est déterminé par le nombre de contraintes indépendantes que le type de roues et leur configuration imposent au mouvement du robot. Les contraintes sur le mouvement ne sont imposées que par des roues motrices (fixes ou directrices). Les roues omnidirectionnelles n’imposent aucune contrainte sur la mobilité du robot. Le degré maximal de mobilité D_m est de 3, soit juste lorsqu’aucune contrainte cinématique n’est

imposée. C'est le cas lorsque toutes les roues du WMR sont omnidirectionnelles. En général, le degré de mobilité est égal à :

$$D_m = 3 - N_c \quad (1.4)$$

N_c est le nombre de contraintes indépendantes.

— **Degré d'orientabilité (maniabilité) :**

Le degré de maniabilité (steerability) noté D_s dépend du nombre de paramètres de direction indépendamment contrôlables et se situe dans l'intervalle $[0, 2]$.

Si le robot ne possède pas de roues orientables (directrices) $D_s=0$. Si le robot n'a pas de roue standard fixe $D_s = 2$. En fait, $D_s = 2$ signifie que le WMR peut placer son ICR à n'importe quel point du plan. Le cas le plus courant est $D_s=1$ qui est obtenu lorsque la configuration du robot a une ou plusieurs roues conventionnelles orientables.

Deux autres paramètres caractéristiques des robots mobiles à roues sont les degrés de liberté DOF et le degré de liberté différentiel $DDOF$, qui vérifient la relation :

$$DDOF \leq M_{wmr} \leq DOF \quad (1.5)$$

Le $DDOF$ est égal à D_m et représente le nombre de vitesses indépendantes qui peuvent être atteintes. DOF représente la capacité d'un WMR à atteindre divers position dans son environnement.

1.4 La perception en robotique mobile

La détection est essentielle pour les robots. Les capteurs conçus pour les robots ressemblent aux systèmes sensoriels humains qui fournissent des signaux d'entrée au cerveau. L'utilisation de capteurs en robotique est d'une importance primordiale pour fermer la boucle de contrôle qui garantis des performances efficaces et automatisées au fonctionnement autonome des robots dans des applications réelles. Le transfert de données du capteur vers l'ordinateur (CPU) peut être effectué par l'ordinateur ou par le capteur lui-même. Certains de ces capteurs sont adaptés à la mesure de position d'un robot, tandis que d'autres conviennent à la mesure de position lors d'un mouvement du robot. Les capteurs à système

mécanique nécessitent un contact physique entre le robot et le capteur. Souvent, ils sont intégrés dans la carcasse du robot. Les capteurs acoustique et électromagnétique utilisent la directivité et la mesure du temps de vol des signaux envoyés et signaux reçus afin de calculer la position de l'objet d'intérêt [65]. Les capteurs magnétiques utilisent la configuration spatiale du champ magnétique statique de la terre et des solénoïdes pour le calcul de la position. Enfin, les capteurs comprennent aussi les caméras avec tous leurs équipements auxiliaires nécessaires à la vision artificielle. Les autres classifications des capteurs utilisés en robotique sont les suivantes :

D'un point de vue robotique :

- Capteurs embarqués : Capteurs montés sur le robot.
- Des capteurs déportés : Capteurs montés à l'extérieur du robot dans l'environnement et envoient les données au système de contrôle du robot.

D'un point de vue passif/actif :

- Les capteurs passifs : Ils surveillent l'environnement sans l'affecter par exemple, gyroscope, caméra .
- Des capteurs actifs : Ils stimulent l'environnement pour le surveiller par exemple, capteur infrarouge, scanner laser, capteur sonar.

Capteurs des robots mobiles

- Capteurs proprioceptifs : Capteurs qui surveillent l'état interne du robot
- Capteurs extéroceptifs : Capteurs qui surveillent l'environnement des robots

Les capteurs internes comprennent les capteurs qui mesurent la vitesse du moteur, la charge des roues, les angles des articulations des bras de robot manipulateurs et la tension de batterie. Les capteurs externes comprennent des capteurs qui mesurent la distance, l'amplitude sonore et l'intensité lumineuse. Des exemples de capteurs passifs comprennent des sondes de température, des microphones et des caméras. Des exemples de capteurs actifs, qui émettent de l'énergie dans l'environnement, sont les encodeurs quadratiques de roue et les télémètres laser[66].

1.4.1 Caractéristiques des capteurs utilisés en robotique

Dans tous les cas, les caractéristiques souhaitées du capteur sont une résolution élevée, une large plage de fonctionnement, une réponse rapide, un étalonnage facile, une haute fiabilité et un faible coût d'achat et d'entretien. Les capteurs utilisés dans la robotique stationnaire ou mobile présentent des caractéristiques de performances variées. Ces caractéristiques sont différentes en fonction des environnements contrôlés (environnement intérieur, laboratoire) et des environnements non contrôlés (environnement extérieur, monde réel). Les caractéristiques de base des capteurs sont les suivantes [1] :

— **La plage dynamique :**

la plage dynamique d'un capteur est l'écart entre les limites inférieure et supérieure des valeurs d'entrée pour lesquelles le capteur fonctionne normalement. Généralement, afin de couvrir à la fois les très petites et les très grandes plages du signal, le logarithme du rapport des valeurs maximales et minimales est utilisé.

$$Range = \frac{20 \log (maximum\ input)}{minimum\ input} \quad (dB) \quad (1.6)$$

— **La résolution :**

La résolution d'un capteur est la différence minimale de la variable mesurée qui peut être reconnue par le capteur. Typiquement, dans les capteurs analogiques, la résolution coïncide avec la limite inférieure de la plage de fonctionnement du capteur. Cependant, ce n'est pas vrai pour les capteurs numériques, où l'entrée analogique est convertie sous forme binaire.

— **La linéarité :**

La linéarité implique que la valeur de sortie $f(in1+in2)$ d'un capteur à une somme d'entrées $in1$ et $in2$ est égale à la somme $f(in1) + f(in2)$ des valeurs de sortie du capteur obtenues séparément par chaque entrée. Dans une formulation plus générale, la linéarité implique $f(k_1 in1+k_2 in2)=k_1 f(in1)+k_2 f(in2)$, où k_1 et k_2 sont des paramètres constants.

— **La bande passante :**

Le débit ou la fréquence maximale de lecture de données que le capteur peut fournir. Le nombre de lectures (mesures) par seconde fournit la fréquence du capteur mesuré en hertz.

— **La sensibilité :**

C'est le degré auquel les changements du signal d'entrée affectent le signal de sortie.

— **L'exactitude (accuracy) :**

C'est-à-dire dans quelle mesure la lecture des capteurs coïncide avec la vraie valeur de l'entrée.

Si

$$error = \text{la lecture du capteur} - \text{la vraie valeur}$$

alors l'exactitude est donnée par :

$$Exactitude = 1 - |error| / (\text{la vraie valeur}) \quad (1.7)$$

— **La précision :**

s'applique dans le cas où les erreurs sont aléatoires. La précision est définie comme

$$Precision = \frac{Range}{\sigma} \quad (1.8)$$

Où σ est l'écart type de l'erreur par rapport à une valeur moyenne

1.4.2 Les différents capteurs utilisés en robotique

1.4.2.1 Capteurs de position et de vitesse :

Les capteurs de position sont utilisés pour déterminer si les articulations des liaisons robotiques ou des plates-formes mobiles sont placées à la position/orientation souhaitée de l'espace cartésien. De même, les capteurs de vitesse mesurent la vitesse du mouvement linéaire ou angulaire des articulations ou de la plate-forme du robot, parmi les capteurs de position les plus utilisés nous pouvons citer :

— **Les résolveurs**

Le résolveur est un capteur analogique qui donne une sortie proportionnelle à l'angle de rotation d'un objet par rapport à un élément fixe.

— **Les encodeurs :**

L'encodeur constitue un capteur de rétroaction de base pour la commande des moteurs. On distingue deux types d'encodeurs : encodeurs différentiels (incrémentaux) et encodeurs

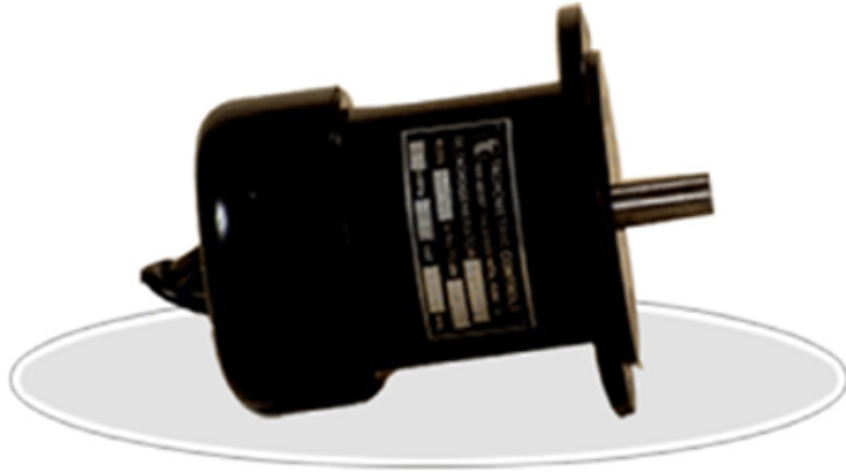


FIGURE 1.6 – Tachymètre à courant continu [1]

absolus. Les deux techniques typiques de construction d'un encodeur sont le capteur à effet Hall (capteur magnétique) et l'encodeur optique (un disque de secteur avec des segments noirs et blancs avec une diode électroluminescente (LED) et une photo-diode.

— Les tachymètres

En règle générale la vitesse des articulations des robots est mesurée directement à l'aide d'un tachymètre. La méthode indirecte de mesure de la vitesse par différenciation numérique du signal de position n'est pas préférée en raison de la production du bruit de différenciation. Les tachymètres se distinguent en tachymètres à courant continu (DC) et à courant alternatif (AC). En robotique, le tachymètre DC donné par la figure (1.6) est le capteur le plus utilisé[1]. Le tachymètre DC appelée aussi génératrice tachymétrique génère une tension continue, à sa sortie, qui est proportionnel à la vitesse du moteur qui lui est relié. Le champ magnétique permanent (aimant permanent) utilisé élimine le besoin d'excitation externe et offre une sortie très fiable et stable. Les tachymètres à courant continu utilisent un commutateur et donc une petite ondulation apparaît dans la sortie qui ne peut pas être entièrement filtrée. L'ondulation n'apparaît pas dans les tachymètres AC.

1.4.2.2 Capteurs de distance

Les capteurs qui mesurent la distance entre un robot et les obstacles qui l'entourent peuvent être des capteurs sonars, des Capteurs laser ou des Capteurs infrarouges.

— Capteurs sonars

Les capteurs sonars ont un angle de cône relativement étroit, ce qui fait pour assurer une couverture de 360° , une configuration typique de capteurs de robot mobile consiste à utiliser 24 capteurs, dont chacun cartographie un cône d'environ de 15° . En fait, il existe des sonars commerciaux disponibles avec une large gamme d'angles de cône, de manière à répondre à toutes les applications pratiques possibles. Le principe de fonctionnement des capteurs sonar comprend l'émission d'un signal acoustique court (d'une durée d'environ 1 ms) à une fréquence ultrasonique de 50 kHz et la mesure du temps écoulé entre l'émission du signal et le retour de l'écho vers le capteur. Le temps de vol mesuré est proportionnel à deux fois la distance de l'obstacle le plus proche dans l'angle du cône du capteur. Si aucun signal n'est reçu dans un délai maximum, alors aucun obstacle n'est détecté. Les mesures sont répétées environ 20 fois par seconde (ce qui correspond à son bruit typique). La distance l_{sonar} d'un objet qui provoque la réflexion de l'onde est donnée par

$$l_{sonar} = \frac{1}{2}v_{son}t_0 \quad (1.9)$$

Où v_{son} est la vitesse du son (environ 330 350 m/s dans l'air) et t_{vol} est le temps de vol.

La vitesse t_{vol} du son dans l'air est donnée par

$$t_{vol} = \sqrt{\gamma RT} \quad (1.10)$$

Où R est la constante des gaz, γ est le rapport de la chaleur spécifique et T est la valeur de la température absolue (K). La portée effective de la plupart des sonars utilisés en robotique mobile varie environ entre 12 cm et 5 m, avec une précision de 98 à 99,1%.

— Capteurs laser

Les capteurs laser constituent un cas particulier des capteurs optiques qui ont une portée de mesure allant de centimètre au mètre. Ils sont communément appelés radars laser

(LIDAR : light direction and ranging sensors). L'énergie est émise en impulsions. La distance est calculée à partir du temps de vol. Ils peuvent également être utilisés comme laser altimètres pour l'évitement d'obstacles ou pour la détection de véhicules sur les autoroutes. Un typique télémètre laser est illustré à la figure (1.7)[1]. Les capteurs laser fournissent la vitesse et la hauteur. Les lasers rotatifs sont basés sur la rotation de l'onde 360° à plus de 1 ou 2 rpm (révolution par minute), et un miroir à 45° . Cela surmonte le problème des zones cachées. Très souvent, ils travaillent en conjonction avec la réflexion des balises. Les lasers rotatifs mesurent la position angulaire. Télémètres laser à balayage combinent le lidar avec les lasers rotatifs en fournissant à la fois la portée et la position angulaire de l'objet détecté. Ils n'ont pas besoin de balises réfléchissantes et sont donc très utiles car ils peuvent fonctionner dans des environnements non structurés. Malheureusement, les capteurs laser sont très grands et lourds et aussi trop chers pour les petits robots mobiles. Pour cette raison, les capteurs de distance infrarouges sont très populaires dans les applications de robots mobiles. La mesure du temps de vol d'un télémètre laser se fait à l'aide d'un laser pulsé et mesurant le temps écoulé directement comme dans le capteur sonar. Le moyen le plus simple consiste à mesurer le déphasage de la lumière réfléchi.

— Capteurs infrarouges

La lumière proche d'infrarouge peut être produite par une LED ou un laser. La longueur d'onde typique de la lumière infrarouge émise se situe entre 820 et 880 nm. Par conséquent, la plupart des surfaces ont une rugosité supérieure à la longueur d'onde de la lumière incidente et ainsi la réflexion aura lieu, c'est-à-dire que la lumière est réfléchi de manière presque isotrope. La composante de la lumière infrarouge qui tombe dans l'ouverture du capteur renvoie presque parallèlement au faisceau transmis pour les objets éloignés. Le capteur envoie une lumière modulée en amplitude à 100% à une fréquence donnée f et mesure le déphasage entre le signal transmis et le signal réfléchi. Si φ est le déphasage mesuré et λ la longueur d'onde, la distance est égale à $(\frac{\varphi}{4\pi})\lambda$.

1.4.2.3 Quelques autres capteurs utilisés en robotiques

Dans cette section, d'autres capteurs utilisés en robotiques sont brièvement décrits :



FIGURE 1.7 – Télémètre laser[1].

— Gyroscope

Les gyroscopes sont des appareils mesurant la position angulaire en préservant leur orientation par rapport à un repère universel. Ainsi, ils sont utilisés en aéronautique, et en navigation des robots mobiles. Le gyroscope se distingue en gyroscope libre et gyroscope à deux degrés de liberté (figure 1.8). Le disque tourne à très grande vitesse angulaire (de l'ordre de 10 000 rad/min jusqu'à 25 000 rad/min). Le système de montage via les deux cardans permet l'axe du moment cinétique du disque rotatif pour avoir trois DOF, mouvement autour les axes x , y et z . Le gyroscope libre peut indiquer en permanence soit la direction sud-nord (gyroscope à direction horizontale) ou la direction verticale si l'axe du moment cinétique est aligné avec la direction de la gravité (direction verticale gyroscope), c'est-à-dire que si le gyroscope est incliné, les cardans se réorienteront pour garder l'axe de rotation du disque dans le même sens. C'est l'implication du principe de conservation de l'angle du moment cinétique. Le gyroscope à deux DOF est obtenu à partir de gyroscope libre en plaçant un ressort de torsion selon l'axe x entre les deux cardans [2].

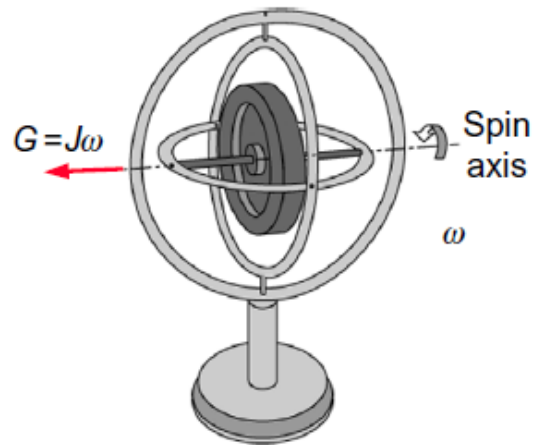


FIGURE 1.8 – Schéma illustratif d'un gyroscope [2]

— Boussole

La Boussole magnétique a été utilisée pour la première fois en Chine en 2634 av. J.-C., lorsqu'un morceau de fer magnétique était suspendu par un fil de soie pour permettre son alignement avec les lignes du champ magnétique terrestre, qui sont horizontales à l'équateur et vertical aux pôles magnétiques. Aujourd'hui, des boussoles électroniques sont disponibles pour une utilisation embarquée des robots mobiles. Les boussoles analogiques peuvent être facilement intégrées aux contrôleurs de robot. Une simple boussole analogique ne peut distinguer que huit directions qui sont représentées par niveaux de tension respectifs. Ceci est utilisé dans de nombreux systèmes à quatre roues motrices. Les Boussole électroniques sont beaucoup plus complexes que les modules analogiques et offrent une résolution directionnelle plus élevée (par exemple, une résolution dans les applications intérieures). Certaines boussoles disponibles dans le commerce pour les robots mobiles incluent la boussole magnétique Dinsmore Starguide, la boussole fluxgate, boussole Zemco, boussole gyroscope Watson, boussoles KVA, boussole magnéto résistive Philips, et d'autres

— Les capteurs de force

Il existe de nombreux types de capteurs qui mesurent la force (charge) ou le couple. Il existe de nombreuses raisons pour lesquelles nous aurions besoin de mesurer directement les forces dans un robot, par exemple, mesure du poids, quantification de la force, optimisation des

paramètres. Un exemple notable est un humanoïde, où un capteur de force permet de savoir combien le poids est sur chaque jambe. Une autre consiste à placer un capteur de force dans le bras d'un manipulateur mobile. Pince pour contrôler la friction de la pince afin de ne pas écraser ou laisser tomber l'objet à saisir et de savoir si le robot a atteint son poids de charge maximum ou combien peut-il porter. Un capteur de force habituel est la soi-disant une jauge de contrainte qui est une minuscule bobine plate de fil conducteur qui change de résistance lorsqu'il est plié. La déformation est directement liée à la force appliquée pour plier la poutre. Si les forces anticipées sont faibles, un conducteur élastomère de mousse peut être utilisé comme jauge de contrainte d'une mousse conductrice. La compression de la mousse diminue la résistance électrique. Pour faire face aux problèmes d'une jauge de contrainte typique, en raison de laquelle elle a un très faible changement de résistance lorsqu'elle est pliée, les jauges de contrainte sont connectées à l'intérieur du capteur de force, généralement un multiple de quatre, dans une configuration en pont de Wheatstone. De cette façon, de très petit changement de résistance est converti en un signal électrique utilisable. Les composants passifs tels que les résistances et des fils dépendant de la température sont utilisés pour compenser et calibrer le signal de sortie du pont. La principale raison de la défaillance du transducteur est la force de surcharge qui doit être correctement sélectionnée. La plage de valeurs de masse pour laquelle le résultat n'est pas affectée par une autre erreur de limite est appelée plage de mesure. Deux autres paramètres qui assure le bon fonctionnement du capteur de force sont la limite de charge/couple de sécurité (la charge/couple pouvant être appliqué sans produire de décalage permanent) et la charge latérale sûre (c'est-à-dire la charge maximale qui ne peut pas agir à 90° par rapport à l'axe le long du capteur et ne peut produire un changement permanent dans la performance au-delà de ce qui est spécifié).

— Capteurs tactiles

Les capteurs tactiles et les détecteurs de contact sont des appareils qui mesurent les paramètres d'un contact entre le capteur et un objet, confiné dans une petite région bien définie. Ceci est en contraste avec un capteur de force qui mesure les forces totales qui sont appliquées à un objet. La détection de contact est la détection et la mesure d'une

force de contact à un point (qui peut aussi être binaire, c'est-à-dire tactile, sans toucher). La détection tactile est la mesure de la distribution spatiale de forces perpendiculaires à une zone sensorielle définie, suivie d'une interprétation de cette répartition des forces. Un réseau de détection tactile est obtenu par la coordination d'un groupe de capteurs tactiles. Le glissement est la détection et la mesure du mouvement d'un objet par rapport à un capteur. Cela peut être fait soit par un capteur de glissement spécialement conçu, soit par interprétation des données obtenues par un capteur tactile ou un réseau tactile. Quelques propriétés recherchées de capteurs tactiles sont les suivants :

- Une bonne zone sensorielle (idéalement un seul point de contact)
- Une bonne sensibilité du capteur selon l'application concernée (sensibilité typique plage étant de 0,4 à 10 N)
- Une bande passante minimale du capteur de 100 Hz
- Caractéristiques du capteur stables et reproductibles avec une faible hystérésis
- Robustesse et protection contre les grands changements environnementaux

Une autre façon de construire des capteurs tactile incluent des capteurs à base de fibre optique (utilisant la micro courbure à l'état interne des fibres optiques), capteurs piézoélectriques (utilisant des matériaux polymères tels que le fluorure de polyvinylidene (PVDF)), à base de silicium (le silicium possède une résistance à la traction comparable à l'acier, et élastique à point de rupture avec une très faible hystérésis mécanique), et tactile à base de capteur à jauge de contrainte. La figure (1.9) montre le capteur tactile de Shadow monté sur les doigts d'une main robotique, la sensibilité des éléments tactiles du capteur de 0,1 à 25 N, convient à la fois pour les manipulations délicates et les prises de force

— **Le GPS (Global Positioning System)**

Le GPS est un système spatial de positionnement radio et de transfert de temps. Les satellites GPS transmettent des signaux à l'équipement approprié au sol. Ces signaux fournissent des informations précises sur la position, la vitesse et le temps à un nombre illimité d'utilisateurs sur terre, mer, dans l'air et dans l'espace. Les récepteurs GPS ont besoin d'une vue dégagée sur le ciel, de sorte qu'ils ne peuvent être utilisés qu'à l'extérieur, et ils ne fonctionnent généralement pas bien à proximité des grands immeubles ou dans



FIGURE 1.9 – Capteur tactile monté sur un bras manipulateur [1]

des zones forestières. Les trois parties primaires des GPS sont les segments d'espace, le contrôle et l'utilisateur. Chaque satellite transmet des données en continu indiquant son emplacement et l'heure actuelle, et tous les signaux satellites sont envoyés en même temps (transition synchrone). Le récepteur GPS dispose d'une horloge à quartz, mais bien que trois satellites puissent donner la position 3-D, le quatrième satellite pour la correction du temps [67]. En connaissant sa distance par rapport à un satellite, chaque récepteur "sait" également qu'il se trouve quelque part à la surface d'une sphère imaginaire centrée sur le satellite. Par la détermination des rayons des sphères, une pour chaque satellite, le récepteur trouve son emplacement au point d'intersection de ces sphères. Le segment spatial implique une constellation minimale de 24 satellites. Le contrôle segment consiste en un réseau d'installations de surveillance et de contrôle (stations principales, stations de surveillance, stations de téléchargement) qui sont utilisées pour gérer la constellation de satellites et pour mettre à jour les messages de données de navigation par satellite. Le segment des utilisateurs implique tous les récepteurs de radionavigation qui peuvent recevoir, décoder et traiter les Codes de télémétrie des satellites GPS et messages de données de navigation. Étant donné que les satellites GPS ne sont qu'une source d'informations, la résolution de localisation qu'ils fournissent dépend fortement sur les stratégies employées. La stratégie de base appelée stratégie pseudo range donne une résolution de 15 m et peut être améliorée à 1 m si un deuxième récepteur, qui est statique et à une position exacte connue, est employé.

Cette technique est dite différentielle GPS ou GPS bi-fréquence. En général, les paramètres de performance de base qui sont utilisés pour comparer différents récepteurs GPS sont les suivants : la précision de positionnement la précision de vitesse ,la précision de l'heure et le délai avant le premier échec.

1.5 La perception visuelle

La vision en robotique est la capacité d'un robot à extraire des informations à partir des images, et de les interpréter. La vision robotique utilise des capteurs optiques ou visuels (caméras) et équipement électronique approprié pour traiter/analyser les images visuelles et reconnaître les objets dans chaque application. L'image ou l'image électronique est un ensemble de pixels qui a été numérisé dans la mémoire d'un ordinateur. Un nombre binaire est stocké dans chaque pixel pour représenter l'intensité et la longueur d'onde de la lumière tombant sur cette partie de l'image. Un pixel, qui est un vecteur de nombres binaires, représente une couleur particulière. Une image numérique est une matrice de pixels (vecteurs) remplie de couleurs correspondantes crée une image de la scène. Il est évident que les caméras convertissent en fait la réalité 2D ou 3D en représentations 2D, bien que via le post-traitement, ils puissent reconstruire des représentations 3D à partir des premières images 2D [68]. Les caméras ne sont pas des appareils sans erreur. D'une manière générale, les distorsions de la caméra se distinguent comme suit :

- Géométriques : représentation des objets se déplaçant sur le plan du capteur.
- Radiométriques : erreurs dans les "valeurs de luminosité" des pixels dues aux variations de la sensibilité des pixels.
- Spectrales :erreurs dans les valeurs de luminosité des pixels en raison de la réponse variable du capteur à différentes longueurs d'onde de lumière.

Le principe physique qui convertie une scène en une image plane : est un échantillonnage spatial de l'information et un échantillonnage temporel (flux vidéo). L'échantillonnage spatial est réalisé par des photos cites (pixel) de forme carré en général, chaque photo sites va capter de la lumière pendant un certain temps et la transforme en charges.

1.5.1 Classes des systèmes de vision

Les systèmes de vision peuvent être classés en systèmes à stéréo-vision et en systèmes à vision monoculaire

1.5.1.1 La stéréovision

Un grand nombre de systèmes de vision, en robotique et dans la nature, utilisent la stéréovision. Cela consiste à mettre en correspondance deux images ou plus, issues de capteurs visuels distants de quelques centimètres, afin d'extraire une information de distance. Sachant que les systèmes de stéréo-vision sont plus chers et plus complexes à mettre en place vu qu'ils nécessitent plusieurs caméras, bien calibrées et bien positionnées.

1.5.1.2 La vision monoculaire

La nature nous montre qu'il est tout à fait possible de se déplacer dans un environnement sans utiliser la stéréovision. Un animal ou un être humain borgne est capable de s'orienter dans un environnement inconnu. Beaucoup de problèmes en robotique ont été réglés et beaucoup de tâches ont été réalisées en utilisant des systèmes de vision monoculaire comme la navigation, la localisation, l'évitement d'obstacle et d'autres.

1.5.2 Technologies des caméras

Les caméras peuvent se distinguer en CCD et en CMOS. Le capteur CCD commence à disparaître sur le marché au profit du capteur CMOS. Les deux capteurs ont le même principe physique de la conversion des photons en charges, la différence est dans la façon de convertir les puits de charges en tension [69].

1.5.2.1 Capteurs CCD

La technologie CCD est ancienne et tend à disparaître. Dans ce type de capteur, tous les pixels sont échantillonnés en même temps et la conversion des charges en tension est faite séquentiellement. Les charges sont transférées par des registres à décalages analogiques

par conséquent, pour accéder à un pixel donné il faut décaler tout d'abord tous les pixels qui le précèdent. Ce type de capteurs consomme beaucoup d'énergie.

1.5.2.2 Capteurs CMOS

La différence réside dans la façon de convertir les charges en tension .Avec la caméra CCD, les charges sont transférées puis converties séquentiellement en tensions, dans la technologie CMOS(Complementary Metal Oxide Semi-conductor) , les charges sont converties en tension directement au niveau du pixel grâce à un réseau de transistors CMOS, ainsi il est possible d'accéder directement à la tension de chaque pixel grâce à un système d'adressage, ce qui permet la réalisation de pré-traitement au niveau de chaque pixel. Parmi les avantages de l'utilisation d'un capteur CMOS ; La réduction des coûts et une faible consommation d'énergie (100 fois moins que la technologie CCD).Ce type de caméra permet l'élimination d'effet Smear dans les images et la possibilité de réaliser des fonctions évoluées sur le même substrat : comme le filtrage,la génération d'adresse, la conversion analogiques numériques. Les images prises par le type de caméra est caractérisé par un facteur de remplissage moyen et cela implique une perte d'information et un bruit sur les pixels.

1.5.2.3 Autres capteurs particulier

Il existe d'autre capteurs spéciaux de vision tel que

La vision chip caméra qui est un capteur spécial mais qui tend à disparaître car il est difficile à programmer. Dans ce capteur chaque pixel embarque une CPU et un processeur de signal 2D parallèle.

