



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire



وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة غرداية

N° d'enregistrement

Université de Ghardaïa

كلية العلوم والتكنولوجيا

Faculté des Sciences et de la Technologie

قسم الآلية والكهروميكانيك

Département de d'automatique et d'électromécanique

Mémoire de fin d'étude, en vue de l'obtention du diplôme

Master

Domaine: Sciences et Technologies

Filière: *Automatique*

Spécialité: *Automatique et systèmes*

Thème

**Commande floue pour le suivi de trajectoire d'un robot
mobile (Quanser Qbot): Validation expérimentale**

Présenté par :

BOUTOULA Mouhammed

LASSAKEUR Abdeslam

Soutenue publiquement le : 07/09/2023

Devant le jury composé de :

HACENE Nacer

MCB

Univ. Ghardaïa

Président

AZZAOUI Mohamed

MBA

Univ. Ghardaïa

Examineur

BENCHABANE Achour

MCB

Univ. Ghardaïa

Examineur

BEKKAR Belgacem

MBA

Univ. Ghardaïa

Encadreur

Année universitaire 2022/2023

Dédicace:

Nous avons le grand plaisir de dédier ce modeste travail à:

*Nos parents pour leur soutien pendant toutes nos années
d'études..*

Toutes nos familles.

Tous nos amis, camarades et collègues sans exception.

*Abdeslam
Mouhammed*

Remerciements

Tout d'abord, nous remercions notre Dieu de nous avoir accepté
avait la santé et la patience nécessaires pour faire le travail.

Nous tenons

à remercier notre encadrant Dr. exprimer nos sincères remerciements.

BEKKAR Balgacem et Dr. Hacem Nacer pour son soutien et ses
conseils tout au long de ce travail. Ses encouragements et ses conseils ont
toujours été une source de motivation. Ce fut un honneur pour moi de travailler
sous sa direction.

Remerciements particuliers à nos enseignants et chercheurs
et personnel de l'Université de Ghardaïa,

Avant tout, nous sommes reconnaissants à nos familles qui
Ils nous ont soutenus chaque fois que nous en avons besoin, toujours à nos
côtés,

Gérez nos vies. Nous tenons à les remercier pour leur amour inconditionnel
nous motive à aller plus haut.

Summary :

In this final thesis which we studied, is to make a Trajectory control model of an autonomous mobile robot, and solve the problem of tracking the trajectory for a mobile Robot precisely the QUANSER QBOT2 robot. And to do this, we use a fuzzy controller which allows the robot to follow the desired trajectory.

Kinematics and dynamics Firstly, we made the kinematic model and the kinetic model, were constructed in detail.

Secondly, the two models of the robot were created and tested using Matlab Simulink. And we choose two models of given trajectories such as, the circle and the infinite. This is done by using another control model such as the PID controller, and comparing them and showing the effectiveness of the fuzzy controller.

Third, for the robot to follow the desired trajectory. The simulation results showed the effectiveness of the proposed controller in the trajectory tracking problem.

Résumé :

Dans cette thèse de fin d'études qui nous avons étudié, est de faire un modèle de contrôle de Trajectoire d'un robot mobile autonome, et résoudre le problème de suivi la trajectoire pour un Robot mobile précisément le robot QUANSER QBOT2. Et Pour ce faire là, on utilise un contrôleur flou qui permette au robot de suivre la trajectoire souhaitée.

La cinématique et la dynamique Dans un premier temps, on fait le modèle cinématique et le modèle cinétique, ont été construits de manière détaillée.

Deuxièmement, on fait les deux modèles du robot ont été réalisées et testées à l'aide de Matlab Simulink. Et on fait comme choix deux modèles de trajectoires donne telle que, le cercle et l'infinie. Ceci est fait en utilisant un autre modèle de contrôle telle-que le contrôleur PID, et en les comparant et en montrant l'efficacité du contrôleur flou.

Troisièmement, pour que le robot suive la trajectoire souhaitée. Les résultats de la simulation ont montré l'efficacité du contrôleur proposé dans le problème de suivi de trajectoire.

ملخص:

في هذه الأطروحة النهائية التي قمنا بدراستها، هو صنع نموذج للتحكم في المسار لروبوت متحرك مستقل، وحل مشكلة تتبع المسار للروبوت المتحرك على وجه التحديد الروبوت QUANSER QBOT2 وللقيام بذلك، نستخدم وحدة تحكم غامضة تسمح للروبوت باتباع المسار المطلوب. الحركية والديناميات

في البداية قمنا بصنع النموذج الحركي، وقد تم بناء النموذج الحركي بالتفصيل .

ثانياً، تم تصنيع نموذجي الروبوت واختبارهما باستخدام من ماتلاب سيمولينك. ونختار نموذجين للمسارات المعطاة، مثل الدائرة واللانهاية. ويتم ذلك من خلال استخدام نموذج تحكم آخر مثل المتحكم PID ومقارنتها وبيان مدى فعالية المتحكم المضرب.

ثالثاً، أن يتبع الروبوت المسار المطلوب. أظهرت نتائج المحاكاة فعالية المتحكم المقترح في مشكلة تتبع المسار .

Table des matières

Remerciements	iii
Résumé :	iv
ملخص.....	iv
Table des matières	vi
Liste des figures.....	viii
Liste des tableaux	xi
Liste des abréviations	xii
Introduction générale.....	1
Chapitre 1 : Étude de l'état de l'art	
1.1 Introduction.....	3
1.2 Commande floues	3
1.3 Suivi de trajectoire	4
1.4 Définition d'un robot mobile.....	5
1.5 Applications	8
1.6 Conclusion	11
Chapitre 2 : Modélisation du système.....	
2.1 Introduction.....	13
2.2 Dynamique des robots.....	14
2.3 Principaux composants matériels.....	14
2.3.1 Développement du modèle dynamique du Quanser Qbot	16
2.3.2 Description détaillée des composants Quanser Qbot.....	17
2.3.3 Spécifications du système et paramètres du modèle :	20
2.4 Cinématique d'entraînement différentiel.....	20
2.4.1 Equations de mouvement et les équations de couple.....	21

2.5	Discussion des défis et des solutions utilisées	26
2.6	Cinématique du Quanser Qbot	27
2.6.1	Cinématique et de son importance dans le contrôle des robots	27
2.6.2	Développement du modèle cinématique et la représentation mathématique du Quanser Qbot :	28
2.6.3	Discussion des défis associés à la modélisation de la cinématique du robot... 31	
2.7	Simulation et Validation des Modèles	32
2.7.1	Réponse indicielle du modèle cinématique	32
2.7.2	Réponse indicielle du modèle dynamique	34
2.8	Analyse de l'impact de différents paramètres	37
2.8.1	Evaluation des résultats	37
2.9	Conclusion :	38
Chapitre 3 : Conception de la commande floue		
2.1	Introduction	40
2.2	Conception de la commande floue	41
2.3	Le contrôleur PID	43
2.4	Le contrôleur flou	47
2.5	Résultats de simulation	53
2.6	Résultats expérimentaux	56
2.6.1	Comparaison analytique	60
2.7	Conclusion	61
Conclusion générale		62
Bibliographie		64

Liste des figures

Figure 1-1: Le robot mobile "Séjourner" de la NASA [12].....	6
Figure 1-2: La classification des robots mobiles.....	6
Figure 1-3 : Les moyens de transport	9
Figure 1-4 : Cybersécurité des systèmes industriels.....	9
Figure 1-5 : Robot de service	9
Figure 1-6 : ces de doctorat	10
Figure 1-7 : Localisation d'un robot mobile autonome en environnements naturels	10
Figure 1-8 : Environnement de réalité augmentée 3D.....	11
Figure 2-1 : Le Quanser Qbot.....	13
Figure 2-2 : Vues du dessous et du dessus du robot mobile QBot	14
Figure 2-3 : Bouton marche/arrêt et prise pour le chargeur de batterie.....	15
Figure 2-4 : Raspberry Pi 3 Model B+ avec WiFi intégré embarqué sur le robot QBot.....	15
Figure 2-5 : Communication entre un ordinateur avec QUARC installé et le QBot.....	16
Figure 2-6 : caméra.....	17
Figure 2-7 : Batterie.....	18
Figure 2-8 : Vue de dessous du Qbot, mise en évidence avec l'emplacement de la batterie	19
Figure 2-9 : SHARP GP2D12 IR Sensor.....	19
Figure 2-10 :LV-MaxSonar-EZ0 Télémètre sonar	19
Figure 2-11: Définitions du cadre de référence QBot	21
Figure 2-12 : Schéma des vitesses appliquées au robot.....	22
Figure 2-13 : Les forces sur les roues : une représentation équivalente.....	28
Figure 2-14 : Modèles de la réponse indicielle de la cinématique du robot sur SIMULINK	32
Figure 2-15 : La réponse indicielle ($V_R=2\text{m/s}$ et $V_L=-2\text{m/s}$ avec $\varphi =0$).....	33

Figure 2-16 : La réponse indicielle ($V_R=2\text{m/s}$ et $V_L=2\text{m/s}$ avec $\varphi = \pi/3\text{rad}$).....	33
Figure 2-17 : La réponse indicielle ($V_R =2\text{m/s}$ et $V_L=0.6\text{m/s}$ avec $\varphi = \pi/3\text{rad}$)	34
Figure 2-18 : Modèle de la réponse indicielle de la dynamique du robot sur SIMULINK..	34
Figure 2-19 : La réponse indicielle ($V_R =5\text{V}$ et $V_L=5\text{V}$)	35
Figure 2-20 : La réponse indicielle ($E_R =3\text{V}$ et $E_L = -1 \text{ V}$)	35
Figure 2-21: Modèle de la réponse indicielle de la dynamique avec la cinématique du robot sur SIMULINK.....	35
Figure 2-22 : La réponse indicielle ($E_R =5\text{V}$ et $E_L = 5\text{V}$ avec $\varphi = \pi/4 \text{ rad}$).....	36
Figure 2-23 : La réponse indicielle ($E_R = 5\text{V}$ et $E_L=3\text{V}$ avec $\varphi = \pi/4 \text{ rad}$).....	36
Figure 3-1: Contrôleur flou du Robot Qbot2.....	41
Figure 3-2 : Le problème de suivi de trajectoire	42
Figure 3-3 : Correcteur PID.....	43
Figure 3-4 : Le problème de suivi de trajectoire	44
Figure 3-5 : : L'entrées et la sortie de PID	44
Figure 3-6 : Système générale de PID contrôle pour le problème du suivi de trajectoires ..	45
Figure 3-7: Suivre une trajectoire circulaire.....	46
Figure 3-8 : l'erreur sur la : a) distance (Dis), b) orientation(β).....	46
Figure 3-9 : La commande appliqué pour la trajectoire circulaire	47
Figure 3-10 : Les fonctions d'appartenance de l'entrée Dis.....	48
Figure 3-11 : Les fonctions d'appartenance de l'entrée β	49
Figure 3-12 : Les fonctions d'appartenance de sortie V_R et V_L	49
Figure 3-13 : Schéma générale de robot avec un contrôleur floue pour le problème du suivi de trajectoires.....	50
Figure 3-14 : Suivi de trajectoire circulaire.....	53
Figure 3-15 : Erreur de distance du suivi de trajectoire circulaire	53
Figure 3-16 : L'erreur de relèvement du suivi de trajectoire circulaire	54
Figure 3-17 : Suivi de trajectoire infini	54

Figure 3-18 Erreur de distance du suivi de trajectoire infini.....	55
Figure 3-19 : L'erreur de relèvement du suivi de trajectoire infini.....	55
Figure 3-20 : Suivi de trajectoire circulaire.....	56
Figure 3-21 : l'erreur sur la : a) distance (Dis), b) orientation(β).....	56
Figure 3-22 : Résultats expérimentales, trajectoire circulaire (Modèle 1).....	57
Figure 3-23 : (a) erreur de l'angle β . (b) erreur de distance DIS trajectoire circulaire (Modèle 1).....	58
Figure 3-24 : Résultats expérimentales, trajectoire ∞ (Modèle 1).....	58
Figure 3-25 : (a) erreur de l'angle β . (b) erreur de distance DIS trajectoire ∞ (Modèle 1) ...	58
Figure 3-26 : Résultats expérimentales, trajectoire circulaire (Modèle 2).....	59
Figure 3-27 : (a) erreur de l'angle β . (b) erreur de distance DIS trajectoire ∞ (Modèle 2) ...	59
Figure 3-28 : Résultats expérimentales, trajectoire ∞ (Modèle 2).....	60
Figure 3-29 : (a) erreur de l'angle β . (b) erreur de distance DIS trajectoire ∞ (Modèle 2) ...	60

Liste des tableaux

Tableau 2-1 : Paramètres du modèle Qbot et spécifications du système.....	20
Tableau 3-1 : Variables linguistiques et floues pour les 3 entrées et les 3 sorties.....	49
Tableau 3-2 : Base de règles du contrôleur flou modèle 1	50
Tableau 3-3 : Base de règles du contrôleur flou modèle 2	51
Tableau 3-4 : Base de règles du contrôleur flou modèle 3	52
Tableau 3-5 : Comparaison analytique entre les contrôleurs flou proposés.....	61

Liste des abréviations

AUAV	Autonomous Unmanned Aerial Vehicle
GPS	Global Positioning Systems
VGA	Véhicule à Guidage Automatique
DDMR	Differential Drive Mobile Robot
ICC	Instantaneous Center of Curvature
PID	Proportionnelle Intégrale Dérivée
IDE	Integrated Development Environment
AC-DC	Direct Current-Alternating-Current
IN/OUT	Entrée (Input)/Sortie (Output)
ICSP	In-Circuit Serial Programmer
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
EEPROM	Electrically Erasable Programmable Read-Only Memory
Dis	la distance entre le robot et le point désiré
β	l'angle d'orientation relative du robot par rapport au point désiré
V_R	La vitesse de motor droite
V_L	la vitesse de motor gauche
E_R	La tension de motor droite
E_L	La tension de motor gauche

Introduction générale

Dans cette dernière années, grâce à la science de l'automatisation ou la robotique, l'homme a commencé à compter sur la machine pour accomplir des tâches, en particulier complexes et précises, et a cédé du besoin humain à la machine et de son rôle de facilité et d'augmenter la productivité et la flexibilité, et d'améliorer la qualité surtout dans l'industriel.

La robotique s'intéresse à tous les aspects de l'ingénierie, de la construction et de l'exploitation de robots. En général, il s'agit d'entreprendre un système technologique aussi rapidement que possible, avec le moins d'effort et le moins coûteux. Le but est toujours de faciliter l'exécution des différentes tâches de la vie.

Actuellement, les robots à bras sont les plus utilisés dans les chaînes de production (soudure, assemblage de composants, peinture, ...), ainsi que sur les robots mobiles qui se déplacent automatiquement dans les usines pour remplir certaines fonctions [1]. Les robots mobiles aussi étaient utilisés dans la plupart des domaines, en particulier ceux caractérisés par des zones rugueuses et dangereuses telles que les utilisations militaires, les mines [1].

La navigation autonome est une tâche essentielle pour les robots mobiles. Parce qu'ils seront confrontés à différents types d'obstacles mobiles et statiques. Cela nécessite que ces robots mobiles doivent bien contrôler et faire ses tâches données en toute sécurité [3].

Dans ce mémoire, le suivi de trajectoire du robot QBot2 a été abordé en utilisant des techniques de contrôleur flou. La tâche fondamentale attribuée au robot mobile consiste à suivre une trajectoire depuis une configuration donnée, en minimisant les erreurs au maximum.

Le mémoire est organisé comme suit :

Après l'introduction, le premier chapitre présente quelques notions principales de la commande floue. Le deuxième chapitre aborde en détail la modélisation du robot, y compris la simulation et la validation des différents modèles de logique floue sur MATLAB.

Le troisième chapitre concerne la conception de la commande floue pour le suivi de trajectoire du robot, ainsi que l'implémentation pratique de cette commande et la discussion des résultats.

Enfin, la conclusion générale présente le bilan de ce travail, ainsi que ses limites et les perspectives d'améliorations envisagées

Chapitre 1

Étude de l'état de l'art

1.1 Introduction

Les commandes floues ont été largement utilisées pour la commande de robots mobiles en raison de leur capacité à gérer les incertitudes et les variations dans les environnements complexes. Les commandes floues permettent aux robots de prendre des décisions et de réaliser des tâches en utilisant des règles linguistiques plutôt que des équations mathématiques précises. Cela permet aux robots de traiter des informations incertaines ou ambiguës et de s'adapter aux variations de l'environnement en temps réel [3].

L'objectif de cette étude est utilisation de la commandes floues pour suivi le trajectoire des robots mobiles (précisément QUANSER QBO2), en se concentrant , sur simulation au MATLAB, et l'applications pratiques et les résultats obtenu . Les questions de recherche qui seront abordées incluent :

- C'est quoi la commandes floues ?
- Quelles sont les applications pratiques des commandes floues dans la commande de robots mobiles ?
- Quels sont les résultats de recherche récents sur l'utilisation des commandes floues pour la commande de robots mobiles ?
- Comment les commandes floues se comparent-elles à d'autres approches de commande pour les robots mobiles ?

Dans les sections suivantes de ce chapitre, nous examinerons en détail les avantages et les limites des commandes floues pour la commande de robots mobiles, les applications pratiques de cette approche, les résultats de recherche récents, et la comparaison des commandes floues avec d'autres approches de commande pour les robots mobiles.

1.2 Commande floues :

La commande floue est une méthode de contrôle de systèmes dynamiques qui utilise des concepts de logique floue pour traiter les informations incertaines ou vagues. Il existe plusieurs approches pour implémenter des commandes floues, qui sont adaptées aux différents types de problèmes [4].

Les commandes floues basées sur les règles sont une approche courante pour la commande floue. Cette approche utilise des règles linguistiques pour décrire la relation entre les entrées et

les sorties du système. Les règles sont généralement exprimées en langage naturel, par exemple "si la température est froide et l'humidité est élevée, alors augmenter le chauffage". Ces règles sont ensuite traduites en formes mathématiques et utilisées pour régler les paramètres du système de commande floue [5].

Les commandes floues basées sur les modèles sont une autre approche pour la commande floue. Cette approche utilise un modèle mathématique du système pour calculer les actions de commande en fonction des entrées du système. Le modèle est généralement basé sur des équations différentielles ou des équations d'état. Les commandes floues basées sur les modèles sont souvent utilisées dans des systèmes dynamiques complexes tels que les robots ou les véhicules autonomes [2].

Les commandes floues basées sur les données sont une approche relativement nouvelle pour la commande floue. Cette approche utilise des techniques d'apprentissage automatique pour déterminer les règles de commande en fonction des données d'entrée et de sortie du système. Les données sont collectées à partir du système réel ou d'un modèle de simulation. Les commandes floues basées sur les données sont souvent utilisées pour des systèmes dynamiques pour lesquels il est difficile de modéliser le comportement du système [6].

Chaque approche de commande floue a ses avantages et ses limites. Les commandes floues basées sur les règles sont faciles à comprendre et à mettre en œuvre, mais peuvent être limitées par la qualité des règles et des connaissances des experts. Les commandes floues basées sur les modèles sont plus précises, mais nécessitent une modélisation précise du système. Les commandes floues basées sur les données peuvent être très précises et adaptatives, mais nécessitent une quantité importante de données pour l'apprentissage [7].

Les méthodes couramment utilisées pour implémenter ces approches incluent des algorithmes tels que le contrôleur proportionnel-intégral-dérivé flou (Fuzzy PID), la commande floue adaptative, la commande floue de Takagi-Sugeno, et la logique floue génétique. Les techniques d'optimisation telles que l'algorithme génétique, l'optimisation par essaim de particules et la recherche de gradient sont également souvent utilisées pour optimiser les paramètres des systèmes de commande floue [8].

1.3 Suivi de trajectoire

Le suivi de trajectoire est une tâche essentielle pour les robots mobiles car il leur permet de se déplacer de manière autonome dans leur environnement. Il existe plusieurs techniques de

suivi de trajectoire, dont les plus courantes sont le suivi de trajectoire basé sur les modèles, le suivi de trajectoire basé sur les données et le suivi de trajectoire hybride [9].

Le suivi de trajectoire basé sur les modèles consiste à utiliser un modèle mathématique pour prédire la trajectoire du robot. Ce modèle est généralement basé sur des équations de mouvement ou des modèles cinématiques. Le suivi de trajectoire basé sur les modèles est précis et peut être utilisé dans des environnements contrôlés. Cependant, il peut être moins fiable dans des environnements complexes ou en présence d'obstacles imprévus.

Le suivi de trajectoire basé sur les données, quant à lui, utilise des données sensorielles pour suivre la trajectoire du robot. Les données peuvent provenir de capteurs tels que des lidars, des caméras ou des capteurs de mouvement. Cette technique est plus robuste que le suivi de trajectoire basé sur les modèles car elle utilise des données réelles pour suivre la trajectoire du robot. Cependant, le suivi de trajectoire basé sur les données peut être moins précis et peut nécessiter une quantité importante de données pour fonctionner correctement [10].

Le suivi de trajectoire hybride combine les deux approches précédentes pour obtenir une technique plus fiable et précise. Cette technique utilise à la fois des modèles mathématiques et des données sensorielles pour suivre la trajectoire du robot. Le suivi de trajectoire hybride peut être utilisé dans des environnements complexes et est plus fiable que les autres techniques [11].

