REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
--T L E M C E N--

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique
Spécialité : Automatique

Présenté par :
BOUROUBA Meriem
BOUZIANI Wassen Ikhlasse

Thème

Intégration de l'Apprentissage Automatique pour la Navigation d'un Robot Mobile

Soutenu publiquement, le 03 / 10 / 2024 , devant le jury composé de :

M. CHIALI Anis	MCA	ESSA. Tlemcen	Président
M/.BOUKLI HACENE Fazil	MAB	ESSA. Tlemcen	Directeur de mémoire
Mme. NEDJAR Imane	MCA	ESSA. Tlemcen	Examinateur 1
M. MOKHTARI Reda	MCA	ESSA. Tlemcen	Examinateur 2
M. DJAMAL Abdounassir	Invite	Naftal Tlemcen	Partenaire socio-economique

Année universitaire : 2023/2024

REMERCIEMENT

Au nom de Dieu,le Tout-Puissant, sans qui rien n'aurait été possible. Nous tenons à exprimer notre profonde gratitude pour la force, la patience et la sagesse qu'Il nous a accordées tout au long de ce travail. C'est par Sa grâce que nous avons pu mener à bien ce projet.

Nous exprimons notre profonde gratitude à M. Boukli Hacene Fazil Lotfi pour son expertise, sa disponibilité et son approche pédagogique. Il nous guidant avec rigueur tout en nous offrant la liberté d'explorer nos projets académiques et personnels. Son engagement tout au long du projet a été essentiel pour orienter notre recherche.

Nous souhaitons exprimer notre reconnaissance à toutes les personnes qui nous ont aidés, directement ou indirectement. Un remerciement particulier à M. Ramz Eddine et M. Abdelatif pour leur soutien constant et leur aide précieuse tout au long de ce projet.

Nous souhaitons adresser nos sincères remerciements à l'ensemble de nos collègues, qui nous ont accompagnés tout au long de notre parcours éducatif. Un remerciement tout particulier à notre collègue et ami, l'ingénieur Korrichi Mohammed Yasser, notre collègue, major de promo et ingénieur BEMMARZOUK Lynda pour leur disponibilité et ses précieux conseils. leurs sens de la collaboration et leur professionnalisme ont grandement contribué à la réussite de ce travail.

Aux membres du jury, pour l'intérêt qu'ils ont porté à notre projet en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Enfin, Nous exprimons notre plus profonde gratitude à notre famille, dont le soutien constant a été un pilier tout au long de ce parcours. Nous tenons à remercier particulièrement BELARBI Nadjia, maman de BOUZIANI Wassen, pour sa présence bienveillante et ses encouragements constants, ainsi que BOUROUBA Mohammed et BENAINI Halima, parents de BOUROUBA Meriem, dont la générosité, la compréhension et le soutien indéfectible nous ont permis de mener à bien ce projet. Leur confiance en nous a été une source inestimable de motivation, et nous leur en sommes infiniment reconnaissants.

Dedicace

First and foremost, I dedicate this thesis to God, the Almighty, whose guidance, wisdom, and strength have been with me throughout this journey. I am deeply grateful for the blessings, courage, and resilience He has bestowed upon me.

To the greatest mother in the universe, to the woman who i got my confidence from, who teach me how to love and how live.To my beloved mother BELARBI Nadjia, Your endless sacrifices, prayers, and unwavering belief in me have been the light guiding me through every challenge. You are the foundation upon which I have built my dreams, and I am forever indebted to you. This accomplishment is a reflection of your strength and the values you have instilled in me.

To my dear sisters,the family's doctor chifaa, little angles Sara, Intissar, Racha my confidantes, and my closest allies, who have been my constant companions through the many phases of life. You have been there for me with open arms and open hearts, sharing in my joys and comforting me in my sorrows. You are my greatest cheerleaders, lifting me up with words of encouragement, and reminding me of my potential when I needed it most. I am grateful beyond words for your presence in my life, and I cherish the love and support you have always given me so freely.I dicade this work also to my first baby and my smart cousin Assil and my grandmother Saliha . I am blessed to have each of you in my life.

To my loving husband EL HADJ MIMOUNE Hamza, whose patience, understanding, and encouragement have been invaluable throughout this journey. Your love and support have been my greatest comfort, and I am truly blessed to have you by my side.

To my dearest friends, Souhila, Sara, Chams, Hind are like the family I have chosen. You have stood by me through thick and thin, offering not only your time and wisdom but also your hearts. Your friendship has been a source of immense joy and a shelter in times of storm. Thank you for being my strength when I was weak, my laughter in times of sorrow, and my courage when I doubted myself. I am blessed to have each of you in my life.

Wassen Ikhlasse BOUZIANI

Dedicace

À ceux qui ont fait de moi ce que je suis aujourd'hui..À ma mère, **BENAINI Halima**, source inépuisable de tendresse et de dévouement. Tes sacrifices, ta patience et ton amour ont guidé chaque étape de mon parcours. Grâce à toi, j'ai appris à rêver grand et à persévérer, même dans les moments difficiles. Ce travail est le reflet de ton soutien silencieux et de tes encouragements constants. Merci d'avoir été ma force et ma plus grande motivation.