Capteur plénoptique est un capteur visuel spécial couvert de micro-lentilles avec une grande profondeur de champs, lui permettant des acquisitions à grande ouverture donc à faible lumière .La mise au point peut être logicielle donc la correction logicielle d'aberrations et l'obtention l'image 3D sont possibles.

Conclusion

Dans ce chapitre, nous avons exposé des généralités sur les robots mobiles. Une classe de robots qui peuvent se déplacer sur le sol, dans l'eau ou dans les airs. Cependant, ces robots mobiles sont très similaires en termes de ce qu'ils font et comment ils le font. Un robot mobile doit être équipé de moyen de perception lui permettant à recueillir des informations qui lui sont utile pour agir et réagir dans son environnement et nous avons donné quelques généralités sur les capteurs utilisés en robotique.

Chapitre 2

Modélisation cinématique et dynamique d'un robot mobile

Introduction

En robotique mobile agir sur les actionneurs c'est permettre au robot mobile d'atteindre tout point de l'espace à l'intérieur de son environnement de travail. Pour ce faire le concepteur étudie la cinématique des robots mobiles qui est donnée sous forme de configuration des robots mobiles dans leur espace de travail, des relations entre ses paramètres géométriques, et les contraintes imposées dans leurs trajectoires. Généralement la synthèse des équations cinématiques appropriées au robot mobile dépend de la structure géométrique de ce dernier. Par ailleurs un robot mobile peut avoir deux ou plusieurs roues avec ou sans contrainte de mouvement. Le développement de nouvelles structures cinématiques robotiques spécialisées est toujours d'actualité, afin de construire des robots qui peuvent effectuer plus de tâches sophistiquées et complexes dans des applications industrielles et sociétales [70, 71]. Dans ce chapitre nous allons présenter les concepts analytiques fondamentaux nécessaires à l'étude de la cinématique et de la dynamique des robots mobiles, en particulier le robot mobile différentiel non holonome. Un robot mobile à roues est capable de se déplacer sur une surface uniquement par l'actionnement de l'ensemble de roues montées sur le robot en contact avec la surface. Une roue est un dispositif qui fournit ou permet un mouvement relatif entre son support et une surface sur laquelle il est destiné à avoir un seul point de contact. Nous introduisons les hypothèses pratiques suivantes pour

qu'on puisse traiter le problème de modélisation[71].

1. Le robot mobile est considéré comme un objet solide et ne contient pas de parties flexibles.
2. Il n'y a pas de biellette de direction.
3. Tous les axes de direction sont perpendiculaires à la surface.
4. Le Robot mobile à roue se déplace sur une surface plane.

2.1 Modélisation cinématique d'un robot mobile

2.1.1 Robot à deux roues à entraînement différentiel

Le modèle cinématique approprié au robot mobile à deux roues fixes et à entraînement différentiel et une roue pivotante passive est l'un des modèles les plus populaires. Ce type de robot peut être classé comme robot mobile de type (2.0) et son nombre de roues peut être étendu à quatre en ajoutant une deuxième roue pivotante passive. Les principaux avantages de ce robot sont : une structure mécanique simple, un modèle cinématique simple et un faible coût de fabrication. Le rayon de braquage nul est possible. Pour un robot cylindrique, l'espace libre d'obstacles peut facilement être calculé en étendant les limites des obstacles par le rayon du robot. Enfin les erreurs systématiques sont faciles à calibrer. En revanche, il a aussi des inconvénients comme, la difficulté de déplacer sur des surfaces irrégulières. Lorsque le robot passe sur des surfaces irrégulières, son orientation peut changer brusquement si une des roues actives perd le contact avec le sol. Et un seul mouvement bidirectionnel est disponible [72]. La synthèse d'une loi de commande nécessite au préalable de modéliser le système que nous désirons commander. Dans cette section nous présentons la géométrie du système, les différents repères associés au robot ainsi que les matrices de passage homogènes associées. Nous introduisons une définition des repères du robot et de la caméra.

2.1.2 Définition des repères

La modélisation du robot s'appuie sur différents repères figure (2.1) définis comme suit :

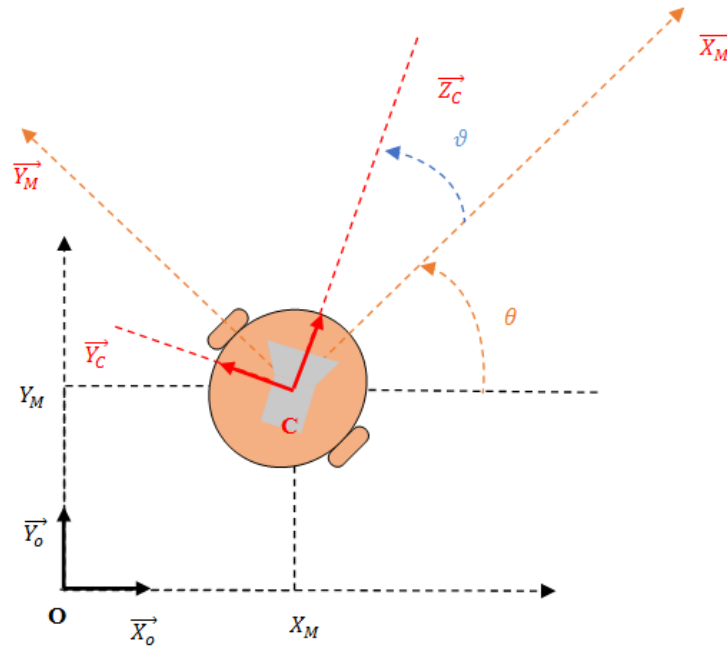


FIGURE 2.1 – Schéma représentant les différents repères associés à un robot mobile et à la caméra

- $R_O(O, \vec{O}_x, \vec{O}_y, \vec{O}_z)$: repère lié à l'environnement
- $R_M(M, \vec{M}_x, \vec{M}_y, \vec{M}_z)$: repère lié au robot mobile
- $R_C(C, \vec{C}_x, \vec{C}_y, \vec{C}_z)$: repère lié à la caméra

La position du robot mobile est représentée par les coordonnées (X_M, Y_M) du point M dans le repère R_O tandis que l'orientation est donnée par l'angle $\theta = (\vec{X}_M \vec{O}_M)$. La position de la caméra dans le repère R_O est la même que celle du robot puisque l'axe de rotation de la caméra est fixe sur le centre du robot, par contre son orientation est donnée par l'angle $\vartheta = (\vec{X}_M \vec{Z}_C)$.

2.1.3 Les matrices de passage homogènes, le vecteur d'état et le vecteur de commande

Les matrices de passages homogènes entre les différents repères liés au système robotique sont définies comme suit :

$$H_{(R_o/R_M)} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & X_M \\ \sin\theta & \cos\theta & 0 & Y_M \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$H_{(R_M/R_C)} = \begin{bmatrix} \cos\vartheta & -\sin\vartheta & 0 & X_M \\ \sin\vartheta & \cos\vartheta & 0 & Y_M \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$H_{(R_o/R_M)}$ est la matrice de passage du repère lié au robot mobile vers le repère lié à l'environnement de travail.

$H_{(R_M/R_C)}$ est la matrice de passage du repère lié à la caméra vers le repère lié au robot mobile.

Le vecteur d'état du robot à l'instant t peut être défini comme étant :

$$x(t) = [X_M(t) \quad Y_M(t) \quad \theta(t) \quad \vartheta(t)] \quad (2.3)$$

Ce vecteur permet de caractériser de manière unique la configuration du système à chaque instant [73]. Nous introduisons le vecteur d'état de la caméra de la manière suivante

$$x_C(t) = [X_C(t) \quad Y_C(t) \quad \theta_T(t)] \quad (2.4)$$

$X_C(t)$ et $Y_C(t)$ sont les coordonnées à l'instant t de la caméra dans le repère R_O et $\theta_T(t) = \theta(t) + \vartheta(t)$ est son angle d'orientation dans le même repère.

Le but dans tout système de commande est d'élaborer un vecteur de commande permettant d'atteindre les performances désirées. Ce vecteur de commande dans le cas de contrôle d'un système robotique mobile par asservissement visuel, est le vecteur de commande en vitesse, déduit de la dérivée du vecteur suivant :

$$q(t) = [l(t) \quad \theta(t) \quad \vartheta(t)] \quad (2.5)$$

$l(t)$ Représente l'abscisse curviligne du point M par rapport au repère R_O

La dérivée de $q(t)$ par rapport au temps donne :

$$\dot{q}(t) = [v(t) \quad \omega(t) \quad \varpi(t)] \quad (2.6)$$

$v(t)$ et $\omega(t)$ sont les vitesses linéaire et angulaire de la base mobile respectivement et $\varpi(t)$ correspond à la vitesse de rotation de la caméra par rapport au robot, $\dot{q}(t)$ représente le vecteur de commande puisque la commande doit être en vitesse.

2.1.4 Le torseur cinématique de la caméra

La synthèse d'un correcteur réalisant un asservissement visuel nécessite de connaître le lien entre le mouvement de la caméra et le vecteur commande. Le torseur cinématique de la caméra noté $T_{(C/R_o)}^{R_c}$, par rapport au repère R_o exprimé dans le repère R_c [74, 75] est défini comme suit :

$$T_{(C/R_o)}^{R_c} = [V_{(C/R_o)}^{R_c} \quad \Omega_{(R_c/R_o)}^{R_c}]^T \quad (2.7)$$

$$V_{(C/R_o)}^{R_c}{}^T = [V_{x_C}^{R_c} \quad V_{y_C}^{R_c} \quad V_{z_C}^{R_c}]^T \quad (2.8)$$

$$\Omega_{(C/R_o)}^{R_c}{}^T = [\Omega_{x_C}^{R_c} \quad \Omega_{y_C}^{R_c} \quad \Omega_{z_C}^{R_c}]^T \quad (2.9)$$

$V_{(C/R_o)}^{R_c}{}^T$, $\Omega_{(C/R_o)}^{R_c}{}^T$ désignent respectivement les vitesses linéaire et angulaire de la caméra par rapport au repère de la scène R_o

2.1.5 Modèle cinématique d'un robot mobile

Un robot est défini comme un robot uni-cycle lorsqu'il possède deux roues indépendantes et éventuellement des roues folles pour assurer sa stabilité. Les roues folles ne sont pas incluses car elles n'affectent pas la cinématique si elles sont correctement placées. Les équations cinématiques d'un robot se déplaçant à une vitesse linéaire v et une vitesse angulaire ω sont fournies par :

$$\begin{cases} \dot{x}(t) = v(t) \cos \theta(t) \\ \dot{y}(t) = v(t) \sin \theta(t) \\ \dot{\theta}(t) = \omega(t) \end{cases} \quad (2.10)$$

x et y indiquent les coordonnées du centre de gravité du robot et θ est l'angle d'orientation. Ce modèle a été utilisé dans plusieurs articles, notamment [15, 18, 76, 77, 78]. Le modèle d'actionnement donne les vitesses linéaire et angulaire du robot mobile en fonction des vitesses de ses roues motrices droite et gauche et de ses propriétés géométriques, ce modèle est exprimé comme suit :

$$\begin{cases} v(t) = \frac{v_r(t)+v_l(t)}{2} \\ \omega(t) = \frac{v_r(t)-v_l(t)}{D} \end{cases} \quad (2.11)$$

La distance entre les roues directrices est notée par la lettre D . v_r et v_l sont les vitesses de la roue droite et gauche respectivement, représentant les entrées du modèle cinématique [16].

Au temps discret $t = nT_0$; $x(t), y(t), v(t)$, and $\omega(t)$ seront respectivement désignés par x_n, y_n, v_n and ω_n [22, 76, 77] Il découle des équations cinématiques que

$$\begin{cases} x_{n+1} = x_n + \int_{nT_0}^{(n+1)T_0} v(t) \cos \theta dt \\ y_{n+1} = y_n + \int_{nT_0}^{(n+1)T_0} v(t) \sin \theta dt \\ \theta_{n+1} = \theta_n + \int_{nT_0}^{(n+1)T_0} \omega(t) dt \end{cases} \quad (2.12)$$

Les équations suivantes sont générées à l'aide de l'approximation d'Euler du modèle cinématique du robot mobile [76]. L'utilisation d'équations à temps discret permet l'implémentation du système de commande en temps réel.

$$\begin{cases} x_{n+1} = x_n + T_0 V_n \cos(\theta) \\ y_{n+1} = y_n + T_0 V_n \sin(\theta) \\ \theta_{n+1} = \theta_n + T_0 \omega_n \end{cases} \quad (2.13)$$

2.1.6 Modélisation de la caméra : le modèle sténopé (pinhole)

Le modèle sténopé est schématisé par la figure (2.2).

Le point 3D P situé à une profondeur Z et à une distance X de l'axe optique sera projeté en un point p tel que présenté sur la figure (2.3). Le modèle sténopé est obtenu tout simplement en appliquant le théorème de Thalès.

$$x = \frac{f}{Z} X \quad (2.14)$$

En 3D le point (X, Y, Z) est projeté dans le plan image tel que illustré par la figure (2.4).

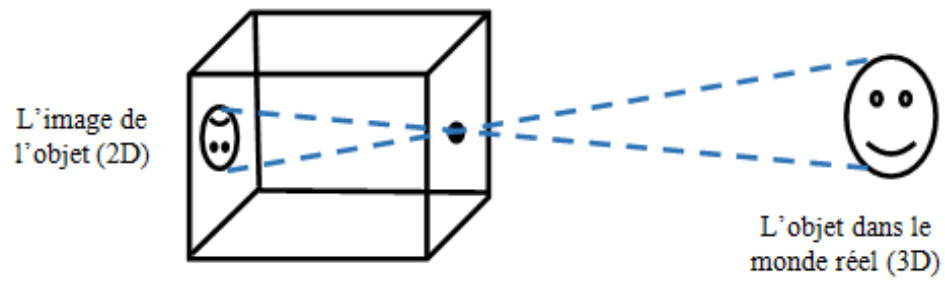


FIGURE 2.2 – Schématisation du modèle sténopé

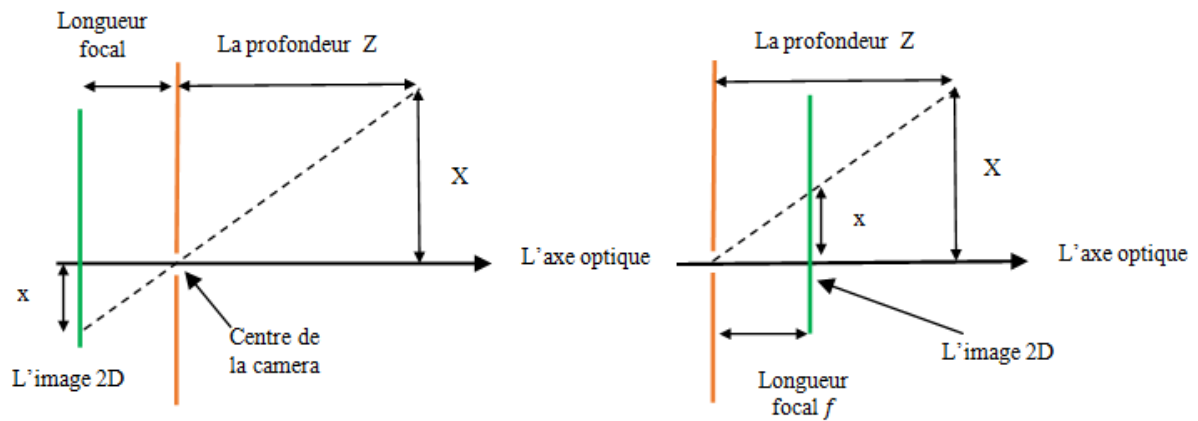


FIGURE 2.3 – Schémas explicatifs pour l'obtention du modèle sténopé

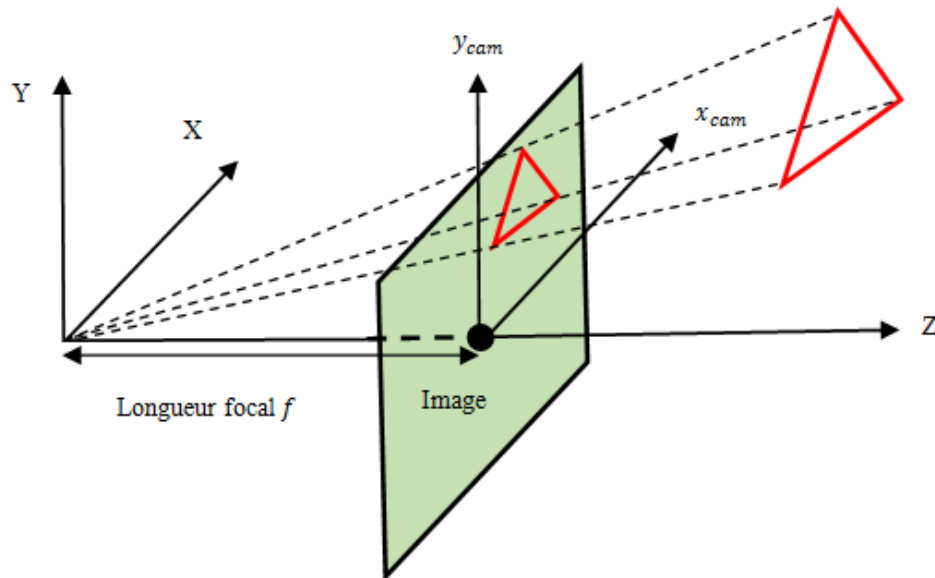


FIGURE 2.4 – Projection d'un point 3D dans le plan image

Cette projection peut s'exprimer en utilisant les coordonnées homogènes, de telle sorte à avoir une expression telle que :

$$x = PX \quad (2.15)$$

Le point (X, Y, Z) est projeté en image plane en (x_{cam}, y_{cam}) avec

$$\begin{cases} x_{cam} = \frac{f}{Z}X \\ y_{cam} = \frac{f}{Z}Y \end{cases} \quad (2.16)$$

Nous pouvons l'écrire :

$$\lambda \begin{bmatrix} x_{cam} \\ y_{cam} \\ f \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.17)$$

Où $\lambda = \frac{Z}{f}$

En utilisant les coordonnées homogènes, il peut être exprimé comme une transformation

linéaire

$$\lambda \begin{bmatrix} x_{cam} \\ y_{cam} \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.18)$$

Les points de caméra sont liés aux coordonnées de pixels dans une image

Dans le cas où les pixels ont la forme carré

$$\begin{cases} x = -f x_{cam} + x_0 \\ y = -f y_{cam} + y_0 \end{cases} \quad (2.19)$$

Dans le cas où les pixels n'ont pas la forme carré

$$\begin{cases} x = -k_x x_{cam} + x_0 \\ y = -k_y y_{cam} + y_0 \end{cases} \quad (2.20)$$

k_x et k_y sont les dimensions d'un pixel

En utilisant les coordonnées homogènes on obtient

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{f} \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{cam} \\ y_{cam} \\ f \end{bmatrix} \quad (2.21)$$

avec $\alpha_x = -fk_x$ et $\alpha_y = -fk_y$

2.2 Modélisation dynamique du robot mobile

Le modèle cinématique direct développé précédemment permet de calculer les évolutions des mouvements élémentaires du robot mobile, mais ne prend pas en charge les facteurs dominants qui caractérisent la dynamique du robot mobile. Cependant, pour tenir compte de ces facteurs il est important de considérer l'influence de la masse, l'inertie et le frottement qui sont présents dans tous les systèmes mécanique réel tel que le robot mobile. Par ailleurs la dynamique des robots mobiles est un domaine difficile et a attiré considérablement l'attention des chercheurs et des ingénieurs au fil des années [79, 80].

En outre, la plupart des robots mobiles, exploités pratiquement utilisent des roues conventionnelles et sont soumis à des contraintes nécessitant un traitement particulier. Des problèmes délicats de stabilité et de contrôle, auxquelles il faut souvent faire face dans la

conception d'un robot mobile, sont dus à l'existence de glissement longitudinal et latéral dans le mouvement du robot mobile à roues [81]. Dans cette partie nous Présenterons les concepts et des techniques générales de modélisation dynamique des robots. Nous allons étudier les modèles dynamiques de Newton Euler et Lagrange des robots mobiles à entraînement différentiel. Nous allons aussi étudier la dynamique des robots mobiles à entraînement différentiel avec des mouvements longitudinaux et latéraux. La modélisation dynamique d'un robot est réalisée par la dérivation des équations dynamiques du mouvement du robot. Cela peut être fait en utilisant les deux méthodologies suivantes :

- Méthode de Newton Euler
- Méthode de Lagrange

Comme la cinématique, la dynamique se distingue en :

- Dynamique directe fournit les équations dynamiques qui décrivent les réponses dynamiques du robot à des forces ou des couples exercés par les moteurs.
- Dynamique inverse fournit les forces ou couples nécessaires pour obtenir les trajectoires souhaitées. Dans le modèle dynamique inverse, les entrées sont les trajectoires souhaitées et les sorties sont des couples moteurs.

2.2.1 Modèle dynamique de Newton-Euler

Le modèle dynamique de Newton Euler est obtenu par l'application du théorème de Newton, du moment cinétique et du théorème d'Euler pour calculer numériquement des forces ou des couples généralisés. Pour ce faire considérons l'objet B_i de la figure (2.5) auquel une force totale f_i est appliquée en son centre de gravité (COG)[1].

Le mouvement de translation du corps solide B_i est décrit par cette relation :

$$\frac{dE_i}{dt} = f_i \quad (2.22)$$

E_i est le mouvement linéaire donnée par :

$$E_i = m_i \dot{s}_i \quad (2.23)$$

m_i est la masse du corps et s_i est la position du COG par rapport au repère $Oxyz$. En supposant que m_i est constant, alors les équations (2.22) et (2.23) donnent le modèle

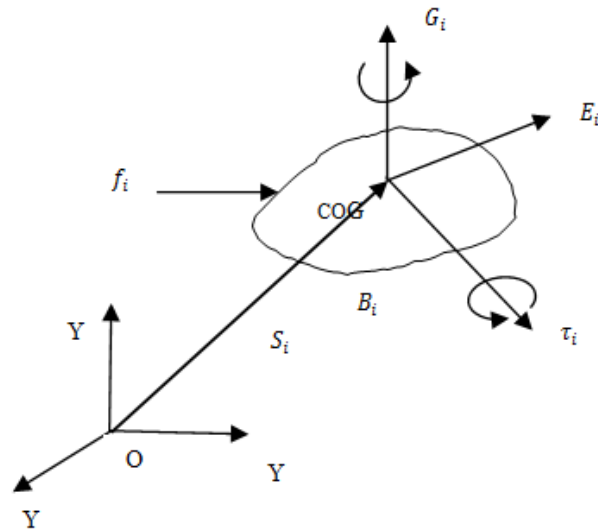


FIGURE 2.5 – Un corps solide B_i dans le repère des coordonnées inertielles $Oxyz$

dynamique de translation suivant.

$$m_i \ddot{s}_i = F_i \quad (2.24)$$

Le mouvement de rotation de l'objet B_i est décrit par :

$$\frac{dG_i}{dt} = \tau_i \quad (2.25)$$

G_i est le moment cinétique total de l'objet B_i par rapport à COG, et τ_i est le couple externe total qui produit le mouvement de rotation du corps. Le moment total G_i est donnée par :

$$G_i = I_i \omega_i \quad (2.26)$$

I_i est le tenseur d'inertie donné par l'intégrale de volume suivantes :

$$I_i = \int_{V_i} [r^T r I_3 - r r^T] \rho_i dV \quad (2.27)$$

ρ_i est la masse volumique de B_i , dV est le volume d'un élément infiniment petit de B_i située à la position r par rapport au COG, ω_i est le vecteur de vitesse angulaire autour de l'axe d'inertie passant par le COG, I_3 est la matrice unitaire 3x3, et V_i est le volume de l'objet B_i .

2.2.2 Modèle dynamique de Lagrange

Le modèle dynamique général de Lagrange d'un corps solide (robot multi bras, robot mobile à roues, etc) est décrit par :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \tau \quad (2.28)$$

$$q = [q_1, q_2, \dots, q_n]^T$$

Où q_i est la variable du i ème degré de liberté, τ est le vecteur de forces ou couples généralisés appliqués au corps (c'est-à-dire la force pour le mouvement de translation et le couple pour le mouvement de rotation), et L est la fonction lagrangienne définie par :

$$L = K - P \quad (2.29)$$

K est l'énergie cinétique totale et P l'énergie potentielle totale du corps .

$$K = K_1 + K_2 + \dots + K_n$$

$$P = P_1 + P_2 + \dots + P_n$$

Où K_i est l'énergie cinétique de liaison (degré de liberté) i , et P_i son énergie potentiel. L'énergie cinétique K du corps est égale à :

$$K = \frac{1}{2}m\dot{s}^T\dot{s} + \frac{1}{2}\omega^T I \omega \quad (2.30)$$

Où s est la vitesse linéaire du COG, ω est la vitesse angulaire de la rotation, m est la masse, et I le tenseur d'inertie du corps.

2.2.3 Modélisation dynamique de robots non holonomes

Le modèle dynamique de Lagrange d'un robot non holonome (fixe ou mobile) a pour forme :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + M^T(q)\lambda = E\tau \quad (2.31)$$

Où $M(q)$ est la matrice des m contraintes non holonomes de dimensions $m \times n$ et λ est le vecteur multiplicateur de Lagrange. Ce modèle conduit à :

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + M^T(q)\lambda = E\tau \quad (2.32)$$

Où E est une matrice de transformation non singulière.

Pour éliminer le terme de contrainte $M^T(q)$ Dans l'équation (2.32) et obtenir un modèle sans contrainte nous utilisons une matrice $n \times (n - m)$

$$B^T(q)M^T(q) = 0 \quad (2.33)$$

À partir des équations (2.31) et (2.33) , on peut vérifier qu'il existe un vecteur $v(t)$ de dimension $(n - m)$ tel que :

$$q(t) = B(q)v(t) \quad (2.34)$$

Maintenant, en multipliant l'équation (2.30) par $B^T(q)$ et en utilisant les équations (2.31), (2.33) et (2.34) on obtient :

$$\bar{D}(q)\dot{v} + \bar{C}(q, \dot{q})v + \bar{g}(q) = \bar{E}\tau \quad (2.35)$$

Où

$$\bar{D} = B^T D B$$

$$\bar{C} = B^T D \dot{B} + B^T C B$$

$$\bar{g} = B^T g$$

$$\bar{E} = B^T E$$

Le modèle réduit (sans contrainte) (2.35), décrit l'évolution dynamique du vecteur $q(t)$ à n dimensions en termes d'évolution dynamique du vecteur $v(t)$ de dimension $(n - m)$.

2.2.4 Le modèle dynamique d'un robot mobile à roue à entraînement différentiel

Le modèle dynamique d'un robot mobile à roue à entraînement différentiel sera présenté par la méthode de Newton Euler et la méthode de Lagrange.

2.2.4.1 Modèle dynamique d'un robot mobile à entraînement différentiel par Newton Euler

Nous utiliserons les équations de Newton Euler :

$$m\dot{v} = F \text{ (Mouvement de translation)} \quad (2.36)$$

$$I\dot{\omega} = N \text{ (Mouvement de rotation)} \quad (2.37)$$

F est la force totale appliquée au COG, N est le couple total par rapport au COG, m est la masse du robot, et I est l'inertie du robot. En supposant que le COG coïncide avec le milieu de l'entraxe distance entre les deux roues, on trouve :

$$F = F_r + F_l \quad (2.38)$$

sachant que

$$\tau_r = rF_r \text{ et } \tau_l = rF_l$$

$$F = \frac{1}{r}\tau_r + \tau_l \quad (2.39)$$

F_r et F_l Sont les forces qui produisent respectivement les couples τ_r et τ_l

Nous avons aussi :

$$N = (F_r - F_l)2a = \frac{a}{r}(\tau_r - \tau_l) \quad (2.40)$$

Par conséquent les équations (2.36) et (2.37) donnent le modèle dynamique du robot mobile à roues :

$$\begin{cases} \dot{v} = \frac{1}{mr}(\tau_r + \tau_l) \\ \dot{\omega} = \frac{2a}{Ir}(\tau_r - \tau_l) \end{cases} \quad (2.41)$$

2.2.4.2 Modèle dynamique d'un robot mobile à entraînement différentiel par la méthode de Lagrange

Nous supposons encore que le centre de gravité COG coïncide avec le milieu de l'entraxe (distance entre les deux roues), la matrice de contraintes non holonome donne

$$M(q) = [-\sin\phi \quad \cos\phi \quad 0] \quad (2.42)$$

Puisque le robot se déplace sur un plan horizontal, les termes $C(q, \dot{q})$ et $g(q)$ dans l'équation (2.32) sont nuls. Par conséquent, le modèle (2.32) devient :

$$D(q)\ddot{q} + M^T(q)\lambda = E\tau \quad (2.43)$$

Où

$$q = \begin{bmatrix} x_Q \\ y_Q \\ \phi \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}$$

$$D(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}$$

$$q = \begin{bmatrix} \cos\phi & \cos\phi \\ \sin\phi & \sin\phi \\ 2a & -2a \end{bmatrix}$$

Pour convertir le modèle (2.43) en modèle non contraint , nous avons besoin de la matrice $B(q)$ dans (2.33), Cette matrice est :

$$B(q) = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix}$$

Qui satisfait l'équation (2.43) Par conséquent

$$\bar{D} = B^T D B = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}$$

$$\bar{E} = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ 2a & -2a \end{bmatrix}$$

Et le modèle (2.35) donne :

$$\begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ 2a & -2a \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (2.44)$$

Notant que v_1 est la vitesse de translation v et v_2 est la vitesse angulaire ω du robot, l'équation (2.44) donne le modèle :

$$\begin{cases} \dot{v} = \frac{1}{mr}(\tau_r + \tau_l) \\ \dot{\omega} = \frac{2a}{I r}(\tau_r - \tau_l) \end{cases} \quad (2.45)$$

Qui est identique à l'équation (2.41), comme prévu. Enfin, en utilisant l'équation (2.35) on obtient

$$\begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi} \end{bmatrix} \begin{bmatrix} \cos & 0 \\ \sin & 0 \\ 0 & 1 \end{bmatrix} \frac{1}{r} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v \cos \phi \\ v \sin \phi \\ \omega \end{bmatrix} \quad (2.46)$$

L'équation (2.46) donne le modèle cinématique du robot. Les équations dynamiques et cinématiques (2.44), (2.45) et (2.46) décrivent complètement le mouvement du robot mobile à roues à entraînement différentiel.