En termes de mise en œuvre, les méthodes couramment utilisées pour implémenter ces techniques comprennent l'utilisation de filtres de Kalman, de réseaux de neurones, de méthodes de Monte Carlo et d'algorithmes de suivi de points de référence. Chacune des méthodes a ses avantages et ses limites, et le choix dépendra du contexte spécifique dans lequel le robot doit être utilisé [3].

1.4 Définition d'un robot mobile

Un robot mobile est une machine mécanique, automatique et programmable capable de se mouvoir dans un environnement donné. On regroupe sous cette appellation tous les robots autonomes (c'est-à-dire non télécommandés) capables de se déplacer, par opposition aux robots attachés à un point fixe, comme les robots manipulateurs en industrie [11].

Il existe plusieurs types de robots mobiles et ceux-ci sont, en général, classifiés selon leur type de locomotion c'est-à-dire, selon l'environnement dans lequel ils opèrent, par exemple les

robots de recherche sur des planètes dans l'espace ou des robots sous-marins dans les mers pour l'exploration, etc.



Figure 1-1: Le robot mobile "Séjourner" de la NASA [12]

Les robots mobiles sont classés en fonction des tâches qu'ils vont effectuer et des Environnements dans lesquels ils sont impliqués. Bien que les robots à roues soient les plus Courants [13], il existe d'autres types de robots mobiles. Ainsi, les robots mobiles peuvent être Classés comme suit :



Robots Mobiles à chenilles



Robots Mobiles à roues



Robots Mobiles hybrides



Robots Mobiles volants



Robots Mobiles à pattes



Robots Mobiles sous-marins

Figure 1-2: La classification des robots mobiles.

- **Robots mobiles à roues :**

Les roues sont de loin la méthode la plus populaire pour assurer la mobilité des robots et sont utilisées pour propulser de nombreux robots de tailles différentes. Les roues peuvent être à peu près n'importe quelle taille de robots et peuvent avoir à peu près n'importe quel nombre de roues, bien que 3 et 4 soient les plus courantes [14].

- **Robots mobiles à chenilles :**

Tout comme les chaînes utilisées dans un char militaire, elles réduisent le glissement, ce qui les rend adaptées aux terrains difficiles.

- **Robots mobiles à pattes :**

Les plus couramment utilisés sont les robots à six, quatre ou deux pattes. Il est très adapté aux terrains irréguliers.

- **Robots mobiles volants :**

Appelé AUAV (Autonomes Numance Aerial Véhicule), Le plus souvent, l'armée s'en sert pour effectuer des missions de reconnaissance, d'espionnage ou de bombardement.

- **Robots mobiles sous-marins :**

Il se caractérise par la présence de moteurs de poussée, de queue ou d'ailettes, et il est encore en cours de développement car de nombreux problèmes doivent être surmontés.

- **Robots mobiles hybrides :**

Les conceptions diverses peuvent inclure l'aéroglisser, les conceptions de serpents, les tourelles et plus encore.

Chaque type de robot mobile présente des avantages et des inconvénients en fonction de l'application spécifique. Par exemple, les robots à roues peuvent être très rapides et efficaces sur des surfaces planes, mais ils peuvent avoir des difficultés à traverser des terrains accidentés. Les robots à chenilles, quant à eux, sont très stables sur des terrains accidentés, mais peuvent être plus lents que les robots à roues. Les robots volants peuvent être très rapides et efficaces pour couvrir de grandes distances, mais ils peuvent avoir des difficultés à opérer dans des environnements venteux ou confinés. Enfin, les robots marins peuvent être très efficaces pour explorer les fonds marins, mais peuvent avoir des difficultés à naviguer dans des zones de fort courant ou de mauvaise visibilité.

L'un des principaux défis associés aux applications de commande floue et de suivi de trajectoire pour les robots mobiles est la robustesse face aux incertitudes. Les capteurs et les actionneurs peuvent être soumis à des perturbations imprévues, telles que des changements brusques de la direction du vent ou des interférences électromagnétiques. Les algorithmes de commande doivent être conçus pour être résistants à ces perturbations et capables de s'adapter en temps réel pour maintenir le robot sur sa trajectoire.

Un autre défi important est de prendre en compte les contraintes de l'environnement dans lequel le robot mobile opère. Par exemple, les robots à roues peuvent avoir des difficultés à franchir des obstacles tels que des marches d'escalier ou des trous dans le sol, tandis que les robots volants peuvent être limités par des zones d'interdiction de vol. Les algorithmes de commande doivent être conçus pour prendre en compte ces contraintes afin de garantir que le robot mobile reste en sécurité et ne cause pas de dommages à son environnement

1.5 Applications

La commande floue, le suivi de trajectoire et les robots mobiles sont des technologies clés dans de nombreux domaines, offrant des avantages tels que la précision, l'efficacité et la sécurité. Voici quelques exemples d'applications de ces technologies dans différents domaines :

-Systèmes de transport : Les systèmes de transport automatisés, tels que les métros, les trains et les tramways, utilisent souvent des commandes floues et des algorithmes de suivi de trajectoire pour optimiser les performances du système et améliorer la sécurité. Les voitures autonomes utilisent également ces technologies pour naviguer sur les routes et éviter les obstacles.



Figure 1-3 : Les moyens de transport

-Systèmes industriels : Les robots industriels utilisent la commande floue et le suivi de trajectoire pour effectuer des tâches répétitives et précises, telles que l'assemblage de pièces et la manipulation de matériaux. Ces technologies permettent également aux robots de s'adapter à des environnements de travail variables et de travailler en collaboration avec les travailleurs humains



Figure 1-4 : Cybersécurité des systèmes industriels

-Robots de service : Les robots de service, tels que les robots de nettoyage et les robots de livraison, utilisent la commande floue et le suivi de trajectoire pour se déplacer efficacement dans des environnements complexes et éviter les obstacles.



Figure 1-5 : Robot de service

-Systèmes de surveillance et de reconnaissance : Les drones utilisent la commande floue et le suivi de trajectoire pour voler de manière autonome et effectuer des missions de surveillance et de reconnaissance. Ces technologies permettent aux drones de s'adapter à des conditions météorologiques variables et de détecter les obstacles en temps réel.



Figure 1-6 : ces de doctorat

-Systèmes de robotique mobile autonome : Les robots mobiles autonomes, tels que les robots de déminage et les robots d'exploration, utilisent la commande floue et le suivi de trajectoire pour naviguer dans des environnements inconnus et éviter les obstacles. Ces technologies permettent également aux robots de prendre des décisions en temps réel pour s'adapter à des situations imprévues.



Figure 1-7 : Localisation d'un robot mobile autonome en environnements naturels

-Systèmes de robotique mobile coopérative : Les robots mobiles coopératifs utilisent la commande floue et le suivi de trajectoire pour collaborer entre eux et effectuer des tâches complexes, telles que le transport de charges lourdes et le nettoyage de grandes surfaces.



Figure 1-8 : Environnement de réalité augmentée 3D

Les résultats obtenus dans ces applications ont été très encourageants, avec des améliorations significatives de la précision, de l'efficacité et de la sécurité. Cependant, il reste encore beaucoup à faire pour améliorer ces technologies, notamment en termes de fiabilité, de robustesse et d'adaptabilité. Les perspectives futures pour ces applications sont donc très prometteuses, avec des opportunités pour développer des systèmes encore plus performants et polyvalents.

1.6 Conclusion

D'après l'étude que nous avons réalisée dans ce chapitre concernant les contrôles flous Avec ces différentes méthodes de mise en œuvre, les plus courantes consistent à contrôler et Surveiller le parcours des robots mobiles.

Cette dernière s'appuie sur une revue de la littérature existante et c'est pourquoi nous avons présenté des techniques de suivi de trajectoire pour les robots mobiles.

Nous avons discuté des avantages et des limites de chaque approche, ainsi que de la définition du robot mobile et du domaine d'application tel que les systèmes de transport, les systèmes industriels, les robots de service, les systèmes de surveillance et de reconnaissance, les systèmes robotiques mobiles autonomes et les systèmes robotiques mobiles collaboratifs. Nous nous sommes concentrés sur les défis actuels et futurs dans ce domaine. L'un des principaux défis associés aux applications de contrôle flou et de suivi de trajectoire pour les robots mobiles est la robustesse aux incertitudes

Chapitre 2

Modélisation du système

2.1 Introduction

Le Quanser Qbot (Figure 1.1) est un robot mobile programmable conçu pour l'apprentissage et la recherche en robotique. Il est largement utilisé dans les universités et les laboratoires de recherche pour étudier la robotique mobile, les systèmes de contrôle, la navigation autonome et la perception. Le robot Qbot est équipé de différents capteurs tels que des capteurs de distance, des capteurs d'inclinaison, des capteurs de vitesse et des caméras, ce qui en fait un outil polyvalent pour la recherche [1].



Figure 2-1 : Le Quanser Qbot

Il est important de modéliser le Quanser Qbot pour comprendre ses dynamiques et sa cinématique de mouvement. La modélisation permet de développer des algorithmes de contrôle, de navigation et de perception pour le robot, ce qui peut améliorer sa précision et son efficacité dans différentes tâches. La modélisation peut également être utilisée pour simuler le comportement du robot dans différents environnements et conditions.

Ce chapitre fournira une modélisation complète du Quanser Qbot, y compris les dynamiques du robot et la cinématique de son mouvement. Les chapitres suivants seront consacrés à la modélisation mathématique et physique du robot, en commençant par la cinématique inverse et directe, puis en abordant les équations de mouvement du robot et les contrôleurs de mouvement.

Les concepts clés utilisés dans ce chapitre incluent la cinématique, les équations de mouvement, les matrices de transformation homogènes et les contrôleurs de mouvement [2].

En somme, ce chapitre fournira une base solide pour comprendre la dynamique et la cinématique du robot mobile Quanser Qbot et simulation et validation des modèles, ce qui est essentiel pour développer des algorithmes de contrôle et de navigation pour le robot dans différentes applications de la robotique mobile

2.2 Dynamique des robots

Quanser Qbot 2e est un robot mobile à roues différentielles conçu pour la recherche et l'éducation en robotique. Il est équipé de plusieurs composants clés qui lui permettent de se déplacer et d'interagir avec son environnement.

2.3 Principaux composants matériels

Comme le montre la figure 2-2, le robot QBot se compose de deux roues motrices gauche et droite montées sur un axe commun. Cette configuration d'entraînement est connue sous le nom d'entraînement différentiel. Deux roues pivotantes à l'avant et à l'arrière du robot stabilisent la plate-forme sans compromettre le mouvement. Ces deux roues motrices sont indépendamment actionnées pour permettre au robot de se déplacer vers l'avant et vers l'arrière [4].



Figure 2-2 : Vues du dessous et du dessus du robot mobile QBot

Le mouvement de chaque roue est mesuré par des encodeurs et le cap ou l'angle de lacet du robot est estimé par le gyroscope intégré. Plus d'informations sur la cinématique et la génération de raccourcis clavier pour obtenir des trajectoires de mouvement spécifiques sont fournies dans

les chapitres suivants. Vous apprendrez également comment les informations sensorielles mesurées sont utilisées pour la localisation odométrique.

En plus des encodeurs et du gyroscope, QBot est livré avec Microsoft Kinect pour la vision [8] par ordinateur qui produit des images en couleur (RVB) et des informations de profondeur. Vous pouvez traiter les données RVB et de profondeur à diverses fins, notamment l'inspection visuelle, la cartographie raster de population 2D et 3D, l'odométrie visuelle, etc. Voir le chapitre Inspection vidéo pour plus d'informations sur ces concepts [4].

Comme le montre la Figure 2.2, le QBot possède également des capteurs d'impact intégrés (gauche, droite et centre) et des capteurs de chute (gauche, droite et centre). Ces capteurs peuvent être utilisés dans l'algorithme de contrôle pour éviter les obstacles ou les dommages au robot. La batterie est située sous le QBot et peut fonctionner en continu pendant environ 3 heures lorsqu'elle est complètement chargée. Le robot peut être chargé avec le chargeur connecté à la prise comme illustré à la Figure 2.3[9].



Figure 2-3 : Bouton marche/arrêt et prise pour le chargeur de batterie

QBot utilise un Raspberry Pi 3 Model B+ comme petit ordinateur de bord avec Wi-Fi intégré qui exécute tous les programmes compilés. Le Raspberry Pi 3 Model B+ se connecte à la Kabuki Mobile Box et au capteur Kinect [9].

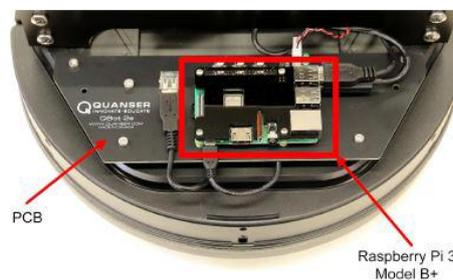


Figure 2-4 : Raspberry Pi 3 Model B+ avec WiFi intégré embarqué sur le robot QBot

2.3.1 Développement du modèle dynamique du Quanser Qbot

Au cours de ce laboratoire, nous utiliserons le logiciel de prototypage Quanser Rapid Control (QUARC) qui s'intègre de manière transparente à MATLAB/Simulink pour fournir une communication et une interface en temps réel avec les plates-formes matérielles.

QUARC est un moyen très efficace de concevoir, développer, déployer et valider des applications en temps réel sur une variété de systèmes cibles matériels à l'aide de Simulink. QUARC générera du code en temps réel directement à partir de Simulink et l'exécutera en temps réel sur le robot QBot, le tout sans traitement numérique du signal ou sans écrire une seule ligne de code [7].

Les contrôleurs seront développés dans Simulink avec QUARC sur l'ordinateur hôte, et ces modèles Simulink seront compilés de manière croisée et téléchargés vers la cible (Raspberry Pi 3 Model B+) de manière transparente en utilisant le routeur sans fil. Un schéma de cette configuration est illustré à la figure 1.5.

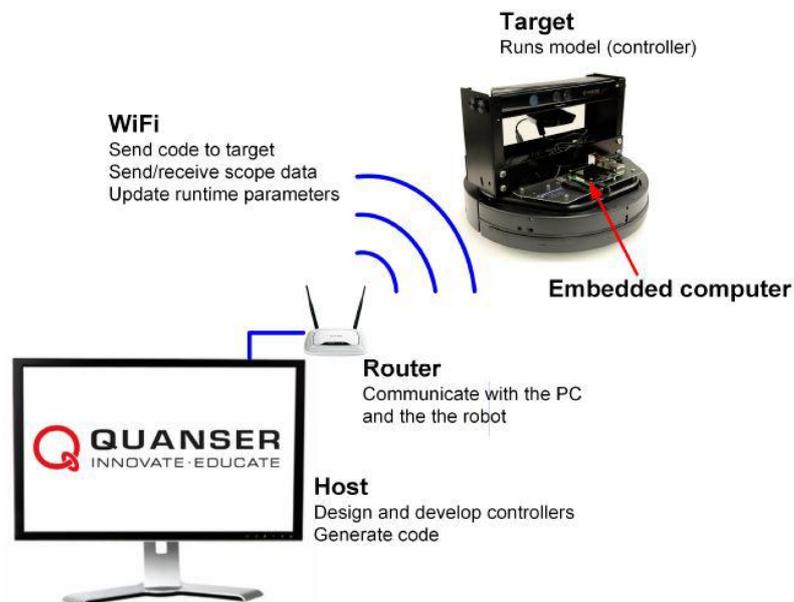


Figure 2-5 : Communication entre un ordinateur avec QUARC installé et le QBot

Pour communiquer avec le QBot, plusieurs blocs QUARC disponibles dans la bibliothèque Quanser de Simulink seront utilisés [7]. Les blocs suivants seront, en particulier, les plus utiles :

1. Bloc Hardware In the Loop (HIL) Initialize : il configure les pilotes et l'interface matérielle pour le QBot.

2. Bloc de lecture HIL : il est utilisé pour lire les données sensorielles.
3. Bloc d'écriture HIL : il est utilisé pour piloter les deux moteurs.
4. Initialisation de l'hôte : il peut être utilisé pour des périphériques d'entrée externes tels qu'un clavier (clavier hôte) ou un joystick (contrôleur de jeu hôte).
5. Bloc Kinect Initialize : il est utilisé pour initialiser le capteur Kinect. La fréquence d'images et la résolution maximales sont définies dans ce bloc.
6. Bloc Kinect Get Image : il capture les données RVB du capteur Kinect.
7. Bloc Kinect Get Depth : il capture les données de profondeur du capteur Kinect.
8. Bloc d'affichage vidéo compressé : il transmet les données d'entrée compressées (RVB ou profondeur) du QBot au PC et les affiche sur le moniteur.

2.3.2 Description détaillée des composants Quanser Qbot

DAC Qbot :

Le DAC Qbot est une carte d'acquisition de données pouvant recevoir des entrées analogiques et d'autres entrées (pour capteurs sonar). Il est également capable de stocker les sorties PWM pour n'importe quel servomoteur. Il se trouve sous le capot noir de Qbot et n'est pas censé être physiquement accessible [3].

Caméra USB :

La caméra USB Logitech Quickcam Pro 9000 est montée sur le Qbot (Figure 1.6). L'ensemble de blocs QUARC qui utilise la bibliothèque Open Source Computer Vision (dans la section 5.3) permet à l'utilisateur de capturer et d'afficher des images en temps réel, de les traiter et de les enregistrer sur un disque pour une analyse future[3].



Figure 2-6 : caméra

Gumstix :

Gumstix est un petit ordinateur entièrement open source. Une fois QUARC installé, le code généré par MATLAB est téléchargé, compilé et exécuté directement sur Gumstix. La carte mère

Gumstix se connecte directement au DAC Qbot. Le Gumstix est également livré avec une carte de connexion Wi-Fi pour permettre une connexion sans fil entre le Gumstix cible et l'ordinateur hôte et/ou d'autres véhicules [3].

Batterie :

Qbot est alimenté par le système d'alimentation avancé "APS" d'iRobot (Figure 1-7). La batterie est située sous le Qbot (Figure 1-8) et peut fonctionner en continu pendant environ 2 heures lorsqu'elle est complètement chargée. L'indicateur d'alimentation de Qbot indique le niveau de la batterie.

La LED est verte lorsque les piles sont neuves puis devient progressivement rouge lorsque les piles sont faibles. Le chargement de la batterie prend moins de 3 heures. Pendant la charge, le voyant d'alimentation clignote lentement en orange. Un chargeur sera fourni.



Figure 2-7 : Batterie

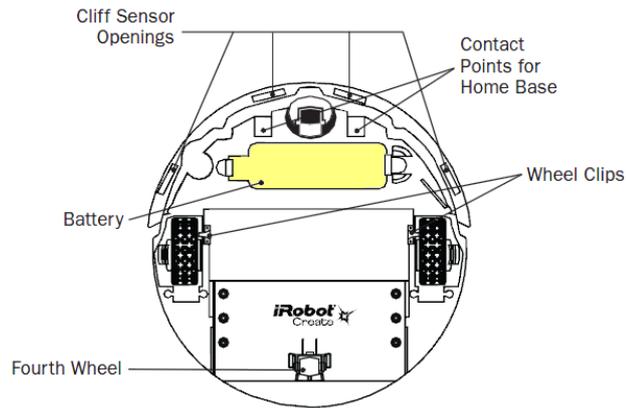


Figure 2-8 : Vue de dessous du Qbot, mise en évidence avec l'emplacement de la batterie

Capteurs infrarouges :

Le SHARP 2Y0A02 [5] est un capteur de portée infrarouge à faible coût (20-150 cm). Il y a cinq de ces capteurs inclus avec le Qbot. Les capteurs sont connectés aux canaux d'entrée analogiques du DAC Qbot, qui peuvent ensuite être lus à l'aide du bloc HIL Read Write



Figure 2-9: SHARP GP2D12 IR Sensor

Capteurs sonars :

MaxSonar-EZ0 [6] est un capteur sonar de détection et de portée très courte à longue (Figure 2-10). Le MaxSonar-EZ0 détecte les cibles de 0 à 254 pouces (6,45 mètres) de longueur et fournit des informations sur la portée du sonar de 6 à 254 pouces avec une résolution de 1 pouce. Les objets entre 0 et 6 pouces se situent dans la plage de 6 pouces. Trois de ces capteurs sont connectés au Qbot. Les capteurs sont connectés à d'autres canaux d'entrée Qbot DAC,

, qui peuvent ensuite être lus à l'aide du bloc HIL Read-Write (Section 5) [3].



Figure 2-10 :LV-MaxSonar-EZ0 Télémètre sonar

2.3.3 Spécifications du système et paramètres du modèle :

Le tableau 2-1 ci- répertorie les principaux paramètres associés au Quanser Qbot.