À mon père, **BOUROUBA Mohammed**, exemple de persévérance et d'honnêteté. Par ton travail et tes conseils, tu m'as appris la valeur de la détermination et de l'exigence. Ta confiance en moi m'a toujours poussé à donner le meilleur. Ce travail est aussi le reflet de tes efforts et de ton soutien indéfectible. Merci pour tout.

À ma famille, mon pilier et ma force. À mes frères Radouane, Yassine et Zakaria, merci pour votre affection, vos encouragements et votre soutien constant. Votre présence, même discrète, m'a donné l'énergie et la motivation nécessaires pour avancer, surtout dans les moments de doute. Merci de toujours croire en moi.

À ma deuxième famille, mes parents Hafida, Zineddine, mes soeurs et frère Mimouna Aya, Nihad, et Abdelkader, merci pour votre accueil, votre soutien constant et votre présence réconfortante. Vous avez été des piliers dans les moments de doute et de fatigue, partageant mes joies et épreuves avec une générosité sans faille. Ce travail vous est dédié, car sans vous, ce chemin aurait été bien plus difficile.

Je dédie ce travail à tous ceux qui ont contribué, de près ou de loin, à mon évolution personnelle et académique. Un remerciement particulier à **AbdelKarim ZIANI KERARTI** pour son soutien inestimable, sa présence bienveillante, et ses précieux conseils tout au long de ce parcours , son encouragement constant m'a donné la force de continuer à travailler et d'atteindre l'efficacité et la qualité requises pour cette réalisation,je te remercie énormément. À mes amis pour leur soutien, à mes enseignants et mentors pour leur savoir et leur passion, ainsi qu'à tous ceux qui m'ont accompagné discrètement. Votre présence, vos encouragements et votre générosité ont été essentiels à cette réussite. Je vous en suis profondément reconnaissant.

Meriem BOUROUBA

Résumé

Dans un contexte où la robotique mobile autonome devient de plus en plus cruciale, ce mémoire explore la navigation des robots mobiles en se concentrant sur l'architecture structurelle et fonctionnelle des robots, ainsi que sur la modélisation géométrique, cinématique et dynamique. L'objectif est de permettre une navigation précise et fiable dans un environnement virtuel, en utilisant le suivi des panneaux de signalisation basé sur des méthodes de soft computing.

Mots-clés

Robotique mobile, Navigation, Modélisation du robot mobile,Suivi des panneaux de signalisation

Abstract

In a context where autonomous mobile robotics is becoming increasingly crucial, this thesis explores the navigation of mobile robots, focusing on the structural and functional architecture of robots, as well as the geometric, kinematic, and dynamic modeling. The goal is to enable precise and reliable navigation in a virtual environment using traffic sign tracking based on soft computing methods.

Keywords

Mobile robotics, Navigation, Modeling the mobile robot, Traffic sign tracking.

ملخص

في سياق أصبح فيه علم الروبوتات المتنقلة الذاتية أكثر أهمية، يستكشف هذا البحث الملاحة للروبوتات المتنقلة، مع التركيز على الهيكلية الوظيفية والمعمارية للروبوتات، بالإضافة إلى النمذجة الهندسية، والحركية، والديناميكية. الهدف هو تمكين الملاحة الدقيقة والموثوقة في بيئة افتراضية باستخدام تتبع إشارات المرور بناءً على طرق الحوسبة اللينة.

الكلمات المفتاحية

الروبوتات المتنقلة، التنقل، نمذجة الروبوتات المتنقلة، تتبع إشارات المرور

Table des matières

Table des figures

Liste des tableaux

Abréviations

— BTP :Bâtiment et travaux publics
— FPGA : Field-Programmable Gate Array
— RNA : réseau neurone artificielle
— IA : Intelligence artificielle
— LTS : Long-Term Support
— RL : renforcement Learning
— ML : Machine Learning
— ROS : robot operating system
— SLAM : simultaneous localization and mapping
— DDS : Data Distribution Service

Introduction Générale

Les systèmes de robots mobiles autonomes représentent un domaine clé de la robotique moderne, en particulier dans le contexte de la navigation et de l'interaction avec des environnements complexes et dynamiques. Ce projet vise à développer un robot mobile capable de détecter et de réagir à des panneaux de signalisation routiers, un défi crucial pour de nombreuses applications, telles que les véhicules autonomes et les robots de surveillance. Notre projet s'appuie sur plusieurs technologies avancées, notamment le système d'exploitation ROS2 (Robot Operating System 2), le simulateur 3D Gazebo, et l'algorithme de détection d'objets YOLOv8n.

Dans ce cadre, l'utilisation de la simulation offre une plateforme efficace pour tester, affiner et valider les comportements du robot avant sa mise en œuvre dans des scénarios réels. La simulation va permettre de recréer des environnements routiers où le robot doit naviguer et interagir avec divers types de panneaux de signalisation, tels que les arrêts (stop), les limitations de vitesse et les indications de direction. Pour cela nous allons effectuer une détection basée sur l'algorithme YOLOv8n, afin que le robot puisse interpréter les données capturées par une caméra virtuelle et d'ajuster ses actions en modifiant sa trajectoire et en adaptant sa vitesse.

Le système à étudier va intégrer des logiciels dans une architecture flexible et robuste, tel que le ROS2 afin de gérer la communication en temps réel entre les différents modules du robot. Le projet a également besoin d'une modélisation d'environnements et des algorithmes de contrôle afin de mieux piloter le robot.