2.3 L'holonomie :

Un robot mobile est dit holonome lorsque le nombre de degrés de liberté contrôlés est égal au nombre total de degrés de liberté. Un robot se déplaçant sur un plan possède trois degrés de liberté (deux translations et une rotation). Une plate-forme holonome doit être capable d'avancer, de se déplacer latéralement et de tourner sur elle-même. Avec ce type de plate-forme tous les mouvements imaginés sont réalisables, cette capacité rend la gestion du robot incroyablement simple (cas des robots omnidirectionnels). Les robots non holonomes ont une complexité mécanique inférieure à celle des robots holonomes, mais cela a un coût. Des algorithmes plus complexes sont utilisés pour la navigation, pour la commande et la planification de trajectoire. Le nombre de mouvements indépendants qu'un robot mobile peut effectuer par rapport à un système de coordonnées particulier est défini comme le nombre de degrés de liberté.

2.3.1 Contraintes non holonomes

La contrainte non holonomes est définie comme une contrainte qui contient des dérivées temporelles des coordonnées généralisées (variables) d'un système et n'est pas intégrable [1]. Pour comprendre ce que cela signifie, nous définissons d'abord une contrainte holonomes comme toute contrainte pouvant être exprimée sous la forme.

$$F(q, t) = 0 \quad (2.47)$$

$q = [q_1, q_2, \dots, q_n]^T$ est le vecteur de coordonnées généralisées.

On suppose la contrainte de forme

$$f(q, \dot{q}, t) = 0 \quad (2.48)$$

Si cette contrainte peut être convertie sous la forme suivante :

$$F(q, t) = 0 \quad (2.49)$$

On dit qu'elle est intégrable

2.3.2 Définition de la contrainte non holonome

Une contrainte de la forme de l'équation (2.48) est dite non holonome si elle ne peut être rendue sous la forme (2.49) telle que pour ne faire intervenir que les variables généralisées. Les systèmes qui sont soumis à des contraintes non holonomes (et donc sont appelés systèmes non holonomes) sont des robots sous-actionnés. Il est souligné que « holonome » ne signifie pas nécessairement sans contrainte. Sûrement, un robot mobile sans contrainte est holonome. Mais un robot mobile capable uniquement de faire une translation est aussi holonome.

2.3.3 Définition de la contrainte Pfaffian

La contrainte non-holonome est appelée la contrainte de Pfaffian si elle est linéaire dans \dot{q} et sera exprimé sous la forme

$$\mu_i(q, \dot{q}) = 0 \quad (2.50)$$

$i = 1, 2, \dots, r$

Où μ_i sont des vecteurs linéairement indépendants et $q = [q_1, q_2, \dots, q_n]^T$ sous forme de matrice compacte, les contraintes Pfaffian ci-dessus peuvent être écrites comme suit :

$$M(q)\dot{q} = 0 \quad (2.51)$$

Avec

$$M(q) = \begin{bmatrix} \mu_1(q) \\ \mu_2(q) \\ \vdots \\ \mu_r(q) \end{bmatrix}$$

La contrainte non holonome en robotique mobile est la contrainte de mouvement d'un disque qui roule sur un plan sans glissement. La condition appropriée sans glissement ne permet pas aux vitesses généralisées \dot{x} , \dot{y} et $\dot{\theta}$ de prendre des valeurs arbitraires.

Soit r le rayon du disque. En raison de la condition de non-glissement, les coordonnées généralisées sont contraintes par les équations suivantes :

$$\dot{x} = r \dot{\theta} \cos\phi \quad (2.52)$$

$$\dot{y} = r \dot{\theta} \sin\phi \quad (2.53)$$

les équations précédentes ne sont pas intégrables. Ces contraintes expriment la condition que le vecteur des vitesses du centre du disque se trouve dans le plan médian du disque.

La suppression de l'expression de la vitesse v dans le système d'équations précédent nous permet d'écrire l'égalité suivante

$$v = r\dot{\theta} = \frac{\dot{x}}{\cos\phi} = \frac{\dot{y}}{\sin\phi} \quad (2.54)$$

Où

$$\dot{x}\sin\phi - \dot{y}\cos\phi = 0 \quad (2.55)$$

L'équation (2.55) représente la contrainte non holonome du mouvement d'un disque.

En raison de la contrainte cinématiques, le disque peut atteindre n'importe quelle configuration finale $(x_2, y_2, \phi_2, \theta_2)$ à partir de toute configuration initiale $(x_1, y_1, \phi_1, \theta_1)$. Cela peut se faire en deux étapes comme suit :

- *Étape 1* : Déplacez le point de contact (x_1, y_1) à (x_2, y_2) en roulant le disque le long d'une ligne de longueur $(2k\pi + \theta_2 - \theta_1)r$, $k = 0, 1, 2, \dots$
- *Étape 2* : Faites pivoter le disque autour de l'axe vertical de ϕ_1 à ϕ_2 .

Étant donné une contrainte cinématique, il faut déterminer si elle est intégrable ou pas. Cela peut être fait via le théorème de Frobenius qui utilise le concept de la géométrie différentielle de distributions.

Conclusion :

Beaucoup de travaux sont consacré au développement des stratégies de contrôle des robots mobiles à roues. L'aspect fondamental de la recherche dans ce domaine implique la manière dont la position et l'orientation du robot mobile à roues sont déduites. La plupart des chercheurs ont utilisé des modèles cinématiques pour accomplir cette tâche. Cependant, comme les robots mobiles à roues sont conçus pour effectuer des déplacements à des vitesses plus élevées, la modélisation dynamique devient de plus en plus importante. Dans ce chapitre nous avons présenté la modélisation cinématique et la modélisation dynamique des robots mobiles à roues. Ainsi le modèle de la caméra sachant que la synthèse d'un asservissement visuel nécessite de connaître le lien entre le mouvement de la caméra et le mouvement du robot ainsi que le vecteur commande du système robotique.

Chapitre 3

Commande par vision d'un robot mobile à roues

Introduction

La commande visuelle d'un robot ou l'asservissement visuel est une approche qui utilise un ou plusieurs capteurs de vision (caméras) pour contrôler le mouvement d'un robot. Plus précisément, les entrées de commande pour les moteurs du robot sont produites en traitant les données de l'image (l'extraction des contours, des caractéristiques, des angles, et d'autres primitives visuelles). David Marr définit la vision comme le processus qui consiste à découvrir à partir d'images ce qui est présent dans le monde et où cela se trouve [68]. Imaginer conduire une voiture avec la vitre avant couverte et regarder simplement le système de navigation GPS. Si vous aviez la route pour vous, vous pourriez probablement conduire avec beaucoup de succès bien que lentement. Cependant, s'il y avait d'autres voitures, des piétons, des feux de circulation ou des travaux routiers, vous seriez alors en difficulté. Pour faire face à cela, vous devez regarder vers l'extérieur et planifier vos actions en conséquence. Pour les humains, c'est facile, de le faire sans penser, mais il n'est pas encore facile de programmer une machine pour exécuter des tâches pareilles. C'est le défi de la vision en robotique. Une caméra est un dispositif passif qui capte les schémas d'énergie optique réfléchis par la scène. Notre propre expérience est que les yeux représentent un capteur très efficace pour la reconnaissance, la navigation et l'évitement d'obstacles. Avec la limitation à une seule caméra, ou à un seul œil, la structure tridimensionnelle de

la scène sera perdue dans l'image bidimensionnelle résultante. En dépit les êtres humains sont particulièrement doués pour déduire la nature tridimensionnelle d'une scène en utilisant plusieurs repères visuels. Les robots ne sont actuellement pas aussi bien développés. Une autre approche appelée stéréo vision exploite deux caméras ou plus, de telle manière à combiner les informations visuelles pour estimer la structure de la scène, il s'agit d'une technique utilisée par les êtres humains et les robots. L'utilisation des ordinateurs permet de traiter les images d'une ou plusieurs caméras et d'extraire des caractéristiques numériques. Par exemple, déterminer les coordonnées d'un objet dans la scène, ou jusqu'où un robot s'est déplacé en fonction de la façon dont le monde semble avoir bougé par rapport au robot. La vision par ordinateur peut regrouper plusieurs sous-domaines comme : La détection, le pré-traitement, la segmentation, la reconnaissance d'objet et l'interprétation. Dans le domaine de la robotique mobile, la tâche que l'opérateur désire confier au contrôleur de vision est transcrite sous forme de programme de telle manière à contrôler la position et l'orientation d'un robot mobile par rapport à certains points du repère. Cependant les délais de détection de la vision interviennent dans la phase de programmation de la tâche à réaliser et permettent de définir les critères et les contraintes de fonctionnement pour assurer la stabilité de suivi d'une trajectoire planifié au préalable. Ce chapitre sera consacré à la présentation des aspects fondamentaux de l'asservissement visuel. A la Planification de trajectoire d'un robot mobile avec évitement d'obstacles. Et à la Commande visuel d'un robot mobile.

3.1 Les sous domaine de la vision en robotique

La vision robotique regroupe les principaux sous-domaines suivants : Détection visuelle, Pré-traitement, Segmentation, Description, Reconnaissance et enfin Interprétation.

3.1.1 La détection visuelle

La détection visuelle représente le processus qui s'occupe de la production d'une image visuelle en poursuivant les étapes : Étalonnage de la caméra, Acquisition d'images, Illumi-

nation et Géométrie de l'image.

3.1.1.1 Étalonage de la caméra

L'étalonnage de la caméra, est une étape très nécessaire pour la détection visuelle, utilisée pour la correction des déplacements des points de l'image qui se produisent en raison des caractéristiques d'orientation intérieure de la caméra. La distorsion de l'objectif est l'une des causes du déplacement des points de l'image par rapport à leur position idéale sur le plan du capteur. On distingue deux méthodes d'étalonnage des caméras :

- Méthode basée sur le modèle : où les caractéristiques spécifiques dominantes de l'erreur sont modélisées et corrigées.
- Méthode basée sur la cartographie : où une cartographie appropriée est produite, sans qu'il soit nécessaire de comprendre le modèle d'erreur.

La distorsion radiale de l'objectif est un facteur contributif de base qui est modélisé. Le modèle généralement utilisé est un polynôme d'ordres 2, 3 ou 4. Les techniques des moindres carrés ou d'interpolation sont utilisées pour déterminer les valeurs des coefficients d'un meilleur modèle de l'erreur observée. Dans l'approche basée sur la cartographie, aucune tentative n'est faite pour comprendre les causes particulières de l'erreur le but est seulement d'avoir une correspondance explicite entre l'image et la réalité. En pratique, les deux approches de l'étalonnage de la caméra peuvent éliminer la plupart des erreurs de déplacement causées par des éléments d'orientation de la caméra intérieure, avant le pré-traitement de l'image.

3.1.1.2 Acquisition d'image

Les informations visuelles sont converties en signaux électriques à travers des capteurs visuels et des équipements électroniques associés. Lorsqu'une image est échantillonnée spatialement et quantifiés en amplitude, ces signaux visuels traduisent une image numérique. Les trois sujets de base inclus dans l'acquisition d'images sont les techniques d'imagerie, les effets de l'échantillonnage sur la résolution spatiale et les effets de la quantification d'amplitude sur la résolution d'intensité. Soit $f(x_i, y_i)$ une image 2D qui est considérée, pour

le traitement informatique, elle est numérisée spatialement et en amplitude (intensité). La numérisation des coordonnées spatiales x_i et y_i est connu sous le nom d'échantillonnage d'image, tandis que la numérisation d'amplitude est appelée intensité ou quantification du niveau de gris . La forme numérisée $f(x_i, y_i)$ (appelée image numérique) est représentée par une matrice de type

$$f(x_i, y_i) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, M - 1) \\ f(1, 0) & f(1, 1) & \dots & f(1, M - 1) \\ \vdots & & & \\ f(N - 1, 0) & f(N - 1, 1) & \dots & f(N - 1, M - 1) \end{bmatrix} \quad (3.1)$$

Chaque élément de cette matrice représente un pixel ou élément d'image. La valeur $f(x_i, y_i)$ représente l'intensité de ce point de l'image.

3.1.1.3 L'illumination

L'illumination (éclairage) d'une scène est un facteur critique qui affecte la complexité des algorithmes de vision. Un bon système d'éclairage illumine une scène de manière à minimiser la complexité de l'image produite et à améliorer (renforcer) les informations nécessaires à la détection et l'extraction d'objets. Les quatre techniques d'éclairage typique qui sont utilisées en robotique sont les suivantes :

1. Éclairage diffus : pour les objets à surface lisse et régulière.
2. Rétro-éclairage : pour les applications où les silhouettes des objets sont suffisantes pour la reconnaissance ou à d'autres fins.
3. Éclairage structuré : c'est-à-dire projection de points, de bandes ou de grilles sur la surface de travail, par exemple lorsqu'un bloc est éclairé par des plans lumineux parallèles qui deviennent des bandes lumineuses lorsqu'ils croisent une surface plane.
4. Éclairage directionnel : un faisceau lumineux ou un laser hautement dirigé est utilisé pour inspecter la surface d'un objet et détecter les défauts tels que les creux et les rayures.

3.1.1.4 Géométrie de l'image

La relation entre la position d'un point dans le plan image 2D et le point qui lui correspond dans le monde réel est déterminé par les lois de l'optique. Le fait que l'ouverture des objectifs de la caméra par laquelle la lumière tombe sur l'image est beaucoup plus petite que la taille des objets étudiés, on peut remplacer les lentilles par le modèle sténopé. Ainsi, des points de l'espace sont projetés sur le plan image par des lignes se coupant en un point commun appelé centre de projection. Dans le modèle de caméra sténopé, le centre de projection est situé au centre des lentilles. La caméra elle-même représente un corps rigide auquel un repère est attribué. Ce repère décrit la pose de la caméra. Le modèle sténopé de la caméra est donné dans la section (2.1.6).

3.1.2 Prétraitement

Le prétraitement d'images peut être effectué dans le domaine spatial ou fréquentiel. Les méthodes de prétraitement dans le domaine spatial utilisent directement les matrices des pixels de l'image. Et la fonction de prétraitement a la forme suivante :

$$p(x, y) = H(f(x, y))$$

Où $f(x, y)$ est l'image d'entrée, $H(\cdot)$ est un opérateur sur $f(x, y)$ et $p(x, y)$ est le résultat du pré-traitement. Dans le cas le plus simple, H a la forme d'une cartographie d'intensité I , à savoir

$$u = I(v)$$

u et v Désignent respectivement les intensités de $f(x, y)$, et $p(x, y)$.

Une autre technique typique de pré-traitement qui utilise des masques de convolution. Les méthodes de pré-traitement dans le domaine fréquentiel utilisent la transformée de Fourier d'une image qui convertit l'image en un agrégat de pixels à valeur complexe. Pour réduire le bruit et les autres effets parasites résultant des opérations d'échantillonnage, de quantification, de transmission, et de toute autre perturbation de l'environnement, des

opérations de lissage appropriées sont employées. Par exemple, La moyenne de voisinage, où l'image lissée est obtenue en faisant la moyenne des valeurs d'intensité des pixels contenus dans une zone prédéfinie. Le filtrage médian (où l'image lissée est obtenue en utilisant la médiane au lieu de la moyenne des pixels contenus dans la région environnante souhaitée) Les autres opérations de pré-traitement sont les suivantes :

- Amélioration de l'image : adaptation automatique aux variations d'illumination.
- Détection des contours : il s'agit de l'étape préliminaire pour de nombreux algorithmes de détection.
- Seuillage de l'image : c'est-à-dire la sélection d'un seuil S_i qui sépare les modes d'intensité, par exemple, dans des images dont les intensités sont regroupées en deux modèles dominants, à savoir, des objets clairs sur un fond sombre.

3.1.3 Segmentation d'images

La segmentation d'image est le processus qui divise une image source en parties ou objets constitutifs. Les régions d'intérêt sont sélectionnées sur la base de plusieurs critères. Par exemple, il peut être nécessaire de trouver une seule pièce dans un bac. Pour la navigation il peut être utile d'extraire uniquement les lignes de sol d'une image. En général, par segmentation, les objets sont extraits d'une scène pour être ensuite reconnus et analysés. Les deux principes de base utilisés dans les algorithmes de segmentation sont, la discontinuité pour la détection de contours et la Similarité pour, seuillage, croissance des régions. Ces processus peuvent être appliqués à des scènes statiques ou variables dans le temps.

3.1.4 Détection des contours

La détection des contours est basée sur les discontinuités d'intensité et permet d'identifier les pixels qui se trouvent à la limite entre les objets et l'arrière-plan. Cependant, le résultat n'est pas toujours satisfaisant en raison du bruit et d'autres effets parasites.

3.1.5 Description d'image

La description d'image est le processus d'extraction des caractéristiques d'un objet à des fins de reconnaissance. Les descripteurs doivent être indépendants vis-à-vis de la taille, de l'emplacement et de l'orientation, en vue de fournir des informations discriminantes suffisantes. Les descripteurs de contours typiques sont les suivants :

- Les codes de chaîne : qui représentent un contour comme un ensemble de segments de ligne droite de longueur et de direction spécifiées.
- Approximations polygonales : dans ce cas, une frontière numérique peut être approximée à n'importe quel degré par un polygone.
- Descripteurs de Fourier : un contour 2D peut être représenté par une transformée 1D, c'est-à-dire qu'un point (x_i, y_i) est représenté par un nombre complexe $x_i + jy_i$

Les descripteurs typiques de régions sont les suivants :

- La texture
- Squelette de la région : un algorithme d'amincissement ou de squelettisation est utilisé pour obtenir le squelette de la région.
- Invariants de moment : les moments centraux normalisés sont utilisés comme descripteurs car ils sont invariants par rapport à la translation, à la rotation et l'échelle.

Les méthodes de segmentation et de description mentionnées dans la section précédente et dans la présente section sont applicables aux données de scènes 2D. Les processus correspondants pour les scènes 3D sont plus complexes et comprennent les constructions de patches planaires, les techniques du gradient et les cônes ou cylindres généralisés.

3.1.6 Reconnaissance d'images

La reconnaissance d'image est appelée le processus d'étiquetage appliqué à un objet segmenté d'une scène. Autrement dit, la reconnaissance d'image suppose que les objets d'une scène ont été segmenté en éléments individuels (par exemple, un boulon, un joint, une clé). La contrainte typique c'est que les images sont acquises dans une géométrie de visualisation connue, souvent perpendiculaire à l'espace de travail. Les méthodologies de

reconnaissance d'images se distinguent comme suit :

- Méthodes théoriques de décision : ces méthodes utilisent des fonctions de décision ou discriminantes appropriées pour faire correspondre les objets à l'un de plusieurs prototypes.
- Méthodes structurelles : un objet est décomposé en un ensemble de motifs d'éléments primitifs de longueur et direction prédéfinies.

3.1.7 Interprétation

L'interprétation d'images est un processus de niveau supérieur qui utilise des combinaisons des méthodes décrites précédemment, à savoir la détection, le pré-traitement, la segmentation, la description, et la reconnaissance. Un système de vision artificielle est classé en fonction de sa capacité à extraire des informations utiles d'une scène, nécessitant une connaissance minimale des objets à portée de main. Les facteurs qui rendent l'interprétation des images difficile, sont la variation dans les conditions d'éclairage, visualisation de la géométrie et occlusion des corps. Les problèmes d'occlusion surviennent lorsque nous avons une multitude d'objets dans l'espace de travail. D'autres processus qui aident à une meilleure interprétation de l'image sont comme suit :

- La possibilité d'obtenir une image à plusieurs échelles.
- Recherche séquentielle : chaque pixel est examiné une fois. Nouveaux pixels sont comparés à des groupes de pixels précédemment trouvés et insérés dans le groupe correspondant. S'il s'avère qu'un pixel appartient à deux groupes, les groupes doivent être combinés.

3.2 Description de l'asservissement visuel d'un robot mobile

Le but de l'asservissement visuel est la modélisation de la boucle de vision comme une commande numérique en utilisant un capteur visuel (caméra) en vue de donner la vision au robot. La caméra est un capteur qui fournit l'information la plus riche par rapport à

tous les autres capteurs, la contrepartie de cette richesse est le traitement très lourd des informations provenant de ce capteur.

3.2.1 Définition de l'asservissement visuel

Asservissement visuel désigne un asservissement où les grandeurs asservies sont issues d'une ou plusieurs images produites par une ou plusieurs caméras fixant une région de l'environnement du robot par rapport à laquelle une tâche doit être effectuée. La boucle d'un asservissement visuel est composée d'un capteur visuel, un dispositif d'acquisition/traitement et un système mécanique actionné. Une caméra est un exemple courant de capteur visuel, bien que d'autres dispositifs d'imagerie puissent également être utilisés (scanner, IRM, etc.). Le dispositif d'acquisition/traitement, souvent appelé ordinateur, est un dispositif capable de traiter les données visuelles numérisées. Souvent, le même système est chargé à la fois de la commande et du traitement de l'image. Le système mécanique qui est activé par l'utilisateur reçoit cette commande qui est générée par le retour visuel. Au cours des 20 dernières années, la recherche sur l'asservissement visuel a connu une grande activité. En réalité, elle est continuellement renforcée via les nouvelles possibilités offertes par l'augmentation de la puissance du système en termes de capacité de calcul, de performances de l'imageur ou de nouvelles applications robotiques.

3.2.2 Historique sur les asservissements visuels

Dans cette partie nous allons donner quelques dates marquantes de l'asservissement visuel [69].

- 1973, Shirai et Inoue ont réalisé la tâche de saisie d'un objet cubique en utilisant une caméra fixe qui observe la scène avec une période d'asservissement de 10s.
- 1979, Hill et Parks ont aussi réalisé une tâche de suivi, de saisie et de déplacement par rapport à une scène en utilisant la projection d'un marqueur lumineux pour reconstruire la profondeur et l'orientation.
- 1979, Prajoux a utilisé un observateur pour anticiper le mouvement de la cible dans une tâche de suivi de cible à 2 degrés de liberté où le temps d'établissement était

de 1 s.

- 1984, Weiss a réalisé le premier asservissement visuel 2D mais il ne l'a pas formalisé.
- 1990, Venkatesan et Archibald ont aussi utilisé la projection de marqueurs lumineux à l'aide de 2 scanners LASER pour une commande par vision suivant 5 degrés de liberté. La bande passante de la boucle fermée était de 4.5 Hz.
- 1991, Papanikolopoulos a réalisé la tâche de suivi de cible par estimation de la mesure par flot optique.
- 1992, Espiau, Chaumette et Rives ont formalisé la commande 2D à travers l'utilisation de la notion de fonction de tâche tout en prouvant la stabilité du système. Ils ont par la suite validé expérimentalement la tâche de suivi de cible.
- 1996, Wilson a réalisé un asservissement 3D, par le filtre de Kalman de la mesure pour une tâche de suivi de cible à 5 degrés de liberté, ainsi la fréquence de l'asservissement est de 60 Hz.
- 1996, Peter Corke a utilisé le modèle dynamique du manipulateur pour la synthèse du correcteur pour la réalisation d'une tâche de suivi de cible à 2 degrés de liberté avec une fréquence de l'asservissement de 60 Hz.
- 1998, Gangloff et al. ont utilisé le modèle dynamique d'un asservissement visuel à 6 DDL pour un asservissement visuel à 120 Hz d'un robot à 6 DDL dans une tâche de suivi de cible.
- 1999, Malis et Chaumette ont parlé pour la première fois de l'asservissement visuel 2D $\frac{1}{2}$ pour la réalisation d'une tâche de suivi de cible.
- 1999, Nakabo est le premier qui a utilisé la camera vision chip pour le suivi de cible avec une caméra orientable à 2 DDL pour un asservissement visuel à 1000 Hz.
- 2005, Ginhoux et al. ont réalisé un asservissement visuel dynamique dans le domaine médical pour une boucle de vision à 500Hz par Commande répétitive pour le suivi de cœur battant

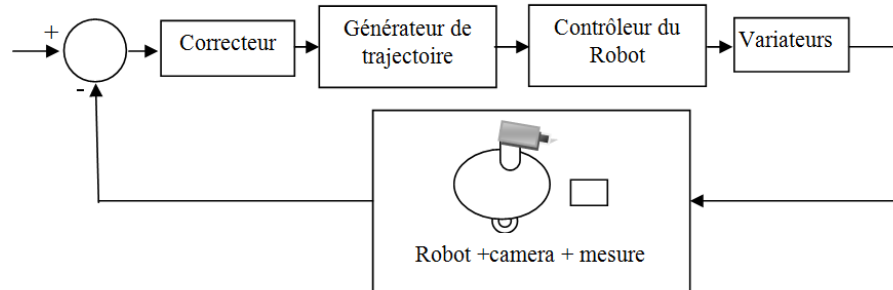


FIGURE 3.1 – schéma de la boucle d'asservissement visuel indirect.

3.2.3 Classification des asservissements visuels

Les asservissements visuels peuvent être classés selon quatre critères : la position de la (les) camera(s), le type de la commande, le type de la mesure, et la rapidité d'échantillonnage.

3.2.3.1 Classification en fonction de la position de la caméra

L'asservissement peut être classé en deux catégories :

- caméra embarquée (eye in hand) la caméra est fixée sur le robot
- caméra déportée (eye to hand) la caméra est fixée dans l'environnement de travail du robot.

3.2.3.2 Classification en fonction du type de la commande

L'asservissement visuel peut être direct ou indirect

- **Asservissement visuel indirect :**

Dans ce type d'asservissement visuel on conserve la boucle de position active et on n'ouvre pas les boucles de vitesse. C'est un asservissement lent, si on ajoute la boucle de vitesse, le traitement devient très lent en faisant des dérivées pour avoir les vitesses. Par ailleurs dans ce type de commande on envoie les consignes directement au générateur de trajectoire comme montré sur la figure (3.1). Ce type est utilisé pour les asservissements lents. Sachant qu'un asservissement commence à être rapide à partir de 50 Hz pour un robot classique moyen.

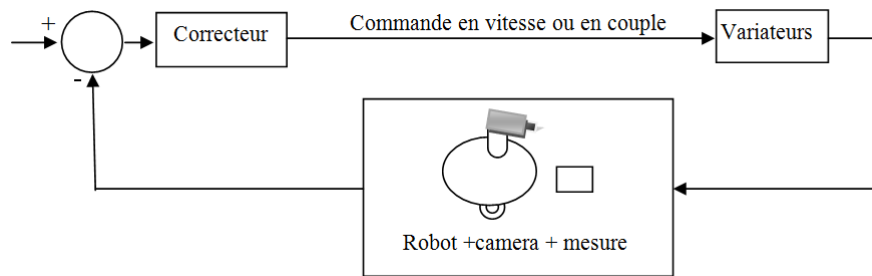


FIGURE 3.2 – Schéma de la boucle d'asservissement visuel direct

— **Asservissement visuel direct :**

Il est utilisé pour les asservissements rapides, de telle manière quand on a les moyens pour faire un asservissement rapide, on peut se passer de la commande numérique du robot (on peut désactiver l'ordinateur qui fait l'asservissement numérique de la position du robot). La seule contre réaction de l'asservissement visuel est la caméra, c'est-à-dire attaquer les variateurs de vitesse avec le correcteur de vision comme montré sur la figure (3.2) (pouvoir lire les positions sur les encodeurs). La désactivation de la boucle de position nous permet d'aller plus vite, nous permet aussi d'éliminer une partie de traitement (partie du code) qu'on ne maîtrise pas forcément, et d'éliminer les retards ou au moins les minimiser.