Tableau 2-1 : Paramètres du modèle Qbot et spécifications du système

<i>Symbol</i>	<i>Description</i>	<i>Value</i>	<i>Unit</i>
<i>d</i>	Diamètre du Qbot	0.34	<i>M</i>
<i>h</i>	Hauteur du Qbot (avec caméra pièce jointe)	0.19	<i>M</i>
<i>Vmax</i>	Vitesse maximale du Qbot	0.5	<i>m/s</i>
<i>m</i>	Masse totale du Qbot	2.92	<i>kg</i>

2.4 Cinématique d'entraînement différentiel

La cinématique d'entraînement différentiel est la relation mathématique qui mappe le mouvement indépendant des roues au mouvement global du châssis du robot. Ce sujet fondamental est le fondement de tout contrôle de robot mobile, en ce sens qu'il est principalement responsable de la mobilité prévisible du robot. Dans cette expérience, vous étudierez la cinématique d'entraînement différentiel de la plate-forme mobile Quanser Qbot [10].

Le QBot est entraîné par un ensemble de deux roues coaxiales. Ces roues sont actionnées à l'aide de moteurs à courant continu hautes performances avec encodeurs et capteurs de chute. Pour déterminer la relation entre le mouvement indépendant des deux roues et le mouvement du robot global, nous commençons par modéliser le mouvement du robot autour d'un point commun.

Soit le rayon des roues soit noté r , et la vitesse de rotation de la roue soit notée w_L et w_R pour la roue gauche et droite respectivement. La vitesse linéaire des deux roues le long du sol est alors donnée par les équations suivantes [4] :

$$v_L = w_L r \quad (2-1)$$

$$v_R = w_R r \quad (2-2)$$

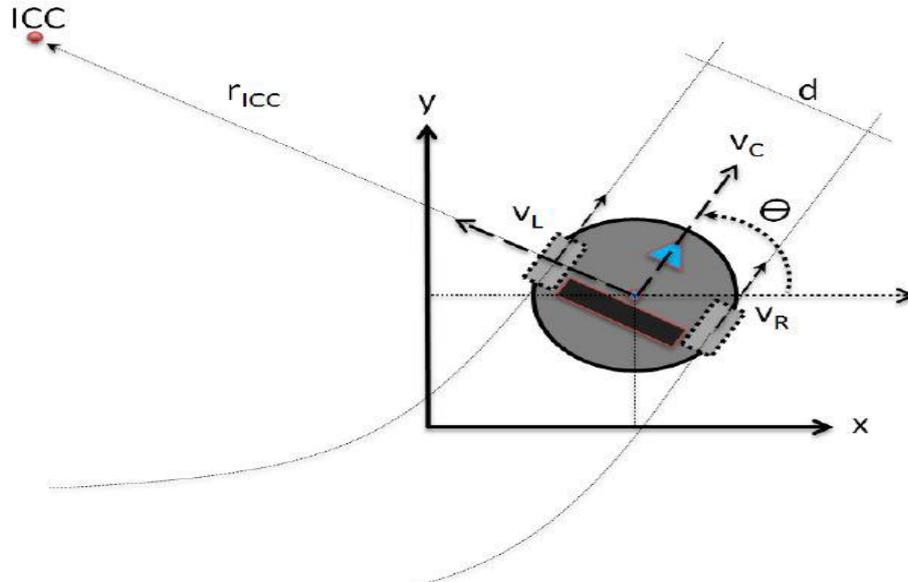


Figure 2-11 : Définitions du cadre de référence QBot

En supposant qu'il n'y a pas de patinage des roues, le QBot peut se déplacer le long du plan horizontal en trajectoire droite ou courbe, ainsi que tourner sur place, en faisant varier la vitesse relative entre les roues gauche et droite.

Puisque nous supposons que les roues ne sont pas sujettes au patinage, le mouvement des roues est contraint de se déplacer le long de leurs directions avant et arrière. Ceci, combiné à la contrainte inhérente imposée par le châssis du robot couplant les deux roues ensemble, signifie que toutes les rotations du châssis du robot doivent se faire autour d'un point situé le long de l'axe de roue commun. Par exemple, si une seule des deux roues tourne, le robot tournera (pivotera) autour de la roue immobile. Par contre, si les deux roues tournent à la même vitesse, le robot tourne autour d'un point infiniment éloigné du robot. Ce centre de rotation est connu sous le nom de centre de courbure instantané (ICC) comme le montre la figure 2.11[4].

2.4.1 Equations de mouvement et les équations de couple

En appliquant la deuxième loi de Newton au système, on obtient :

$$\begin{cases} \sum F_i = m\dot{V} \\ \sum M_i = j\dot{\omega} \end{cases} \quad (2-3)$$

Alors,

$$\begin{cases} F + F_s = m\dot{V} \\ M_r + M_l + M_s = I\dot{\omega} \end{cases} \quad (2-4)$$

Où, $m[\text{kg}]$ et $I [\text{kg.m}^2]$ sont la masse et l'inertie de rotation du robot.

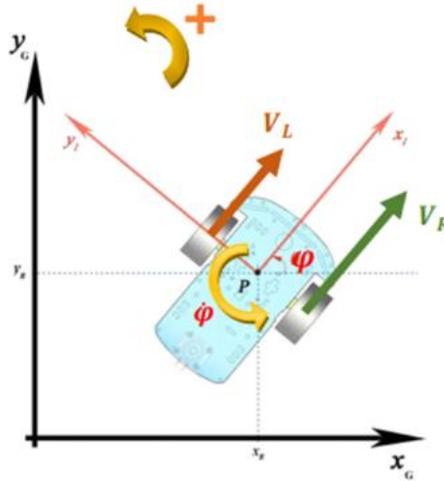


Figure 2-12 : Schéma des vitesses appliquées au robot

Donc,

$$\begin{cases} \frac{1}{2}FR + \frac{1}{2}FL + kvV = mV \\ Frl + Fll + k_\omega\omega = I\omega \end{cases} \quad (2-5)$$

kv est le coefficient de résistance au mouvement linéaire ; et kx est le coefficient de résistance au mouvement de rotation.

En prenant la dérivée temporelle de l'équation. [20], on obtient :

$$\begin{cases} \dot{v} = \frac{1}{2}\dot{v}_R + \frac{1}{2}\dot{v}_L \\ \omega = \frac{1}{2l}\dot{v}_R + \frac{1}{2l}\dot{v}_L \end{cases} \quad (2-6)$$

En remplaçant l'éq. [2-5] dans l'éq. [2-6], on obtient :

$$\begin{cases} \frac{1}{2}F_R + \frac{1}{2}F_L - k_v v = \frac{m}{2}\dot{v}_R + \frac{m}{2}\dot{v}_L \\ F_\tau l - F_l l - k_\omega \omega = \frac{l}{2l}\dot{v}_R - \frac{l}{2l}\dot{v}_L \end{cases} \quad (2-7)$$

Alors,

$$\begin{cases} \frac{1}{2}F_R + \frac{1}{2}F_L - k_v v = \frac{m}{2}\dot{v}_R + \frac{m}{2}\dot{v}_L \\ F_\tau l - F_l l - k_\omega \omega = \frac{l}{2l}\dot{v}_R - \frac{l}{2l}\dot{v}_L \end{cases} \quad (2-8)$$

On obtient après résolution de l'Eq. [23]:

$$\begin{cases} F_R = \left(\frac{2l^2m + l}{4l^2}\right)\dot{v}_R + \left(\frac{2l^2m - l}{4l^2}\right)\dot{v}_L + \left(\frac{2l^2K_V + K_\omega}{4l^2}\right)v_R + \left(\frac{2l^2K_V - K_\omega}{4l^2}\right)v_L \\ F_L = \left(\frac{2l^2m - l}{4l^2}\right)\dot{v}_R + \left(\frac{2l^2m + l}{4l^2}\right)\dot{v}_L + \left(\frac{2l^2K_V - K_\omega}{4l^2}\right)v_R + \left(\frac{2l^2K_V + K_\omega}{4l^2}\right)v_L \end{cases} \quad (2-9)$$

Les relations entre les forces de traction et les couples des roues $\frac{1}{2}T_wR$; T_wL en $\frac{1}{2}N:m$ sont les suivants :

$$\begin{cases} T_{wR} = l_w \left(\frac{\dot{v}_R}{r}\right) + f_w \left(\frac{V_R}{r}\right) + f_{Rr} \\ T_{wL} = l_w \left(\frac{\dot{v}_L}{r}\right) + f_w \left(\frac{V_L}{r}\right) + f_{Lr} \end{cases} \quad (2-10)$$

où $l_m[Kg:m^2]$ est le moment d'inertie des roues ; $f_w[N:s:m]$ est le coefficient de frottement visqueux, et $r[m]$ est le rayon de la roue. $\left(\frac{V_R}{r}, \frac{V_L}{r}\right)$ et $\left(\frac{\dot{v}_R}{r}, \frac{\dot{v}_L}{r}\right)$ sont les vitesses angulaires et les accélérations angulaires des roues, respectivement.

La relation entre le couple de roue et la couple moteur $[T_mR ; T_mL]$ en $\frac{1}{2}N:m$ est donnée comme suit :

$$\begin{cases} T_{mR} = l_m \left(n \frac{\dot{v}_R}{r}\right) + f_m \left(n \frac{V_R}{r}\right) + \frac{T_{wR}}{n} \\ T_{mL} = l_m \left(n \frac{\dot{v}_L}{r}\right) + f_m \left(n \frac{V_L}{r}\right) + \frac{T_{wL}}{n} \end{cases} \quad (2-11)$$

où $l_m[Kg:m^2]$ est le moment d'inertie du moteur, $f_m[N:s:m]$ est le frottement visqueux du moteur, et n est le rapport de démultiplication entre la roue et le moteur. $\left(n \frac{v_R}{r}, n \frac{v_L}{r}\right)$ et $\left(n \frac{\dot{v}_R}{r}, n \frac{\dot{v}_L}{r}\right)$ sont respectivement les vitesses angulaires et les accélérations angulaires des moteurs.

En remplaçant l'Eq. [24] dans l'éq. [25], on obtient :

$$\begin{cases} T_{mR} = \left(nl_m + \frac{l_w}{n}\right)\frac{\dot{v}_R}{r} + \left(nf_m + \frac{f_w}{n}\right)\frac{v_R}{r} + \frac{f_{Rr}}{n} \\ T_{mL} = \left(nl_m + \frac{l_w}{n}\right)\frac{\dot{v}_L}{r} + \left(nf_m + \frac{f_w}{n}\right)\frac{v_L}{r} + \frac{f_{Lr}}{n} \end{cases} \quad (2-12)$$

En substituant les forces de l'Eq. [26] dans l'équation précédente Eq. [27], on obtient après l'arrangement :

$$\left\{ \begin{array}{l} T_{mR} = \left(\frac{n^2 I_m + I_w}{nr} + \frac{2l^2 mr + Ir}{4l^2 n} \right) \dot{V}_R + \left(\frac{n^2 f_m + f_w}{nr} + \frac{2l^2 k_v r + k_\omega r}{4l^2 n} \right) V_R + \left(\frac{2l^2 mr - 1r}{4l^2 n} \right) \dot{V}_L + \\ \quad \left(\frac{2l^2 k_v r - k_\omega r}{4l^2 n} \right) V_L \\ T_{mL} = \left(\frac{n^2 I_m + I_w}{nr} + \frac{2l^2 mr + Ir}{4l^2 n} \right) V_L + \left(\frac{n^2 f_m + f_w}{nr} + \frac{2l^2 k_v r + k_\omega r}{4l^2 n} \right) \dot{V}_L + \left(\frac{2l^2 mr - lr}{4l^2 n} \right) \dot{V}_R + \\ \quad \left(\frac{2l^2 k_v r - k_\omega r}{4l^2 n} \right) V_R \end{array} \right. \quad (2-14)$$

Nous avons modélisé la partie mécanique du robot décrit par l'éq. [28], qui est la dynamique du robot qui relie le couple moteur à la vitesse des roues. L'étape suivante consiste à modéliser la partie électrique du robot moteurs.

Pour un moteur à courant continu, nous avons :

$$\left\{ \begin{array}{l} L_a \frac{dI_{aR}}{dt} + R_a i_{aR} + k_E \left(n \frac{V_R}{r} \right) = E_R \\ L_a \frac{dI_{aL}}{dt} + R_a i_{aL} + k_E \left(n \frac{V_L}{r} \right) = E_L \end{array} \right. \quad (2-15)$$

E_R et E_L [Volts] sont les tensions d'entrée des moteurs, c'est-à-dire

[A] est le courant d'induit, L_a [Henries] est l'inductance d'induit, R_a [Ω] est la résistance d'induit, K_E [$V, \frac{s}{rad}$] est la constante de force contre-électromotrice du moteur. Puisque l'inductance du moteur est omise, car elle est petite et généralement ignorée dans la dynamique du robot, nous pouvons négliger la dynamique du circuit électrique du moteur, $L_a \frac{di_a}{dt} = 0$, qui conduit à :

$$\left\{ \begin{array}{l} R_a i_{aR} + K_E \left(n \frac{V_R}{r} \right) = E_R \\ R_a i_{aL} + K_E \left(n \frac{V_L}{r} \right) = E_L \end{array} \right. \quad (2-16)$$

Le couple T_m développé par les moteurs est donné par la équation suivante :

$$\left\{ \begin{array}{l} T_{mR} = K_m i_{aR} \\ T_{mL} = K_m i_{aL} \end{array} \right. \quad (2-17)$$

où k_m [N.m/A] est la constante de couple du moteur À partir des éqs. [2-16] et [2-17], on obtient

$$\begin{aligned} T_{mR} &= \frac{k_m}{R_a} E_R - \frac{k_m k_E n}{R_a r} V_R \\ T_{mL} &= \frac{k_m}{R_a} E_L - \frac{k_m k_E n}{R_a r} V_L \end{aligned} \quad (2-18)$$

En remplaçant l'éq. [2-17] en éq. [2-18], on trouve :

$$(0-19) \left\{ \begin{aligned} &\left(\frac{n^2 l_m + l_w}{nr} + \frac{2l^2 mr + lr}{4l^2 n} \right) \dot{V}_R + \left(\frac{n^2 f_m + f_w}{nr} + \frac{2l^2 k_v r + k_\omega r}{4l^2 n} + \frac{k_m k_E n}{R_a r} \right) V_R + \left(\frac{2l^2 mr - lr}{4l^2 n} \right) \dot{V}_L + \left(\frac{2l^2 k_v r - k_\omega r}{4l^2 n} \right) V_L \\ &= \frac{k_m}{R_a} E_R \\ &\left(\frac{n^2 l_m + l_w}{nr} + \frac{2l^2 mr + lr}{4l^2 n} \right) \dot{V}_L + \left(\frac{n^2 f_m + f_w}{nr} + \frac{2l^2 k_v r + k_\omega r}{4l^2 n} + \frac{k_m k_E n}{R_a r} \right) V_L + \left(\frac{2l^2 mr - lr}{4l^2 n} \right) \dot{V}_R + \left(\frac{2l^2 k_v r - k_\omega r}{4l^2 n} \right) V_R \\ &= \frac{k_m}{R_a} E_L \end{aligned} \right.$$

la dynamique de l'entraînement différentiel robot qui est un ensemble de deux équations non linéaires couplées avec les tensions E_R et E_L comme grandeurs d'entrée et les vitesses de roue V_R et V_L comme variables de sortie. Cette équation peut s'écrire comme suit :

$$\begin{cases} K_1 \dot{V}_R + K_2 \dot{V}_L + K_3 V_R + K_4 V_L = \frac{k_m}{R_a} E_R \\ K_1 \dot{V}_L + K_2 \dot{V}_R + K_3 V_L + K_4 V_R = \frac{k_m}{R_a} E_L \end{cases} \quad (2-20)$$

Dans,

$$\begin{cases} \dot{V}_R = \frac{1}{K_1} \left[-K_2 \dot{V}_L - K_3 V_R - K_4 V_L + \frac{k_m}{R_a} E_R \right] \\ \dot{V}_L = \frac{1}{K_1} \left[-K_2 \dot{V}_R - K_3 V_L - K_4 V_R + \frac{k_m}{R_a} E_L \right] \end{cases} \quad (2-21)$$

Alor ,

$$K_1 = \left(\frac{n^2 l_m + l_w}{nr} + \frac{2l^2 mr + lr}{4l^2 n} \right) \quad (2-22)$$

$$K_2 = \left(\frac{2l^2 mr - lr}{4l^2 n} \right) \quad (2-23)$$

$$K_3 = \left(\frac{n^2 f_m + f_w}{nr} + \frac{2l^2 k_v r + k_w r}{4l^2 n} + \frac{K_m k_E n}{R_a r} \right) \quad (2-24)$$

$$K_3 = \left(\frac{2l^2 k_v r + k_w r}{4l^2 n} \right) \quad (2-25)$$

2.5 Discussion des défis et des solutions utilisées

La modélisation de la dynamique du Qbot 2e peut être un défi, mais il existe des solutions pour surmonter ces défis. Voici quelques-uns des défis associés à la modélisation de la dynamique [15] du robot Quanser Qbot 2e :

Non-linéarités : Le Qbot 2e est un système non-linéaire, ce qui signifie que sa dynamique ne peut pas être modélisée de manière linéaire. Les non-linéarités peuvent rendre la modélisation plus complexe et augmenter la difficulté de contrôle.

Bruit et incertitudes : Le Qbot 2e est soumis à des perturbations environnementales telles que les frottements, les vibrations, les changements de température, etc. Les mesures peuvent être affectées par des incertitudes [16] et du bruit, ce qui peut rendre difficile la prédiction de la dynamique du robot.

Dynamique des roues : Le Qbot 2e utilise des roues pour se déplacer, ce qui peut ajouter une complexité supplémentaire à la modélisation de la dynamique.

Pour surmonter ces défis, voici quelques solutions possibles :

Modélisation non-linéaire : Pour modéliser la dynamique non linéaire du Qbot 2e, il peut être nécessaire d'utiliser des outils mathématiques avancés tels que la théorie du contrôle non linéaire ou la méthode des équations [17] différentielles. Ces outils peuvent aider à modéliser le comportement complexe du robot.

Filtres de Kalman : Les filtres de Kalman peuvent être utilisés pour réduire les effets du bruit et des incertitudes dans les mesures. Ces filtres utilisent des estimations statistiques pour filtrer les données bruyantes et améliorer la précision des mesures.

Modélisation de la dynamique des roues : Pour modéliser la dynamique des roues, il peut être utile de développer des modèles de frottement et de glissement pour les roues. Ces modèles

peuvent aider à comprendre comment les forces appliquées aux roues affectent la dynamique globale du robot.

En résumé, la modélisation de la dynamique du robot Quanser Qbot 2e peut être un défi en raison de la non-linéarité, du bruit et des incertitudes, ainsi que de la dynamique des roues. Cependant, des outils mathématiques avancés tels que la théorie du contrôle non linéaire, les filtres de Kalman et la modélisation de la dynamique des roues peuvent être utilisés pour surmonter ces défis et améliorer la précision des modèles de dynamique [18].

2.6 Cinématique du Quanser Qbot

2.6.1 Cinématique et de son importance dans le contrôle des robots

La cinématique dans le contrôle des robots est l'étude des mouvements et des positions des robots. Elle joue un rôle essentiel dans la planification, la simulation et le contrôle des robots. La cinématique permet de décrire et de prédire le mouvement d'un robot en fonction de ses paramètres géométriques et des contraintes physiques auxquelles il est soumis.

La cinématique des robots 4, également connus sous le nom de robots à quatre degrés de liberté (DDL), concerne les robots qui peuvent effectuer des mouvements dans trois dimensions ainsi qu'une rotation autour d'un axe. Les DDL d'un robot correspondent aux différents mouvements indépendants qu'il peut réaliser. Par exemple, les articulations d'un robot peuvent lui permettre de se déplacer en translation et en rotation.

L'importance de la cinématique dans le contrôle des robots 4 réside dans les aspects suivants :

- Planification de trajectoire : La cinématique permet de planifier la trajectoire d'un robot en déterminant les positions et les orientations nécessaires pour atteindre une cible donnée. En connaissant les mouvements possibles du robot, il est possible de calculer les trajectoires les plus efficaces pour réaliser une tâche spécifique.
- Contrôle de mouvement : La cinématique fournit les outils nécessaires pour contrôler les mouvements du robot en temps réel. En utilisant des modèles cinématiques précises, il est possible de développer des algorithmes de contrôle qui ajustent les paramètres du robot pour atteindre les positions et les vitesses désirées.
- Évitement d'obstacles : La cinématique permet de prendre en compte les contraintes d'environnement lors de la planification des mouvements. En analysant les positions des obstacles et les capacités de mouvement du robot, il est possible de générer des trajectoires qui évitent les collisions.

- Simulation : La cinématique est utilisée pour simuler le comportement d'un robot avant de le mettre en action. Les modèles cinématiques permettent de prédire comment un robot se déplacera dans un environnement donné, ce qui est utile pour évaluer les performances du robot et optimiser ses mouvements.

2.6.2 Développement du modèle cinématique et la représentation mathématique du Quanser Qbot :

En tenant compte des forces agissant sur les roues droite et gauche, qui résultent des deux vitesses linéaires V_R et V_L (Figure 2-13), la vitesse du robot est déterminée par la vitesse linéaire V et la vitesse angulaire ω .