Chapitre 1

Chapitre 1 : Initiation à la robotique mobile

1.1 Introduction

Un Robot Mobile est un ensemble de systèmes issus du domaine de la mécatronique qui est une technologie qui relie la mécanique, l'électronique, l'automatique et l'informatique. Cet assemblage agit physiquement dans un environnement de telle sorte de réaliser un objectif désiré. Cette Machine est dotée d'une capacité de fonctionnement de perception et de décision et d'action lui permettant d'effectuer de différentes tâches même en rencontrant des situations inattendues

1.2 Historique de la Robotique Mobile

En 1921 l'écrivain tchèque Karel Capek a introduit le mot "Robot" dans sa pièce intitulé (Rossuum's Universal Robots). "Robot" en tchèque vient du mot "robota", qui signifie "travail obligatoire ou forcé".

En 1942 le mot « robotique » est apparu pour la première fois dans un roman sous le nom « Habillage » par le scientifique Américain Isaac Asimov.

Deux sources distinctes ont donné naissance au concept de robot mobile autonome vers la fin des années soixante : tout d'abord des recherches menées au Stanford Research Institute sur les possibilités d'équiper des machines de capacités de déduction et de réaction logique à des événements extérieurs.

Shakey a été construit de cette manière, une machine à roues connectée à un ordinateur et équipée d'une caméra pour capter des images de son environnement.

D'un autre côté, l'industrie nucléaire a besoin de machines capables d'agir à distance dans des environnements encombrés et inaccessibles à l'homme. Tandis que les projets Luna et Mars Rover s'échafaudent dans le but d'explorer des planètes sans que l'homme ne prenne part au voyage, en parallèle, les laboratoires, les industriels, les informaticiens et les mécaniciens vont travailler la dessus pendant plusieurs années.

Par contre, la télé-opération est une partie de la robotique classique qui est accessible dans le domaine industriel, tandis que des progrès voient le jour avec l'arrivée de l'intelligence artificielle qui se base essentiellement dans le domaine de l'informatique. Vers la fin des années soixante-dix, trois pays émergent tel que la France, le Japon et les États-Unis ont dominé l'innovation, la réalisation et la commercialisation des robots mobiles autonomes dans divers domaines domestiques et militaires[2].

1.3 Définition de la Robotique Mobile

La robotique en générale est une branche de la science et de la technologie qui traite la conception, la fabrication, l'exploitation et l'utilisation de Machine capable de

réaliser des tâches. Elle combine plusieurs disciplines telles que : la robotique, l'informatique, l'électronique et la mécanique. Les robots mobiles sont des machines automatiques capables de se déplacer dans l'environnement où ils sont développés afin d'accomplir un travail bien déterminé. Ces derniers peuvent être semi-autonomes, ou ça nécessite une certaine interaction humaine pour fonctionner, ou autonomes, c'est-à-dire qu'ils peuvent prendre leurs propres décisions et agir sans l'intervention humaine.

1.4 Classification des robots mobiles

La classification des robots ce fait selon divers critères, mais principalement selon trois critères : **le degré d'autonomie, le domaine d'application et le système de locomotion.**

1.4.1 Classification selon le degré d'autonomie

Robot télécommandé :

Ce sont des robots commandés par un opérateur (machine ou être humain), qui leur donnent les instructions nécessaires pour effectuer des mouvements basiques tels qu'avancer, reculer, tourner à droite...etc.

Robot semi-autonome :

Ce type de robot effectue diverses tâches de manière complètement autonome, mais il peut être interrompu pour recevoir des commandes de contrôle d'un opérateur.

Robot autonome :

Un robot est autonome s'il est capable de s'adapter à son environnement en ayant la capacité de capter, percevoir, analyser, communiquer, planifier, et prendre des décisions afin d'atteindre les buts qui lui ont été assignés par un opérateur humain à l'aide d'une interface homme-machine dédiée.

1.4.2 Classification selon le domaine d'application

Les robots industriels et de service :

Les robots mobiles sont utilisés dans des applications industrielles telles que le transport, la distribution, le nettoyage, la maintenance, la surveillance et l'entretien. Ils sont conçus pour soutenir les personnes handicapées, aider les personnes âgées et conduire des véhicules automatiques.

Les robots militaires :

Les robots mobiles ont de nombreuses applications militaires, offrant des spécifications strictes telles que la vitesse du véhicule, les capacités de franchissement d'obstacles et la rapidité de réaction.

Les robots de laboratoire :

De nombreux laboratoires de robotique travaillent pour valider des travaux théoriques sur la perception ou la planification de mouvement. Au départ, tous les robots ont été créés dans un laboratoire de recherches pour des différents domaines à étudier.