3.2.3.3 Classification en fonction du type de la mesure :

L'asservissement visuel aussi se classe en trois en fonction du type de la mesure :

— **L'asservissement visuel 3D**

C'est une approche classique ,c'est un asservissement basée sur la position. La consigne et la mesure sont des coordonnées d'attitude (pose) comme le montre la figure (3.3). Afin d'estimer la mesure il est nécessaire de connaître la géométrie de l'objet d'intérêt. L'inconvénient principal de l'asservissement visuel 3D est quand l'erreur est grande (l'objet est à une position loin du robot) .Le robot va se déplacer en ligne droite en 3D pour atteindre la position désirée dans tous les sens, tout en choisissant le chemin le plus optimal avec un minimum d'énergie. Néanmoins lorsqu'on déplace la caméra, on peut ne pas voir la cible à tout moment, car on peut perdre un point et on ne peut plus faire la reconstruction 3D.

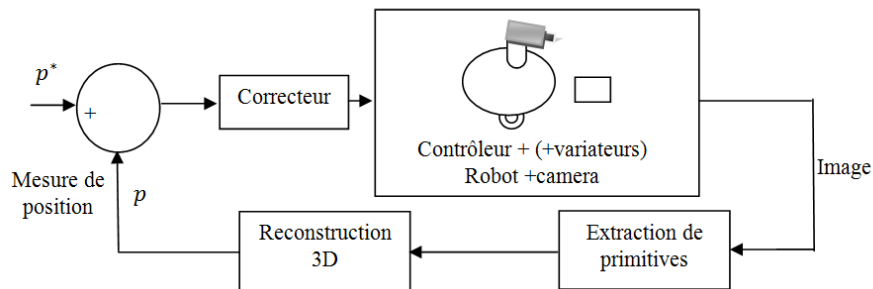


FIGURE 3.3 – Schéma de la boucle de commande d'un asservissement 3D

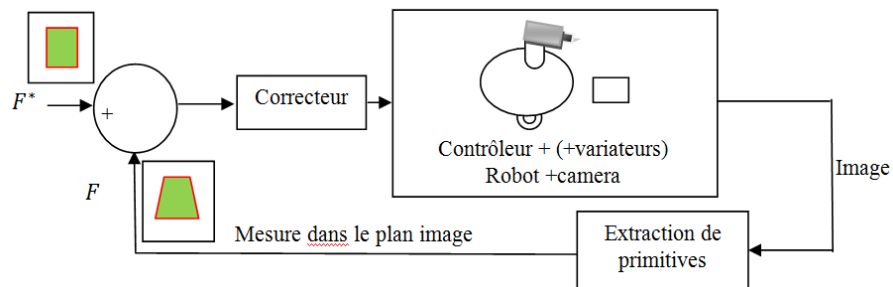


FIGURE 3.4 – schéma de la boucle de commande d'un asservissement 2D

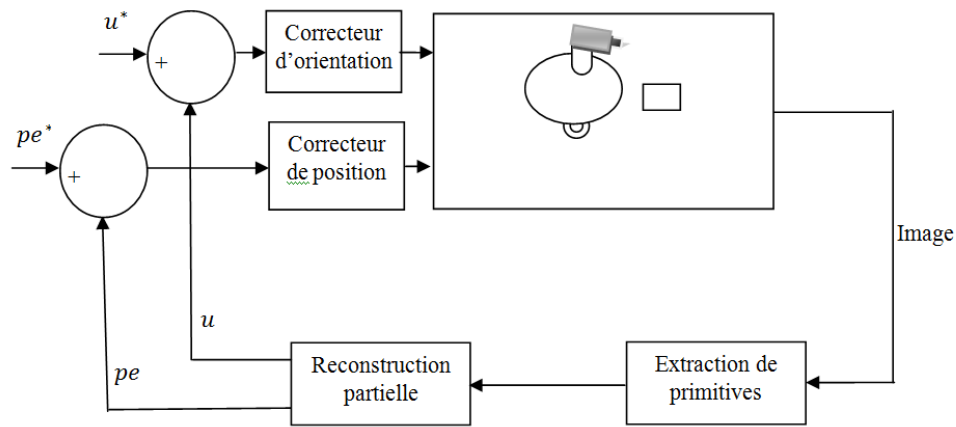
— L'asservissement visuel 2D

C'est un asservissement basé sur l'image (image based). La consigne et la mesure sont des coordonnées de primitives visuelles, l'asservissement 2D ne contient pas de reconstruction 3D, il n'est pas nécessaire de connaître la géométrie du robot[74]. Le schéma de la boucle de commande est donné par la figure (3.4).

L'inconvénient principal de ce type est que les points se déplacent dans l'image en lignes droites ce qui fait que l'optimisation n'est pas garantie (phénomène de Chaumette Conundrum).

— L'asservissement visuel 2D $\frac{1}{2}$

En raison des inconvénients de chacun des asservissements 2D et 3D les chercheurs ont pensé à un schéma regroupant les deux boucles de commande précédentes afin de tirer profit de leurs avantages. L'asservissement de l'orientation est un asservissement 3D et l'asservissement de position est un asservissement 2D. Les 2 boucles d'asservissement sont

FIGURE 3.5 – Schéma de l'asservissement visuel 2D $\frac{1}{2}$

découplées. Comme illustré par la figure (3.5)

- La boucle 2D assure la visibilité et la translation
- La boucle 3D assure la visibilité et l'orientation, cette boucle assure le suivi de trajectoire en 3D et cela sans utiliser le modèle CAO.

3.2.3.4 Classification en fonction De la rapidité

Les asservissements visuels sont classés en lent et rapide en fonction de période d'échantillonnage

— Asservissement visuel lent

Les asservissements visuels lents sont appelé asservissement visuel cinématique. On approxime la fonction de transfert du robot entre consigne de vitesse et mesure de vitesse par un simple gain. Cette approximation est valable pour des déplacements lents du robot et lorsque la commande est du type indirecte. Le correcteur est un simple gain ce qui permet de démontrer une convergence exponentielle de la mesure vers la consigne. Ce type est utilisé généralement dans le domaine nucléaire.

— Asservissement visuel rapide

Les asservissements rapides sont des asservissements visuels dynamiques. Le modèle dynamique du robot est pris en compte dans la synthèse du correcteur. Étant donné le comportement dynamique moyen de la plupart des robots mobiles, on considère que 50 Hz

est la limite à partir de laquelle il est nécessaire de tenir compte du modèle dynamique pour améliorer les performances. La commande est du type direct soit en vitesse ou en couple.

3.3 Planification de trajectoire d'un robot mobile avec évitement d'obstacles visuel

Les robots mobiles sont devenus de plus en plus populaires, offrant un large éventail d'applications dans divers domaines tels que l'industrie, l'agriculture, la recherche et le sauvetage et bien plus encore. Cela a été réalisé principalement grâce à des efforts extrêmement actifs des chercheurs pour le développement de la technologie de l'autonomie. Nous sommes encore confrontés à de nombreux défis, tel que la navigation d'un robot mobile de manière efficace et fiable dans un environnement sans aucune aide humaine. Le robot doit être capable d'extraire les informations nécessaires de l'environnement et prendre les mesures nécessaires pour planifier un chemin sûr pour que le mouvement sans collision atteigne son objectif. Pour qu'un robot puisse naviguer avec succès, il doit être capable de trouver un chemin entre sa position actuelle et sa destination. Il doit également être capable d'éviter les obstacles le long du chemin statiques ou dynamiques. Dans cette partie, nous passons en revue des méthodologies de planification de trajectoire les plus couramment utilisées qui ont été appliquées pour la navigation des robots mobiles. Nous présenterons des approches globales et locales de planification de trajectoire ainsi que des techniques classiques et heuristiques et en deuxième partie nous présenteront les différentes méthodes pour l'évitement d'obstacles par vision. Un robot doit avoir la capacité d'optimiser sa trajectoire en calculant l'itinéraire le plus court et le plus sûr vers sa destination afin de gagner du temps et de réserver de l'énergie. La complexité de la tâche de planification du chemin augmente plus l'espace de configuration devient grand. Il s'agit encore d'un problème fondamental dans la robotique mobile et de nombreuses recherches ont été menées pour trouver la meilleure solution. Les approches de navigation des robots mobiles ont été classées comme ; Planification globale, et planification locale. Dans la planification

de trajectoire globale, le robot possède des informations sur son environnement. Cela inclut sa taille, sa forme, l'orientation, ainsi que les positions d'obstacles dans l'environnement. L'approche de planification globale utilise généralement une certaine forme d'optimisation afin de maximiser l'efficacité et les performances de la recherche. La principale contrainte est cependant dans laquelle cette approche ne peut pas être invoquée est l'environnement dynamique qui contient plusieurs objets en mouvement le long de son chemin.

Dans la planification de trajectoire locale, en revanche, le robot a peu ou aucune information sur son environnement et doit plutôt adopter une approche réactive telle que l'utilisation d'algorithmes génétiques (GA) ou des réseaux de neurones artificiels (ANN) pour l'aider à prendre les meilleures décisions sur la façon de planifier son chemin en fonction de sa perception du milieu environnant. Les données collectées des capteurs embarqués est crucial pour déterminer les prochains plans d'action du robot. Dans cette partie, nous passons en revue les méthodologies de planification de trajectoire les plus couramment utilisées et appliquées pour la navigation de robots mobiles en environnement statique ou dynamique. Nous présenterons à la fois les techniques classiques et heuristiques tout en soulignant les avantages et les inconvénients associés.

3.3.1 Les méthodes de planification de trajectoire globales

Parmi les méthodes de planification globales nous pouvons citer ; la méthode Roadmap (la feuille de route), La méthode du champ potentiel, et la Méthode de décomposition en cellules.

3.3.1.1 Méthode Roadmap

appelé aussi feuille de route, cette approche implique la construction d'un réseau de chemins sans collision entre la position initiale du robot et la position finale. Cette méthode peut être définie comme un algorithme de recherche de graphe. L'espace de travail du robot, est défini comme un ensemble de mouvements réalisables, est réduit en un réseau de lignes unidimensionnelles [82]. Il existe plusieurs approches pour trouver le chemin le plus optimal et de nombreuses méthodes diffèrent dans la façon dont les séries de chemins et de nœuds

sont définis. Parmi les feuilles de route les plus connues ; les graphes de visibilité et le diagramme de Voronoï.

— **Les graphes de visibilité**

Dans les graphes de visibilité les sommets sont reliés au départ, l’objectif à attendre et les obstacles sont polygonaux, les arêtes représentent les bords de l’obstacle et relient les deux. Les bords de l’obstacle et les bords reliant tous les ensembles de sommets qui peuvent se voir les uns les autres. Par conséquent, cette méthode présente une solution de chemin de longueur minimale. Les graphes de visibilité sont considérés comme très utiles dans des environnements où les objets suivent des formes polygonales dans des espaces continus ou détachés [83]. Un problème courant avec les graphes de visibilité, c’est que les chemins qui en résultent entrent en contact avec les obstacles aux sommets et aux arêtes. Ce qui peut provoquer des collisions potentielles pour le robot. Cela peut rendre sa mise en œuvre assez difficile. Les diagrammes de Voronoï, en revanche d’autre part, sont capables de résoudre ce problème.

— **Diagramme de Voronoï**

Le diagramme de Voronoï est souvent utilisé pour calculer la proximité de points. Pour un ensemble donné d’objets, (appelés sites), chaque cellule de Voronoï définit l’ensemble des points du plan qui sont plus proches de chaque site que de tout autre site. Les contours de toutes les cellules de Voronoï forment le diagramme de Voronoï. Cette approche garantit la construction du chemin le plus sûr pour le robot par défaut car les arêtes du graphique sont situées à une distance maximale disponible de tous les obstacles proches [3]. Ces distances ne peuvent être calculées que sur la base des capacités et de la précision des capteurs embarqués (par exemple LiDAR, Radar) qui sont utilisés pour la localisation. Par conséquent, les capteurs à courte portée tels que les capteurs à ultrasons ne conviennent pas à ce type d’approche de planification de trajectoire.

3.3.1.2 Méthode du champ potentiel

Cette méthode a été proposée pour la première fois par Khatib [84] pour résoudre la tâche d’évitement d’obstacles par robot mobile. Elle consiste à traiter le robot comme

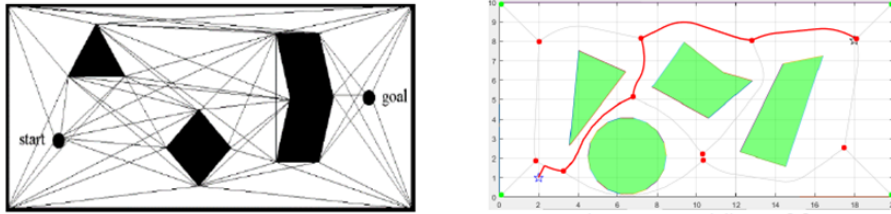


FIGURE 3.6 – Les graphes de visibilité[3] et le diagramme de Voronoi[4]

une particule sous l'influence d'un champ potentiel artificiel et représenté comme un point dans l'espace de travail. L'objectif de cette méthode est de faire naviguer le robot vers des champs attractifs et loin des champs répulsifs. Les obstacles sont représentés par des forces répulsives, tandis que le point de destination du robot est attribué à la force positive. Cette méthode s'est avérée très populaire au fil des ans en raison de sa simplicité. Mais elle n'est efficace que dans un environnement stationnaire avec des obstacles stationnaires. Un autre problème courant qui se produit est que le robot peut être piégé dans un minimum local du potentiel. Diverses solutions à ce problème ont été proposées, telles que la marche aléatoire et le retour en arrière [85]. Il existe de nombreuses versions modifiées de la méthode du champ potentiel, dont la plupart sont utilisées comme des contrôleurs de mouvement pour les robots. Subramanian, George et Thondiyath [86] ont proposé une approche de champs potentiel multipoints pour l'évitement d'obstacles dans des environnements statiques et dynamiques. Cette approche observe des erreurs minimales de suivi de position qui tendent à générer un chemin plus lisse. Valbuena et Tanner [87] ont proposé l'utilisation de cette méthode pour la planification des mouvements et l'évitement d'obstacle des robots non ergonomiques.

3.3.1.3 Méthode de décomposition cellulaire

L'approche de la décomposition cellulaire est bien connue dans la littérature pour résoudre des problèmes de planification de trajectoire statiques et dynamiques. Elle consiste à la représentation de l'espace de travail du robot qui a été divisé en cellules. Les relations entre chacune des cellules adjacentes seront ensuite calculées dans le but de créer un che-

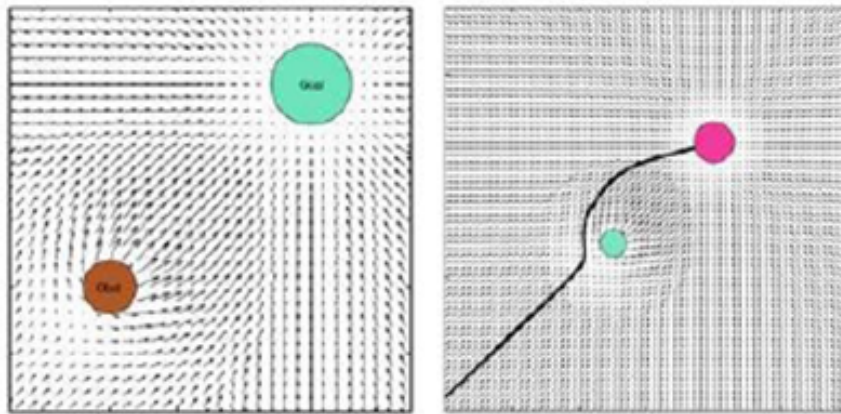


FIGURE 3.7 – Représentation du champs potentiel avec obstacle et objectif à atteindre par le robot[5]

min sans collision du point de départ jusqu'à l'objectif. Les cellules qui contiennent des obstacles sont divisées en deux nouvelles cellules, puis les cellules qui ne contiennent pas d'obstacles sont ajoutées à la séquence [83]. Le processus commence par l'identification des deux cellules contenant le point de départ et le point d'arrivée. Un premier exemple d'application de cette approche est présenté dans [88]. Une nouvelle méthode d'approximation de décomposition cellulaire est proposée par [89]. Cette approche représente les cibles, les obstacles, la plate-forme du capteur et le champ de vision (FOV) comme des sous-ensembles fermés d'un espace de travail euclidien. Chenghui Cai et Ferrari [90] ont proposé une méthode d'approximation de décomposition cellulaire hiérarchique qui inclut la dimension du modèle du monde. Cette approche permet aux obstacles dynamiques d'être considérés comme statiques dans un monde étendu.

3.3.2 Les méthodes de planification de trajectoire locales

3.3.2.1 Algorithmes des bugs

— Bug 1

Dans le cas où le robot détecte un obstacle sur sa trajectoire, cet algorithme permet au robot d'arrêter, puis se déplacer en suivant le contour de l'obstacle détecté[91]. Il continuera à contourner l'obstacle jusqu'à ce qu'il revienne au point où il a commencé à suivre l'obstacle.

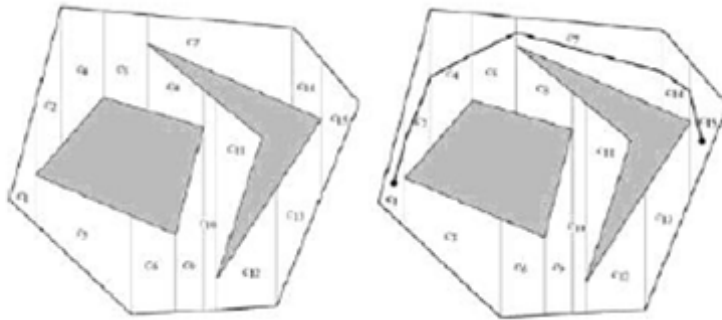


FIGURE 3.8 – Méthode de décomposition cellulaire[6].

Pendant que ce processus se déroule le robot essaie aussi de calculer la distance entre la position actuelle et la position de l'emplacement du but dans l'espoir d'estimer le point exact sur la ligne de bord qui représente la distance la plus courte du but. Une fois le point enregistré en mémoire, le robot sait qu'il s'agira du point où il s'écartera de l'obstacle lorsqu'il commencera son prochain cycle autour de celui-ci [7].

— Bug 2

L'algorithme bug 2 est une version plus efficace que l'algorithme bug 1[91], dans la mesure où le robot n'est pas obligé de faire un cycle complet autour de l'obstacle. Peut donc atteindre sa destination en moins de temps. Ceci peut être accompli en calculant d'abord un chemin initial de la source au but et en stockant la pente du chemin dans sa mémoire de comportement. Dans le cas où le robot détecte un obstacle, il changera automatiquement de comportement et commence à suivre le bord comme il le fait dans l'algorithme bug 1, sauf que cette fois, il calcule constamment la ligne de pente actuelle pour voir si elle correspond à la ligne de pente mémorisée. Une fois que les deux pentes sont égales, le robot repasse en mode de comportement au " déplacement vers le but " et s'éloigne de l'obstacle. Il existe de nombreuses variantes et améliorations de l'algorithme du bug. L'algorithme DistBug [92] qui utilise un capteur de distance. L'algorithme Tangent Bug permet au robot d'obtenir une sélection de chemins optimaux en créant un graphe local de tangentes [93].

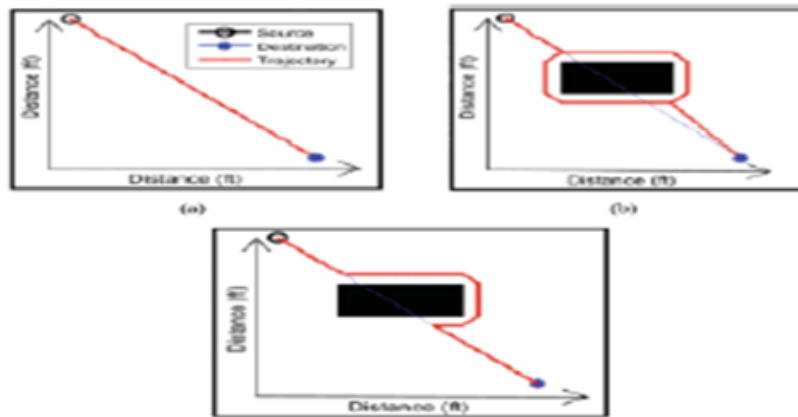


FIGURE 3.9 – Les méthodes de planification de trajectoire par les Algorithmes des bugs [7]

3.3.2.2 Histogramme de champ vectoriel

Cette méthode a été initialement introduite par Borenstein et Koren [94] et considérée comme une version améliorée de la méthode de champ potentiel. La méthode de champ vectoriel est une approche d'évitement d'obstacles en temps réel qui permet à un robot de détecter et d'éviter des obstacles inconnus tout en continuant à se diriger vers la cible. Un modèle de son environnement est représenté par une grille d'histogrammes cartésiens à 2 dimensions qui sont constamment mise à jour avec les données obtenues via des capteurs de distance embarqués. La méthode utilise ensuite un processus de réduction des données en deux étapes afin de calculer les commandes de contrôle requises pour le robot. La première étape consiste à prendre un sous-ensemble de taille fixe de la grille d'histogrammes à 2 dimensions autour de l'emplacement du robot et de le réduire à un histogramme polaire unidimensionnel. Une valeur représentant la fonction de densité d'obstacles polaires dans cette direction est contenue dans chaque secteur de cet histogramme polaire. Dans la deuxième étape, le processus de sélection de l'environnement qui contient la densité d'obstacles polaires la plus faible est calculée et puis la direction du robot est calculée. Il existe un certain nombre de variantes de cette méthode qui ont été développées afin d'améliorer l'algorithme original. La méthode de champ vectoriel + (VFH+) a été développée pour améliorer la fiabilité dans les applications de robots à déplacement rapide [95]. Une autre

variante est VFH*, qui adopte la méthode heuristique de recherche A* [96].

3.3.3 Les approches heuristiques de planification de trajectoire

Les approches heuristiques de planification de trajectoire sont devenues très actives ces dernières années par rapport aux approches classiques. La principale différence réside dans le potentiel d'apprentissage basé sur le comportement humain dans une approche heuristique. La section suivante examine certaines de ces approches les plus populaires, notamment les réseaux de neurones artificiels (ANN), la logique floue (FL), l'algorithme génétique (GA) et la PSO (particule swarm optimisation).

3.3.3.1 Les réseaux neurones artificiels

Les réseaux de neurones artificiels (ANN) s'inspirent de l'étude du fonctionnement interne du cerveau humain. Il s'agit de modèles artificiels qui ont la capacité d'apprendre sur la base d'un comportement courant. Le réseau est composé de neurones, comme celui du cerveau humain, et chaque neurone représente une unité de calcul qui peut reproduire des informations basées sur l'expérience et l'observation. La capacité globale ou l'intelligence de l'ANN est définie en fonction de la nature et du potentiel de chaque neurone et de la façon dont ils sont inter-connectés. Les ANN sont largement utilisées dans le domaine de l'optimisation, de l'apprentissage et de la reconnaissance des formes en raison de leur capacité à fournir des solutions simples et optimales [83]. Zou, Hou, Fu et Tan [97] ont suggéré que de nombreux problèmes actuels concernant la navigation des robots dans une application réelle peuvent être résolus en utilisant une approche hybride qui combine d'autres méthodes d'intelligence artificielle telles que les systèmes basés sur la connaissance ou la logique floue. Li et Chen [98] ont proposé une approche hybride qui combine un ANN avec un algorithme PSO où la PSO crée une trajectoire plus lisse pour atteindre l'objectif. L'un des principaux inconvénients de l'utilisation des ANN est que la quantité de données d'apprentissage requise peut être très importante. Un autre problème commun concerne la difficulté de minimiser l'erreur entre les données calculées et la sortie requise dans un algorithme de ANN à apprentissage supervisé.

3.3.3.2 La logique floue

L'idée de la logique floue a été proposée pour la première fois en 1965 par Zadeh [99]. Elle est basée sur la logique traditionnelle connue sous le nom de booléenne, où quelque chose peut être représenté comme "vrai" ou "faux", (1 ou 0). L'approche floue offre un "degré de liberté" entre ces deux états. Cette approche a été largement utilisée pour un large éventail d'applications de planification de trajectoire des robots, car elle offre au robot la possibilité d'acquérir une connaissance heuristique de son environnement. L'utilisation d'un contrôleur à logique floue (FLC) est bien adaptée au contrôle d'un robot en raison de sa capacité à faire des déductions, même en cas d'incertitude. Et peut aider à la prise de décision ainsi qu'à la génération de règles. Une approche floue basée sur les données pour créer un cadre pour résoudre les problèmes de mouvement dynamique d'un robot a été développée par [84]. L'un des principaux résultats de ce développement est qu'elle permet au robot de naviguer avec succès dans un environnement contenant de nombreux obstacles dynamiques et il montre également une réduction du temps de déplacement par rapport aux approches précédentes. Un nouveau FLC pour un robot de type voiture a été conçu par [100]. Il est capable de fournir des services autonomes dans un garage et un parking parallèle en incorporant un traitement d'image en temps réel. Un algorithme hybride de planification de trajectoire pour un robot mobile utilisant un FLC et un GA est proposé par [101]. L'algorithme génétique (GA) est incorporé afin d'effectuer des modifications sur les fonctions d'entrée et de sortie du FLC.

3.3.3.3 Les algorithmes génétiques

Les algorithmes génétiques GA sont considérés comme étant des outils d'optimisation les plus couramment utilisés pour produire des résultats précis et de haute qualité. Les GA s'inspirent de l'idée de la sélection naturelle qui commence sans aucune connaissance préalable de ce qui pourrait être la bonne solution. Au lieu de cela, un algorithme génétique prend des connaissances de l'environnement et ensuite utilise des opérateurs évolutifs tels que la mutation, le croisement et la sélection pour calculer la meilleure solution [102]. L'une

des toutes premières utilisations des AG pour un robot mobile a été réalisée par [103] en 1989. Il a utilisé pour essayer de résoudre le problème de cinématique inverse associé aux robots redondants. Les exigences pour l'application d'un GA à un problème de planification de trajectoire sont ; la représentation de chemin par un chromosome approprié, la création d'une méthode de guidage le long du chemin, la recherche d'une solution pour l'évitement des obstacles, ainsi que la sélection d'une méthode appropriée de définition des contraintes afin de réduire le nombre d'obstacles. La sélection d'une définition de contrainte appropriée fournissant une méthode pour réduire la distance du chemin [82]. UN GA utilisant un chromosome de longueur variable est proposé dans [104].

3.3.3.4 La PSO

La PSO (particule swarm optimisation) a été introduite pour la première fois par Kennedy et Eberhart en 1995[105], comme un concept d'optimisation de fonctions non linéaires. L'idée a été inspirée de la capacité intrigante d'animaux sauvages comme les oiseaux, les troupeaux d'animaux et les bancs de poissons à s'adapter naturellement à leur environnement. Une étude socio-cognitive a examiné l'idée d'une "intelligence collective" associée à de tels groupes. Dans la PSO, un essaim initial représente un ensemble de solutions aléatoires qui se propagent dans l'espace de recherche vers la solution optimale au cours d'une série d'itérations fondées sur la vaste base de données concernant l'espace de recherche qui ont été recueillies par chaque membre (particule) de l'essaim. Chaque particule est définie par sa position et sa vitesse dans l'espace de recherche, et se déplace vers la position qui représente sa meilleure découverte personnelle (locale) et la meilleure découverte de l'essaim (globale)[106]. Il existe de nombreuses variantes de PSO utilisées dans le domaine de la recherche en robotique, plus précisément dans l'évitement d'obstacles et la planification de trajectoire. Min, Zhu et Zheng [107] ont proposé une technique d'évitement d'obstacles dans un environnement dynamique en utilisant l'optimisation multi-objectif par PSO. Pugh, Martinoli et Zhang [108] ont utilisé des versions de PSO avec quartier local pour multi-robots, où il a été constaté que la méthode proposée est plus performante que les AG.

Dans cette partie nous avons donné un aperçu de la variété des méthodologies de planification de trajectoire utilisées pour la navigation des robots mobiles. Ainsi qu'une classification de ces méthodes en globale et locale, et en méthodes classiques et heuristiques. Nous avons aussi discuté les avantages et les inconvénients associés à chaque méthode. Les méthodes classiques abordées sont roadmap, les Méthode du champ potentiel et les méthodes de décomposition cellulaire. Ces méthodes cherchent soit à trouver une solution, soit à prouver qu'il n'y a pas de solution. Elles peuvent ne pas être fiables dans les applications en temps réel à cause de leurs incapacités à fonctionner dans des environnements dynamiques. Les méthodes heuristiques examinées comprennent les ANN, la FL, les GA et la PSO. Ces méthodes sont parmi les plus couramment utilisées pour la navigation des robots mobiles. Contrairement aux méthodes classiques, ces algorithmes ne garantissent pas la découverte d'une solution, mais s'ils y parviennent, ils la trouveront en beaucoup moins de temps et de calcul. Cela rend les méthodes heuristiques beaucoup plus adaptées aux applications en temps réel, notamment dans les environnements dynamiques. Les performances globales de ces méthodes dépendent toutefois entièrement de la précision et de la fiabilité des informations fournies.

3.4 Commande par vision d'un robot mobile à roues

Différentes méthodes de commande sont utilisées pour les robots mobiles. Dans cette partie nous allons donner un aperçu sur des techniques de contrôle basé sur la vision. Le comportement des robots mobiles peut être géré à l'aide de diverses méthodes.