Si la roue droite patine alors que la roue gauche ne contribue pas et est à l'arrêt, elle commencera immédiatement à tourner à mi-vitesse puisque le point est à mi-chemin entre les deux roues. Il peut être défini dans la relation suivante :

$$V_{CR} = \frac{V_R}{2} \quad (2-26)$$

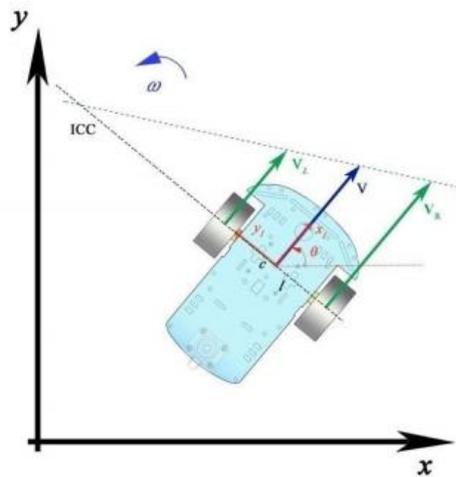


Figure 2-13 : Les forces sur les roues : une représentation équivalente

A l'inverse, si la roue gauche tourne alors que la roue droite est à l'arrêt, elle se met immédiatement à tourner à mi-vitesse :

$$V_{CL} = \frac{V_L}{2} \quad (2-27)$$

En ajoutant les deux dernières relations, nous pouvons calculer la vitesse V du point c :

$$V = V_{cR} + V_{cL} = \frac{V_R + V_L}{2} \quad (2-28)$$

De la même manière on peut calculer la vitesse angulaire ω si l'on considère la roue droite. Lorsque la roue droite tourne toute seule, le robot tourne vers la gauche. La vitesse de rotation en c peut être calculée car la roue suit immédiatement un arc de rayon $2l$:

$$\omega_{cR} = \frac{V_R}{2l} \quad (2-29)$$

En ajoutant les deux dernières relations, nous pouvons calculer la vitesse angulaire ω du point c :

$$\omega = \omega_{cR} + \omega_{cL} = \frac{V_R + V_L}{2l} \quad (2-30)$$

Nous avons obtenu les équations ci-dessus pour le cas où l'une des roues du robot est immobile et l'autre tourne. Et nous allons regarder le cas où les deux roues patinent.

En faisant varier la vitesse de chaque roue, le robot peut rouler ; Ce mouvement s'effectue autour d'un point commun, appelé centre de courbure instantané (ICC), qui se situe sur l'axe commun des cercles gauche et droit (Fig. 2-13). Comme la vitesse angulaire ω autour de l'ICC est la même pour les deux roues, on peut écrire l'équation suivante :

$$\omega = \frac{V_R}{R + l} = \frac{V_L}{R - l} \quad (2-31)$$

De cela nous obtenons :

$$\begin{aligned} V_R &= \omega(R + l) \\ V_L &= \omega(R - l) \end{aligned} \quad (2-31)$$

Où V_R est la vitesse de rotation de la roue droite, V_L est la vitesse de rotation de la roue gauche, R est la distance entre l'ICC et le centre du robot, et l est la moitié de la distance entre les roues DROITE et GAUCHE.

De l'équation (2-31) on soustrait l'équation (2-32), on obtient la vitesse angulaire :

$$\omega = \frac{V_R - V_L}{2l} \quad (2-32)$$

En ajoutant les mêmes deux équations, nous obtenons :

$$V_R + V_L = 2\omega R \quad (2-33)$$

Donc à partir l'équation (2-26) la vitesse de translation :

$$V = \frac{V_R + V_L}{2} = \omega R \quad (2-34)$$

À partir de là, nous pouvons calculer R :

$$R = \frac{V}{\omega} = l \frac{V_R + V_L}{V_R - V_L} \quad (2-35)$$

Alors que les coordonnées de l'ICC sont :

$$\text{ICC} = \begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix} = \begin{bmatrix} x_R - R \cos \theta \\ y_R - R \sin \theta \end{bmatrix} \quad (2-36)$$

La cinématique avant dans le repère local est donnée par :

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2l} & -\frac{1}{2l} \end{bmatrix} \begin{bmatrix} V_R \\ V_L \end{bmatrix} \quad (2-37)$$

La cinématique inverse est donnée par :

$$\begin{bmatrix} V_R \\ V_L \end{bmatrix} = \begin{bmatrix} 1 & l \\ 1 & -l \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix} \quad (2-38)$$

Puisque la vitesse de translation V est toujours dans la direction x_l , $x_l = V$ et $y_l=0$. La cinématique directe peut s'écrire :

$$\begin{bmatrix} \dot{x}_l \\ \dot{y}_l \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \\ \frac{1}{2l} & -\frac{1}{2l} \end{bmatrix} \begin{bmatrix} V_R \\ V_L \end{bmatrix} \quad (2-39)$$

À partir de l'équation (2-24), la transformation entre le système de coordonnées local et le système de coordonnées global peut être représentée comme suit :

$$\begin{bmatrix} x_G^l \\ y_G^l \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cos \theta & \frac{1}{2} \cos \theta \\ \frac{1}{2} \sin \theta & \frac{1}{2} \sin \theta \\ \frac{1}{2l} & -\frac{1}{2l} \end{bmatrix} \begin{bmatrix} V_R \\ V_L \end{bmatrix} \quad (2-40)$$

V_R et V_L sont les entrées du contrôleur, où:

$$\begin{aligned} V_R &= \omega_R r \\ V_L &= \omega_L r \end{aligned} \quad (2-41)$$

Donc, le modèle cinématique du robot est donnée par :

$$\begin{bmatrix} \dot{x}_G^l \\ \dot{y}_G^l \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{2l} & -\frac{r}{2l} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad (2-42)$$

2.6.3 Discussion des défis associés à la modélisation de la cinématique du robot

La modélisation de la cinématique d'un robot est un processus essentiel pour comprendre et contrôler les mouvements de celui-ci. Cependant, il existe plusieurs défis associés à cette tâche, et des solutions ont été développées pour les surmonter. Voici quelques-unes de ces défis et solutions :

- Cinématique directe : Le défi principal consiste à calculer la position et l'orientation finales du robot en fonction de ses articulations et de ses paramètres géométriques. La solution couramment utilisée est d'appliquer les transformations homogènes successives pour relier les articulations du robot et obtenir la pose finale. Les matrices de transformation homogènes sont utilisées pour représenter les rotations et les translations.
- Cinématique inverse : Ce défi implique de déterminer les configurations articulaires nécessaires pour atteindre une position et une orientation spécifiques de l'outil ou de l'effecteur final du robot. La résolution de ce problème peut être complexe, en particulier pour les robots avec de nombreux degrés de liberté. Des méthodes numériques, telles que la méthode de Newton-Raphson ou des approches basées sur des algorithmes génétiques, sont souvent utilisées pour résoudre la cinématique inverse.
- Singularités : Les singularités se produisent lorsque le robot atteint une configuration où le calcul de la cinématique inverse devient difficile ou impossible. Cela peut entraîner des

problèmes de stabilité ou de contrôle du robot. Une approche courante pour surmonter ce défi est d'éviter les configurations singulières en planifiant les trajectoires ou en utilisant des méthodes de résolution itérative qui s'éloignent des singularités.

-Interférences et limites de mouvement : Certains robots peuvent rencontrer des interférences entre leurs articulations, leurs liens ou leur environnement lorsqu'ils se déplacent dans certaines configurations. La modélisation précise des limites de mouvement des articulations et des contraintes d'évitement des collisions est nécessaire pour éviter ces problèmes. Des techniques telles que les limites articulaires virtuelles, les champs de forces artificiels ou les algorithmes de planification de trajectoire sont utilisées pour garantir des mouvements sûrs et éviter les interférences.

-Erreurs de modélisation et imprécisions : La modélisation de la cinématique du robot repose souvent sur des approximations et des paramètres idéalisés. Cependant, dans la réalité, il peut y avoir des erreurs de modélisation, des jeux mécaniques, des flexions des liens, des tolérances d'usinage, etc. Ces facteurs peuvent entraîner des imprécisions dans les calculs cinématiques. Pour minimiser ces erreurs, des techniques de calibration et de recalibration sont utilisées pour ajuster les modèles cinématiques aux caractéristiques réelles du robot.

2.7 Simulation et Validation des Modèles

2.7.1 Réponse indicielle du modèle cinématique

Nous appliquons un modèle d'entrée cinématique aux deux entrées comme le montre la Figure 2 -14: Le robot est initialement à $(X_G, Y_G,) = (0m, 0m, 0^\circ)$.

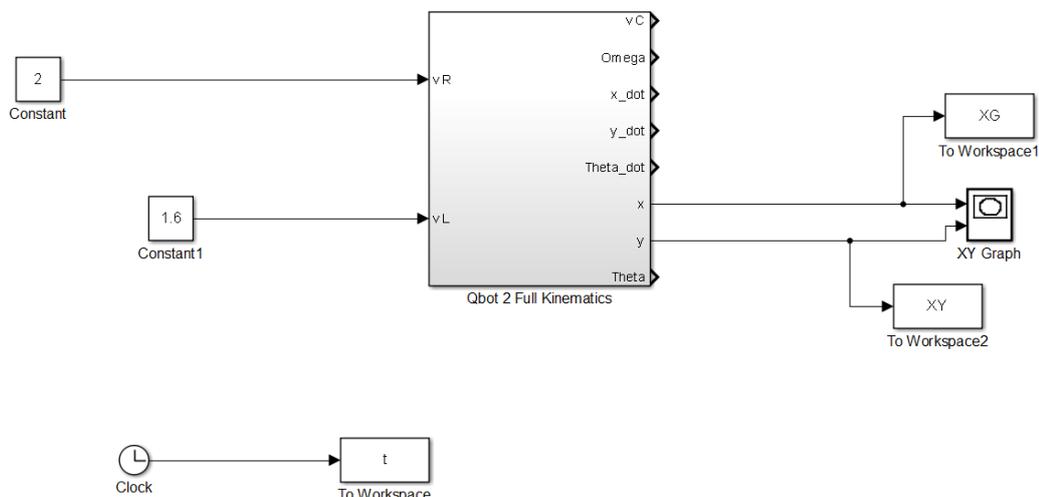


Figure 2-14 : Modèles de la réponse indicielle de la cinématique du robot sur SIMULINK

La même valeur de vitesse est appliquée aux deux entrées du robot (V_R, V_L). Les résultats obtenus sont présentés dans la Figure 2-15. Le robot se déplace en ligne droite sans changer son orientation d'origine $\varphi = 0$

Graphique 2-16. Le robot se déplace en ligne droite sans changer son orientation initiale $\varphi = \frac{\pi}{3}$ rad.

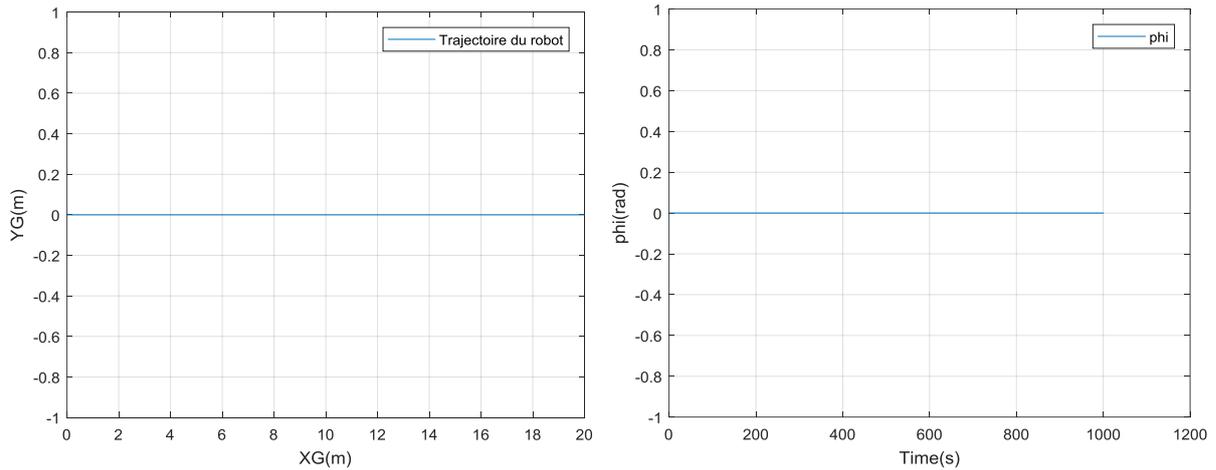


Figure 2-15 : La réponse indicielle ($V_R=2\text{m/s}$ et $V_L=-2\text{m/s}$ avec $\varphi=0$)

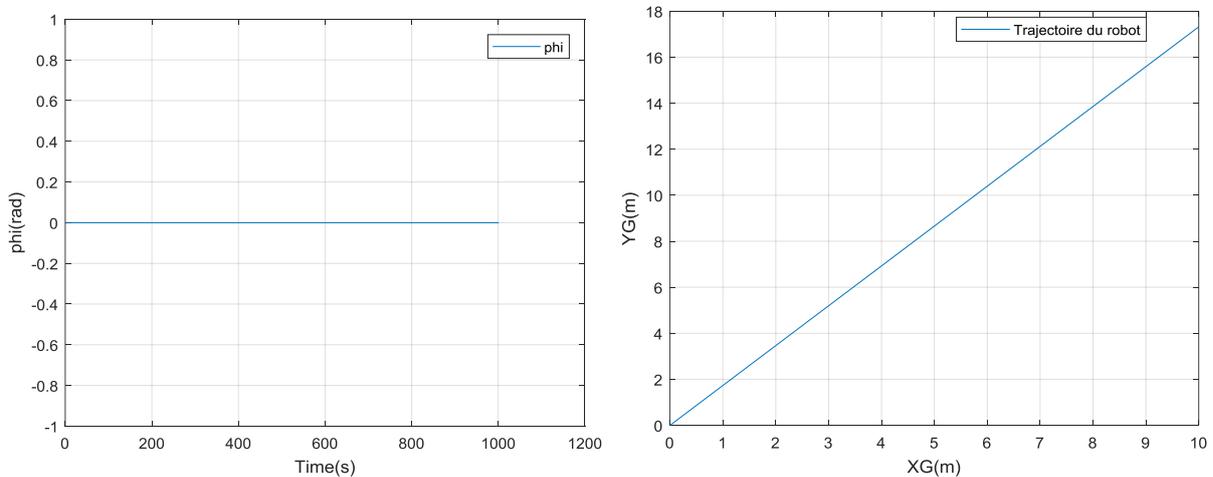


Figure 2-16 : La réponse indicielle ($V_R=2\text{m/s}$ et $V_L=2\text{m/s}$ avec $\varphi = \frac{\pi}{3}$ rad)

Les résultats obtenus à partir des Figures 2-15 et 16 montrent que le robot se déplace également en ligne droite.

Maintenant, nous appliquons différentes valeurs aux entrées du modèle ($V_R = 2 \text{ m/s}$ et $V_L = 0.6 \text{ m/s}$) avec $\varphi = \frac{\pi}{3}$ rad.

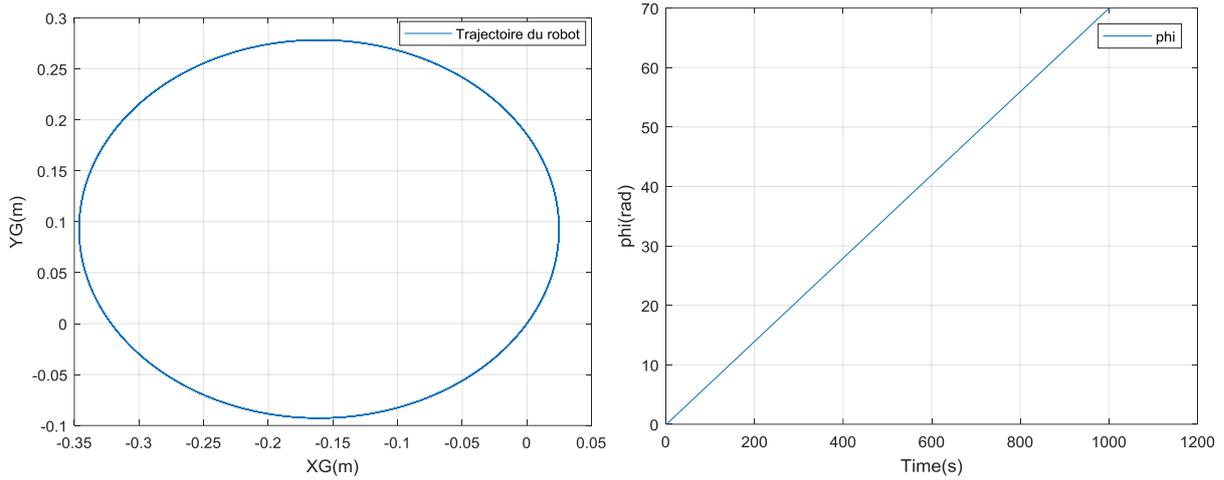


Figure 2-17 : La réponse indicielle ($V_R = 2\text{m/s}$ et $V_L = 0.6\text{m/s}$ avec $\varphi = \pi 3\text{rad}$)

Le robot dessine un cercle et l'orientation augmente linéairement, comme le montre la Figure 2.17.

2.7.2 Réponse indicielle du modèle dynamique

De même, différents types de données d'entrée sont utilisés pour tester un système dynamique (Figure II.18).

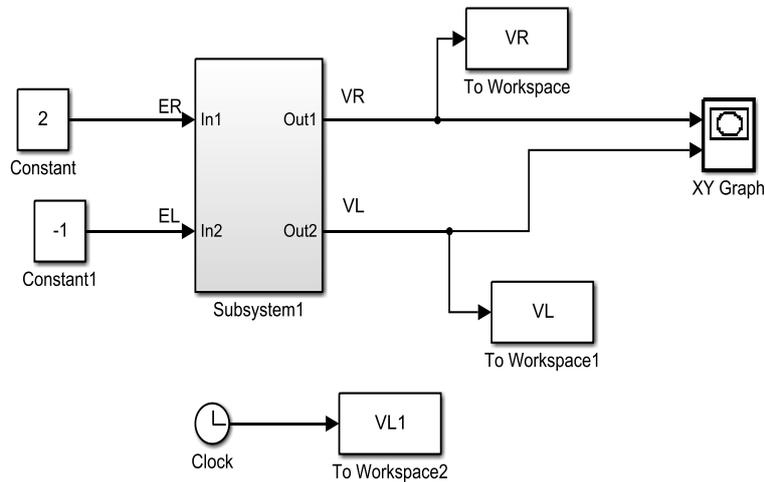


Figure 2-18 : Modèle de la réponse indicielle de la dynamique du robot sur SIMULINK

Les résultats sont donnés dans les Figures 2-19 et 2-20.

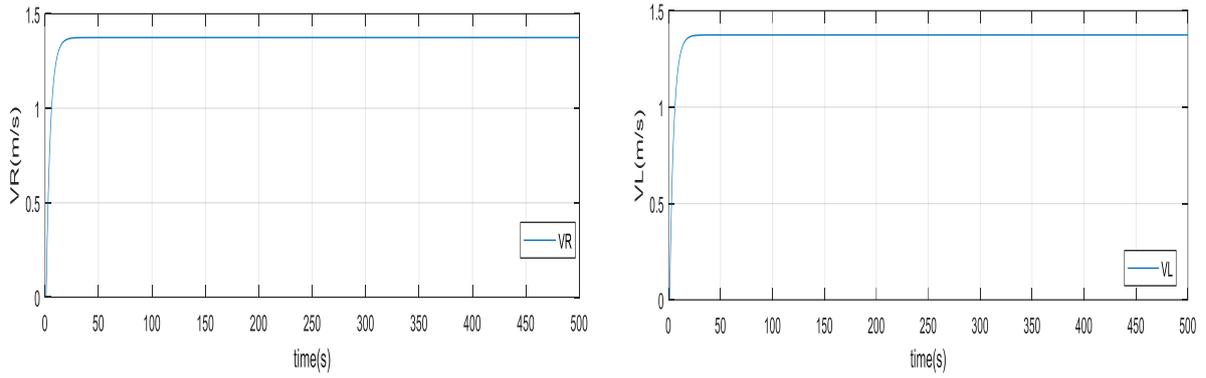


Figure 2-19 : La réponse indicielle ($V_R = 5V$ et $V_L = 5V$)

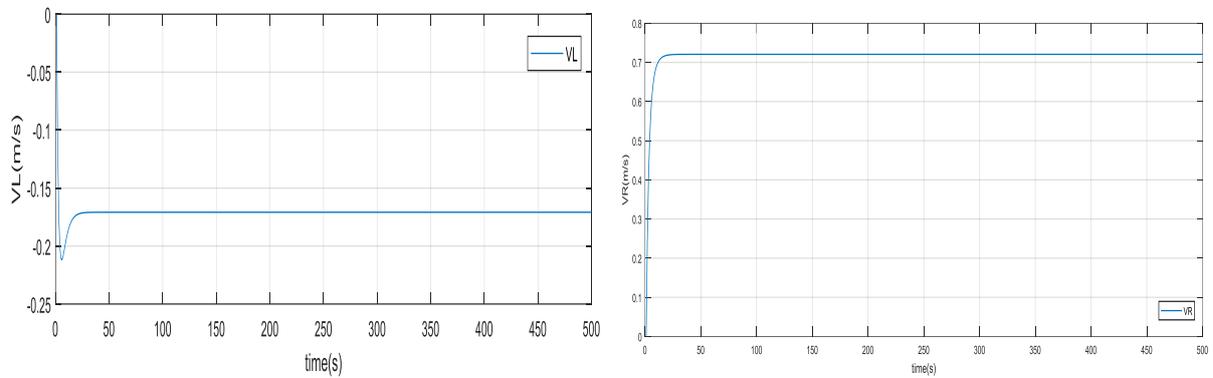


Figure 2-20 : La réponse indicielle ($E_R = 3V$ et $E_L = -1 V$)

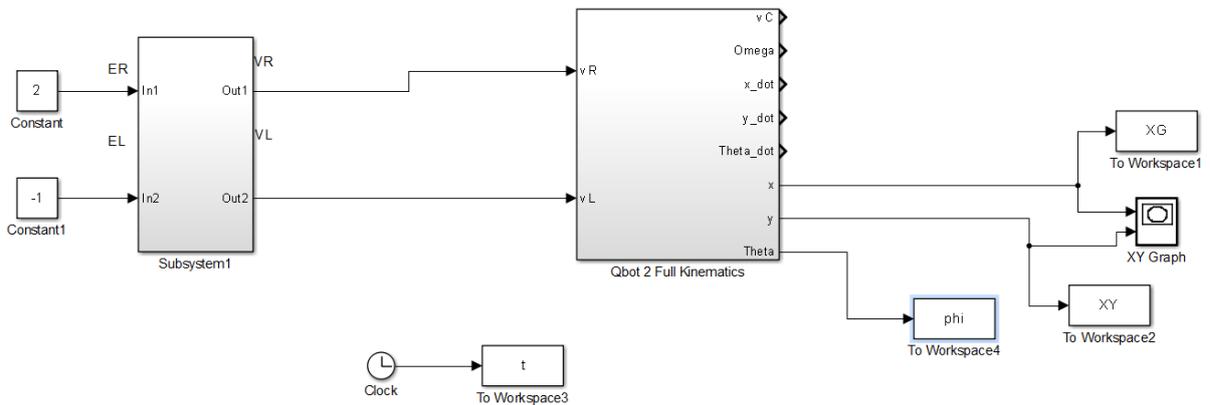


Figure 2-21: Modèle de la réponse indicielle de la dynamique avec la cinématique du robot sur SIMULINK

Nous combinerons le modèle cinématique avec le modèle dynamique pour voir le modèle complet testé (Figure 2-21). La configuration initiale du robot est $(X_G, Y_G, \varphi) = (0 \text{ m}, 0 \text{ m}, 45^\circ)$.