1.4.3 Classification selon locomotion

Les robots mobiles à roues :

La structure mécanique la plus fréquemment utilisée est la mobilité par roues. Ce type de robot facilite le déplacement, mais il nécessite un sol plutôt plat. Les robots à roues sont classés en fonction de la position et du nombre de roues utilisées. Nous citerons ici les quatre principales classes de robots à roues :

- **Robots à unicycle :** à deux roues fixes.
- **Robots tricycle :** avec deux roues fixes et une roue orientable centrée.
- **Robots de voiture :** avec deux roues fixes et deux roues orientables centrées.
- **Robot mobile omnidirectionnel :** est dit omnidirectionnel si l'on peut agir indépendamment sur les vitesses tel que la vitesse de translation selon les axes x et y et la vitesse de rotation autour de z

Les robots mobiles à chenilles

L'avantage des robots mobiles à chenilles est leur bonne adhérence au sol et leur capacité à surmonter les obstacles. L'utilisation est axée sur l'utilisation sur un sol accidenté ou de mauvaise qualité au niveau de l'adhérence (boue, herbe...). Ils sont généralement utilisés pour des fins militaires, principalement comme des robots de surveillance ou de démineurs.

Les robots mobiles marcheurs

Les robots marcheurs mobiles sont destinés à réaliser diverses tâches dont l'accès au site est difficile, dangereux ou impossible à l'homme, leur anatomie à plusieurs degrés de liberté les rapproche des robots manipulateurs. La conduite de la locomotion est basée sur les coordonnées articulaires et le concept d'allure qui garantit le déplacement stable de l'ensemble est défini par des méthodes de commande des articulations.

Les robots mobiles rampants

Les robots mobiles rampants utilisent la reptation comme solution de locomotion pour les environnements de tunnels, ce qui permet la création de structures filiformes. Le système se compose de modules à mobilités multiples, dérivés des méthodes de locomotion des serpents. Le type scolopendre est une structure inextensible, tandis que le type lombaire comprend trois articulations, rotations orthogonales et traduction, le type péristaltique implique le mouvement de module à vaisseau[11].

1.5 Application

Le tableau ci-après résume de manière non exhaustive les diverses applications des robots mobiles[11].

Industrie nucléaire	surveillance de sites manipulation de matériaux radio-actifs démantèlement de centrales
Sécurité civile	neutralisation d'activité terroriste déminage pose d'explosif surveillance de munitions
Militaire	surveillance, patrouille pose d'explosifs manipulation de munitions
Chimique	surveillance de site manipulation de matériaux toxiques
Médecine	assistance d'urgence aide aux handicapés physiques, aux aveugles
Lutte contre l'incendie	localisation d'une source d'incendie détection de fumée suppression de flammes
Sous-marine	pose de câbles recherche de nodules recherche de navires immergés inspection des fonds marins
Agricole	cueillette de fruits traite, moisson, traitement des vignes...
Construction BTP	projection mortier lissage du béton
Nettoyage	coque de navire nettoyage industriel
Espace	exploration
Industriel	convoyage surveillance

TABLE 1.1 – Le rôle des robots dans divers domaines

1.6 Conclusion

En tant que domaine multidisciplinaire, la robotique offre un large éventail de possibilités fascinantes pour la conception, la fabrication et l'utilisation de robots.

Nous pouvons apprécier la diversité et la complexité des robots mobiles en les classant en fonction de leur système de locomotion, leur degré d'autonomie et leur domaine d'application.

Les robots mobiles sont utilisés dans une variété de domaines, tel que l'industrie à la défense en passant par la recherche en laboratoire, qu'ils soient télécommandés, semi-autonomes ou autonomes.

Chaque mode de déplacement répond à des besoins spécifiques, démontrant l'adaptabilité et l'ingéniosité de la robotique. La classification des robots mobiles reflète l'évolution et la diversification constante de ce domaine, promettant un avenir où ces machines joueront un rôle de plus en plus important dans notre société.

Chapitre 2

Chapitre 2 : La Navigation d'un robot mobile

2.1 Introduction

La robotique est un domaine de recherche qui se situe au carrefour de l'intelligence artificielle, de l'automatique, de l'informatique et d'électronique, cette interdisciplinarité est à l'origine d'une certaine complexité. Des applications dans des domaines aussi variés que l'industrie manufacturière, le spatial, l'automobile ou plus récemment les loisirs et le secteur médical, démontrent aujourd'hui l'intérêt économique et social de ces recherches.

La robotique mobile autonome vise plus spécifiquement à concevoir des systèmes capables de se déplacer de façon autonome. Les applications directes se situent notamment dans les domaines de l'automobile et de la robotique de service par exemple. Ce chapitre a pour cadre général étude de la navigation autonome des robots mobiles, et elle se focalise sur des applications dans le transport automatique industriel.

2.2 Architecture

Les premières architectures définies prenaient en compte de manière descendante la totalité des fonctions nécessaires à la réalisation d'une action et en particulier d'un déplacement, Cette approche conduit à une séparation modulaire des fonctions, créant une disposition horizontale des modules liés en série.

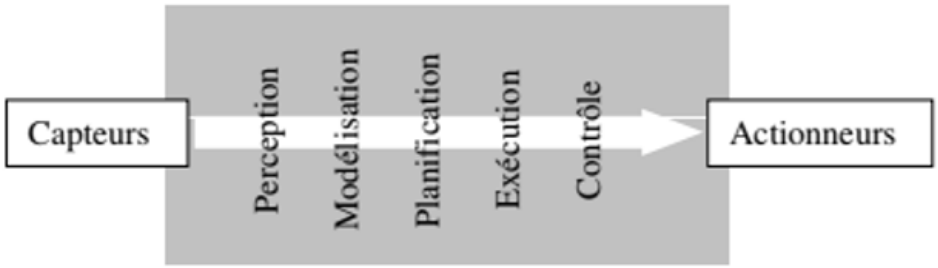

FIGURE 2.1 – Architecture structurelle et fonctionnelle d'un robot mobile

Un robot est décomposé de 02 architectures essentielles assurant son fonctionnement : **architecture structurelle** et**une autre fonctionnelle.**

2.2.1 Architecture structurelle

L'architecture structurelle d'un robot mobile se concentre sur la manière dont les composants matériels et logiciels sont connectés et organisés physiquement, elle ex-

plique la disposition physique des capteurs, des actionneurs, de l'unité de contrôle, de l'alimentation, etc.