3.4.1 Les différents types de commande visuelle

Dans cette partie nous présenterons les différentes stratégies de commande visuelle, appliquées aux robots mobiles à roues [69] :

3.4.1.1 Commande séquentielle

Désigné sous le nom de look then move, cette méthode ne nécessite ni le modèle du robot, ni le modèle de l'environnement. Dans ce type de commande la caméra observe

la scène et fait l'acquisition de l'image ou d'un flux d'images, après l'analyse de l'image et l'extraction des informations nécessaires, une trajectoire à suivre est générée par la commande, le robot fait son déplacement en aveugle suivant une trajectoire prédéfinie, cette approche est bien adaptée dans les systèmes où l'environnement de travail du robot est statique et bien connu. Par ailleurs cette méthode quoiqu'elle soit simple, elle manque de précision et de robustesse.

3.4.1.2 Commande cinématique

Dans ce type de commande le robot est considéré comme un intégrateur simple, le comportement dynamique du robot est simplifié; les effets dynamiques comme les retards sont négligés. Cette approche est utilisée dans les situations où il n'est pas nécessaire que le robot se déplace rapidement. Le but de cette commande est la robustesse non pas la rapidité. La commande cinématique peut être une commande 2D, 3D ou 2D1/2. Dans la commande 2D, les mesures sont des paramètres issus de l'image comme les points, les barycentres des points et autres primitives dans l'image. Les mesures dans une commande cinématique 3D sont des positions dans l'environnement du robot. Par contre la commande 2D1/2 regroupe les avantages des deux boucles de commande précédente. L'asservissement de l'orientation est un asservissement 3D et l'asservissement de position est un asservissement 2D.

3.4.1.3 Commande dynamique

Dans ce type de commande il faut tenir compte de tous les effets dynamiques du système robotique, comme les retards modélisant les temps de transfert et de traitement de l'image. Il est nécessaire de tenir compte de l'effet de l'échantillonnage et du bloqueur d'ordre 0, puisque l'asservissement visuel est un asservissement numérique, sans oublier la modélisation du comportement dynamique du robot nécessitant généralement une connaissance de tous les paramètres mécanique du système robotique. En général les effets dynamiques sont pris en compte dans des systèmes de robotique simple.

Conclusion

Dans ce chapitre nous avons présenté les sous-domaines de la vision par ordinateur nécessaires pour réaliser une commande par vision. Nous avons ensuite donné un bref historique sur l'évolution des asservissements visuels au fil des années et leur classification selon divers critères. Enfin, nous avons évoquées les stratégies de planification de trajectoire avec évitement d'obstacles visuel et les différentes stratégies de contrôle visuel des robots mobiles, à savoir le contrôle séquentiel, le contrôle cinématique et le contrôle dynamique.

Chapitre 4

Résultats expérimentaux

Introduction

L'utilisation de robots a considérablement augmenté ces dernières années. Ils sont maintenant utilisés à des fins très diverses et peuvent être trouvés dans une variété de contextes, y compris les milieux industriels, domestiques, éducatifs et de santé. Dans le cas d'un robot mobile autonome, il doit être capable de naviguer dans une zone inconnue tout en remplissant la tâche pour laquelle il a été construit. Par conséquent, le robot mobile doit être capable de créer un modèle d'environnement, qui sera utilisé en vue de calculer sa position et son orientation relatives à sa navigation dans l'environnement pour atteindre les positions désirées. Parmi les robots mobiles, nous avons retenu : le robot mobile différentiel dont la conception classique comporte deux roues motrices, ainsi qu'au moins une roue libre (généralement une ou deux) pour maintenir la stabilité de la plate-forme. Cette disposition a l'avantage de permettre au véhicule de tourner sur place, dans le cas où les vitesses des deux roues motrices sont de signes opposés. En conséquence, le robot peut tourner rapidement. Cette structure n'est cependant pas holonome puisque le déplacement latéral n'est pas directement réalisable.



FIGURE 4.1 – Photographie de dr robot i90

4.1 Description et modélisation de la plate-forme utilisée dans la partie expérimentale

4.1.1 Description de dr robot i90

Dr Robot i90 est un robot mobile à roues à entraînement différentiel conçu pour le développement d'applications avec une connexion entièrement sans fil. Il ne pèse que 5 kg mais peut transporter une charge utile allant jusqu'à 15 kg. Il mesure 43 cm de largeur, 38 cm de longueur et 30 cm de hauteur. Il est capable d'atteindre une vitesse maximale de 75 centimètres par seconde. La figure (4.1) représente une photographie de dr Robot i90. L'élément pilote de dr Robot i90 est la carte PMS5005, qui est conçue pour fonctionner avec le système WiRobot. Le kit de développement logiciel WiRobot permet la communication entre le PC et le robot [109, 110]. La figure (4.2) présente l'architecture de communication entre le robot et un système de contrôle comme par exemple un ordinateur

4.1.1.1 Capteurs et composants externes

La plate-forme utilisée dans la partie expérimentale est équipée de trois capteurs à ultrasons, neuf capteurs infrarouges, deux capteurs de détections de mouvement humain.

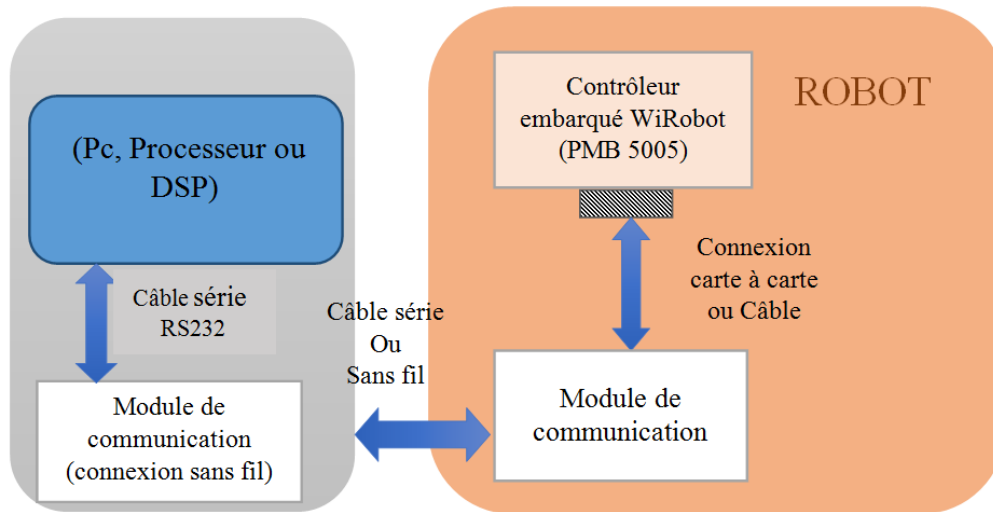


FIGURE 4.2 – L’architecture de communication du système dr Robot i90.

Ce robot dispose d’une caméra à haute résolution et de deux moteurs à courant continu de 12V qui lui permettent de naviguer dans son environnement, ainsi que deux encodeurs quadratiques placés sur les axes des roues motrices qui offrent des mesures relatives incrémentales [111]. La figure (4.3) illustre les principaux composants fonctionnels disposés à l’extérieur de dr robot i90.

4.1.1.2 Fonctionnement et connexion du système dr robot i90

Avec son module sans fil Wifi 802.11G à large bande passante intégré de 54Mbps, le système peut envoyer les données mesurées par des capteurs vers un PC ou un serveur à des fréquences supérieures à 10 Hz. Les commandes et les instructions envoyées au robot via la même liaison sans fil passent également à des fréquences supérieures à 10 Hz, en offrant un contrôle et un accès en temps réel. Le système i90 comprend aussi les composants logiciels de développement WiRobot pour MS Windows 2000 et plus, lui permettant un accès à toutes les données et informations dans un environnement de programmation Microsoft Windows standard. Le schéma de la figure (4.4) montre le scénario de fonctionnement typique de connexion de Dr robot i90. Le PC serveur exécutant le programme de contrôle pourrait se connecter à ce réseau via soit :

- Câble réseau : Connexion de PC à l’un des ports LAN du routeur.

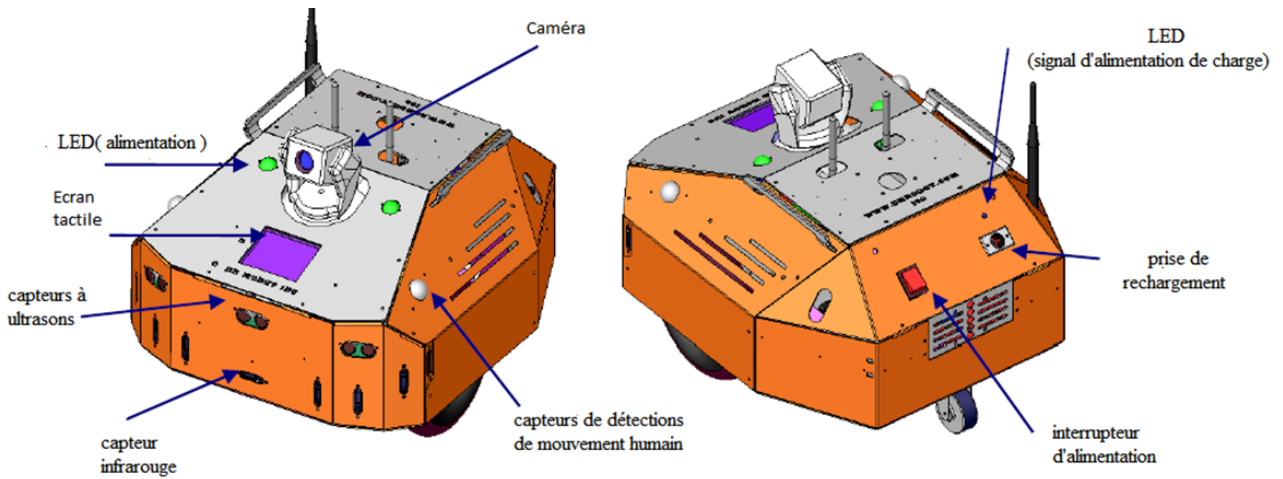


FIGURE 4.3 – Capteurs et composants externes de dr robot i90

- Sans fil : connexion du PC au routeur sans fil, en configurant les paramètres sans fil du PC à l'aide des paramètres de configuration de connexion sans fil.

4.1.1.3 Présentation du SDK WiRobot

Le kit de développement de logiciel de WiRobot (SDK) fait partie du système de développement WiRobot. Le SDK permet la communication du système à l'interface utilisateur, ainsi l'acquisition de données des capteurs et le contrôle des mouvements du robot. Les SDK comprennent des API (interfaces de programmation d'applications), des échantillons de code, des bibliothèques, ainsi que des guides qui peuvent être utilisés et intégrés dans des applications de commande et de contrôle de robots. Les API constituent l'interface entre la partie logicielle et le système matériel de WiRobot. Les programmes développés à l'aide du SDK WiRobot s'exécutent sur le PC en envoyant et en recevant des données vers et depuis le matériel WiRobot via une connexion filaire ou sans fil. Les micro logiciel sur les contrôleurs embarqués s'occupe de toutes les opérations de niveau bas des modules fonctionnels du système, comme l'acquisition de données, le contrôle de mouvement de bas niveau, la capture et la compression d'images et la communication sans fil. L'API existe en tant que composant MS ActiveX, appelé "WiRobot SDK ActiveX Module". Le programme utilisateur utilise ce composant dans un programme pour communiquer avec les



FIGURE 4.4 – Connexion sans fil du système de robot i90

contrôleurs WiRobot PMS5005. Les Données entre le matériel WiRobot et le "WiRobot SDK ActiveX Module" sont gérées et transférées par le programme WiRobot Gateway (WiRobotGateway.exe) avec la mémoire partagée, comme le montre la figure (4.5). Les API du SDK de WiRobot pour le PMS5005 sont des API pour périphériques de capteur, Contrôle de mouvement, Contrôle multimédia et Événements. De nombreux robots sont fournis avec une bibliothèque API ou un SDK pour faciliter la programmation et certains fournissent un environnement de haut niveau basé sur des blocs pour les programmeurs non experts afin de développer des algorithmes et des comportements pour ces robots. La gestion de ces fonctionnalités détourne l'attention des programmeurs de se concentrer sur le développement d'algorithmes de haut niveau.

4.1.2 L'interface de commande

L'unité de contrôle est un ordinateur de bureau avec un processeur de type Intel(R) Pentium(R) CPU G645@ 2.90GHz avec une RAM de 4 Go et un système d'exploitation Windows 7 à 32 bit. Le système de contrôle est implémenté en utilisant le langage MATLAB. L'environnement de développement d'interface utilisateur graphique (GUIDE) a été utilisé pour développer des interfaces interactives permettant aux utilisateurs d'apporter

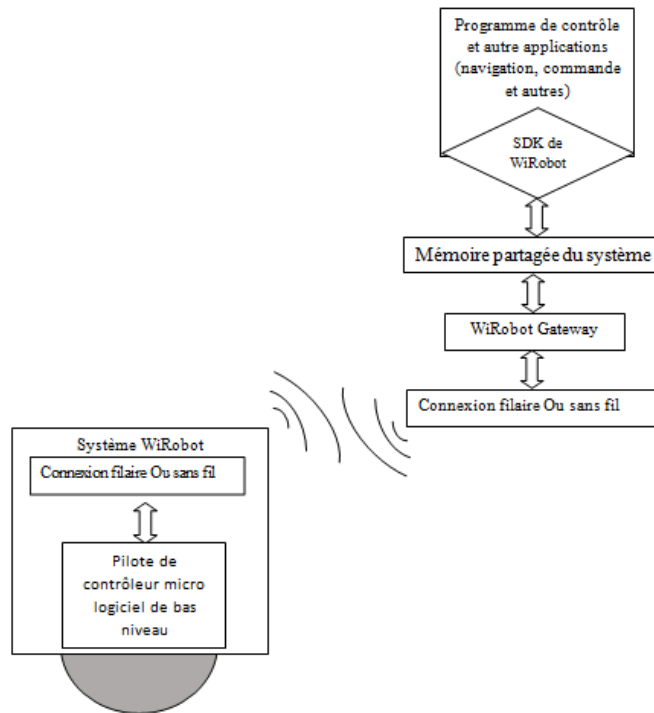


FIGURE 4.5 – Architecture logicielle de WiRobot

des modifications au programme de commande ou d'envoyer les consignes et d'afficher des résultats. Cette interface rend le système plus simple à modifier et à visualiser en temps réel. Chaque composant ou interface graphique est associé à un ou plusieurs codes écrits par l'utilisateur, appelées callbacks. L'exécution de chaque callback est déclenchée par une action particulière de l'utilisateur, telle qu'une pression sur un bouton, un clic de souris, la sélection d'un élément de menu ou le passage du curseur sur un composant. Ce type de programmation est appelé programmation pilotée par les événements.

4.1.3 La description d'un robot mobile de type uni-cycle

Un robot est défini comme uni-cycle lorsqu'il possède deux roues indépendantes et éventuellement des roues folles pour assurer sa stabilité. La figure (4.7) représente un robot de type uni-cycle. Les roues folles ne sont pas incluses car elles n'affectent pas la cinématique si elles sont correctement placées.

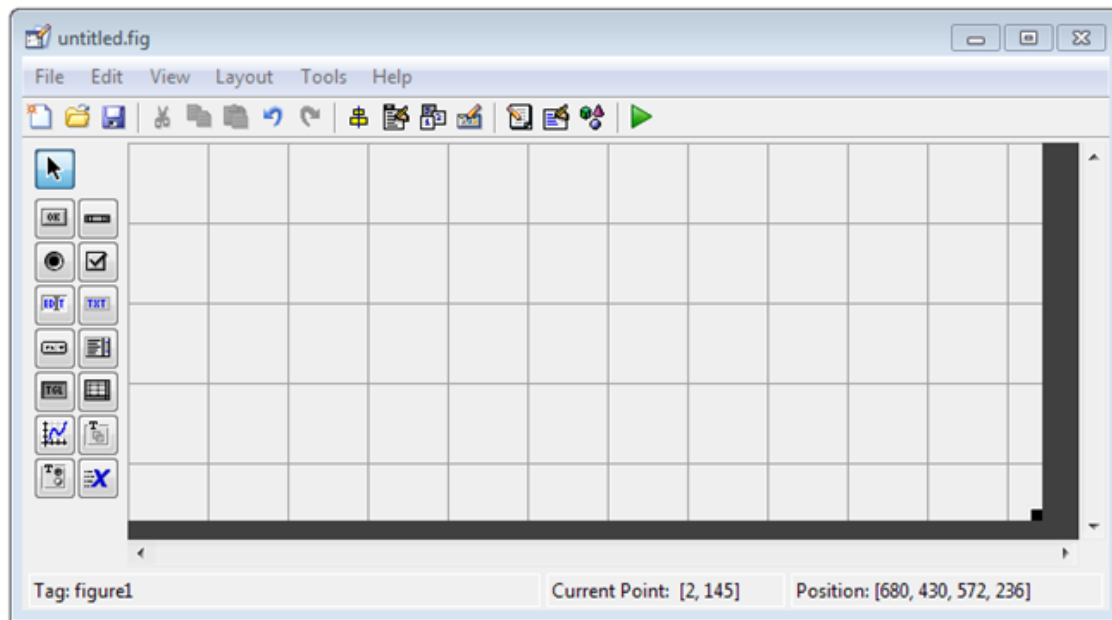


FIGURE 4.6 – L’environnement de création d’interface graphique de MATLAB

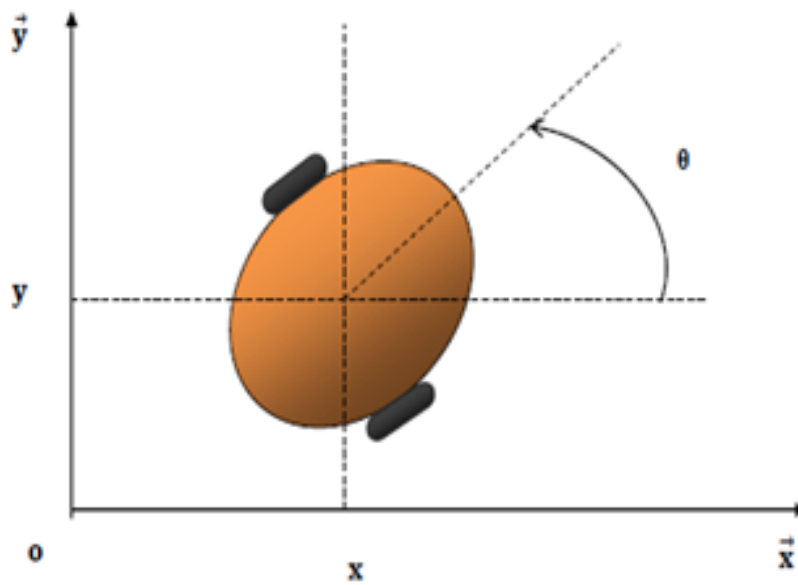


FIGURE 4.7 – Robot mobile uni-cycle

4.1.4 Modèle cinématique

Un robot mobile uni-cycle est une plate-forme à deux roues à entraînement différentiel. Les moteurs à courant continu servent d'actionneurs pour les deux roues du robot mobile, lui permettant de se déplacer et de s'orienter. Pour maintenir l'équilibre de la plate-forme, une roue folle est utilisée. Les équations cinématiques d'un robot se déplaçant à une vitesse linéaire v et une vitesse angulaire ω sont fournies par

$$\begin{cases} \dot{x}(t) = v(t) \cos \theta(t) \\ \dot{y}(t) = v(t) \sin \theta(t) \\ \dot{\theta}(t) = \omega(t) \end{cases} \quad (4.1)$$

x et y indiquent les coordonnées du centre de gravité du robot et θ est l'angle d'orientation. Ce modèle a été utilisé par plusieurs références, notamment [15, 18, 76, 77, 78]. Le modèle d'actionnement donne les vitesses linéaire et angulaire du robot mobile en fonction des vitesses de ses roues motrices droite et gauche et de ses propriétés géométriques, ce modèle est exprimé comme suit :

$$\begin{cases} v(t) = \frac{v_r(t) + v_l(t)}{2} \\ \omega(t) = \frac{v_r(t) - v_l(t)}{D} \end{cases} \quad (4.2)$$

La distance entre les roues directrices est notée par la lettre D . v_r et v_l sont les vitesses de la roue droite et gauche respectivement, représentant les entrées du modèle cinématique [16]. La mobilité du robot est définie par deux restrictions non holonomes [18, 68] qui sont déterminées par deux hypothèses principales :

Hypothèse 1 : Il n'y a pas de glissement latéral. Cela indique simplement que le robot est incapable de se déplacer latéralement dans son système de coordonnées, comme exprimé par l'équation (4.3) :

$$\dot{y}_{robot}(t) = 0 \quad (4.3)$$

Hypothèse 2 : Chacune des deux roues motrices du robot conserve un point de contact avec le sol. La roue ne coulisse dans aucun de ses axes longitudinaux ou orthogonaux.

Au temps discret $t = nT_0$, $x(t), y(t), v(t)$, et $\omega(t)$ seront respectivement désignés par x_n

, y_n , v_n et ω_n [22, 76, 77]. Ainsi il découle des équations cinématiques le modèle suivant :

$$\begin{cases} x_{n+1} = x_n + \int_{nT_0}^{(n+1)T_0} v(t) \cos \theta_n dt \\ y_{n+1} = y_n + \int_{nT_0}^{(n+1)T_0} v(t) \sin \theta_n dt \\ \theta_{n+1} = \theta_n + \int_{nT_0}^{(n+1)T_0} \omega(t) dt \end{cases} \quad (4.4)$$

Le modèle d'équations (4.5) est généré à l'aide de l'approximation d'Euler du modèle cinématique du robot mobile [76]. L'utilisation d'équations à temps discret permet l'implémentation du système de commande en temps réel.

$$\begin{cases} x_{n+1} = x_n + T_0 V_n \cos(\theta_n) \\ y_{n+1} = y_n + T_0 V_n \sin(\theta_n) \\ \theta_{n+1} = \theta_n + T_0 \omega_n \end{cases} \quad (4.5)$$

4.2 Description des approches de contrôles proposés

De nombreuses avancées significatives dans la conception de systèmes non linéaires pour une large gamme d'applications pratiques ont eu lieu au cours des dernières décennies. Plusieurs approches inspirantes, telles que le contrôle optimal, le contrôle adaptatif, le contrôle par mode glissant, la dynamique non linéaire, le contrôle du chaos, le contrôle de la synchronisation du chaos, le contrôle par logique floue, le contrôle adaptatif flou, le contrôle par ordre fractionnaire et le contrôle robuste, ainsi que leur intégrations, ont été proposées.

4.2.1 Commande en vitesse d'un robot mobile par un régulateur PID

La commande par PID est une stratégie de contrôle en boucle fermée largement utilisée dans la recherche industrielle et universitaire car elle est simple à appliquer et donne de bons résultats. Les actions proportionnelle, intégrale et dérivée constituent la loi de commande PID. L'objectif de la commande est de réduire le signal d'erreur représenté par la différence entre la valeur mesurée et la valeur désirée ou la consigne. L'action proportionnelle consiste à générer une action qui varie proportionnellement au signal d'erreur [112]. L'avantage du régulateur intégral est qu'il élimine l'erreur de régulation qui persistait avec un régulateur

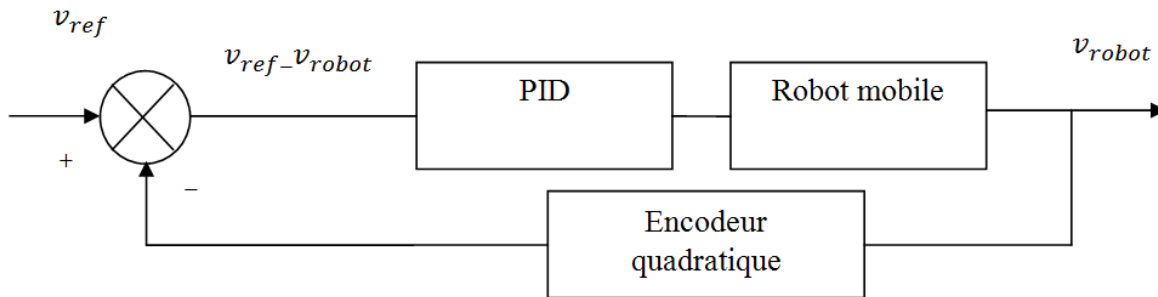


FIGURE 4.8 – Schéma de contrôle par PID

proportionnel seul. L'action dérivée permet d'anticiper les variations futures du signal de mesure en appliquant une action proportionnelle à sa vitesse de variation. Par conséquent l'action dérivée a un effet prédictif [112]. La figure (4.8) montre le schéma de contrôle.

Le régulateur traditionnel PID (Proportionnel-Intégral-Dérivé) a une structure simple ; il est largement utilisé dans la communauté de contrôle en raison de sa robustesse et de sa familiarité. Les coefficients d'un PID k_p , k_i et k_d peuvent être modifiés expérimentalement par test et erreur [23, 58]. La forme mathématique du contrôleur PID est donnée comme suit :

$$c(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt} \quad (4.6)$$

Où k_p , k_i et k_d représentent les paramètres de contrôleur PID. $e(t)$ est l'erreur entre la valeur souhaitée et la valeur mesurée et $c(t)$ est le signal de commande appliqué au système [113].

Le contrôle par un régulateur PID consiste à mémoriser l'erreur, la somme des erreurs et la différence entre l'erreur courante et l'erreur précédente. La régulation PID consiste en le choix des paramètres du régulateur de manière à réduire l'erreur à zéro et à maintenir le système rapide et stable. Pour le choix des coefficients du régulateur on ne peut pas appliquer Nichols-Ziegler. Pour cette approche le système doit déjà être réglé en boucle fermée et le fait de mettre le système dans un état oscillatoire risque de détruire notre robot. Pour cela, nous avons suivi l'organigramme donné par la figure (4.9) pour concevoir notre régulateur PID et mieux choisir ses paramètres.

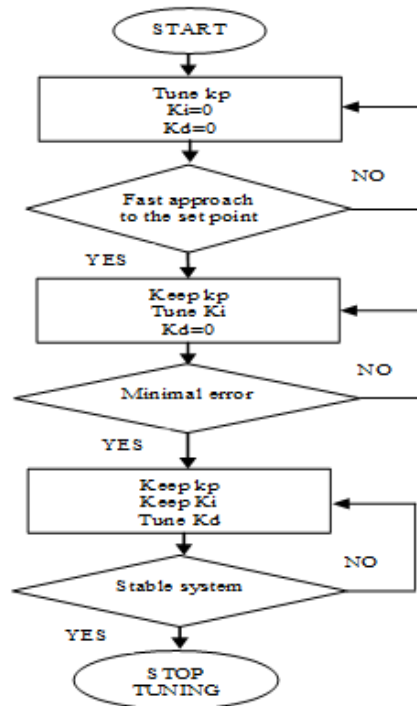


FIGURE 4.9 – Organigramme représentant le Processus de conception du régulateur PID

4.2.2 Commande en vitesse d'un robot mobile par un régulateur PID flou

Le contrôleur PID traditionnel présente les difficultés à régler ses paramètres pour atteindre l'état optimal. Tenter de modifier les trois coefficients en même temps est inutile. Il y a beaucoup de combinaisons envisageables et obtenir un bon triplet serait difficile. Cependant, il est difficile de faire correspondre les paramètres optimaux. Dans la deuxième approche de contrôle proposée, les paramètres du contrôleur PID sont déterminés par le contrôleur à logique floue auto-ajustable. Outre les mathématiques, la logique floue offre une approche beaucoup plus pragmatique. En particulier, là où les mathématiques échouent en raison de l'impossibilité ou de la complexité de la modélisation, la logique floue fonctionne. Le contrôleur à logique floue (FLC) est un système inspiré des capacités de raisonnement humain et basé sur des données linguistiques. Le concept de commande par logique floue n'est pas nouveau ; La structure de la logique floue est comparable à celle

d'un processus de jugement humain : les signaux d'entrée sont évalués subjectivement et de manière floue. Les signaux de sortie sont influencés par des règles d'inférence, qui sont considérées comme un ensemble de règles floues développées à partir de la connaissance personnelle du sujet d'étude. Des règles floues sont développées en fonction des différentes conditions de fonctionnement du système, en tenant compte des coefficients proportionnel, intégral et dérivé et en modifiant les trois paramètres. La précision d'un système de contrôle flou est influencée par des règles expertes et des fonctions d'appartenance. Par conséquent, ils doivent être bien choisis.