Les résultats sont présentés dans la figure (2-22) La même valeur est présente sur les deux moteurs ($E_R = E_L = 5V$)

la Figure (2-22).deux valeurs différentes est appliquée aux deux moteurs ($E_R = 5V$ et $E_L = 3V$).

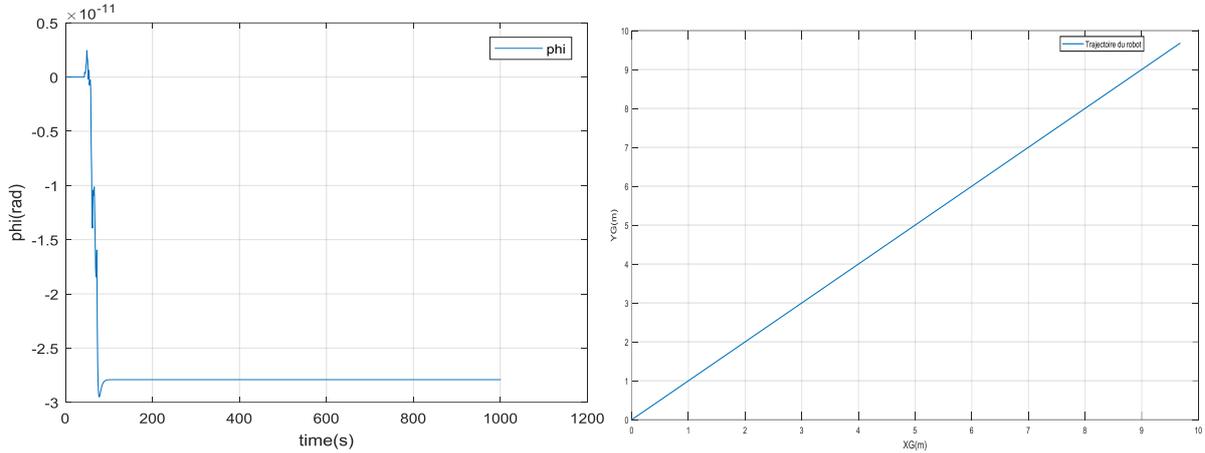


Figure 2-22 : La réponse indicielle ($E_R = 5V$ et $E_L = 5V$ avec $\varphi = \frac{\pi}{4}$ rad)

Les résultats obtenus de la Figure 2-16 montrent que le robot dessine une ligne droite avec l'angle de $\frac{\pi}{4}$.

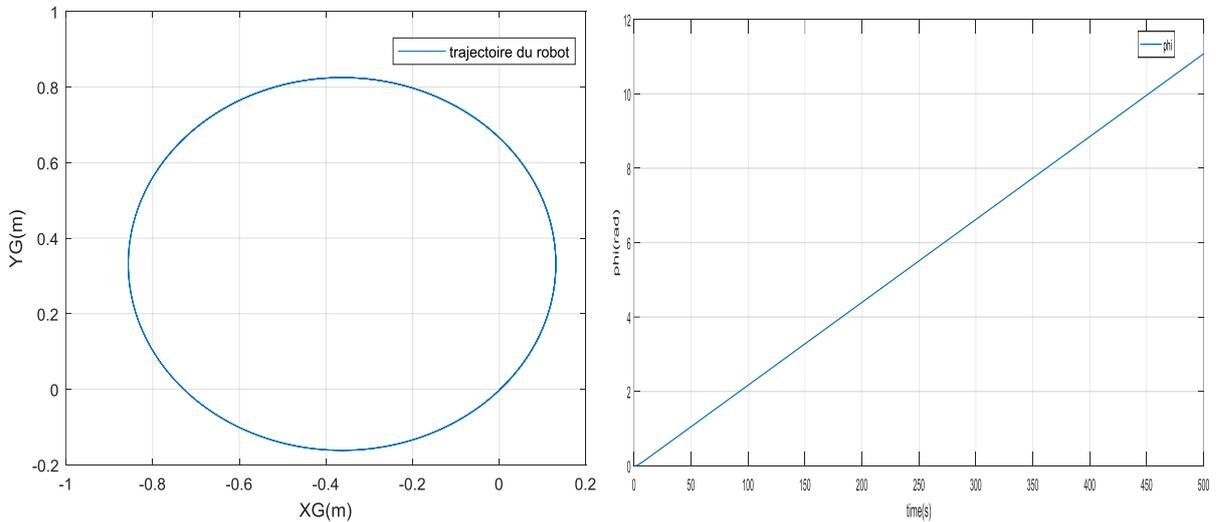


Figure 2-23 : La réponse indicielle ($E_R = 5V$ et $E_L=3V$ avec $\varphi = \frac{\pi}{4}$ rad)

Les résultats obtenus à partir de la Figure 2-23 montrent que le robot trace un cercle. Les réponses échelonnées des modèles cinématiques et dynamiques sont conformes aux attentes. Par conséquent, ce modèle offre une représentation réelle du robot.

2.8 Analyse de l'impact de différents paramètres

L'analyse de l'impact de différents paramètres, tels que la taille du robot QBot Quanser et les mesures des capteurs, sur la validité du modèle peut aider à comprendre comment ces variables affectent les performances du système robotique et la précision des modèles prédictifs.

-Dimensions du Quanser QBot :

Les dimensions d'un robot, telles que sa taille, sa forme et son poids, peuvent affecter sa maniabilité, sa stabilité et sa capacité à effectuer certaines tâches. Par exemple, un robot plus grand peut être plus difficile à contrôler avec précision, tandis qu'un robot plus petit peut avoir une portée ou une charge utile limitée. Par conséquent, la taille du robot peut affecter la validité des modèles en termes de portabilité des performances du entre différentes configurations de robot.

-Caractéristiques de l'environnement :

Les propriétés de l'environnement dans lequel le robot travaille peuvent également affecter la validité des modèles. Par exemple, des changements d'éclairage, des obstacles inattendus ou des surfaces glissantes peuvent entraîner des erreurs dans les lectures du capteur. Par conséquent, il est important de tenir compte de ces facteurs environnementaux lors de la conception des modèles et d'en tenir compte lors de l'évaluation de leur validité.

2.8.1 Evaluation des résultats

Pour évaluer des performances des modèles dans différentes conditions, telles que les changements de vitesse de Quanser QBot, est essentielle pour comprendre la robustesse des modèles prédictifs.

Changement de vitesse du robot :

Lorsque QBot change sa vitesse, il peut affecter sur sa dynamique, sa stabilité et sa précision de contrôle. Évaluez les modèles prédictifs pour voir s'ils peuvent prédire avec précision le comportement du robot à différentes vitesses. Des tests peuvent être effectués en faisant varier la vitesse du robot dans une plage spécifiée et en comparant les prédictions du modèle aux données réelles pour évaluer sa validité et sa précision.

Modifications des conditions environnementales :

En plus des changements de vitesse, d'autres changements dans les conditions environnementales peuvent affecter les performances des modèles. Cela peut inclure, des différences de finition de surface.

2.9 Conclusion :

La modélisation est un principe ou une technique qui permet de créer un modèle explicatif d'un phénomène ou d'un comportement et est une phase très importante dans le contrôle d'un robot.

Deux types de modèles sont généralement utilisés lors de la commande, à savoir un modèle cinématique et un modèle dynamique.

Dans ce chapitre la modélisation d'un robot mobile est présentée. Nous avons également développé les équations cinématiques et dynamiques du robot mobile. Nous avons validé les modèles par simulation avec Simulink Matlab.

Dans le chapitre suivant, nous proposons un système de contrôle basé sur PID et contrôle fuzzy pour le problème de suivi de ligne.

Chapitre 3

Conception de la commande floue

2.1 Introduction

Dans ce chapitre, nous allons rappeler brièvement l'objectif global de notre étude, qui consiste à développer et à valider une commande floue pour le suivi de trajectoire d'un robot mobile Quanser Qbot. Nous avons choisi cette approche de commande floue en raison de sa capacité à gérer des situations où les modèles mathématiques précis du système difficile à obtenir ou à utiliser.

La commande floue est une méthode de contrôle basée sur la logique floue, qui permet de traiter des informations imprécises ou incertaines. Elle utilise des règles linguistiques et une représentation floue des variables d'entrée et de sortie pour générer des commandes adaptées au système. Dans notre cas, le système est le robot mobile Quanser Qbot, et notre objectif est de développer une commande floue qui lui permettra de suivre des trajectoires prédéfinies avec précision.

Dans la section suivante, nous expliquerons les principes fondamentaux de la logique floue et de la commande floue. Nous présenterons les concepts clés tels que les ensembles flous, les fonctions d'appartenance, les opérateurs flous et les règles linguistiques. Nous montrerons également comment ces concepts peuvent être utilisés pour concevoir une commande floue adaptée au suivi de trajectoire du robot Qbot.

Après avoir présenté les principes théoriques, nous passerons à la conception de notre commande floue pour le suivi de trajectoire. Nous expliquerons comment nous avons choisi et défini les variables d'entrée et de sortie, ainsi que les règles linguistiques qui relient ces variables. Nous discuterons également des différentes méthodes de défuzzification que nous avons considérées pour convertir les sorties floues en commandes de contrôle concrètes.

Enfin, nous décrirons les expériences que nous avons menées pour valider notre commande floue. Nous présenterons les scénarios de test, les métriques de performance que nous avons utilisées et les résultats obtenus. Nous analyserons les performances de notre commande floue par rapport à d'autres méthodes de contrôle, mettant en évidence ses avantages et ses limites.

En conclusion de ce chapitre, nous résumerons les principales contributions et les résultats obtenus jusqu'à présent. Nous discuterons également des perspectives d'amélioration et des travaux futurs pour poursuivre le développement de notre commande floue et son application dans d'autres domaines de contrôle des robots mobiles.

2.2 Conception de la commande floue

La conception de la commande floue pour le suivi de trajectoire du Qbot comprend plusieurs étapes. Tout d'abord, permettez-nous de vous expliquer brièvement les principes de base de la logique floue.

La logique floue est une méthode qui permet de traiter des informations imprécises ou incertaines en utilisant des variables linguistiques et des règles floues. Contrairement à la logique classique, qui utilise des valeurs binaires (vrai/faux), la logique floue permet de représenter des degrés de vérité entre 0 et 1. Elle est particulièrement utile lorsque les conditions de contrôle sont difficiles à définir de manière précise.

Maintenant, parlons des variables d'entrée et de sortie de la commande floue pour le suivi de trajectoire du Qbot. Les variables d'entrée sont généralement liées à des informations provenant des capteurs, telles que la distance par rapport à la trajectoire désirée et l'angle d'orientation du Qbot par rapport à la trajectoire. Les variables de sortie sont les commandes de contrôle, telles que la vitesse et l'angle de direction.

Pour représenter ces variables, nous utilisons des ensembles flous, également appelés membres linguistiques. Par exemple, la variable d'entrée "distance par rapport à la trajectoire" pourrait avoir des membres linguistiques tels que "proche", "moyenne" et "loin". De même, la variable de sortie "angle de direction" pourrait avoir des membres linguistiques tels que "gauche", "tout droit" et "droite". Ces membres linguistiques décrivent les différentes valeurs que les variables peuvent prendre.

Maintenant, passons à l'architecture générale de la commande floue. Elle est composée de trois blocs principaux : le bloc de fuzzification, le bloc de règles et le bloc de défuzzification.

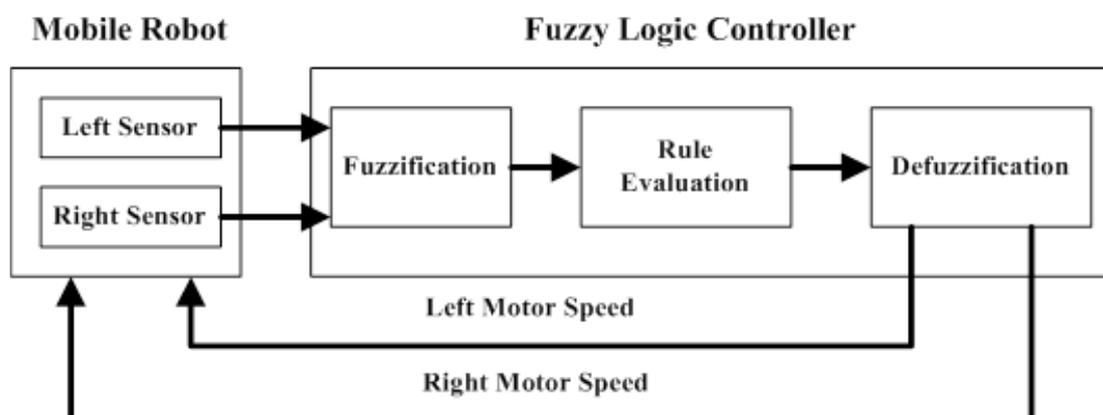


Figure 3-1: Contrôleur flou du Robot Qbot2

- a) **Bloc de fuzzification** : Ce bloc convertit les valeurs numériques des variables d'entrée en degrés d'appartenance aux ensembles flous correspondants. Cela se fait en évaluant à quel point chaque valeur d'entrée appartient à chaque membre linguistique. Ainsi, une valeur d'entrée donnée peut avoir un degré d'appartenance élevé à plusieurs membres linguistiques.
- b) **Bloc de règles** : Ce bloc contient un ensemble de règles qui décrivent la relation entre les membres linguistiques des variables d'entrée et de sortie. Par exemple, une règle pourrait être formulée comme suit : "Si la distance par rapport à la trajectoire est proche ET l'angle d'orientation est à gauche, alors l'angle de direction doit être fort à gauche." Ces règles capturent les connaissances et les heuristiques du domaine.
- c) **Bloc de défuzzification** : Ce bloc convertit les degrés d'appartenance des membres linguistiques de la variable de sortie en une valeur numérique précise. Cela se fait en prenant en compte les règles floues et en calculant un résultat agrégé. Par exemple, on peut utiliser la méthode du centre de gravité pour déterminer la valeur finale de l'angle de direction.

Le problème de suivi de trajectoires est illustré dans la Figure 3-2. À tout instant, le robot doit suivre le point actuel désiré, il doit donc minimiser l'erreur de distance \mathbf{Dis} et l'erreur d'orientation β .

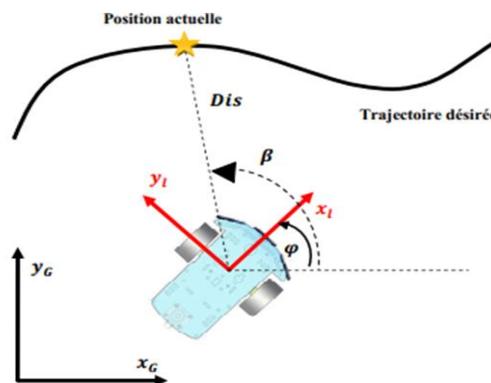


Figure 3-2 : Le problème de suivi de trajectoire

La distance (en mètres) entre le robot et la cible est donnée par :

$$Dis = \sqrt{(y_T - y_R)^2 + (x_T - x_R)^2} \quad (3-1)$$

où $(x_T; y_T)$ et $(x_R; y_R)$ sont respectivement les coordonnées des points réels du robot.

L'angle d'orientation β (en radians) est donné par :

$$\beta = \tan^{-1} (y_T - y_R, x_T - x_R) \quad (3-2)$$

Où $\text{atan2}(y, x)$ est la tangente inverse à quatre quadrants (arc tangente) de y et x , et le résultat appartient au fermé intervalle $[-\pi, \pi]$, tandis qu'à un $\text{atan}\left(\frac{y}{x}\right)$ prend ses valeurs dans l'intervalle $[-\pi/2, \pi/2]$.

Dans la théorie de contrôle flou, les commandes induites sont traitées en trois étapes :

Fuzzification, inférence et défuzzification. Les règles linguistiques peuvent être par exemple :

R1: Si le but est Loin Alors la vitesse est Grande et le braquage est Zéro,

R2: Si l'obstacle est à droite Alors la vitesse est Moyenne et le braquage est à Gauche,

....

Rn: Si le but est Proche Alors la vitesse est Faible et le braquage est Zéro.

2.3 Le contrôleur PID

Le contrôleur proportionnel-intégral-dérivé (PID) est un mécanisme de rétroaction en boucle fermée. Comme son nom l'indique, l'algorithme PID se compose de trois coefficients de base : les coefficients proportionnels, intégraux et dérivés, qui sont variés pour obtenir la réponse optimale [23], en utilisant les valeurs des constantes K_P , K_D et K_I . K_P dépend des erreurs actuelles, K_I dépend de l'accumulation des erreurs passées et K_D est la prédiction des erreurs futures. Ils fournissent signaux de commande qui sont proportionnels à l'erreur entre le signal de référence et la sortie courant (action proportionnelle), l'intégrale de l'erreur (action intégrative) et la dérivée de l'erreur (dérivée de l'action).

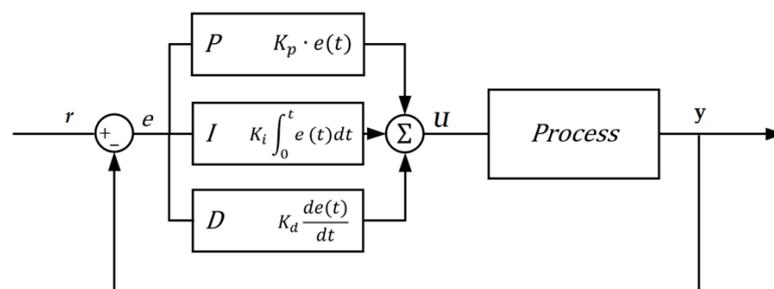


Figure 3-3 : Correcteur PID

Forme :

$$U(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{d}{dt} \quad (3-3)$$

$U(t)$ = signal de commande

$e(t)$ = signal d'erreur

K_P = Gain proportionnel

K_I = gain intégral

K_D = gain dérivé

II.3.1.1 Conception du contrôleur PID :

La figure illustre le problème suivant avec les chemins. Pendant ce temps, le robot doit suivre le point actuel requis, puis réduire l'erreur de distance et l'erreur de direction, le contrôleur PID calcule l'erreur système, qui est la différence entre son état actuel et le point de consigne souhaité, et calcule la commande pour réduire cette erreur. Principalement pour les réglages du gain proportionnel (K_p), i. H le gain d'intégration (K_I) et le gain de différence (K_D) pour obtenir un contrôle optimal, c'est-à-dire Le contrôleur PID

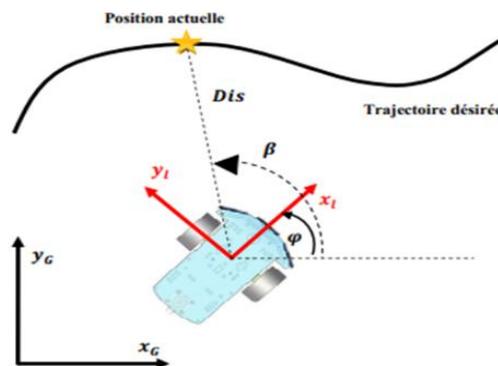


Figure 3-4 : Le problème de suivi de trajectoire

a une entrée et une sortie comme le montre l'image. Les entrées sont :

Somme de la distance Dis et de l'angle d'orientation du robot β .

sortie son :

Tension du moteur VR droit.