Elle se divise en deux parties : **une électrique** et **l'autre mécanique. Partie électrique :**

- **Alimentation électrique :** Fournit l'énergie nécessaire au robot, elle peut être constituée de batteries, de piles à combustible ou d'une source d'alimentation externe.
- **Câblage :** Permet de transporter l'électricité entre les différents composants du robot.
- **Circuit électronique :** Commande les actionneurs et traite les informations provenant des capteurs. Il peut être constitué de microcontrôleur, de microprocesseur, de FPGA ou d'autres composants électroniques.
- **Capteurs :** Détectent l'environnement du robot ; les capteurs les plus courants sont les capteurs de distance, les caméras, les gyroscopes et les accéléromètres.
- **Actionneurs :** Permettent au robot de se déplacer et d'interagir avec son environnement, les actionneurs les plus courants sont les moteurs, les vérins et les pinces.

Partie mécanique :

- **Châssis :** Structure du robot qui supporte tous les autres composants.
- **Roues :** Permettent au robot de se déplacer.
- **Moteurs :** Fournissent la force nécessaire pour déplacer le robot.

2.2.2 Architecture fonctionnelle

L'architecture fonctionnelle d'un robot mobile explique comment ses différentes fonctions sont organisées pour atteindre ses objectifs. Elle englobe la partie informatique du robot

Partie informatique

- **Ordinateur :** Traite les informations provenant des capteurs et commande les actionneurs.
- **Logiciels :** Permettent au robot de fonctionner, les logiciels les plus courants sont les systèmes d'exploitation, les logiciels de navigation, de cartographie, de reconnaissance d'objets.
- **Capteurs intelligents :** Capteurs qui peuvent traiter les informations localement et envoyer des données prétraitées à l'ordinateur.
- **Actionneurs intelligents :** Actionneurs qui peuvent être contrôlés par des instructions de haut niveau et qui peuvent s'adapter à leur environnement[10].

2.3 Modélisation D'un Robot Mobile

La synthèse de la commande d'un système robotique spécifique nécessite généralement le modèle du système. Pour garantir les performances voulues en termes de précision, de rapidité et de robustesse, en particulier le suivi de trajectoire, la méthodologie de modélisation doit s'effectuer d'une manière rigoureuse.

2.3.1 Hypothèses de modélisation

— Les robots mobiles sont considérés comme mono-corps.

— Le châssis des robots, de même que les roues sont supposés rigides (ce qui exclut tout système de suspension). Dans ces conditions, les robots mobiles ne présentent pas :

Une rotation du châssis autour d'un axe parallèle à la direction d'avancement.

Rotation du châssis autour d'un axe perpendiculaire à la direction d'avancement.

Les plans ou chacune des roues restent à tout moment verticaux.

— La dynamique des différents moteurs commandant les roues en rotation et en orientation sont supposées négligeables. Dans ces conditions, les couples de rotation et d'orientation appliqués sur les roues peuvent être considérés comme étant les variables de commande.

— La surface d'évolution des robots est supposée horizontale et parfaitement plane.

— Il est supposé qu'aucune force aérodynamique n'agit sur les robots mobiles.

— Chaque roue est supposée indéformable et la zone de contact roue-sol est supposée ponctuelle.

— La vitesse linéaire du point de contact d'une roue avec le sol est nulle.

2.3.2 Coordonnées généralisées décrivant le châssis d'un robot

Soit O un point immobile dans le domaine d'évolution du robot et P un point fixe sur le châssis et soit (O, X, Y) un repère immobile dans le plan d'évolution du robot et (P, X', Y') un repère attaché au châssis. Nous noterons

— $\mathbf{(X}_p\mathbf{,\ Y}_p\mathbf{)}$ les coordonnées de P dans (O, X, Y).

— φ l'orientation du repère (P, X', Y') par rapport au repère (O, X, Y).

Ces 3 variables seront dans la suite regroupées dans le vecteur ξ :

$$\xi = \begin{pmatrix} X_p \\ Y_p \\ \varphi \end{pmatrix}$$

2.3.3 Modélisation géométrique

Grâce à cette modélisation, les robots mobiles à roues peuvent être représentés dans une variété de configurations, telles que des roues fixes, directrices, désaxées ou omnidirectionnelles.

En outre, cela permet de positionner les différentes composantes mobiles d'un robot les unes par rapport aux autres.