L'objectif principal de cette méthode est de concevoir un régulateur PID auto-ajustable par un régulateur flou à deux entrées, l'erreur de vitesse e et la variation de l'erreur de vitesse Δe acquises via l'encodeur quadratique, à chaque période d'échantillonnage, et trois sorties, à savoir : k_p , k_i et k_d qui représentent les paramètres du régulateur PID ajustés en temps réel à chaque période d'échantillonnage. Chacun des moteurs à courant continu des roues gauche et droite est contrôlé séparément par un contrôleur PID flou. La figure (4.9) représente un schéma du système de contrôle par PID flou appliqué au moteur de chaque roue. La stabilité du contrôleur PID est basée sur la positivité des paramètres PID où la conception d'un contrôleur PID flou stable pour le système de moteurs à courant continu a été réalisée sur la base du théorème de passivité [114].

$$k_p > 0, k_i > 0, k_d > 0$$

Les valeurs linguistiques de e and Δe sont [NB, NM, NS, Z, PS, PM, PB], qui correspondent à [Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, Positive Big]. Les intervalles de variations des valeurs de e and Δe sont [-50 50] et [-100 100], respectivement. Les valeurs linguistiques de k_p , k_i et k_d représentant les sorties du contrôleur flou sont [Z, S, M, B] correspondant à [Zero, Small, Medium, Big]. Les intervalles de variations de k_p , k_i et k_d sont [0 25], [0 50], [0 10], respectivement. Les fonctions d'appartenance triangulaire standard sont utilisées pour les deux entrées et les trois sorties. Les figures (4.11) (4.12) illustrent les fonctions d'appartenance des entrées floues et des sorties floues, respectivement.

Les règles floues utilisées dans le mécanisme d'inférence du contrôleur flou sont du type

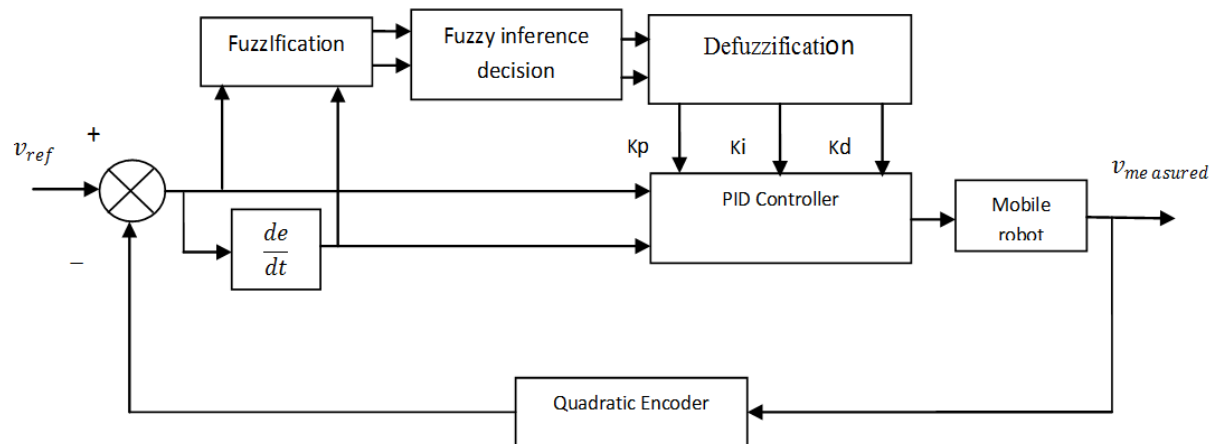


FIGURE 4.10 – Schéma du système de control par PID flou

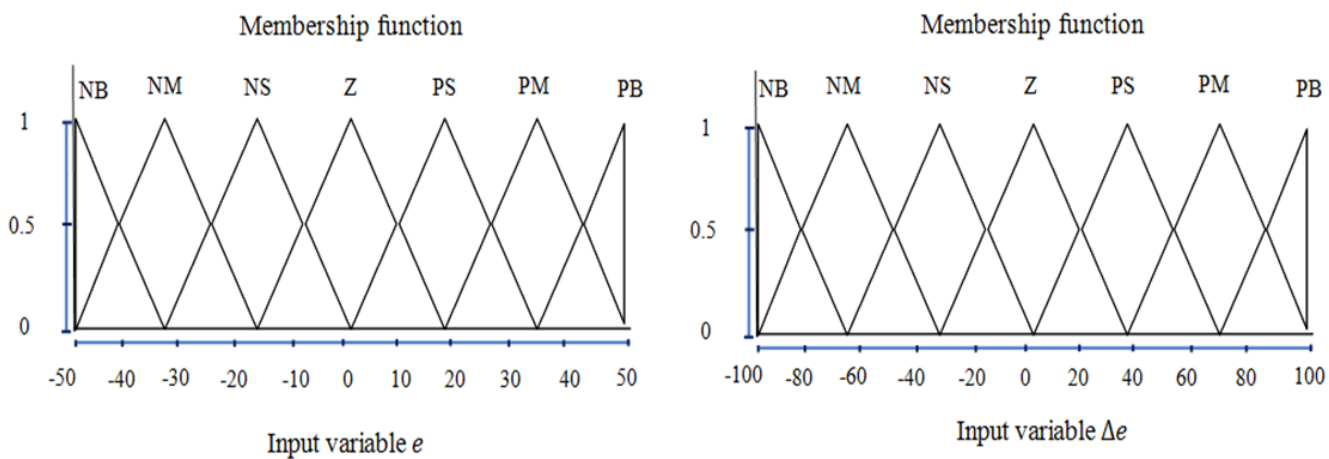


FIGURE 4.11 – Les fonctions d'appartenance de l'erreur de vitesse e et de la variation de l'erreur de vitesse Δe

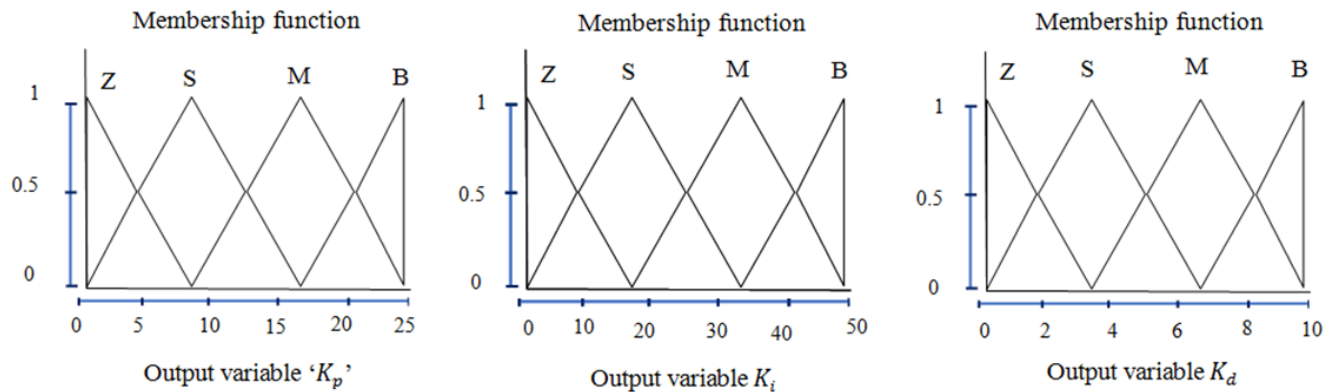


FIGURE 4.12 – Les fonctions d’appartenance des paramètres du régulateur PID k_p , k_i et k_d .

Mamdani [115], et elles sont basées sur le principe If-Then :

si $P1$ est $X1$ et $P2$ est $X2$, alors $C1$ est $Y1$ et $C2$ est $Y2$ et $C3$ est $Y3$

Les valeurs des gains sont déterminées en utilisant l’approche de défuzzification par centre de gravité. Le tableau (4.1) comprend les règles floues utilisés pour le calcul des gains du régulateur flou [116, 117].

TABLE 4.1 – Table des règles floues utilisées dans le calcul de $k_p / k_i / k_d$

e	Δe	NB	NM	NS	Z	PS	PM	PB
	NB	B/B/S	B/M/S	M/S/B	M/M/B	S/S/B	Z/Z/M	Z/Z/S
	NM	B/B/S	B/B/S	M/M/B	S/S/M	S/S/M	Z/Z/S	S/Z/Z
	NS	M/B/Z	M/M/S	M/S/M	S/S/M	S/Z/S	Z/S/S	S/S/Z
	Z	M/M/Z	M/M/S	S/S/S	Z/Z/S	S/S/S	M/M/S	M/M/Z
	PS	S/M/Z	S/S/Z	Z/Z/Z	S/S/Z	S/S/Z	M/M/Z	M/B/Z
	PM	S/Z/B	Z/Z/S	S/S/S	M/S/S	M/M/S	M/B/S	B/B/B
	PB	Z/Z/B	Z/Z/M	M/S/M	M/M/M	M/M/S	B/B/S	B/B/B

4.2.2.1 Suivi de trajectoire par robot mobile

La planification de trajectoire fait référence au mouvement du robot le long d’une trajectoire prédéfinie. Avant que le robot puisse se déplacer, la planification du chemin et la génération de la trajectoire doivent être effectuées. Le robot doit se déplacer d’un endroit du lieu de travail à un autre. Il y a eu un grand débat sur la façon de déterminer la

trajectoire d'un robot mobile autonome. Lorsque l'environnement est parfaitement connu et que le mouvement peut être planifié à l'avance, on dit qu'il est structuré. Lorsqu'il y a des incertitudes et que les mouvements doivent être choisis sur place, il est classé comme partiellement structuré. La capacité d'obtenir des informations sur l'environnement via des capteurs externes tels que des capteurs de vision, de distance ou de proximité est appelée navigation autonome. Les deux grands groupes d'approches que nous reconnaissons généralement sont les approches délibératives et réactives. L'idée derrière les approches délibératives est d'utiliser un modèle de l'environnement dans lequel le système opère pour déterminer un mouvement complet (un itinéraire ou une trajectoire) entre la position initiale et la position finale. Les approches réactives reposent uniquement sur les données collectées à travers les capteurs par le système robotique en vue de décider sur le mouvement du robot au prochain pas d'échantillonnage [117]. L'objectif de suivi de trajectoire consiste à trouver les actions de contrôle qui amènent le robot mobile vers la position souhaitée avec une orientation prédéterminée à chaque période d'échantillonnage. À la suite de ces opérations combinées, le robot mobile poursuit la trajectoire prévue. Pour atteindre cet objectif, seules deux variables de contrôle sont disponibles : la vitesse linéaire du robot v et la vitesse de rotation ω . Ainsi le problème à résoudre est d'identifier des valeurs qui permettent au robot mobile de suivre un chemin prédéterminé avec le moins d'erreurs possibles. Le vecteur des vitesses de référence a la forme suivante :

$$\dot{P}_{ref} = \begin{bmatrix} \dot{x}_{ref} \\ \dot{y}_{ref} \\ \dot{\theta}_{ref} \end{bmatrix} = \begin{bmatrix} v_{ref} \cos \theta_{ref} \\ v_{ref} \sin \theta_{ref} \\ \omega_{ref} \end{bmatrix} \quad (4.7)$$

Les coordonnées du robot mobile sont dans le repère global et peuvent être transformées en coordonnées locales à l'aide de la matrice de rotation ci-dessous [11] :

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

Dans le repère local du robot mobile , le vecteur d'erreur de position et d'orientation $E(\theta)$ est donné par l'équation (4.9)[118] :

$$E(\theta) = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = R(\theta)(p_{ref} - p) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_{ref} - x \\ y_{ref} - y \\ \theta_{ref} - \theta \end{bmatrix} \quad (4.9)$$

Où v_{ref} et ω_{ref} sont respectivement la vitesse linéaire de référence et la vitesse angulaire de référence du robot, p_{ref} et p sont respectivement les vecteurs de position et d'orientation désirée et mesurée du robot mobile.

Pour assurer la stabilité asymptotique du système, considérons la fonction de Lyapunov candidate suivante [42] :

$$L = \frac{e_x^2 + e_y^2}{2} + \frac{1 - \cos e_\theta}{k_2} \quad (4.10)$$

Avec $k_2 > 0$ et $L > 0$.

Le système prévu atteint la stabilité asymptotique lorsque $\dot{L} < 0$.

$$\dot{L} = (v_{ref} \cos e_\theta - v)e_x - \left(\frac{\omega - \omega_{ref}}{k_2} - v_{ref}e_y\right) \sin e_\theta \quad (4.11)$$

La loi de commande sera conçue comme

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1 e_x + v_{ref} \cos e_\theta \\ \omega_{ref} + k_2 v_{ref} e_y + k_3 \sin e_\theta \end{bmatrix} \quad (4.12)$$

avec :

$$\dot{L} = -k_1 e_x^2 - \frac{k_3}{k_2} \sin^2(e_\theta) < 0 \quad (4.13)$$

k_1 , k_2 and k_3 sont des paramètres réels positifs

4.2.3 Évitement d'obstacles par vision

Le robot mobile doit naviguer dans son lieu de travail sans heurter ou entrer en collision avec toute forme d'obstacle. Cela nécessite une stratégie de contrôle efficace qui garantit que l'imprévisibilité inhérente à l'environnement soit surmontée. L'évitement d'obstacle est réalisé par un processus perception-action (look then move) qui se répète à haute fréquence. La caméra recueille des données sur le robot et son environnement, qui seront utilisées pour

calculer le mouvement d'évitement d'obstacle. En conséquence, une série de mouvements conduit le robot mobile au but sans collision. Une simple méthode de commande par vision est utilisée dans cette partie pour tenter de répondre à cette question.

4.2.3.1 Retour visuel

Dans de nombreuses applications, il est très avantageux d'équiper les robots mobiles de caméras car toutes les informations sur l'environnement sont stockées dans l'image. La méthode de contrôle basée sur les informations fournies par une ou plusieurs caméras est appelée asservissement visuel. Nous avons proposé une méthode d'évitement d'obstacles en temps réel qui permet au robot de se déplacer en toute sécurité dans un environnement d'intérieur. La partie la plus importante de l'asservissement visuel est le traitement d'image. Le traitement d'image permet d'exploiter des informations extraites d'une ou plusieurs images comme données de sortie. Dans cette partie, les données visuelles sont collectées à l'aide d'une caméra monoculaire montée directement sur le robot mobile. Comme le montre la figure (4.13), son mouvement est induit par la mobilité du robot mobile. Pour poursuivre les caractéristiques dans des séquences d'images, un équilibre doit être atteint entre la précision de la prédiction de mouvement et le coût de calcul associé. Malgré le fait que les performances des systèmes informatiques se sont considérablement améliorées ces dernières années, les systèmes de surveillance sont capables d'enregistrer, d'analyser et d'afficher des résultats en temps réel en utilisant généralement une sorte de simplification pour accélérer le processus. Le traitement d'image nous permet de modifier le contenu des images afin d'extraire des informations importantes dans un but précis. Les images de la caméra ont une taille de 640 x 400 pixels, mais il n'est pas nécessaire de visualiser l'image complète. En conséquence, l'image est coupée pour créer une zone d'intérêt. Il est seulement intéressant de détecter les obstacles sur le chemin du robot sachant que la caméra est fixe et focalisée sur la zone de l'environnement devant le robot mobile. Cette zone d'intérêt est choisie d'une manière expérimentale. Comme illustré par la figure (4.14), la longueur I est déterminée par la taille du robot, tandis que la largeur i est déterminée par la vitesse du robot mobile.

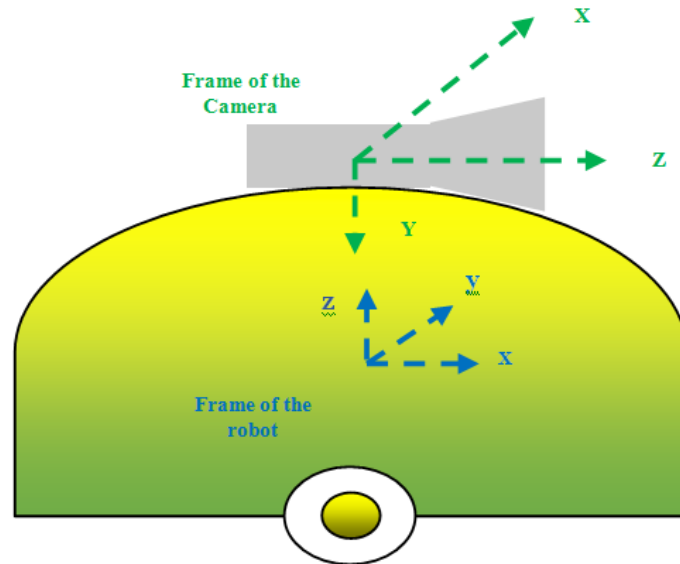


FIGURE 4.13 – Systèmes de coordonnées du robot mobile et de la caméra monoculaire

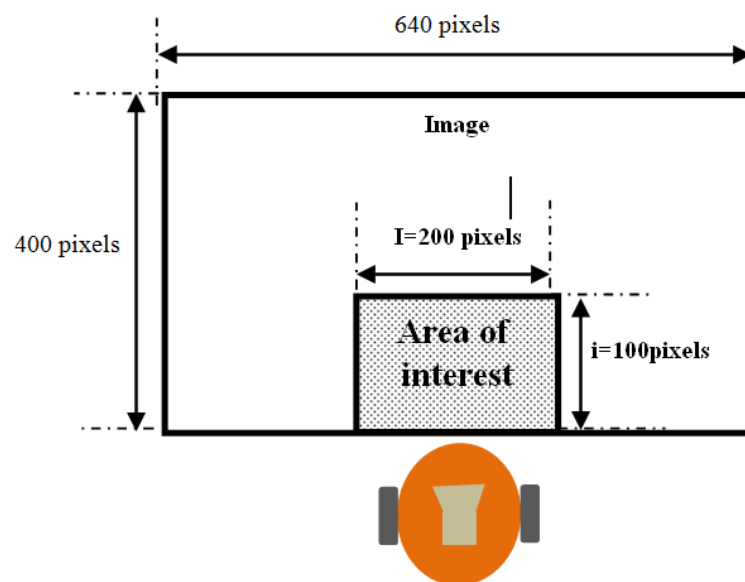


FIGURE 4.14 – Représentation de la zone d'intérêt dans l'image .

4.2.3.2 Détection d'obstacles

Le traitement d'image est le processus de modification des propriétés d'une image afin d'améliorer ses informations graphiques pour l'interprétation humaine ou de la rendre plus adaptée à la perception par la machine autonome. La différence pixel par pixel entre l'image courante et l'image de référence, qui est l'image du sol sur lequel le robot se déplace sans obstacles, est faite. l'image résultante est binarisée et convertie en valeur absolue. Le procédé de détection d'obstacle fonctionne sur la base du calcul de la somme S des intensités des pixels sur la surface blanche et de sa comparaison à un seuil déterminé empiriquement indiquant l'existence d'un obstacle Sp . Après détection de l'objet, l'image binaire a été soumise à deux opérations morphologiques : une érosion suivie d'une dilatation pour éliminer le bruit créer par le mouvement du robot et des variations d'illumination. Le centre de gravité de l'obstacle est localisé et l'obstacle est entouré d'un rayon R_{ob} qui aidera à calculer le rayon R_m du mouvement d'évitement d'obstacle.

4.2.3.3 Processus d'évitement d'obstacles

Le processus d'évitement se fait en quatre étapes suivantes

- Étape 1 : Détecter l'obstacle
- Étape 2 : Calculer le rayon du mouvement d'évitement, qui est sous forme circulaire
- Étape 3 : éviter l' obstacle
- Étape 4 : reprendre le chemin

Comme indiqué précédemment, le traitement d'image est utilisé pour détecter les obstacles. Son centre de gravité sera identifié, et l'obstacle sera entouré d'un cercle de rayon R_{ob} , qui est représenté sur la figure (4.15) par la flèche de couleur verte. Le rayon du mouvement d'évitement d'obstacle R_m est calculé à l'aide de l'équation (4.14).

$$R_m = R_o + R_s + \frac{D}{2} \quad (4.14)$$

R_o est le rayon du cercle entourant l'obstacle, R_s est la marge de sécurité et D est la distance entre les roues directrices, comme illustré à la figure (4.15). Le robot mobile suit

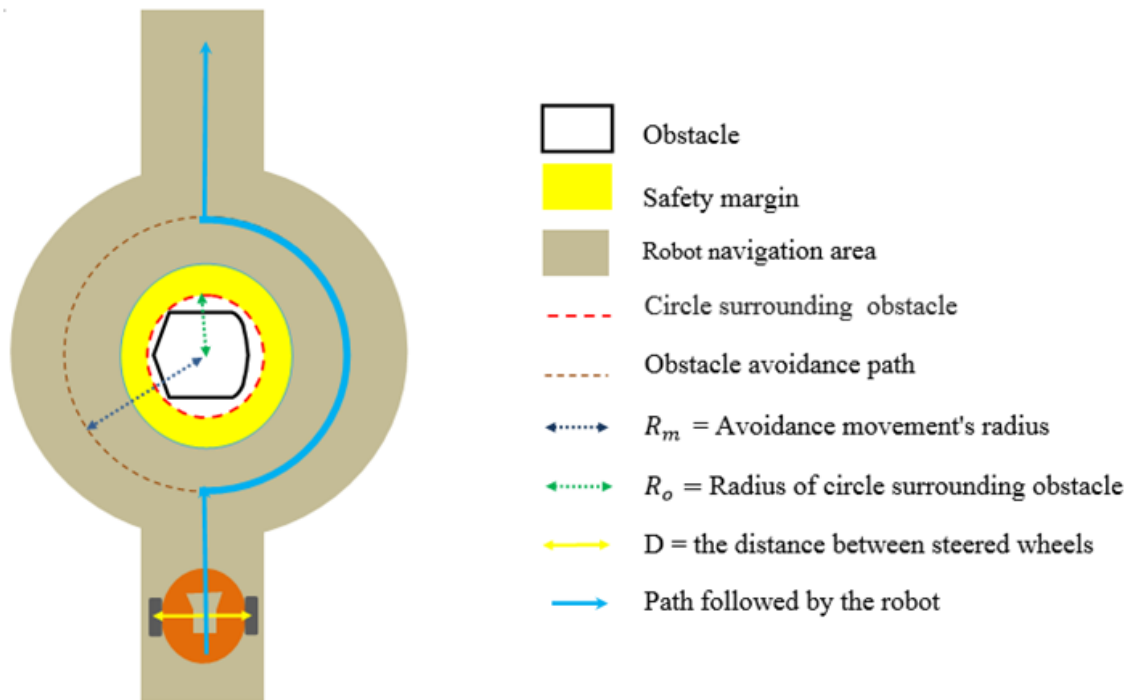


FIGURE 4.15 – Illustration de la stratégie d'évitement d'obstacles .

le chemin tracé par la ligne bleue pour éviter l'obstacle et reprend sa trajectoire initiale après l'évitement d'obstacle .

4.3 Résultats expérimentaux

Le dr Robot i90 est utilisé pour tester l'efficacité des méthodes proposées. Le robot possède deux encodeurs quadratiques qui mesurent les vitesses des deux roues droite et gauche qui sont représentées respectivement par v_r et v_l à chaque pas d'échantillonnage $T_o = 0.05$ seconde. Il dispose également d'un appareil photo à haute résolution capable de prendre 25 photos de 400 x 640 pixels par seconde. A l'aide du modèle d'actionnement donné par l' équation (4.2) , la vitesse angulaire ω et la vitesse linéaire v peuvent être calculées sachant que la distance entre roues directrices est notée D égale à 0,315 m.

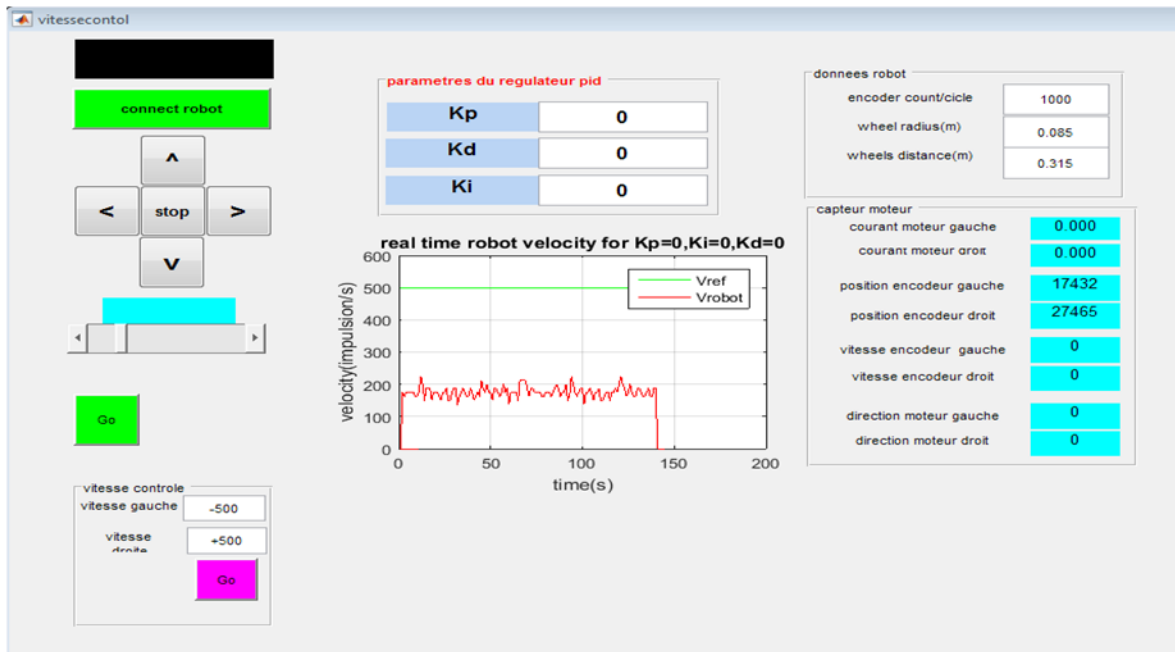


FIGURE 4.16 – Interface utilisateur de commande par PID .

4.3.1 Première expérience

Cette commande est implémentée à l'aide de MATLAB et testée sur un système de robot mobile réel Dr Robot i90 dans un environnement intérieur. MATLAB permet de construire ces interfaces grâce à GUIDE (Graphical User Interface Development Environment). Cet outil est capable de construire des applications de haut niveau. Une interface graphique permet de contrôler une application de manière interactive avec la souris plutôt qu'en lançant les commandes avec le clavier. Il permet également de cliquer sur des images, des graphiques ou des objets pour modifier la valeur d'une variable, activer des fonctions ou simplement faire apparaître des informations. L'interface utilisateur créée pour ce test est donnée par la figure (4.16)

Dans un premier temps, nous avons réalisé un test en boucle ouverte . Nous avons remarqué que la vitesse du robot mobile est loin de la vitesse désirée avec un système très instable. Les résultats sont présentés dans la figure (4.17) pour $v_{ref} = 100$ impulsions/s et pour $v_{ref} = 500$ impulsions/s dans la figure (4.18)

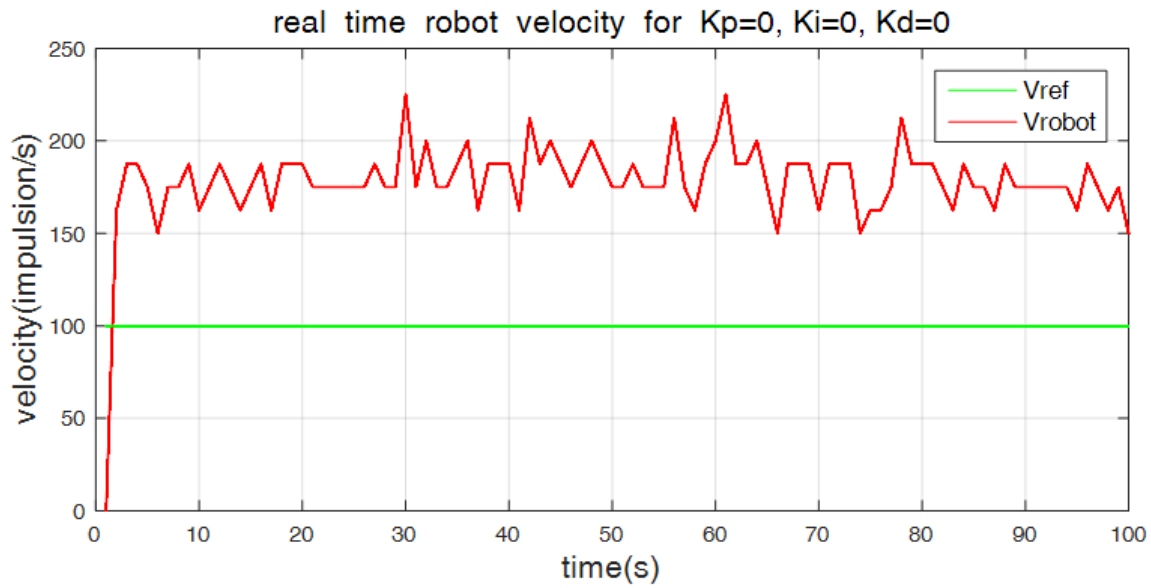


FIGURE 4.17 – Contrôle de vitesse avec $K_p=0, K_d=0$ and $K_i=0$ for $v_{ref}=100$ impulsion/s

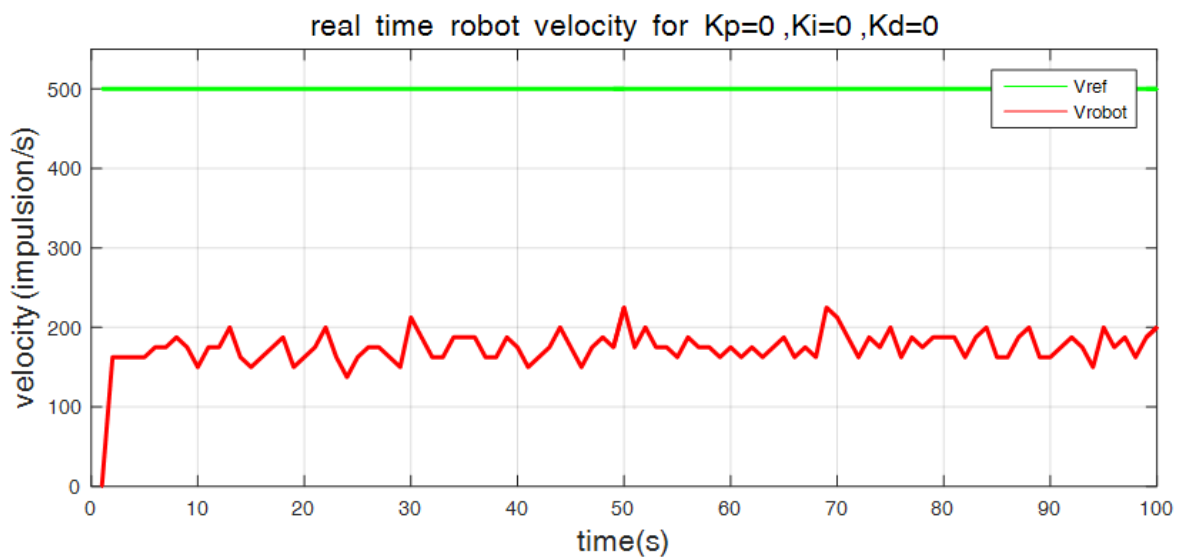


FIGURE 4.18 – Contrôle de vitesse avec $K_p=0, K_d=0$ and $K_i=0$ for $v_{ref}=500$ impulsion/s.