Tension moteur gauche VL.

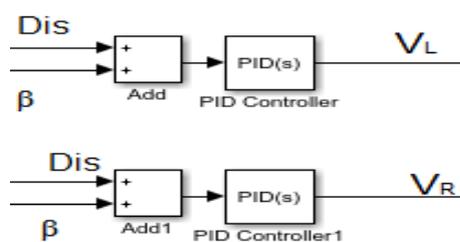


Figure 3-5 : : L'entrées et la sortie de PID

II.4.1.2 Simulation:

Le système robotique mobile est défini par un modèle cinématique et dynamique. Les valeurs d'orientation et de position du robot sont calculées et utilisées comme rétroaction pour l'ensemble du système.

Les entrées du contrôleur PID sont la distance entre le robot et le point souhaité Dis et l'angle d'orientation relatif du robot par rapport au point souhaité β [19]

$$Dis = \sqrt{(y_{traj} - y_{robot})^2 + (x_{traj} - x_{robot})^2} \tag{3-4}$$

$$\beta = \tan^{-1} \left(\frac{y_{traj} - y_{robot}}{x_{traj} - x_{robot}} \right) \tag{3-5}$$

Le système général de control est illustrée dans la Figure

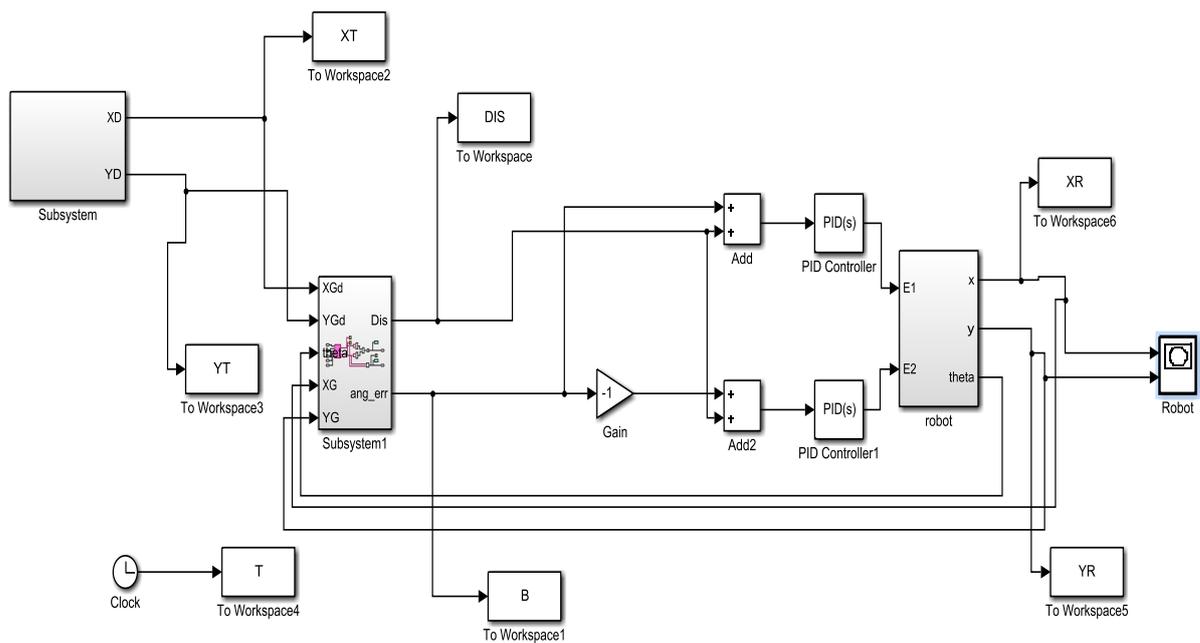


Figure 3-6 : Système générale de PID contrôle pour le problème du suivi de trajectoires

Pour vérifier l'efficacité du contrôleur PID dans le suivi de la trajectoire, deux trajectoires ont été testées : une trajectoire circulaire et une trajectoire en forme de pétale

Les paramètres de contrôle suivants ont été définis par essais et erreurs :

$P=15, I=0.6, D=0.8$

- Trajectoire circulaire

Le robot doit suivre une trajectoire circulaire comme illustré à la Figure 3-. Le robot est initialement à (0 m, 0 m, 0 rad).

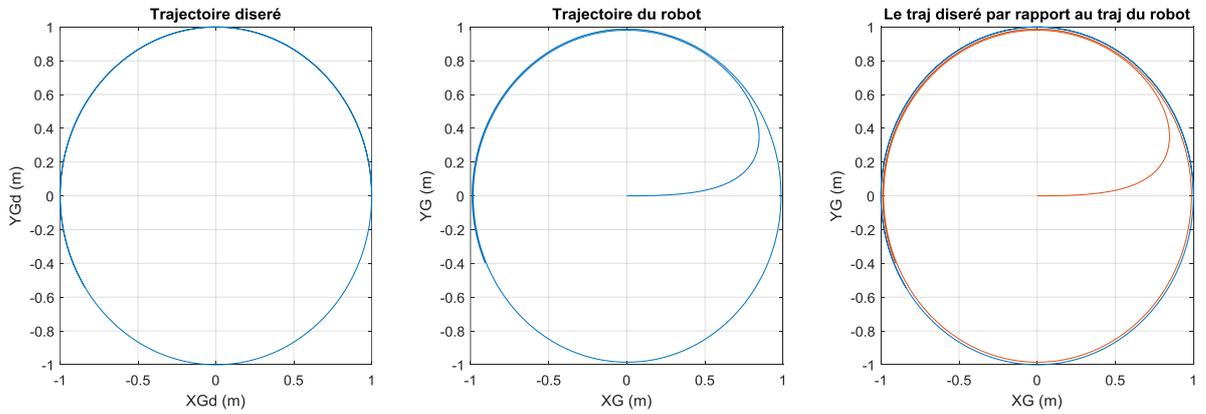


Figure 3-7: Suivre une trajectoire circulaire

Les erreurs de distance entre le robot et le point souhaité et l'angle d'orientation sont illustrés à la Figure 3-.

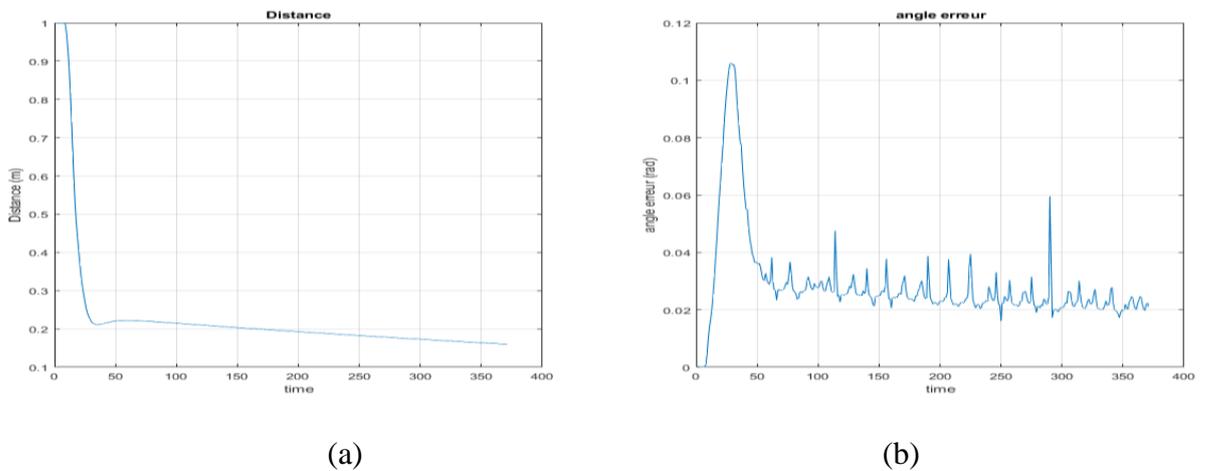


Figure 3-8 : l'erreur sur la : a) distance (Dis), b) orientation(β)

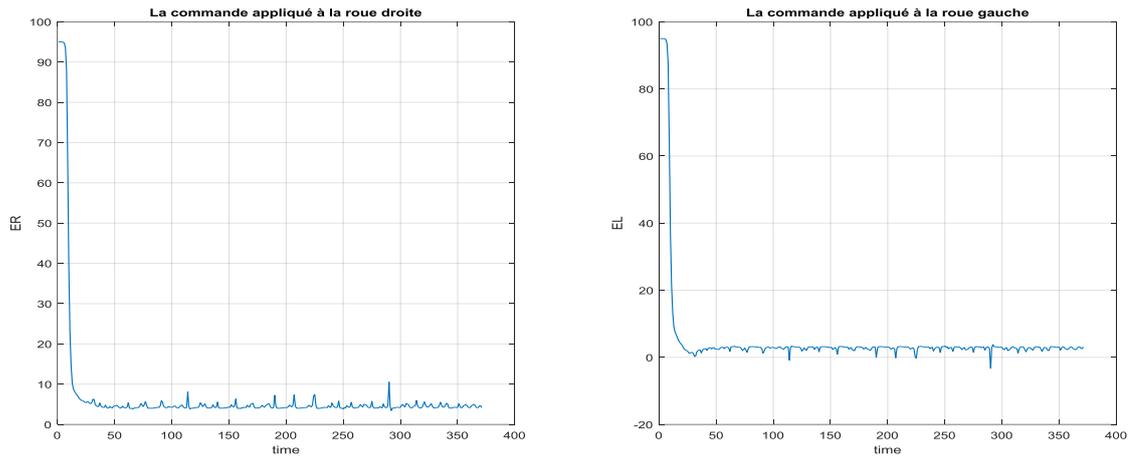
a) Tension du moteur Droit E_R b) Tension du moteur Gauche E_L

Figure 3-9 : La commande appliqué pour la trajectoire circulaire

Le contrôleur PID proposé a montré un bon suivi et une bonne stabilité pour différentes trajectoires données, et les résultats de la simulation montrent l'efficacité de l'approche proposée et que le contrôleur proposé peut être développé efficacement avec les paramètres de contrôle. Proportionnelle (P), Intégrale (I), Dérivée (D).

2.4 Le contrôleur flou

Le robot est un système multidimensionnel avec deux équations de mouvement non linéairement couplées, chaque sortie est couplée à l'autre. Traiter un degré élevé de non-linéarité nécessite un contrôleur qui traite cette non-linéarité comme un contrôleur flou, capable de contrôler un système multidimensionnel et de gérer à la fois la non-linéarité et l'incertitude. C'est donc le bon choix pour notre problème.

Le contrôleur flou est une application statique non linéaire entre ses entrées et ses sorties (c'est-à-dire qu'il ne s'agit pas d'un système dynamique). Les entrées et les sorties sont "nettes", c'est-à-dire qu'il s'agit de nombres réels et non d'ensembles flous. Il comporte quatre composants principaux : Le bloc de fuzzification, la base de règles, le mécanisme d'inférence et le bloc de défuzzification.

Le contrôleur flou a deux entrées et deux sorties. Les entrées sont la distance \mathbf{Dis} entre le robot et le point actuel de la trajectoire, et l'angle d'orientation $\mathbf{\beta}$ (l'angle de relèvement) du robot par rapport au point actuel désiré. Les sorties sont Tension moteur droit \mathbf{VR} et tension moteur gauche \mathbf{VL} .

2.4.1.1 La procédure de fuzzification

Les fonctions d'appartenance triangulaire et trapézoïdale sont les plus couramment utilisées dans la littérature. Alors que la raison la plus évidente de leur utilisation est la simplicité surprenante de cette forme de fonction d'appartenance, de nombreuses recherches ont abordé ce sujet et ont montré que les fonctions d'appartenance triangulaires et trapézoïdales sont supérieures aux autres types de fonctions d'appartenance [30-31].

Ainsi, les fonctions d'appartenance triangulaire et trapézoïdale sont utilisées pour représenter les valeurs floues d'entrée et de sortie des variables linguistiques : distance Dis, l'angle β , et vitesses des roues VR et VL. La portée de la "distance" Dis de l'univers flou est $[0m, 1m]$ et est suffisante pour toute distance supérieure à 0. β est limité à $[-p, p]$. Étant donné que les deux moteurs sont alimentés à 5 V dans cette étude, la gamme vocale pour les deux sorties (tension du moteur gauche et droit) est $[-1 V, 1 V]$.

Les fonctions d'appartenance d'entrée du contrôleur sont illustrées aux figures 3-10 et 3-11 tandis que les sorties sont illustrées à la figure 3-12.

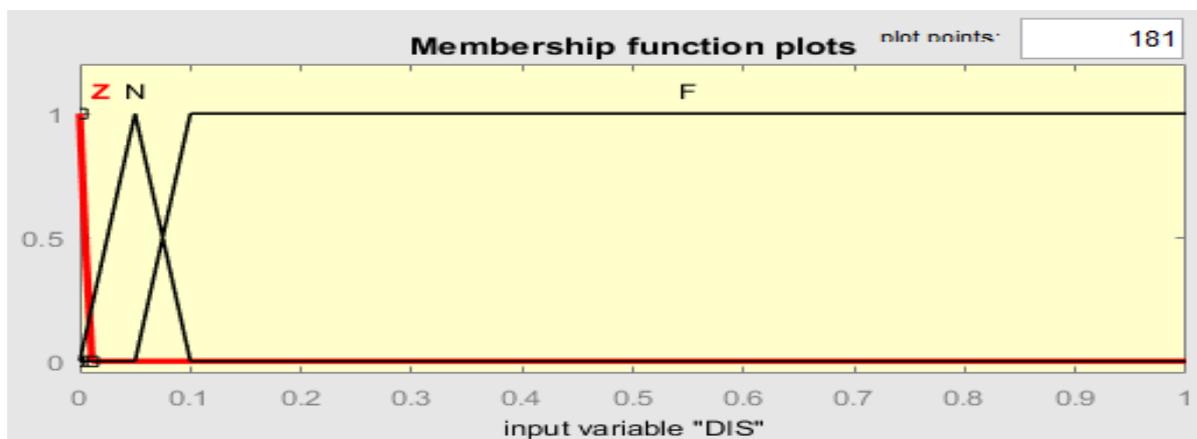


Figure 3-10 : Les fonctions d'appartenance de l'entrée Dis

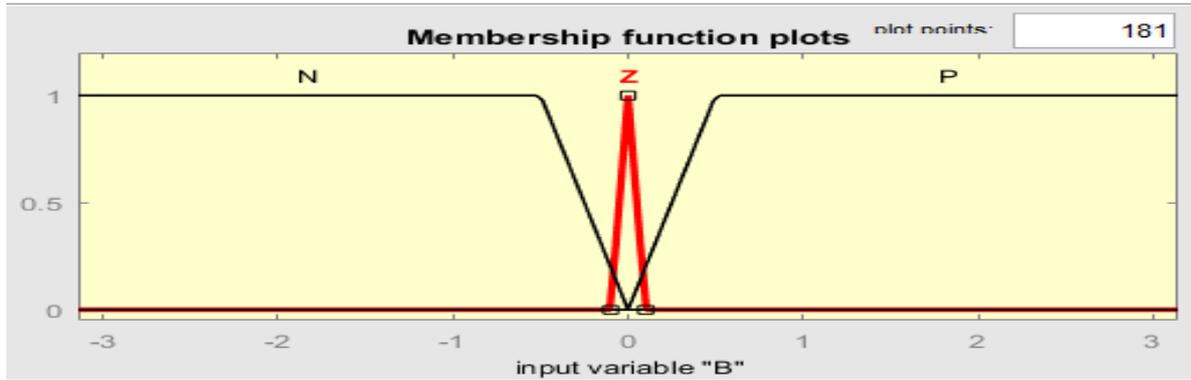


Figure 3-11 : Les fonctions d'appartenance de l'entrée β

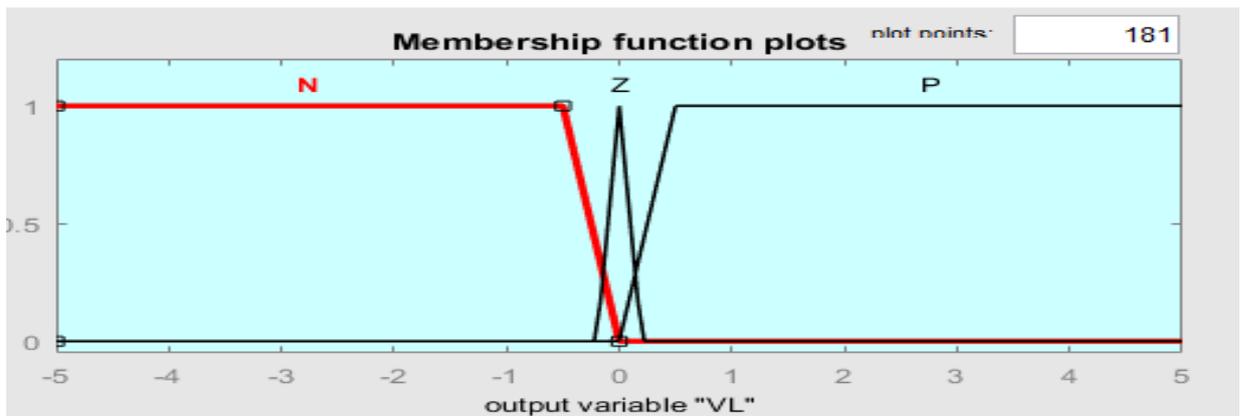


Figure 3-12 : Les fonctions d'appartenance de sortie V_R et V_L

Tableau 3-1 : Variables linguistiques et floues pour les 3 entrées et les 3 sorties

Dis (m)	Z Zéro	N Near	F Far
β (rad)	N Négative	Z Zéro	P Positive
VR et VL	N Négative	Z Zéro	P Positive

Tableau 3-2 : Variables linguistiques et floues pour les 3 entrées et les 5 sorties

Dis (m)	Z Zéro		N Near	F Far	
β (rad)	N Négative		Z Zéro	P Positive	
VR et VL	N Négative	SN Small Négative	Z Zéro	P Positive	SP Small Positive

Le système général de contrôle est illustrée dans la Figure 3-13

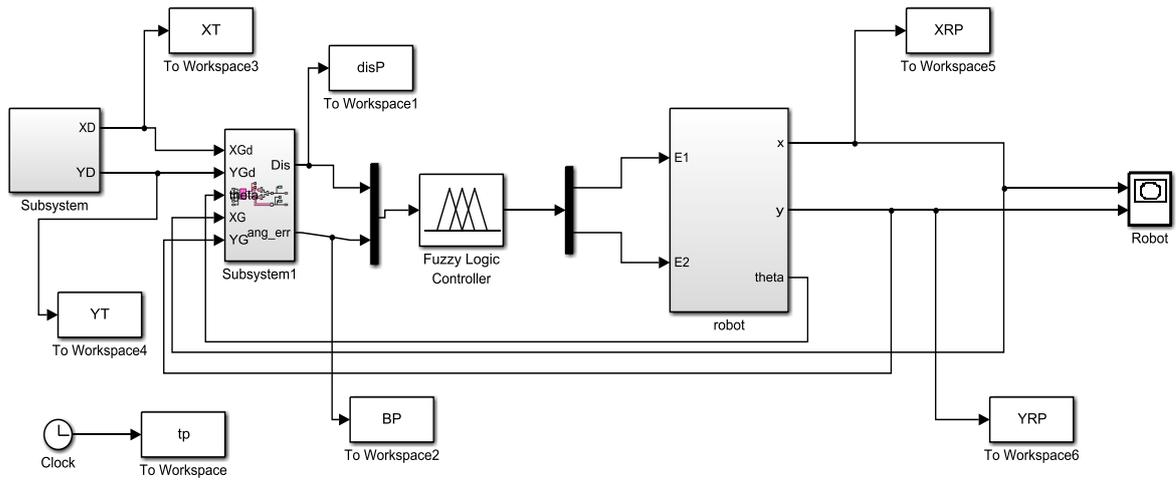


Figure 3-13 : Schéma générale de robot avec un contrôleur flou pour le problème du suivi de trajectoires

2.4.1.2 Conception de la base de règles

Modèle 1 :

Contrôleur flou à 3 variables linguistiques pour chaque entrée et 5 variables linguistiques pour chaque sortie

Tableau 3-2 : Base de règles du contrôleur flou modèle 1

		β		
		N	Z	P
<i>Dis</i>	Z	z/z	z/z	z/z
	N	N/P	P/P	P/N
	F	N/P	P/P	P/N

Les règles présentées : Type1

1. If (Dis is Z) and (B is N) then (VR is Z)(VL is Z)
2. If (Dis is Z) and (B is Z) then (VR is Z)(VL is Z)
3. If (Dis is Z) and (B is P) then (VR is Z)(VL is Z)
4. If (Dis is N) and (B is N) then (VR is N)(VL is P)
5. If (Dis is N) and (B is Z) then (VR is P)(VL is P)
6. If (Dis is N) and (B is P) then (VR is P)(VL is N)
7. If (Dis is F) and (B is N) then (VR is N)(VL is P)
8. If (Dis is F) and (B is Z) then (VR is P)(VL is P)
9. If (Dis is F) and (B is P) then (VR is P)(VL is N)

Modèle 2 :

Contrôleur flou à 3 variables linguistiques pour chaque entrée et 3 variables linguistiques pour chaque sortie

Tableau 3-3 : Base de règles du contrôleur flou modèle 2

		β		
		N	Z	P
<i>Dis</i>	Z	<i>Z/Z</i>		
	N	N/P	P/P	P/N
	F			

Les règles peuvent être écrites comme suit: Type2

1. If (Dis is Z) then (VR is Z)(VL is Z)
2. If (Dis is not Z) and (B is N) then (VR is N)(VL is P)
3. If (Dis is not Z) and (B is Z) then (VR is P)(VL is P)
4. If (Dis is not Z) and (B is P) then (VR is P)(VL is N)

Modèle 3 :

Contrôleur flou à 3 variables linguistiques pour chaque entrée et 3 variables linguistiques pour chaque sortie

Tableau 3-4 : Base de règles du contrôleur flou modèle 3

		β		
		N	Z	P
<i>Dis</i>	Z	z/z	z/z	z/z
	N	SN/SP	SP/SP	SP/SN
	F	N/SP	P/P	P/SN

les règles présentées: Type 3

1. If (Dis is Z) and (B is N) then (VR is Z)(VL is Z)
2. If (Dis is Z) and (B is Z) then (VR is Z)(VL is Z)
3. If (Dis is Z) and (B is P) then (VR is Z)(VL is Z)
4. If (Dis is N) and (B is N) then (VR is SN)(VL is SP)
5. If (Dis is N) and (B is Z) then (VR is SP)(VL is SP)
6. If (Dis is N) and (B is P) then (VR is SP)(VL is SN)
7. If (Dis is F) and (B is N) then (VR is N)(VL is SP)
8. If (Dis is F) and (B is Z) then (VR is P)(VL is P)
9. If (Dis is F) and (B is P) then (VR is P)(VL is SN)

2.4.1.3 La procédure de défuzzification

La méthode utilisée pour la défuzzification est la méthode "bissectrice", du nom de la ligne verticale qui divise la région en deux sous-régions d'aire égale.