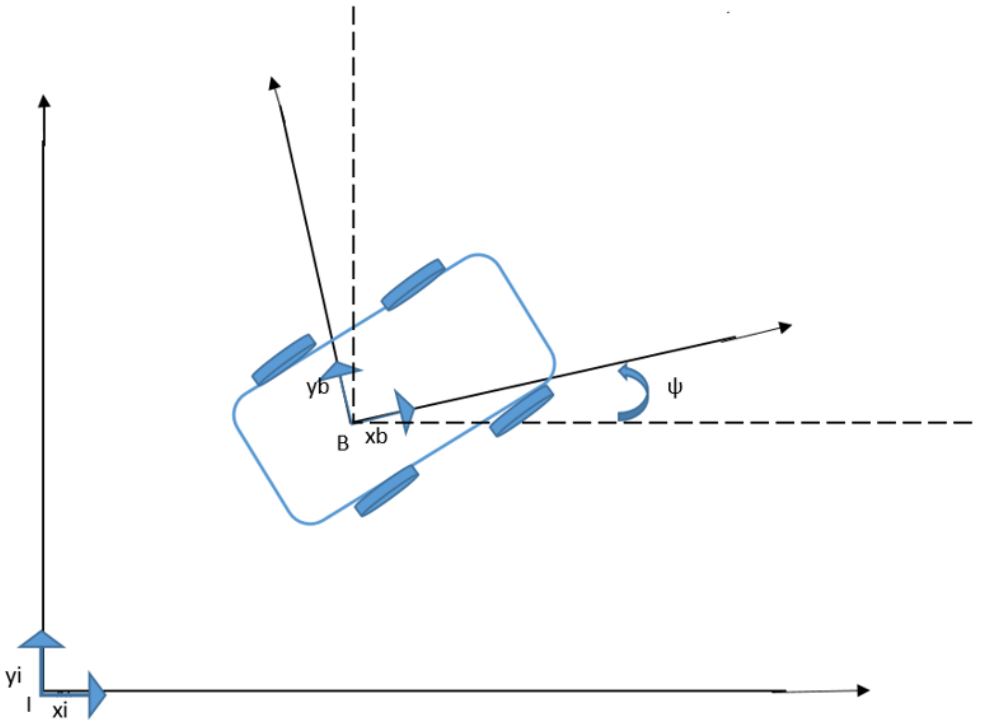

FIGURE 2.2 – Modélisation géométrique du robot mobile dans les repères global et local

Dans le cas des robots mobiles, tout l'ensemble du robot est considéré comme un seul corps solide qui se déplace sur un plan horizontal. Cela simplifie la géométrie et nous permet de trouver une relation entre le repère de référence globale, également connu sous le nom de repère mondial $F\,(I, \overrightarrow{x_i}, \overrightarrow{y_i}, \overrightarrow{z_i})$, et le repère de référence local du robot $F'\,(B, \overrightarrow{x_b}, \overrightarrow{y_b}, \overrightarrow{z_b})$, comme le montre clairement la figure 2.

Un point B sur le châssis du robot est choisi pour déterminer sa position dans le plan, c'est le point de référence de sa position et l'origine de son repère local $F'\,(B, \overrightarrow{x_b}, \overrightarrow{y_b}, \overrightarrow{z_b})$.

La position du point B en repère monde est spécifiée à l'aide des coordonnées x_b, y_b et l'angle Ψ qui définit l'orientation entre le repère global et le repère local. On peut décrire la pose du robot par un vecteur de ces trois éléments $\eta = (x, y, \Psi)^T$:

$$\eta = \begin{pmatrix} x \\ y \\ \Psi \end{pmatrix}$$

2.1) où :

- x : représente l'abscisse du centre de masse du robot dans le repère de référence globale $F\,(I, \overrightarrow{x_i}, \overrightarrow{y_i}, \overrightarrow{z_i})$,
- y : représente l'ordonnée du centre de masse du robot dans le repère de référence globale $F\,(I, \overrightarrow{x_i}, \overrightarrow{y_i}, \overrightarrow{z_i})$,
- Ψ : représente l'angle d'orientation du robot par rapport au repère de référence globale $F\,(I, \overrightarrow{x_i}, \overrightarrow{y_i}, \overrightarrow{z_i})$.

À ce stade, il est clair que la relation entre le repère du monde et le repère du robot est la rotation d'angle Ψ, et cette relation est exprimée par :

$$\eta = (\Psi) \cdot \eta' \tag{2.2}$$

Avec (Ψ) la matrice de rotation orthogonale :

$$R(\Psi) = \begin{pmatrix} \cos(\Psi) & -\sin(\Psi) & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

Et η' est la pose du robot dans le repère local :

$$\eta' = \begin{pmatrix} x_b \\ y_b \\ \Psi \end{pmatrix} \tag{2.4}$$

Ψ est l'angle de rotation entre le repère local et le repère monde.