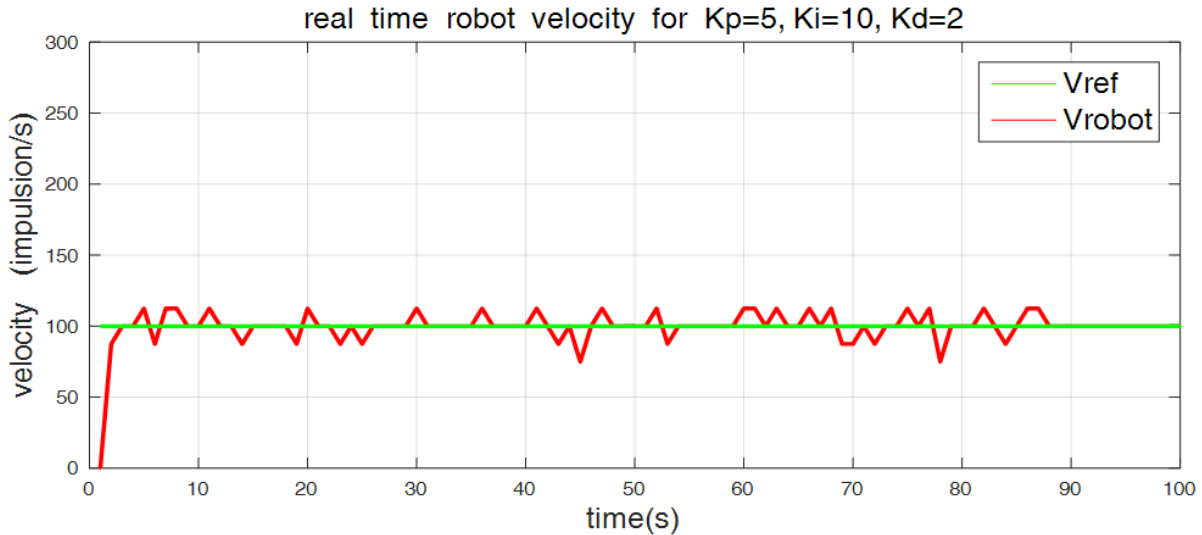


FIGURE 4.19 – Contrôle de vitesse avec $K_p=5$, $K_d=2$ and $K_i=10$ for $v_{ref}=100$ impulsions/s.

Pour le choix de nos paramètres nous avons procédé comme suit. Nous avons augmenté la valeur de K_p en gardant la valeur de K_i et K_d égale à zéro. Nous avons remarqué que la valeur de la vitesse est loin de la valeur de vitesse référence mais elle devient plus stable pour une valeur de $K_p = 5$. Ensuite on a procédé à l'augmentation de la valeur de K_i en gardant la valeur de K_d égale à zéro et on a atteint la consigne de vitesse pour $K_i = 5$. Pour les valeurs de $K_i > 10$ on note l'existence de dépassement et le robot devient instable. Nous avons opté pour le choix de la valeur de K_d de la même manière pour une valeur de $K_d = 2$ et nous constatons que le système perd sa stabilité en augmentant K_d avec des dépassements de la consigne de vitesse. Les figures (4.19) (4.20) montrent les résultats pour des valeurs de paramètres bien choisies et la figure(4.21) montre les résultats pour des paramètres mal choisis.

Nous avons proposé un contrôleur qui peut être appliqué à une large classe de systèmes. L'application de cette commande sur un robot mobile non holonome (robot réel) a permis de mettre en évidence la robustesse du contrôleur PID. En essayant d'améliorer les performances de vitesse du robot mobile Dr robot i90 et nous avons implémenté le régulateur PID sans calcul mais en modifiant les paramètres PID tout en essayant de garder notre système stable et sain. Mais l'application d'un régulateur flou pour le choix de ces paramètres peut

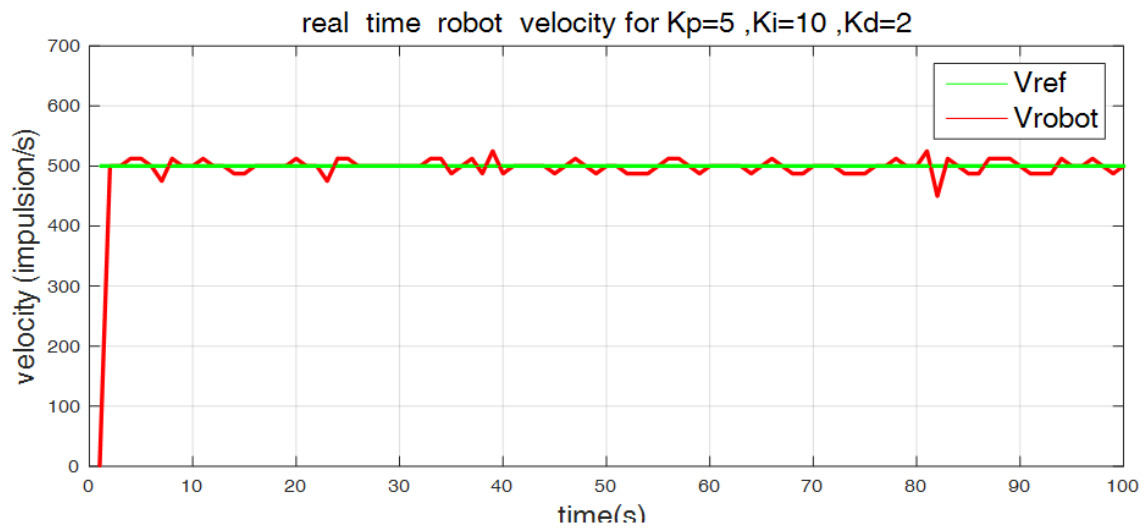


FIGURE 4.20 – Contrôle de vitesse avec $K_p=5$, $K_d=2$ and $K_i=10$ for $v_{ref}=500$ impulses/s.

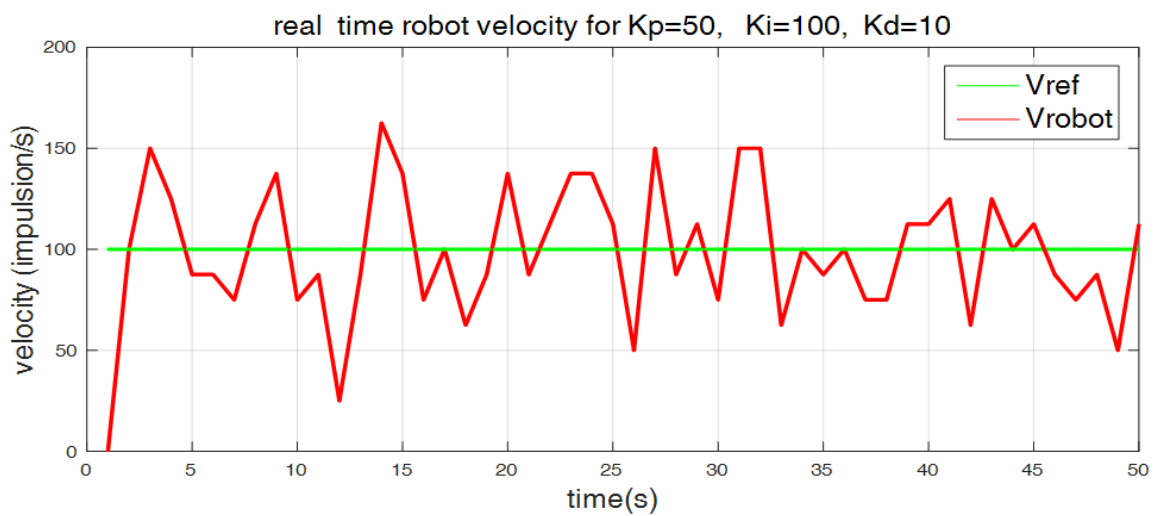


FIGURE 4.21 – Contrôle de vitesse avec $K_p=50$, $K_d=10$ and $K_i=100$ for $v_{ref}=100$ impulses/s.

être une meilleure solution.

4.3.2 Deuxième expérience

Cet expérience a été réalisée pour valider l'efficacité du contrôleur PID flou à maintenir la vitesse du robot mobile constante et égale à la valeur désirée. L'interface de utilisateur pour le contrôle de vitesse par le PID flou est illustrée à la figure (4.22). Étant donné que certains robots mobiles perdent leurs performances pour des vitesses élevées, deux vitesses linéaires ont été choisies. Les courbes de vitesse du robot sont données sur la figure(4.23) pour $v_{ref} = 200$ impulsions par seconde et sur la figure (4.24) pour $v_{ref} = 500$ impulsions par seconde.

Un deuxième test a été réalisé pour évaluer l'efficacité du contrôleur proposé. Le but est de permettre au robot mobile de suivre une trajectoire prédéfinie, et pour cela un ensemble d'équations a été créé en utilisant l'approximation d'Euler du modèle cinématique du robot mobile. L'utilisation d'un modèle cinématique discret pour calculer l'emplacement du robot mobile simplifie la mise en œuvre de l'approche suggérée. Une trajectoire circulaire a été utilisée comme itinéraire planifié. Le robot commence sa trajectoire à $(0, 0)$ avec une orientation initiale de zéro degré. Il se déplace avec $v_r = 500$ impulsions/seconde et $v_l = 350$ impulsions par seconde, où v_r et v_l sont les vitesses des roues droite et gauche, respectivement, avec un pas d'échantillonnage de $T_o = 0,05$ seconde. Dans cette expérience, le contrôle de la vitesse du robot mobile Dr robot i90 a été comparé entre le contrôle par un simple PID et le contrôle par un PID flou, et les courbes de vitesse indiquent clairement l'amélioration que le contrôle flou apporte en termes de stabilité et de précision du robot à maintenir la vitesse requise. Nous remarquons également que, même à grande vitesse, le robot fonctionne mieux avec un contrôleur PID flou. Les courbes de vitesse du robot contrôlé par contrôleur PID flou présentent des oscillations mineures, qui pourraient être causées par une petite différence de rayon entre les roues gauche et droite, ou causées par le sol poussiéreux et qui n'est pas complètement de niveau. Deux tests ont été effectués pour valider le suivi de trajectoire en comparant le suivi de trajectoire par dr robot i90 contrôlé par PID conventionnel au suivi d'un robot contrôlé par PID flou. Les résultats



FIGURE 4.22 – Interface utilisateur créée pour le contrôle avec un PID flou.

montrent que le suivi de trajectoire par le robot est réalisé efficacement lors des tests. Ce test montre la différence entre les contrôles par PID de la figure(4.25) (a) et le contrôle PID flou de la figure (4.25)(b), ainsi que le suivi presque parfait de la trajectoire souhaitée dans le test de contrôle par un PID flou de la figure(4.25)(b) .

4.3.3 Troisième expérience

Comme indiqué précédemment, le dr robot i90 est équipé d'une caméra à haute résolution. Pour valider la méthode proposée d'évitement d'obstacles par vision et pour prouver que l'utilisation de la vision comme outil de détection du système est une bonne idée, il fallait laisser le robot naviguer en toute sécurité d'un point de départ à une position finale en suivant un chemin bien défini sans heurter des objets ou des obstacle se trouvant dans son environnement. Le but de ce test est de faire naviguer le robot en suivant un itinéraire en ligne droite tout en reconnaissant simultanément l'existence d'un obstacle sur son chemin. Lorsqu'un obstacle est détecté, le robot s'arrête et fait l'extraction des informa-

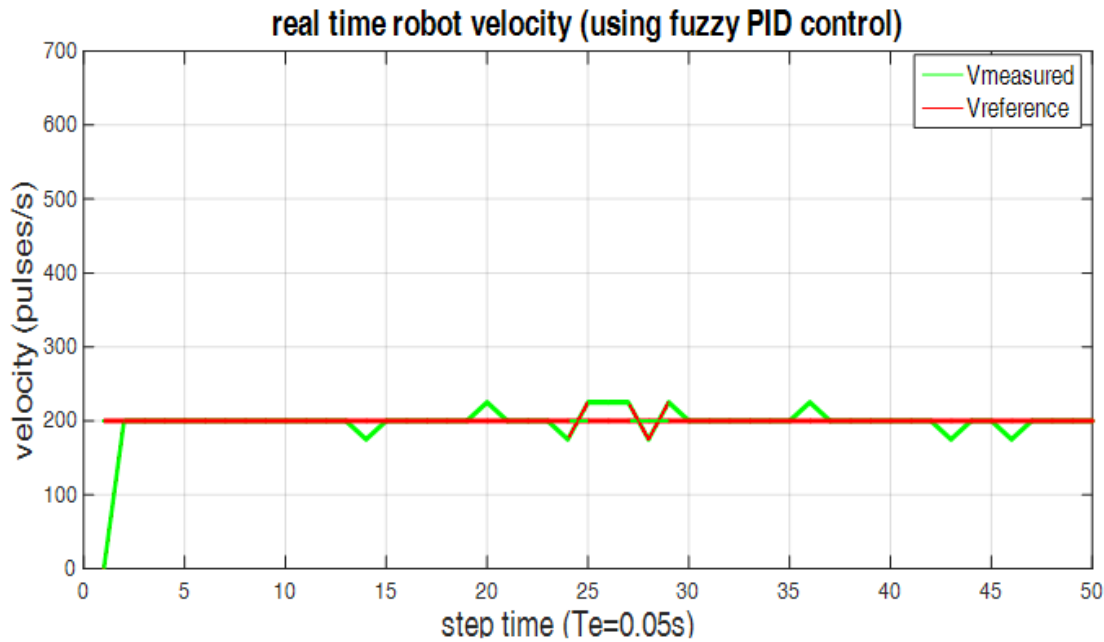


FIGURE 4.23 – La vitesse du robot en temps réel utilisant le contrôle PID flou pour la valeur désirée $v_{ref}=200$ impulsions/sec

tions visuelles nécessaires au processus de l'évitement d'obstacle. La figure (4.26) montre l'interface utilisateur créée par Guide de MATLAB pour ce test.

L'approche de soustraction de fond est utilisée pour détecter les obstacles. La figure (4.14) montre comment l'image a été coupée pour créer une zone d'intérêt avec une taille fixe de 200x100 pixels. La figure (4.27) représente l'environnement sous forme d'image d'arrière-plan, d'image avec l'obstacle, de sous-image de référence binaire et de sous-image binaire avec l'obstacle, ainsi que le résultat de la différence entre les sous-images binaires et le résultat après l'application des filtres morphologiques. La figure (4.28) affiche l'identification de l'obstacle ainsi que l'emplacement de son centre de gravité tel qu'indiqué par le cercle qui l'entoure.

La figure (4.29) montre des photos prises lors de l'expérience réalisée pour valider la méthode d'évitement d'obstacle suggérée. Il montre une série d'images prises tout au long de l'expérience qui décrivent la méthode d'évitement d'obstacles effectuée par le robot i90. Dans l'image 1, le robot avance et suit sa trajectoire pré-planifiée, dans l'image 2, le

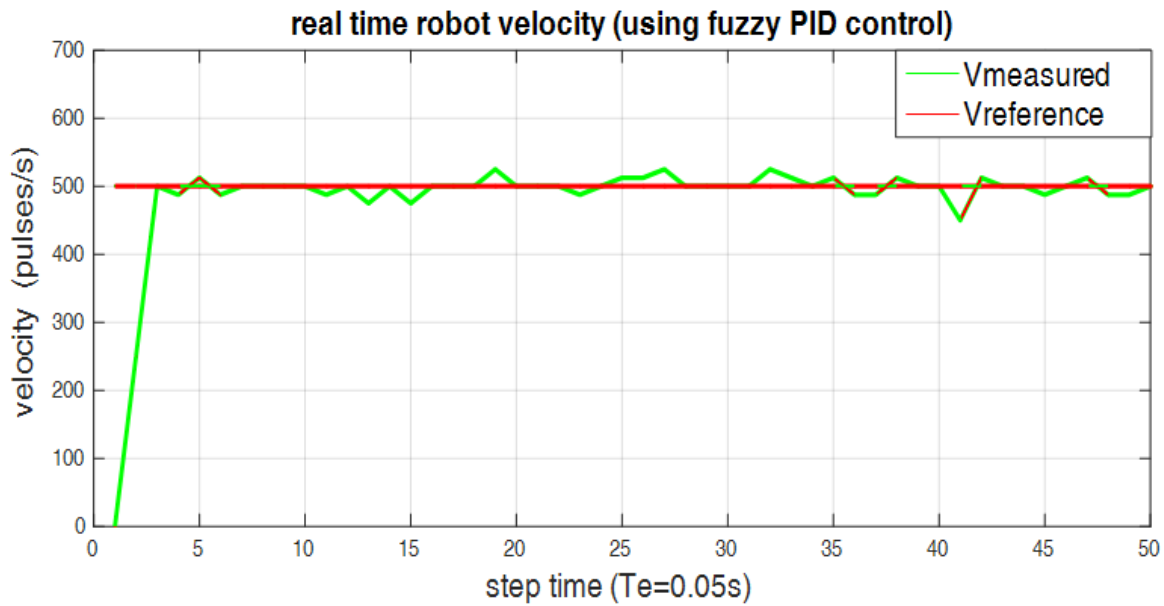


FIGURE 4.24 – La vitesse du robot en temps réel utilisant le contrôle PID flou pour la valeur désirée $v_{ref}=500$ impulsions/s.

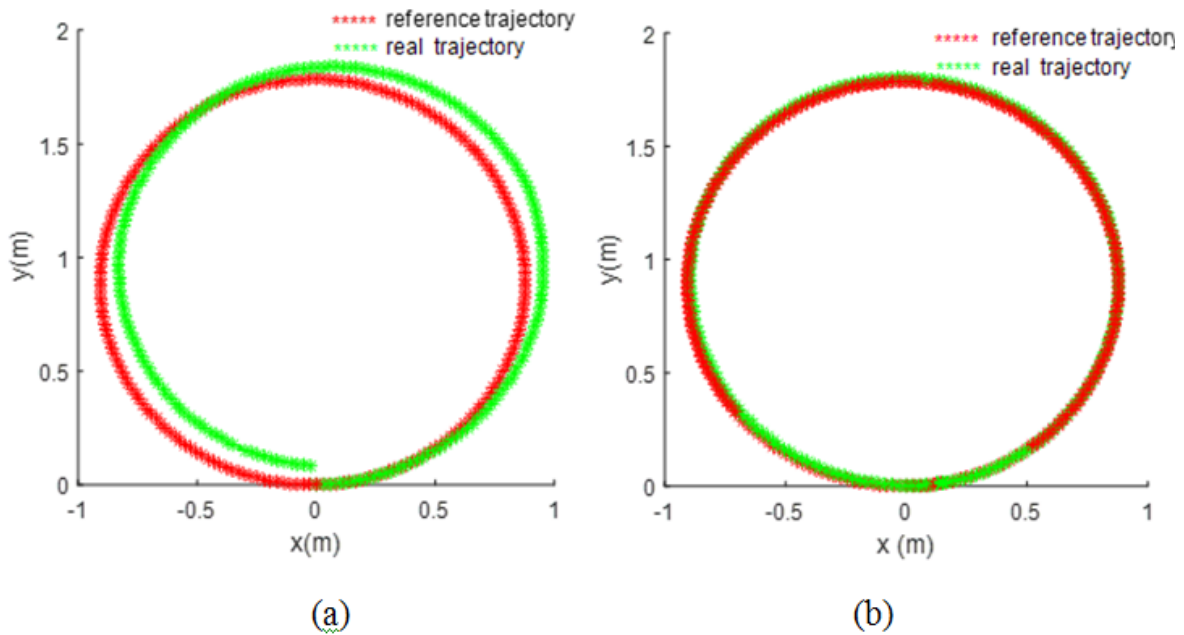


FIGURE 4.25 – Suivi de trajectoire du robot dr i90 à l'aide du contrôle PID (a). Suivi de trajectoire de dr robot i90 à l'aide d'un contrôle PID flou (b).



FIGURE 4.26 – Interface utilisateur créée pour la navigation du robot i90 avec évitement d’obstacles

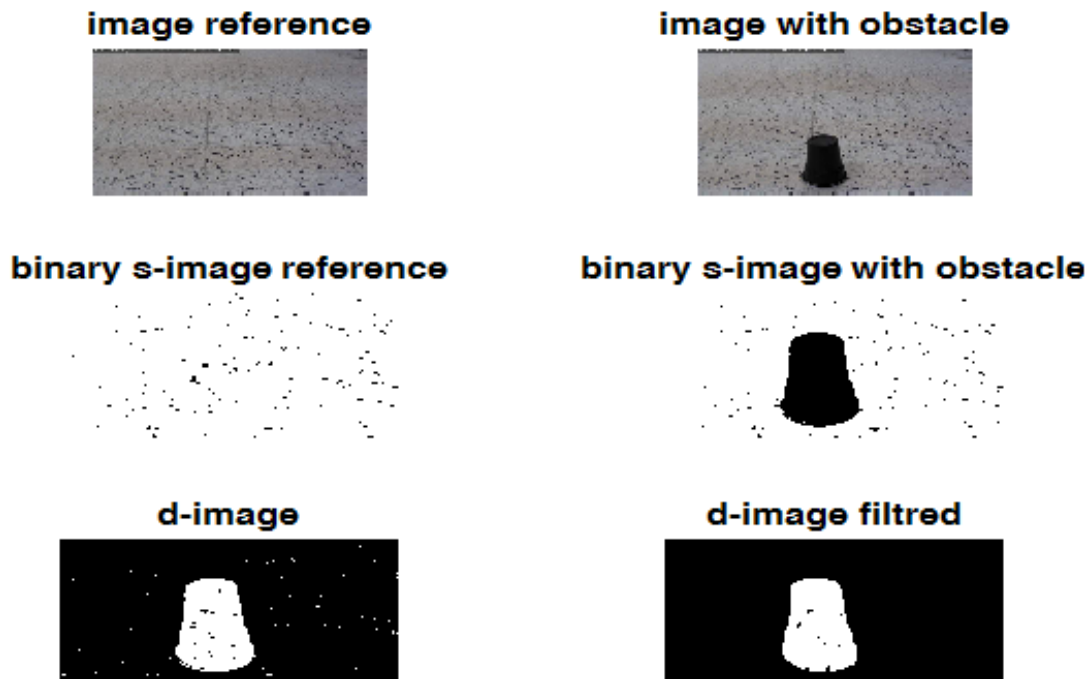


FIGURE 4.27 – Les étapes de traitement d’image utilisée pour la détection d’obstacle

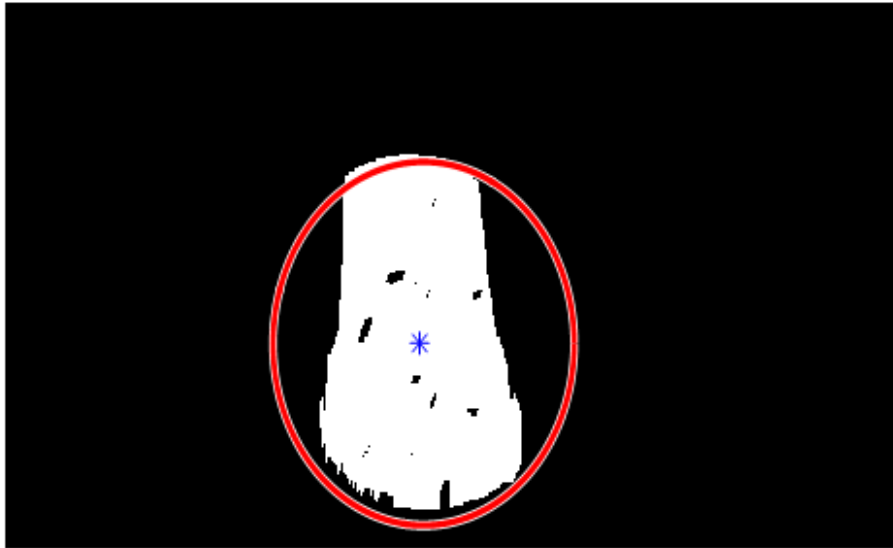


FIGURE 4.28 – Identification du centre de gravité de l’obstacle détecté et de sa circonférence

robot détecte l’obstacle et s’arrête . Dans la trame 3, le robot change de direction .Il suit la trajectoire circulaire calculée à partir des informations extraites de l’image de l’obstacle détecté dans les trames 4 et 5. Le robot s’arrête dans l’image 6, il change de direction dans l’image 7. le robot poursuit sa trajectoire initiale dans l’image 8. Les photographies montrent que l’évitement d’obstacles a été réalisé avec succès.

Conclusion

L’implémentation de contrôleur PID floue en vitesse sur un robot mobile réel non autonome a permis le réajustement automatique des paramètres du PID en temps réel, augmentant la sensibilité et la précision du système. Les résultats révèlent que le régulateur PID flou surpasse le régulateur PID simple. Les travaux futurs tentent d’améliorer le système en raccourcissant la durée de l’échantillon, malgré le fait qu’un système de traitement plus puissant est nécessaire. La deuxième expérience, basée sur le traitement d’images par région d’intérêt, permet au robot mobile de se déplacer de manière autonome. Le principal avantage de cette méthode est qu’elle ne nécessite pas de modèles mathématiques du robot et de son environnement, ni l’utilisation d’un autre système odométrique. Malgré le fait

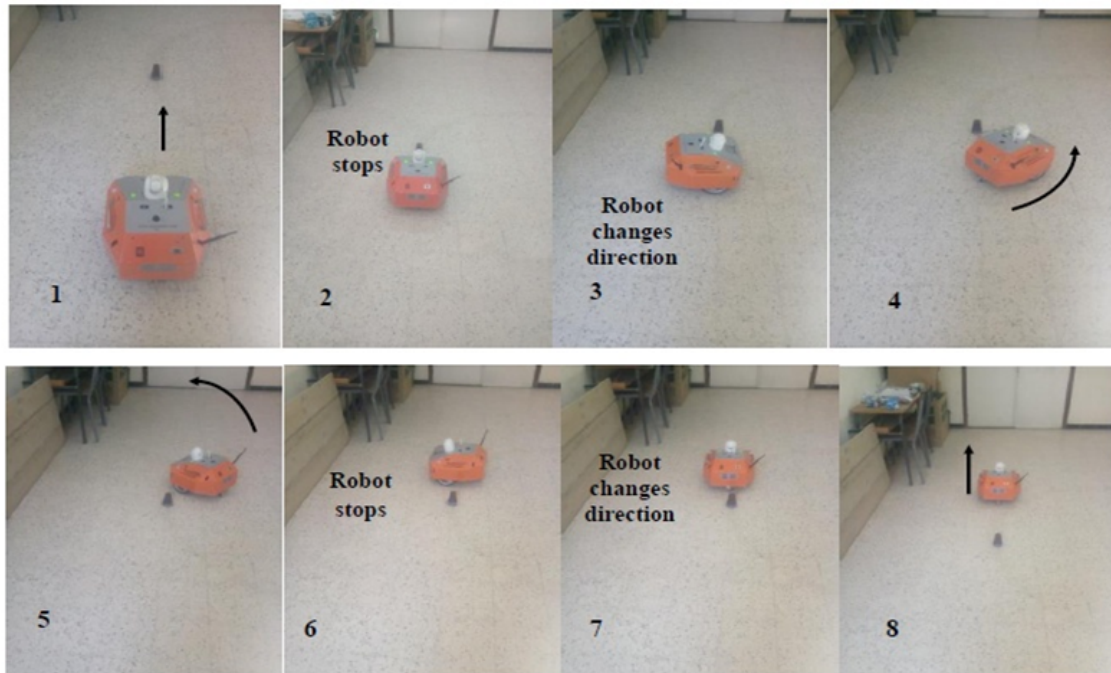


FIGURE 4.29 – Série de photographies montrant le dr robot i90 évitant un obstacle

que cette méthode soit simple, les résultats des essais ont prouvé son utilité. Les recherches futures devraient se concentrer sur l'extension de cette méthode à la navigation dans un environnement dynamiques.