2.5 Résultats de simulation

Matlab Simulink a été utilisé pour créer les résultats de simulation. Trajectoire circulaire, trajectoire en L'infini.

Dans le premier cas, le robot doit suivre une trajectoire circulaire qui a pour origine (0,0) et un rayon de 0,01 m. Le robot est initialement situé à l'origine (0,0) avec l'orientation $\varphi = 0$ rad.

D'après la figure 14, tous les contrôleurs fuzzy ont suivi avec succès la trajectoire circulaire. Les erreurs de relèvement et de distance sont illustrées aux figure. 15 et 16, respectivement.

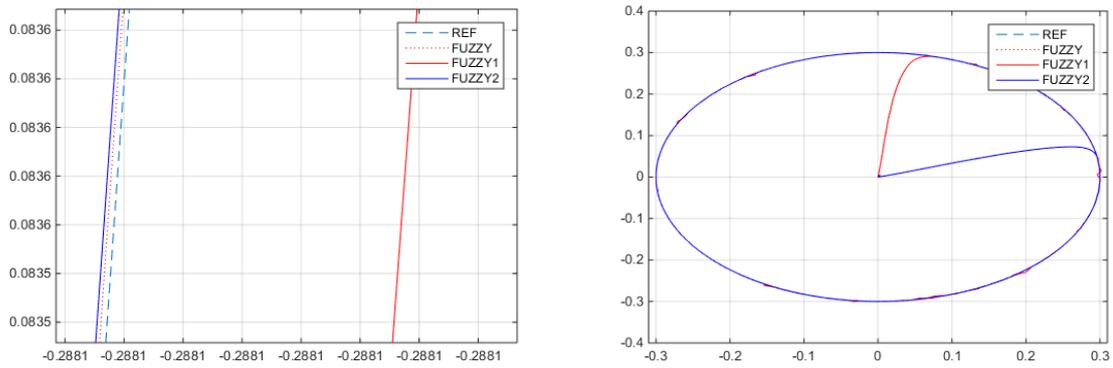


Figure 3-14 : Suivi de trajectoire circulaire

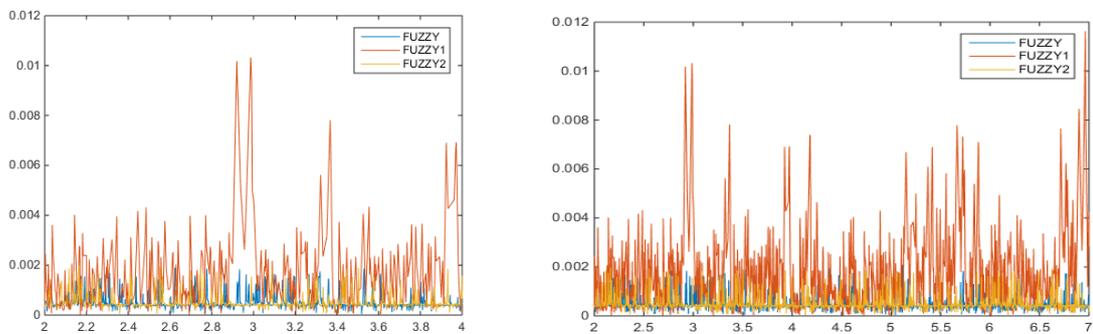


Figure 3-15 : Erreur de distance du suivi de trajectoire circulaire

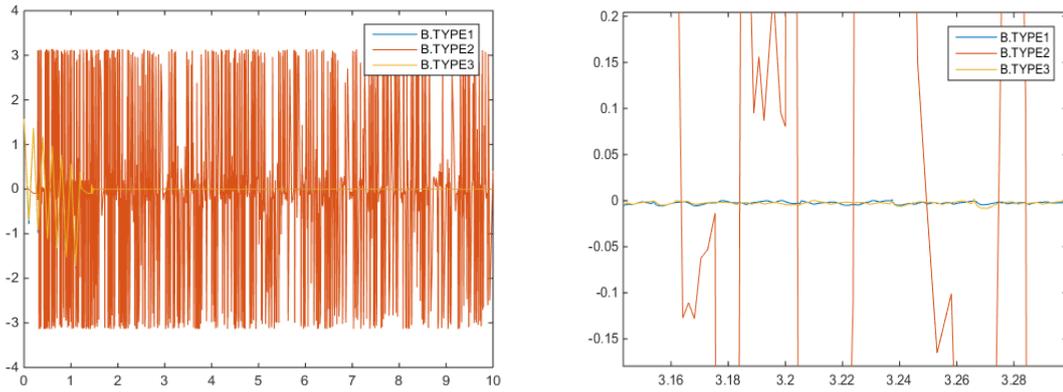


Figure 3-16 : L'erreur de relèvement du suivi de trajectoire circulaire

On peut voir sur la figure 3-15 que l'erreur de distance est inférieure à 2 cm pour tous les contrôleurs. Le contrôleur des règles ordinaires a meilleur résultat, suivi du contrôleur des règles complètes. Il en va de même pour l'erreur de relèvement (figure 3-16).

La deuxième trajectoire est la trajectoire en forme de L'infini, où le robot doit suivre une trajectoire en forme de L'infini (figure 3-17), en partant de l'origine (0,0). Tous les contrôleurs peuvent suivre avec succès la trajectoire en forme de L'infini.

Une comparaison entre les différents contrôleurs en termes d'erreur de distance est donnée sur la figure 3-18. Les courbes des différents contrôleurs sont proches les unes des autres et l'erreur est d'environ 2 cm.

L'erreur sur l'angle de relèvement est illustrée ci-dessous dans la figure 3-19.

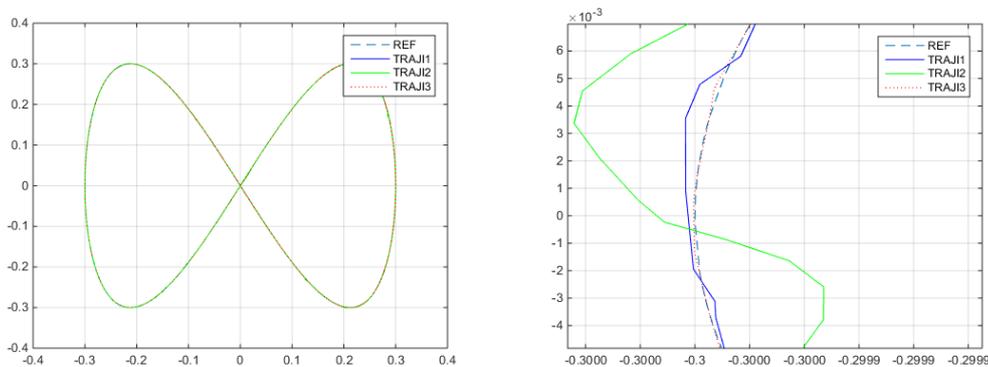


Figure 3-17 : Suivi de trajectoire infini

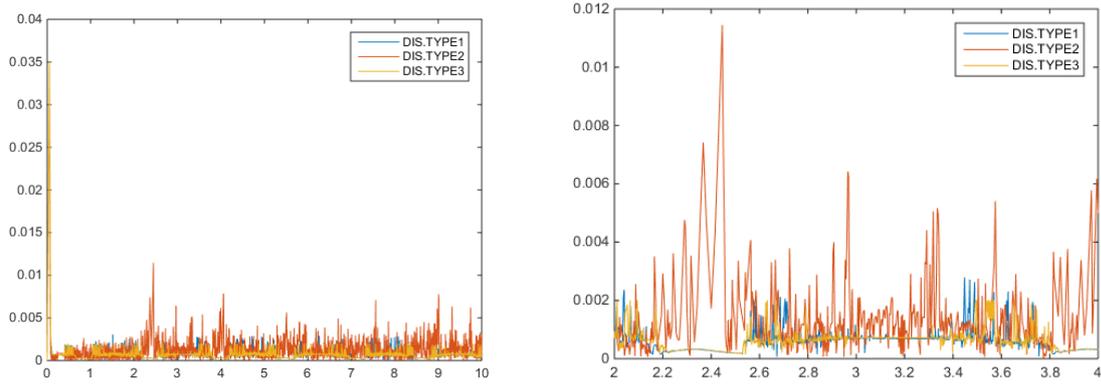


Figure 3-18 Erreur de distance du suivi de trajectoire infini

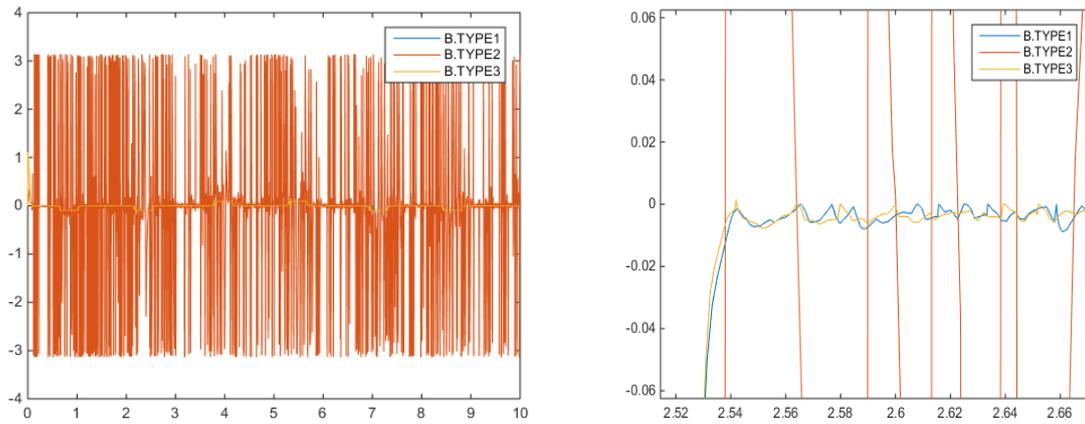


Figure 3-19 : L'erreur de relèvement du suivi de trajectoire infini

III.4.2.4 Comparaison fuzzy logique(type 3) avec contrôleur PID :

D'après la figure 3-20, le contrôleur flou du fuzzy (type 3) suivi avec bonne résultat la trajectoire circulaire par rapport au contrôleur PID, et les erreurs de relèvement de distance (DIS) et l'angle β sont très faibles, et illustrées aux figures.3-21.

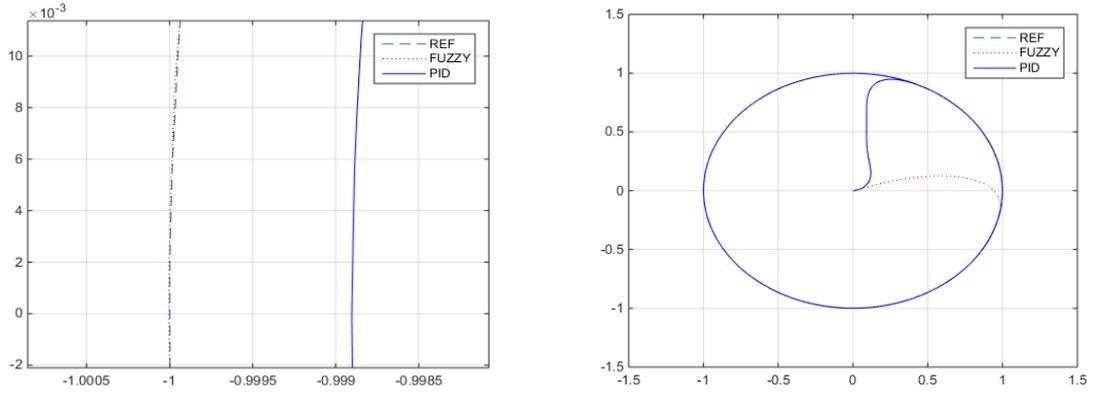


Figure 3-20 : Suivi de trajectoire circulaire

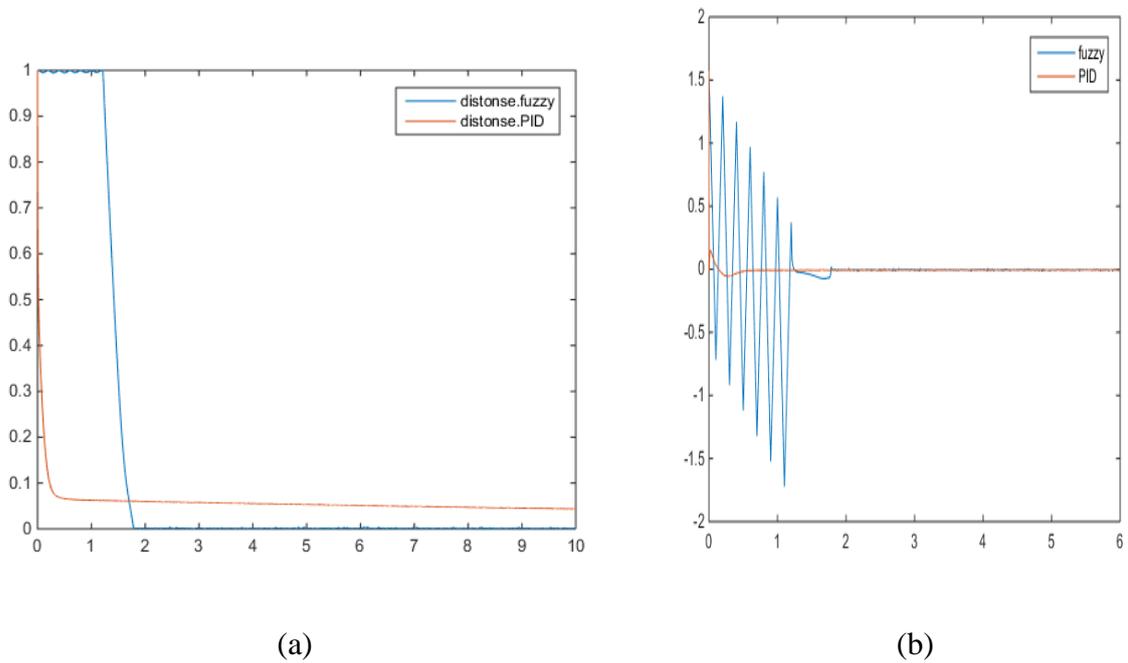


Figure 3-21 : l'erreur sur la : a) distance (Dis), b) orientation(\beta)

2.6 Résultats expérimentaux

Dans cette section, les résultats expérimentaux du contrôle flou du suivi de trajectoire de Qbot sont présentés. Plusieurs expériences ont été réalisées pour évaluer les performances du contrôle flou dans cette tâche particulière. Les variables de performance mesurées comprenaient la précision du suivi de trajectoire, l'erreur entre la trajectoire et la réponse désiré.

Pour évaluer les performances du contrôle flou, plusieurs modèles ont été testés. Dans chaque modèle, le Qbot devait suivre un circuit défini tout en rencontrant des perturbations telles que des changements brusques de direction.

Les résultats obtenus ont montré que la commande floue était capable de maintenir le Qbot sur la trajectoire souhaitée dans la plupart des cas. La précision du suivi de trajectoire était très bonne avec une erreur moyenne inférieure à 10% par rapport à la trajectoire circulaire et l'infini comme la figure suivent :

Modèle 1 :

Contrôleur flou à 3 variables linguistiques pour chaque entrée et 5 variables linguistiques pour chaque sortie

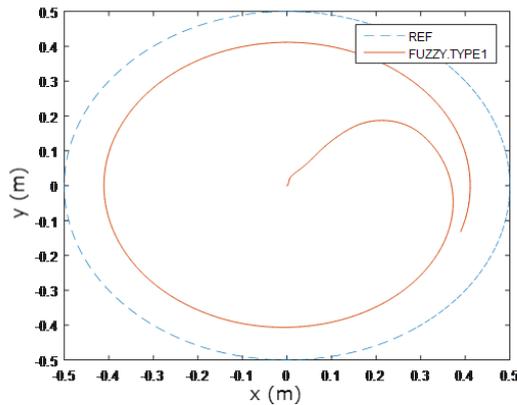
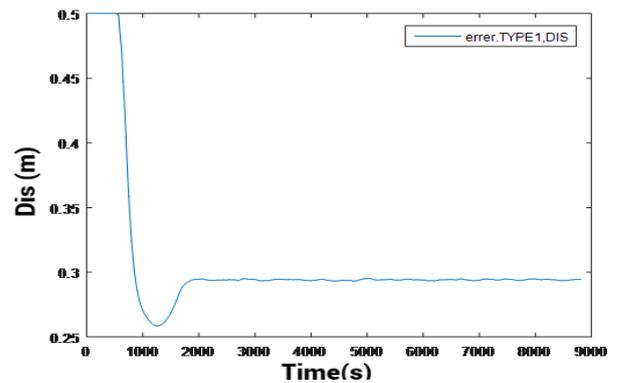
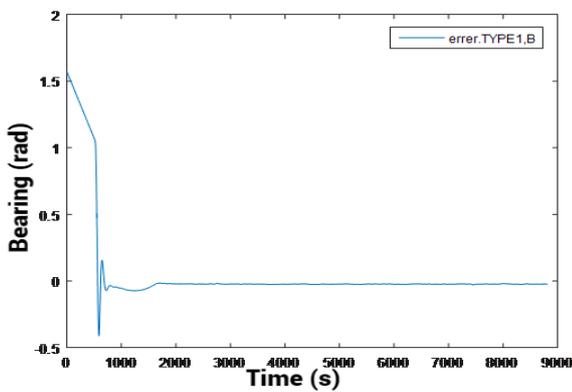


Figure 3-22 : Résultats expérimentales, trajectoire circulaire (Modèle 1)



(a) (b)

Figure 3-23 : (a) erreur de l'angle β . (b) erreur de distance DIS trajectoire circulaire (Modèle 1)

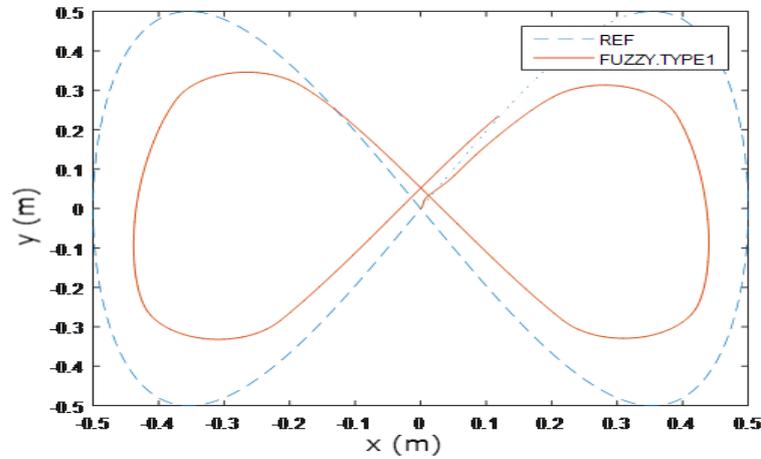


Figure 3-24 : Résultats expérimentales, trajectoire ∞ (Modèle 1)