Cette transformation exprime aussi le déplacement dans les deux repères. $\dot{\eta}$ et ξ décrivent les vitesses dans les deux repères :

$$\dot{\eta} = R(\Psi) \cdot \xi \tag{2.5}$$

2.3.4 Modélisation cinématique

La modélisation cinématique étudie le mouvement d'un système mécanique sans tenir compte des forces qui l'influencent. A ce moment-là, nous ne nous concentrons que sur les vecteurs de vitesse. On choisit généralement pour B un point remarquable de la plate-forme, typiquement le centre de l'axe des roues motrices ou le centre de gravité de robot mobile

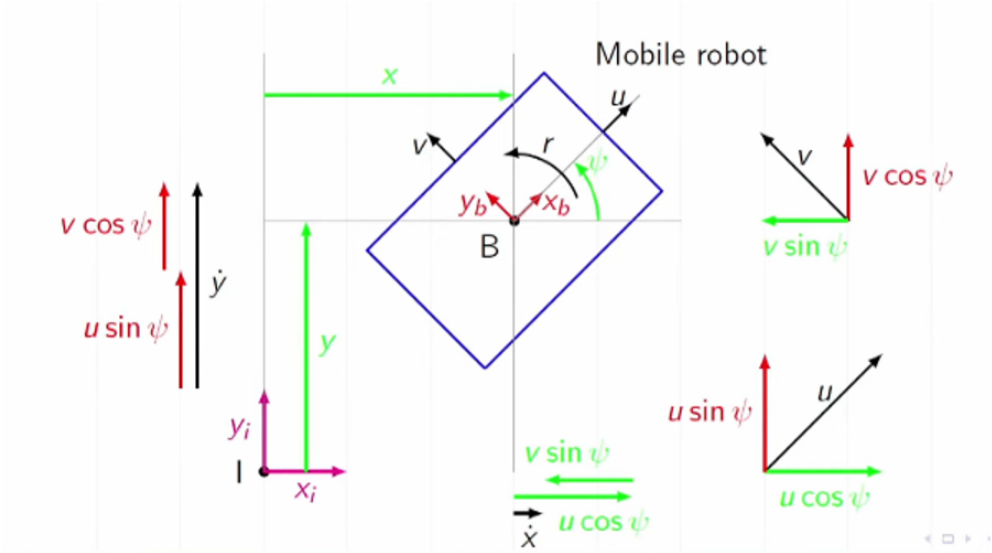

FIGURE 2.3 – Modèle cinématique : relation entre les vitesses locales et globales d'un robot mobile

Soit :

— (B, x_b, y_b) : repère mobile du robot.
— (I, x_i, y_i) : repère monde.
— x : déplacement longitudinal du robot dans le repère monde.
— y : déplacement latéral du robot dans le repère monde.
— Ψ : angle de rotation entre le repère local et le repère monde.
— u : vitesse longitudinale du robot dans le repère mobile.
— v : vitesse latérale du robot dans le repère mobile.
— r : vitesse de rotation du robot autour de z_b.

Nous voulons exprimer les vitesses $\dot{x}, \dot{y}$ et $\dot{\Psi}$ en fonction de u, v et r.

Par projection du vecteur u, v sur le repère monde

— Sur l'axe des x :

$$u_x = u \cdot \cos \Psi \tag{2.6}$$

— Sur l'axe des y :

$$v_x = v \cdot \sin \Psi \tag{2.7}$$

— Sur l'axe des x :

$$u_y = u \cdot \sin \Psi \tag{2.8}$$

— Sur l'axe des y :

$$v_y = v \cdot \cos \Psi \tag{2.9}$$

D'après la figure 03, l'expression des vitesses dans le repère monde est :

$$\dot{x} = u \cdot \cos \Psi - v \cdot \sin \Psi \tag{2.10}$$

$$\dot{y} = u \cdot \sin \Psi + v \cdot \cos \Psi$$

$$\dot{\Psi} = r$$

Sous forme matricielle

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\Psi} \end{pmatrix} = \begin{pmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ r \end{pmatrix} \tag{2.11}$$

Tel que :

— $\dot{x}$: vitesse selon l'axe X du robot mobile par rapport au repère de référence globale $(I, \overrightarrow{x_i}, \overrightarrow{y_i})$.

— $\dot{y}$: vitesse selon l'axe Y du robot mobile par rapport au repère de référence globale $(I, \overrightarrow{x_i}, \overrightarrow{y_i})$.

— $\dot{\Psi}$: vitesse rotationnelle du robot mobile par rapport au repère de référence globale $(I, \overrightarrow{x_i}, \overrightarrow{y_i})$.

— u : vitesse selon l'axe X du robot mobile par rapport au repère local $F(B, \overrightarrow{x_b}, \overrightarrow{y_b}, \overrightarrow{z_b})$.

— v : vitesse selon l'axe Y du robot mobile par rapport au repère local $F(B, \overrightarrow{x_b}, \overrightarrow{y_b}, \overrightarrow{z_b})$.

— r : vitesse rotationnelle du robot mobile par rapport au repère local $F'(B, \overrightarrow{x_b}, \overrightarrow{y_b}, \overrightarrow{z_b})$.

On note par η le vecteur $(\dot{x}, \dot{y}, \dot{\Psi})$ et par ζ le vecteur (u, v, r). Ainsi, le modèle cinématique du robot mobile est :

$$\dot{\eta} = R(\Psi) \cdot \zeta$$

ζ : commande d'entrée.

Modèle généralisé des roues

On passe au modèle généralisé des roues pour pouvoir commander le robot et bien préciser le son mouvement ; essayons d'écrire le vecteur ζ en fonction des vitesses des roues :

Une roue a deux vitesses : une vitesse latérale V_{slide} et une vitesse longitudinale V_{drive} telles que

$$V_{\text{drive}} = \omega_i \cdot a_i \quad \text{et} \quad V_{\text{slide}} = \beta_i \cdot \rho_i$$

avec :

— ω_i : vitesse angulaire de la $i^{\text{ème}}$ roue,
— a_i : rayon de la $i^{\text{ème}}$ roue,
— β_i : vitesse de rotation de la roue passive,
— ρ_i : rayon de la roue passive.