Conclusion générale et perspectives

Un robot tel que défini par Peter Corke est une machine orientée vers un but capable de détecter son environnement et d'utiliser les données pour planifier certaines actions. Le défi fondamental que de nombreux chercheurs ont surmonté dans le monde de la robotique est de savoir contrôler un robot réel.

Les travaux présentés dans cette thèse ont porté sur la navigation efficace et sûre d'un robot mobile à roues non holonome dans un environnement d'intérieur tout en évitant des obstacles. Ainsi, deux méthodes ont été présentées, dont la première consiste à utiliser un contrôleur PID flou pour permettre au robot mobile la navigation à la vitesse désirée, et la seconde consiste à faire naviguer le robot en utilisant les techniques de vision pour assurer l'évitement des obstacles. Le problème est de savoir comment faire déplacer le robot dans une zone connue ou inconnue tout en évitant les obstacles. Il est primordial pour un robot mobile de naviguer à une vitesse précise et bien définie en fonction de l'activité et de l'environnement. Dans ce travail, le robot mobile différentiel est contrôlé en vitesse à l'aide d'un contrôleur PID flou, qui combine un contrôleur flou avec un PID conventionnel dans le but de réajuster les gains de ce dernier. L'environnement peut contenir une gamme d'obstacles de forme et de taille arbitraires. Une caméra monoculaire est utilisée pour détecter les obstacles pendant le processus de navigation. Pour assurer le déplacement d'un robot mobile dans un espace d'intérieur tout en évitant les obstacles, une approche visuelle qui combine deux méthodes dont la première est basée sur le traitement d'image par la zone d'intérêt en vue de détecter les obstacles et la deuxième est inspirée des algorithmes des bugs pour l'évitement d'obstacles. L'objectif de cette recherche est de développer et d'implémenter un régulateur de vitesse appelé PID flou sur un robot réel, ainsi qu'un sys-

tème visuel simple et efficace d'évitement d'obstacles. Le logiciel MATLAB est utilisé pour implémenter le système de contrôle. GUIDE (Graphical User Interface Development Environment) a permis la création d'interfaces utilisateurs graphiques. Ces interfaces facilitent la manipulation et la visualisation du système de contrôle en temps réel. Pour valider les approches proposées, des expériences ont été réalisées en milieu interne avec le robot mobile non holonome de Robot i90.

Le contrôleur proposé pourrait être utilisé pour contrôler une variété de systèmes. Dans notre cas, le contrôleur est utilisé pour contrôler en vitesse la navigation d'un robot mobile de telle manière que les paramètres d'un régulateur PID sont ajustés en temps réel par la logique floue, dans le but d'augmenter la sensibilité et la précision du système. En termes de précision, et de robustesse, les résultats révèlent que le régulateur PID flou surpasse le régulateur PID simple. Les travaux futurs tentent d'améliorer le système en raccourcissant la période d'échantillonnage, et pour cela un système de traitement puissant est nécessaire. La deuxième méthode, permet au robot mobile une navigation visuelle autonome, la méthode de la détection d'obstacles est basée sur le traitement d'image par zone d'intérêt, par contre l'évitement est inspiré des méthodes d'évitement d'obstacle par les algorithmes des Bug. Le principal avantage de cette méthode est qu'elle ne nécessite pas de modèles mathématiques du robot et de son environnement, ni l'utilisation d'un autre système d'odométrie ou un capteur de position. Quoique cette méthode soit simple, les résultats des essais ont prouvé son utilité. Les recherches futures devraient se concentrer sur l'extension de cette méthode aux obstacles dynamiques.

Bibliographie

- [1] Spyros G Tzafestas. *Introduction to mobile robot control*. Elsevier, 2013.
- [2] Surachai Panich and Nitin Afzulpurkar. Mobile robot integrated with gyroscope by using ikf. *International Journal of Advanced Robotic Systems*, 8(2) :22, 2011.
- [3] Howie Choset and Joel Burdick. Sensor based planning. i. the generalized voronoi graph. In *Proceedings of 1995 IEEE international conference on robotics and automation*, volume 2, pages 1649–1655. IEEE, 1995.
- [4] Evgeni Magid, Roman Lavrenov, and Ilya M. Afanasyev. Voronoi-based trajectory optimization for ugv path planning. *2017 International Conference on Mechanical, System and Control Engineering (ICMSC)*, pages 383–387, 2017.
- [5] Michael A Goodrich. Potential fields tutorial. *Class Notes*, 157, 2002.
- [6] SM LaValle and D Leidner. Chapter 6 : Combinatorial motion planning. *Planning Algorithms*, pages 249–310, 2006.
- [7] Muhammad Zohaib, Syed Mustafa Pasha, Nadeem Javaid, and Jamshed Iqbal. Intelligent bug algorithm (iba) : A novel strategy to navigate mobile robots autonomously. *arXiv preprint arXiv :1312.4552*, 2013.
- [8] Alaa Khamis, Jun Meng, Jin Wang, Ahmad Taher Azar, Edson Prestes, Howard Li, Ibrahim A Hameed, and Tamas Haidegger. Ai and robotics in the fight against covid-19 pandemic. *Modeling, control and drug development for COVID-19 outbreak prevention*, pages 57–85, 2022.
- [9] Fatin Hassan Ajeil, Ibraheem Kasim Ibraheem, Ahmad Taher Azar, and Amjad J Humaidi. Autonomous navigation and obstacle avoidance of an omnidirectional mo-

- bile robot using swarm optimization and sensors deployment. *International Journal of Advanced Robotic Systems*, 17(3) :1729881420929498, 2020.
- [10] Fatin Hassan Ajeil, Ibraheem Kasim Ibraheem, Ahmad Taher Azar, and Amjad J Humaidi. Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments. *Sensors*, 20(7) :1880, 2020.
- [11] Ghusn Abdul Redha Ibraheem, Ahmad Taher Azar, Ibraheem Kasim Ibraheem, and Amjad J Humaidi. A novel design of a neural network-based fractional pid controller for mobile robots using hybridized fruit fly and particle swarm optimization. *Complexity*, 2020 :1–18, 2020.
- [12] Hossam Ammar, Mohammed Ibrahim, Ahmad Azar, and Raafat Shalaby. Gray wolf optimization of fractional order control of 3-omni wheels mobile robot : Experimental study. In *2020 16th International Computer Engineering Conference (ICENCO)*, pages 147–152. IEEE, 2020.
- [13] Hossam Hassan Ammar and Ahmad Taher Azar. Robust path tracking of mobile robot using fractional order pid controller. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019) 4*, pages 370–381. Springer, 2020.
- [14] Mohamed Hesham Barakat, Ahmad Taher Azar, and Hossam Hassan Ammar. Agricultural service mobile robot modeling and control using artificial fuzzy logic and machine vision. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019) 4*, pages 453–465. Springer, 2020.
- [15] Senka Krivić, Aida Mrzić, and Nedim Osmić. Building mobile robot and creating applications for 2d map building and trajectory control. In *2011 Proceedings of the 34th International Convention MIPRO*, pages 1712–1717. IEEE, 2011.
- [16] Ahmad A Mahfouz, Ayman A Aly, and Farhan A Salem. Mechatronics design of a mobile robot system. *International Journal of Intelligent Systems and Applications*, 5(3) :23–36, 2013.

- [17] Erkan Kayacan and Girish Chowdhary. Tracking error learning control for precise mobile robot path tracking in outdoor environment. *Journal of Intelligent & Robotic Systems*, 95 :975–986, 2019.
- [18] Fethi Matoui, Boumedyen Boussaid, and Mohamed Naceur Abdelkrim. Distributed path planning of a multi-robot system based on the neighborhood artificial potential field approach. *Simulation*, 95(7) :637–657, 2019.
- [19] Songmin Jia, Hao Yang, Xiuzhi Li, and Wei Fu. Lrf-based data processing algorithm for map building of mobile robot. In *The 2010 IEEE International Conference on Information and Automation*, pages 1924–1929. IEEE, 2010.
- [20] Lan Anh Trinh, Mikael Ekström, and Baran Cürüklü. Dependable navigation for multiple autonomous robots with petri nets based congestion control and dynamic obstacle avoidance. *Journal of Intelligent & Robotic Systems*, 104(4) :69, 2022.
- [21] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking : A survey. *Acm computing surveys (CSUR)*, 38(4) :13–es, 2006.
- [22] KH Ng, CF Yeong, ELM Su, and AR Husain. Implementation of cascade control for wheeled mobile robot straight path navigation. In *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)*, volume 2, pages 503–506. IEEE, 2012.
- [23] Saurabh Sharma and Sheilza Jain. Speed control of mobile robotic system using pi, pid and pole placement controller. In *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pages 1–5. IEEE, 2016.
- [24] Jong Jin Park, Seungwon Lee, and Benjamin Kuipers. Discrete-time dynamic modeling and calibration of differential-drive mobile robots with friction. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6510–6517. IEEE, 2017.

- [25] Hao Sun, Han Zhao, Shengchao Zhen, Kang Huang, Fumin Zhao, Xianmin Chen, and Ye-Hwa Chen. Application of the udwadia–kalaba approach to tracking control of mobile robots. *Nonlinear Dynamics*, 83 :389–400, 2016.
- [26] Haoxiang Lang, Ying Wang, and Clarence W de Silva. Visual servoing with lqr control for mobile robots. In *IEEE ICCA 2010*, pages 317–321. IEEE, 2010.
- [27] Sabrina Mohand Saidi and Rabah Mellah. Mobile robot environment map building, trajectory tracking and collision avoidance applications. In *2019 International Conference on Advanced Electrical Engineering (ICAEE)*, pages 1–5. IEEE, 2019.
- [28] Michel Lopez-Franco, Edgar N Sanchez, Alma Y Alanis, and Carlos López-Franco. Neural control for a differential drive wheeled mobile robot integrating stereo vision feedback. *Computación y Sistemas*, 19(3) :429–443, 2015.
- [29] Nafees Ahamad, Shailesh Uniyal, Afzal Sikander, and Gagan Singh. A comparative study of pid controller tuning techniques for time delay processes. *UPB Scientific Bulletin, Series C : Electrical Engineering and Computer Science*, 81(3) :129–142, 2019.
- [30] Yun Li, Kiam Heong Ang, and Gregory CY Chong. Patents, software, and hardware for pid control : an overview and analysis of the current art. *IEEE Control Systems Magazine*, 26(1) :42–54, 2006.
- [31] Francisco Rossomando and Carlos Soria. Design and implementation of adaptive neural pid for non linear dynamics in mobile robots. *IEEE Latin America Transactions*, 13(4) :913–918, 2015.
- [32] Wei Fan and Tao Chen. Design and implementation of fuzzy pid controller for brushless dc motor control system. In *Applied Mechanics and Materials*, volume 432, pages 472–477. Trans Tech Publ, 2013.
- [33] Tufan Kumbasar and Hani Hagraas. A self-tuning zsllices-based general type-2 fuzzy pi controller. *IEEE Transactions on Fuzzy Systems*, 23(4) :991–1013, 2014.

- [34] Houcine Zerfa and Wahid Nouibat. Fuzzy reactive navigation for autonomous mobile robot with an offline adaptive neuro fuzzy system. In *3rd international conference on systems and control*, pages 950–955. IEEE, 2013.
- [35] Sun Jun, Zhang Meixia, Li Zhengming, and Wu Xiaohong. Simulation of smith fuzzy pid temperature control in enzymatic detection of pesticide residues. *International Journal of Agricultural and Biological Engineering*, 8(1) :50–56, 2015.
- [36] SEYED KAMALEDDIN Mousavi Mashhadi, Hamid Yadollahi, and ABBAS MASHHAD. Design and manufacture of tds measurement and control system for water purification in reverse osmosis by pid fuzzy logic controller with the ability to compensate effects of temperature on measurement. *Turkish Journal of Electrical Engineering and Computer Sciences*, 24(4) :2589–2608, 2016.
- [37] Yan Xiang Wu, Chao Jun Zhang, Hai Bo Huo, and Chao Qun Zhou. Design and simulation of fuzzy-pid dc governor system based on mine hoist. In *Applied Mechanics and Materials*, volume 341, pages 834–838. Trans Tech Publ, 2013.
- [38] Huma Khan, Shahida Khatoun, Prerna Gaur, and Salman Ahmad Khan. Speed control comparison of wheeled mobile robot by anfis, fuzzy and pid controllers. *International Journal of Information Technology*, 14(4) :1893–1899, 2022.
- [39] Najah Yousfi Allagui, Farhan A Salem, Awad M Aljuaid, et al. Artificial fuzzy-pid gain scheduling algorithm design for motion control in differential drive mobile robotic platforms. *Computational Intelligence and Neuroscience*, 2021, 2021.
- [40] Chiraz Ben Jabeur and Hassene Seddik. Design of a pid optimized neural networks and pd fuzzy logic controllers for a two-wheeled mobile robot. *Asian Journal of Control*, 23(1) :23–41, 2021.
- [41] The Anh Mai, Thai Son Dang, Dinh Tu Duong, Van Chuong Le, and Santo Banerjee. A combined backstepping and adaptive fuzzy pid approach for trajectory tracking of autonomous mobile robots. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 43 :1–13, 2021.

- [42] Jean-Blaise Mvondo Zanga, Bertrand Moffo Lonla, Arnaud Nanfak, Gildas Martial Ngaleu, et al. Fuzzy-fopid control for tracking the trajectory of nonholonomic wheeled mobile robot. *Journal of Electrical Engineering, Electronics, Control and Computer Science*, 8(2) :29–38, 2021.
- [43] RMNB Ratnayake, TS De Silva, and CJ Rodrigo. A comparison of fuzzy logic controller and pid controller for differential drive wall-following mobile robot. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 523–528. IEEE, 2019.
- [44] Jefferson Campos, Sebastián Jaramillo, Luis Morales, Oscar Camacho, Danilo Chávez, and David Pozo. Pso tuning for fuzzy pd+ i controller applied to a mobile robot trajectory control. In *2018 International Conference on Information Systems and Computer Science (INCISCOS)*, pages 62–68. IEEE, 2018.
- [45] Mohamed Boukens, Abdelkrim Boukabou, and Mohammed Chadli. A real time self-tuning motion controller for mobile robot systems. *IEEE/CAA Journal of Automatica Sinica*, 6(1) :84–96, 2018.
- [46] R Hartono and TN Nizar. Speed control of a mobile robot using fuzzy logic controller. In *IOP Conference Series : Materials Science and Engineering*, volume 662, page 022063. IOP Publishing, 2019.
- [47] Abdurrahman HALABI, Mert Ezim, Kansu OĞUZ Canbek, and Abdurrahman Eray Baran. A multiple sensor fusion based drift compensation algorithm for mecanumwheeled mobile robots. *Turkish Journal of Electrical Engineering and Computer Sciences*, 29(2) :704–716, 2021.
- [48] Mehmet Serdar Guzel and Robert Bicker. Vision based obstacle avoidance techniques. *Recent advances in mobile robotics*, pages 83–108, 2011.
- [49] Boguslaw Cyganek. *Object detection and recognition in digital images : theory and practice*. John Wiley & Sons, 2013.

- [50] François Pasteau, Vishnu K Narayanan, Marie Babel, and François Chaumette. A visual servoing approach for autonomous corridor following and doorway passing in a wheelchair. *Robotics and Autonomous Systems*, 75 :28–40, 2016.
- [51] Athanasios Tsalatsanis, K Valavanis, and Ali Yalcin. Vision based target tracking and collision avoidance for mobile robots. *Journal of Intelligent and Robotic Systems*, 48 :285–304, 2007.
- [52] Alexander Woodward, David Berry, and James Dunning. Real-time stereo vision on the visionserver framework for robot guidance. In *2010 25th International Conference of Image and Vision Computing New Zealand*, pages 1–8. IEEE, 2010.
- [53] William Benn and Stanislaw Lauria. Robot navigation control based on monocular images : an image processing algorithm for obstacle avoidance decisions. *Mathematical Problems in Engineering*, 2012, 2012.
- [54] Mohamed Boumehraz, Zineb Habba, and Rafia Hassani. Vision based tracking and interception of moving target by mobile robot using fuzzy control. *Journal of Applied Engineering Science & Technology*, 4(2) :159–165, 2018.
- [55] Mohamed Nadour, Mohamed Boumehraz, Lakhmissi Cherroun, and Vicenç Puig. Mobile robot visual navigation based on fuzzy logic and optical flow approaches. *International Journal of System Assurance Engineering and Management*, 10 :1654–1667, 2019.
- [56] Mohamed Nadour, Mohamed Boumehraz, Lakhmissi Cherroun, and Vicenç Puig Cayuela. Hybrid type-2 fuzzy logic obstacle avoidance system based on horn-schunck method. *Electrotehnica, Electronica, Automatica*, 67(3) :45–51, 2019.
- [57] N Adar, A Egrisogut Tiryaki, and Recep Kozan. Real time visual servoing of a 6-dof robotic arm using fuzzy-pid controller. *Acta Physica Polonica A*, 128(2B), 2015.
- [58] Sabrina MohandSaidi and Rabah Mellah. Real-time speed control of a mobile robot using pid controller. In *Artificial Intelligence and Its Applications : Proceeding of the 2nd International Conference on Artificial Intelligence and Its Applications (2021)*, pages 548–556. Springer, 2022.

- [59] Sabrina Mohand Saidi, Rabah Mellah, Arezki Fekik, and Ahmad Taher Azar. Real-time fuzzy-pid for mobile robot control and vision-based obstacle avoidance. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, 13(1) :1–32, 2022.
- [60] Nikos Katevas. *Mobile robotics in healthcare*, volume 7. IOS Press, 2001.
- [61] Thomas Brunl. *Embedded robotics : mobile robot design and applications with embedded systems*, 2008.
- [62] Matthew West and Harry Asada. *Design of ball wheel mechanisms for omnidirectional vehicles with full mobility and invariant kinematics*. 1997.
- [63] Jefri Efendi Mohd Salih, Mohamed Rizon, Sazali Yaacob, Abdul Hamid Adom, and Mohd Rozailan Mamat. Designing omni-directional mobile robot with mecanum wheel. *American Journal of Applied Sciences*, 3(5) :1831–1835, 2006.
- [64] Richard Welch, Daniel Limonadi, and Robert Manning. Systems engineering the curiosity rover : A retrospective. In *2013 8th international conference on system of systems engineering*, pages 70–75. IEEE, 2013.
- [65] Robert H Bishop. *Mechatronic systems, sensors, and actuators : Fundamentals and modeling*. *Taylor & Francis eBooks DRM Free Collection*.
- [66] Spyros Tzafestas. Sensor integration and fusion techniques in robotic applications. *Journal of Intelligent & Robotic Systems*, 43(1), 2005.
- [67] Lindsay Kleeman and Roman Kuc. Mobile robot sonar for target localization and classification. *The International Journal of Robotics Research*, 14(4) :295–318, 1995.
- [68] Peter Corke and Peter Corke. Mobile robot vehicles. *Robotics, Vision and Control : Fundamental Algorithms In MATLAB® Second, Completely Revised, Extended And Updated Edition*, pages 99–124, 2017.
- [69] Jacques Gangloff. *Cours de robotique*. *Université de strasbourg*, 2010.
- [70] Thongchai Phairoh and Keith Williamson. Autonomous mobile robots using real time kinematic signal correction and global positioning system control. In *Proceedings of*

- IAJC-IJME International Conference on Industrial Technology, Nashville, TN, USA, 17-19 November, 2008.*
- [71] Patrick F Muir and Charles P Neuman. Kinematic modeling of wheeled mobile robots. *Journal of robotic systems*, 4(2) :281–340, 1987.
- [72] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008.
- [73] Adrien Durand-Petiteville. *Navigation référencée multi-capteurs d’un robot mobile en environnement encombré*. PhD thesis, Université Paul Sabatier-Toulouse III, 2012.
- [74] François Chaumette. *La relation vision-commande : théorie et application à des tâches robotiques*. PhD thesis, Rennes 1, 1990.
- [75] Roger Pissard-Gibollet and Patrick Rives. Applying visual servoing techniques to control a mobile hand-eye system. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 1, pages 166–171. IEEE, 1995.
- [76] Mario E Serrano, Sebastián A Godoy, Vicente A Mut, Oscar A Ortiz, and Gustavo JE Scaglia. A nonlinear trajectory tracking controller for mobile robots with velocity limitation via parameters regulation. *Robotica*, 34(11) :2546–2565, 2016.
- [77] Gustavo Scaglia, Lucía Quintero Montoya, Vicente Mut, and Fernando di Sciascio. Numerical methods based controller design for mobile robots. *Robotica*, 27(2) :269–279, 2009.
- [78] Farhad Shamsfakhr and Bahram Sadeghi Bigham. A neural network approach to navigation of a mobile robot and obstacle avoidance in dynamic and unknown environments. *Turkish Journal of Electrical Engineering and Computer Sciences*, 25(3) :1629–1642, 2017.
- [79] Gregory Dudek and Michael Jenkin. *Computational principles of mobile robotics*. Cambridge university press, 2010.
- [80] M De Villiers and NS Tlale. Development of a control model for a four wheel meca-num vehicle. 2012.

- [81] Shahrul Naim Sidek. *DYNAMIC MODELING AND CONTROL OF NONHOLONOMIC WHEELED MOBILE ROBOT SUBJECTED TO WHEEL SLIP*. PhD thesis, Vanderbilt University, 2008.
- [82] Ellips Masehian and Davoud Sedighizadeh. Classic and heuristic approaches in robot motion planning—a chronological review. *World Academy of Science, Engineering and Technology*, 23(5) :101–106, 2007.
- [83] Adham Atyabi and David Powers. Review of classical and heuristic-based navigation and path planning approaches. *International Journal of Advancements in Computing Technology (IJACT)*, 5(14), 2013.
- [84] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pages 500–505. IEEE, 1985.
- [85] Wesley H Huang, Brett R Fajen, Jonathan R Fink, and William H Warren. Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems*, 54(4) :288–299, 2006.
- [86] Saravanakumar Subramanian, Thomas George, and Asokan Thondiyath. Obstacle avoidance using multi-point potential field approach for an underactuated flat-fish type auv in dynamic environment. In *Trends in Intelligent Robotics, Automation, and Manufacturing : First International Conference, IRAM 2012, Kuala Lumpur, Malaysia, November 28-30, 2012. Proceedings*, pages 20–27. Springer, 2012.
- [87] Luis Valbuena and Herbert G Tanner. Hybrid potential field based control of differential drive mobile robots. *Journal of intelligent & robotic systems*, 68 :307–322, 2012.
- [88] J Mark Keil and Jorg-R Sack. Minimum decompositions of polygonal objects. In *Machine Intelligence and Pattern Recognition*, volume 2, pages 197–216. Elsevier, 1985.

- [89] Chenghui Cai and Silvia Ferrari. Information-driven sensor path planning by approximate cell decomposition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(3) :672–689, 2009.
- [90] Kikuo Fujimura and Hanan Samet. A hierarchical strategy for path planning among moving obstacles (mobile robot). *IEEE transactions on robotics and Automation*, 5(1) :61–69, 1989.
- [91] Sean Campbell, Niall O’Mahony, Anderson Carvalho, Lenka Krpalkova, Daniel Rioridan, and Joseph Walsh. Path planning techniques for mobile robots a review. In *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pages 12–16. IEEE, 2020.
- [92] Ishay Kamon and Ehud Rivlin. Sensory-based motion planning with global proofs. *IEEE transactions on Robotics and Automation*, 13(6) :814–822, 1997.
- [93] Ishay Kamon, Ehud Rivlin, and Elon Rimon. A new range-sensor based globally convergent navigation algorithm for mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 429–435. IEEE, 1996.
- [94] Johann Borenstein, Yoram Koren, et al. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE transactions on robotics and automation*, 7(3) :278–288, 1991.
- [95] Iwan Ulrich and Johann Borenstein. Vfh+ : Reliable obstacle avoidance for fast mobile robots. In *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*, volume 2, pages 1572–1577. IEEE, 1998.
- [96] Iwan Ulrich and Johann Borenstein. Vfh/sup* : Local obstacle avoidance with look-ahead verification. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 3, pages 2505–2511. IEEE, 2000.
- [97] An-Min Zou, Zeng-Guang Hou, Si-Yao Fu, and Min Tan. Neural networks for mobile robot navigation : a survey. In *Advances in Neural Networks-ISNN 2006 : Third In-*

- ternational Symposium on Neural Networks, Chengdu, China, May 28-June 1, 2006, Proceedings, Part II 3*, pages 1218–1226. Springer, 2006.
- [98] Yangmin Li and Xin Chen. Mobile robot navigation using particle swarm optimization and adaptive nn. In *Advances in Natural Computation : First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part III 1*, pages 628–631. Springer, 2005.
- [99] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3) :338–353, 1965.
- [100] T-HS Li, Shih-Jie Chang, and Yi-Xiang Chen. Implementation of human-like driving skills by autonomous fuzzy behavior control on an fpga-based car-like mobile robot. *IEEE Transactions on Industrial Electronics*, 50(5) :867–880, 2003.
- [101] Iraj Hassanzadeh and Sevil M Sadigh. Path planning for a mobile robot using fuzzy logic controller tuned by ga. In *2009 6th International Symposium on Mechatronics and its Applications*, pages 1–5. IEEE, 2009.
- [102] Anantha Sai Hari Haran V Injarapu and Suresh Kumar Gawre. A survey of autonomous mobile robot path planning approaches. In *2017 International conference on recent innovations in signal processing and embedded systems (RISE)*, pages 624–628. IEEE, 2017.
- [103] Joey K Parker, Ahmad R Khoogar, and David E Goldberg. Inverse kinematics of redundant robots using genetic algorithms. In *1989 IEEE International Conference on Robotics and Automation*, pages 271–272. IEEE Computer Society, 1989.
- [104] Jianping Tu and Simon X Yang. Genetic algorithm based path planning for a mobile robot. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 1, pages 1221–1226. IEEE, 2003.
- [105] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

- [106] Andrew Stacey, Mirjana Jancic, and Ian Grundy. Particle swarm optimization with mutation. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, volume 2, pages 1425–1430. IEEE, 2003.
- [107] Hua-Qing Min, Jin-Hui Zhu, and Xi-Jing Zheng. Obstacle avoidance with multi-objective optimization by pso in dynamic environment. In *2005 international conference on machine learning and cybernetics*, volume 5, pages 2950–2956. IEEE, 2005.
- [108] Jim Pugh, Alcherio Martinoli, and Yizhen Zhang. Particle swarm optimization for unsupervised robotic learning. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 92–99. IEEE, 2005.
- [109] Dr Robot. Wirobot sdk. *application programming interface API reference manual, for MS Windows*, 2010, 2010.
- [110] PMS5005 Sensing and User Manual Motion Controller. Dr robot, 25 valley wood dr. *Unitn20, Markham, ON, L3R2, 5L9 Canada version*, 1(5), 2006.
- [111] Dr Robot. i90. *Wireless networked autonomous mobile robot with high resolution pan-Tilt-zoom camera) quick start guide*, 2013, 2010.
- [112] Jean-Marie Flaus. *La régulation industrielle : régulateurs PID, prédictifs et flous*. Hermes Science Publications, 2000.
- [113] Bin He and Benjamin M Adams. Engineering process control. *Wiley StatsRef : Statistics Reference Online*, 2014.
- [114] KC Sio and CK Lee. Stability of fuzzy pid controllers. *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans*, 28(4) :490–495, 1998.
- [115] EH Mamdani and S Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of human-computer studies*, 51(2) :135–147, 1999.
- [116] Wu Wei, Ma Xiaoning, and Gao Xiuyu. The application of fuzzy pid control in intelligent wheelchair system. In *2010 Second WRI Global Congress on Intelligent Systems*, volume 1, pages 230–232. IEEE, 2010.

- [117] Hamani Miloud and Hassam Abdelouahab. Improving mobile robot navigation by combining fuzzy reasoning and virtual obstacle algorithm. *Journal of Intelligent & Fuzzy Systems*, 30(3) :1499–1509, 2016.
- [118] S Pawlowski, P Dutkiewicz, K Kozłowski, and W Wroblewski. Fuzzy logic implementation in mobile robot control. In *Proceedings of the Second International Workshop on Robot Motion and Control. RoMoCo'01 (IEEE Cat. No. 01EX535)*, pages 65–70. IEEE, 2001.