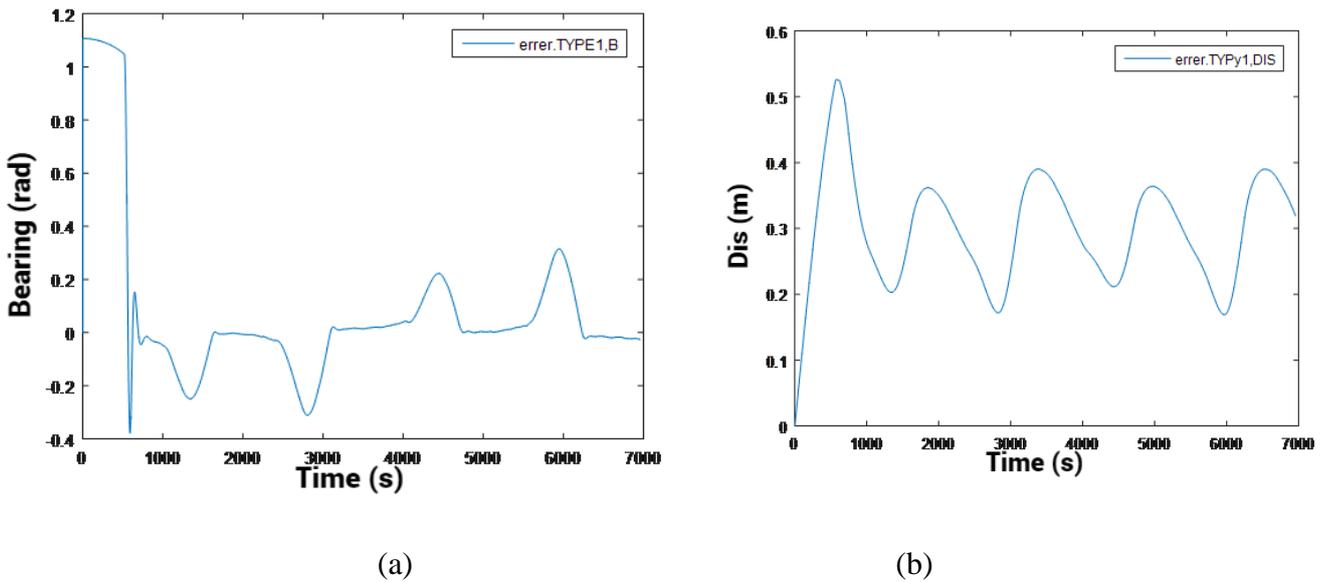


Figure 3-25 : (a) erreur de l'angle β . (b) erreur de distance DIS trajectoire ∞ (Modèle 1)

Modèle 2 :

Contrôleur flou à 3 variables linguistiques pour chaque entrée et 3 variables linguistiques pour chaque sortie

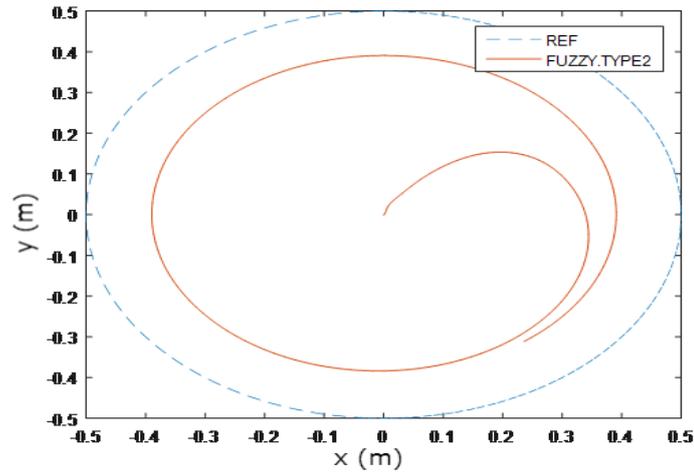


Figure 3-26 : Résultats expérimentales, trajectoire circulaire (Modèle 2)

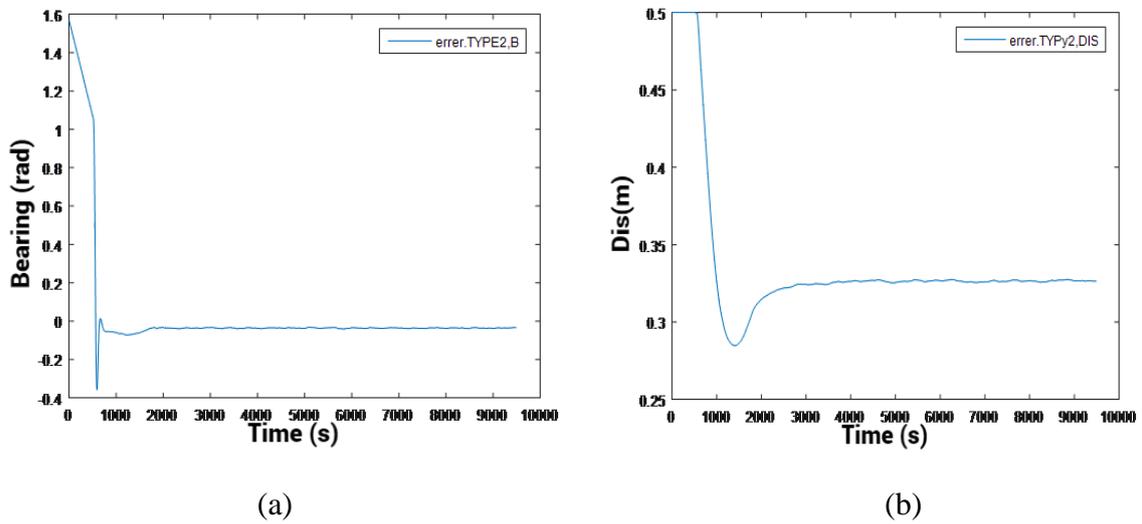


Figure 3-27 : (a) erreur de l'angle β . (b) erreur de distance DIS trajectoire ∞ (Modèle 2)

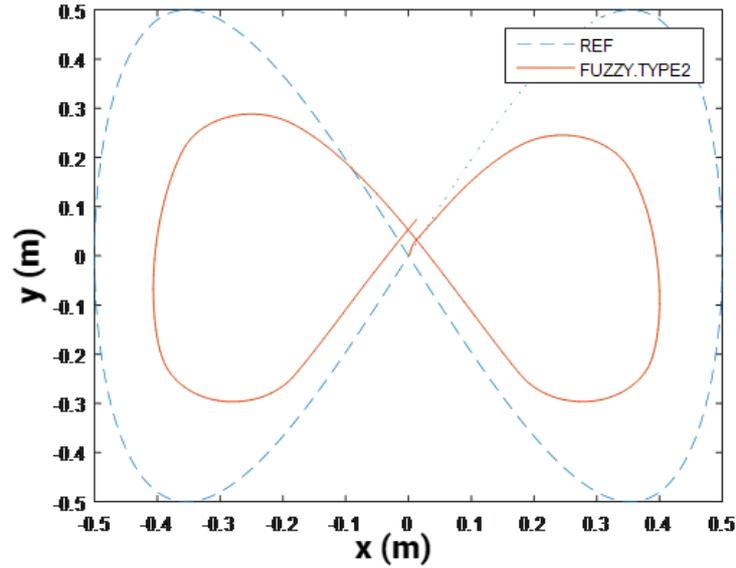


Figure 3-28 : Résultats expérimentales, trajectoire ∞ (Modèle 2)

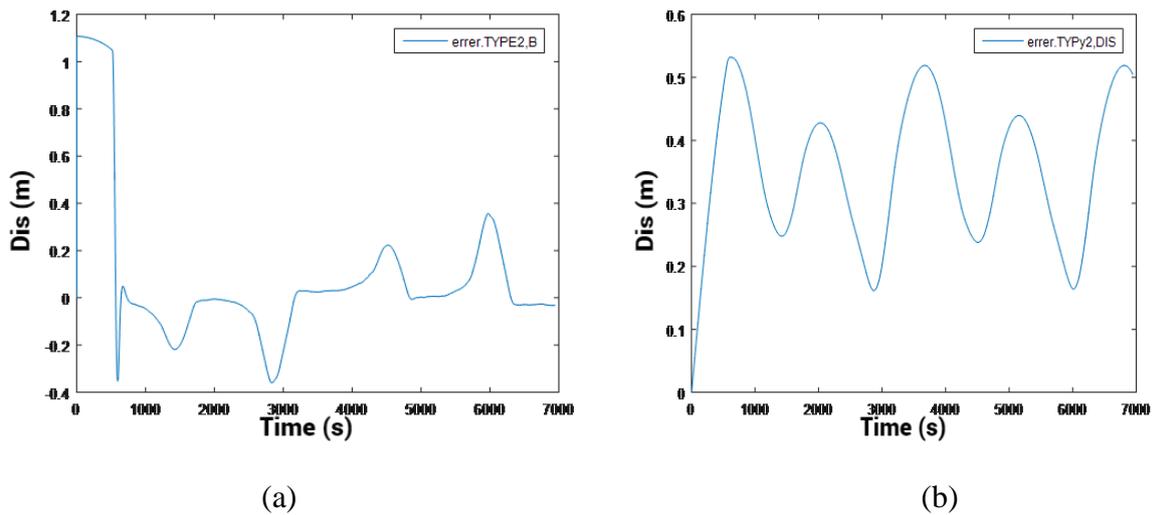


Figure 3-29 : (a) erreur de l'angle β . (b) erreur de distance DIS trajectoire ∞ (Modèle 2)

2.6.1 Comparaison analytique

Pour évaluer et comparer les performances des techniques proposées, deux mesures sont utilisées : l'erreur moyenne de direction (Err_{β}) et l'erreur moyenne de distance Err_{Dis} .

$$\text{erreur moyenne} = \frac{\text{somme des valeur}}{\text{nombre de valeur}}$$

On effectue une comparaison entre les résultats du contrôleur flou de Modèle 1 et du contrôleur flou de Modèle 2, que celle-ci soit circulaire ou infinie (∞). Les résultats de cette comparaison sont présentés dans le tableau 3-3.

Tableau 3-5 : Comparaison analytique entre les contrôleurs flou proposés

	Trajectoire circulaire (0)		Trajectoire l'infini (∞)	
	Contrôleur flou (Modèle 1)	Contrôleur flou (Modèle 2)	Contrôleur flou (Modèle 1)	Contrôleur flou (Modèle 2)
Err_{β}	0,75	0,42	0,61	0,40
Err_{Dis}	0,52	0,3	0,47	0,28

2.7 Conclusion

Dans ce chapitre et après l'expérience que nous avons fait, pour le contrôle et suivi de trajectoire et avec différents modèles de commande floue, Les résultats obtenus ont montré que la commande floue était capable de maintenir le Qbot sur la trajectoire souhaitée dans la plupart des cas. La précision du suivi de trajectoire était très bonne avec une erreur moyenne inférieure à 10% par rapport à la trajectoire circulaire et l'infini, Et l'erreur de distance est inférieure à 2 cm pour tous les contrôleurs.

Le contrôleur des règles ordinaires a meilleur résultat, et le contrôleur fuzzy a suivi avec succès la trajectoire circulaire.

La deuxième trajectoire est la trajectoire en forme de L'infini, aussi le contrôleur a suivi avec succès la trajectoire en forme de L'infini.

Et par comparaison entre les différents contrôleurs en termes d'erreur de distance on donne Les courbes de différents contrôleurs est proches les unes par rapport à l'autres et l'erreur est d'environ 2cm.

D'après le contrôleur flou du fuzzy (type 3) il assure le suivi avec bonne résultat la trajectoire circulaire par rapport au contrôleur PID, et les erreurs de relèvement de distance (DIS) et l'angle sont très faibles.

Enfin, la commande floue est une approche essentielle dans le domaine de la robotique mobile, en particulier pour le suivi de trajectoire du Qbot. Les résultats expérimentaux ont démontré l'efficacité de la commande floue en termes de précision, de stabilité et de réactivité.

Conclusion générale

En conclusion, Dans ce travail le problème de contrôler et de suivi la trajectoire d'un robot mobile a été abordé en utilisant les techniques de la commande floue, et à travers cette étude que nous avons menée, et selon les résultats que nous avons obtenus, nous pouvons conclure ce que : La tâche principale de robot mobile QUANSER doit faire est de suivre la trajectoire souhaitée, à partir d'une configuration donnée, et avec minimum d'erreur, et sans intervention de l'être humain.

Dans un premier temps, nous avons présenté quelques concepts de la logique floue et ces différentes approches de commande suivent leur problème, et donné un aperçu sur les robots mobiles et sont technologies clés dans de nombreux domaines, et quelques exemples d'applications de ces technologies.

Après un aperçu global du robot mobile, et description détaillée des composants QUANSER QBOT2, et leur modélisation complète, la cinématique et la dynamique du robot en détail. Ce type de robot est largement utilisé, notamment dans le domaine de la recherche en raison de sa simplicité et de sa facilité de travail.

Le problème de contrôle et suivi la trajectoire a été considéré. Le robot doit suivre une trajectoire souhaitée. Le premier choix est d'utiliser un fuzzy logique de différents types de règle d'inférence pour contrôle, et un régulateur PID par sa simplicité, et pour nous faire la comparaison entre eux, Alors, la meilleure est le contrôle flou. La logique floue est un raisonnement approximatif qui peut faire face à l'incertitude de l'information. Il peut produire des contrôleurs robustes à l'incertitude. Le contrôleur flou a été conçu à cet effet. Les résultats de simulation montrent l'efficacité de l'approche proposée basée sur l'utilisation de Cinq ou trois nombres de règles.

Enfin, la commande floue présente un potentiel considérable pour le suivi de trajectoire du robot Quanser Qbot. et elle offre une approche flexible et robuste qui permet de prendre en compte les incertitudes et les imprécisions inhérentes à la navigation autonome des robots. Travaux futurs et perspectives Après l'expérience que nous avons faite, qui consiste à contrôler la trajectoire de QBOT2, nous concluons que nous pouvons utiliser le contrôleur flou, et nous obtiendrons de très bons résultats, et il peut fournir une base pour les orientations future.

Travaux futurs et perspectives

Conclusion générale

Après l'expérience que nous avons faite, qui consiste à contrôler la trajectoire de QBOT2, nous concluons que nous pouvons utiliser le contrôleur flou, et nous obtiendrons de très bons résultats, et il peut fournir une base pour les orientations future.

Bibliographie

- [1] G. Antonelli, S. Chiaverini et G. Fusco, « Une approche basée sur la logique floue pour le suivi du chemin des robots mobiles », IEEE Transactions on Fuzzy Systems, vol. 15, ch. 2, pages 211-221, 2007
- [2] D. R. Parhi, « Navigation de robots mobiles à l'aide d'un contrôleur à logique floue », Journal of Intelligent and Robotic Systems, vol. 42, p.253-273, 2005
- [3] K. P. Mohanty et R. P. Dayal « Navigation d'un robot mobile autonome utilisant un système d'inférence floue basé sur un réseau adaptatif », Journal of Mechanical Science and Technology, vol. 28, non. 7, pp. 2861-2868, juillet 2014.
- [4] Q. Feng, Q. Zhu, A. F. T. Winfield et C. Melhuish, « Contrôle adaptatif du mode glissant pour les systèmes non linéaires MIMO basé sur un schéma de logique floue », International Journal of Automation and Computing, vol. 1, pages 51-62, 2004.
- [5] N. Paykari, S. H. Abbasi et F. Shabaninia « Conception de contrôleurs à logique floue MIMO mamdani pour le robot mobile suivant le mur », Advances in Intelligent Systems and Computing, vol. 195, p. 155-164, 2013.
- [6] S. Tong, B. Chen et Y. Wang, « Contrôle de rétroaction de sortie adaptatif flou pour les systèmes non linéaires MIMO », Fuzzy Sets and Systems, vol. 156, non. 2, pages 285-299, 2005.
- [7] www.quanser.com
- [8] Le [mot-valise](#) « Kinect » est issu des mots anglais « *kinetic* » (qu'on peut traduire par « cinétique ») et « *connect* » (qu'on peut traduire par « connecter »). La sortie européenne a eu lieu le 10 novembre 2010.
- [9] Hugues Garnier Mayank Jha January 2023
- [10] ©2014 Quanser Inc. All rights reserved.
- [11] Stéphane Lens. Locomotion d'un robot mobile. 3ELIN. Université de Liège, 2008.
- [12] Figure I.1: <https://www.cite-telecoms.com/>
- [13] C.benson. What Types of Mobile Robots are there?
<https://www.Robotshop.com/.17/09/2018>

- [14] Paul Heney. Robots are software and software is “eating the world”.
<https://www.therobotreport.com/> , December 29, 2017
- [15] Shah, N. (2018). Modeling and Control of the Quanser Qbot 2e: A Mobile Robot Platform. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 7(1), 194-201.
- [16] Balakrishnan, V., & Balasubramanian, A. (2019). Dynamic modeling and control of Quanser Qbot 2e mobile robot. *Journal of Control, Automation and Electrical Systems*, 30(6), 787-798.
- [17] Li, W., Li, H., & Wu, X. (2019). Nonlinear control for Quanser Qbot 2e mobile robot based on extended state observer. *Journal of Intelligent & Robotic Systems*, 96(3-4), 671-682.
- [18] Gholami, M., Miao, Y., Naderan-Tahan, K., & Arbabtafti, M. (2020). Control and navigation of Quanser QBot 2e: A survey of recent advances. *Journal of Intelligent & Robotic Systems*, 98(1), 179-207
- [19] Hwangbo, J., Lee, D., & Kim, J. (2021). Dynamic Modeling and Control of Quanser Qbot 2e Based on Machine Learning for Autonomous Navigation. *Electronics*, 10(2), 158.
- [20] Singh, R., Singh, G., Kumar, V.: Control of closed-loop differential drive mobile robot using forward and reverse Kinematics. In: 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, pp. 430–433 (2020).
- [21] Okuyama, I.F., Maximo, M.R.O.A., Afonso, R.J.M.: Minimum time trajectory planning for a differential drive mobile robot considering non-slipping constraints. *J. Control Automat. Electric. Syst.* 32, 120–131 (2021). <https://doi.org/10.1007/s40313-020-00657-x>
- [22] Martinez-Melchor, J.A., Jimenez-Fernandez, V.M., Vazquez-Lea, H., Filobello-Nino, U.A.: Optimization of collision-free paths in a differential-drive robot by a smoothing piecewise-linear approach. *Compt. Appl. Math.* 37, 4944–4965 (2018)
- [23] Hirpo, B.D., Zhongmin, W.: Design and control for differential drive mobile robot. *Int. J. Eng. Res. Technol (IJERT)* 6(10), 327–334 (2017)
- [24] Khai, T.Q., Ryoo, Y.J., Gill, W.R., Im, D.Y.: Design of kinematic controller based on parameter tuning by fuzzy inference system for trajectory tracking of differential-drive mobile robot. *Int. J. Fuzzy Syst.* 22, 1972–1978 (2020)

- [25] Bouzoualegh, S., Guechi, E., Kelaiaia, R.: Model predictive control of a differential-drive mobile robot. *Acta Universitatis Sapientiae* 10, 20–41 (2018)
- [26] Demirbas, F., Kalyoncu, M.: Differential drive mobile robot trajectory tracking with using pid and kinematic based backstepping controller. *Selcuk Univ. J. Eng. Sci. Technol.* 5(1), 1–15 (2017).
- [27] Macias, V., Becerra, I., Martinez, E., Murrieta-Cid, R., Becerra, H.M.: Single landmark feedback-based time optimal navigation for a differential drive robot. *J. Franklin Inst.* 358(9), 4761–4792 (2021).
- [28] Cristofaro, A., Salaris, P., Pallottino, L., Giannoni, F., Bicchi, A.: On the minimum-time control problem for differential drive robots with bearing constraints. *J Optim Theory Appl.* 173, 967–993 (2017)
- [29] Martins, F.N., Sarcinelli-Filho, M., Carelli, R.: A velocity-based dynamic model and its properties for differential drive mobile robots. *J. Intell. Robot. Syst.* 85, 277–292 (2017). [https://doi.org/ 10.1007/s10846-016-0381-9](https://doi.org/10.1007/s10846-016-0381-9)
- [30] Lopez-Padilla, R., Murrieta-Cid, R., Becerra, I., Laguna, G., LaValle, S.M.: Optimal navigation for a differential drive disc robot: a game against the polygonal environment. *J. Intell. Rob. Syst.* 89, 211–250 (2018)
- [30] 39. Kosheleva, O., Kreinovich, V., Nguyen, T.N.: Why triangular membership functions are successfully used in f-transform applications: a global explanation to supplement the existing local ones. *Axioms* 8(3), 95 (2019). <https://doi.org/10.3390/axioms8030095>
- [31] Pedrycz, W.: Why triangular membership functions? *Fuzzy Sets Syst* 64, 21–30 (1994). [https://doi.org/10.1016/0165-0114\(94\)90003-5](https://doi.org/10.1016/0165-0114(94)90003-5)