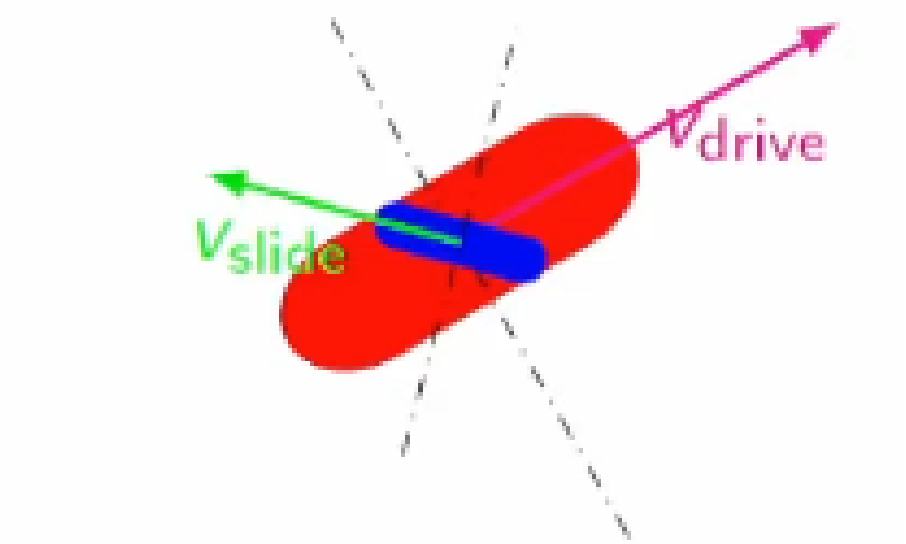

FIGURE 2.4 – Composants du vitesse d'une roue

Considérons le centre de la $i^{\text{ème}}$ roue notée c_i.

Définissons un repère au centre de la roue (c_i, x_{ci}, y_{ci}).

Représentons V_{slide} et V_{drive} sur ce repère.

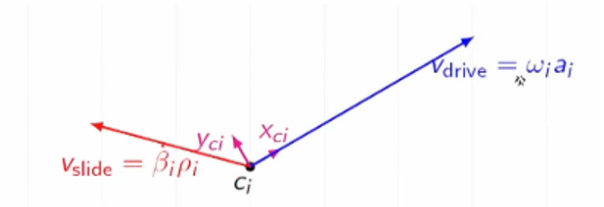

FIGURE 2.5 – Composants du vitesse d'une roue

Pour trouver la relation entre V_{slide} et V_{drive}, on pose un angle noté φ_i.

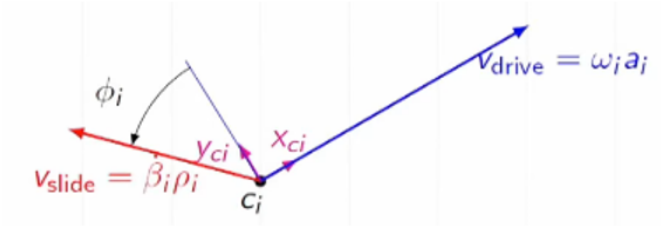

FIGURE 2.6 – Relation entre Vslide et Vdrive pour la roue d'un robot mobile en fonction des angles et des vecteurs de vitesse

Par projection sur l'axe y_{ci} on trouve

$$V_{\text{slide}y_{ci}} = \beta_i \cdot \rho_i \cdot \cos\varphi_i$$

$$V_{\text{drive}y_{ci}} = 0$$

Donc

$$\dot{y}_{ci} = \beta_i \cdot \rho_i \cdot \cos\varphi_i - 0 = \beta_i \cdot \rho_i \cdot \cos\varphi_i \tag{2.12}$$

Par projection sur l'axe x_{ci} on trouve

$$V_{\text{slide}x_{ci}} = \beta_i \cdot \rho_i \cdot \sin\varphi_i \tag{2.1}$$

$$V_{\text{drive}x_{ci}} = \omega_i \cdot a_i \tag{2.2}$$

Donc

$$\dot{x}_{ci} = \omega_i \cdot a_i - \beta_i \cdot \rho_i \cdot \sin\varphi_i \tag{2.13}$$

Essayons d'écrire ω_i en fonction de $\dot{x}_{ci}$ et $\dot{y}_{ci}$ De (2.12) :

$$\beta_i \cdot \rho_i = \frac{\dot{y}_{ci}}{\cos\varphi_i} \tag{2.14}$$

En remplaçant (2.14) dans (2.13) :

$$\omega_i = \frac{\dot{x}_{ci} + \dot{y}_{ci} \cdot \tan\varphi_i}{a_i} \tag{2.15}$$

Avec $\dot{x}_{ci}$ et $\dot{y}_{ci}$ sont la vitesse longitudinale et latérale du repère (c_i, x_{ci}, y_{ci}).

Maintenant essayons de représenter ω_i dans le repère local.

Définissons θ_{bi} comme l'angle entre le repère local et le repère de roue.

— V_{ci} : vecteur de vitesse linéaire du repère c_i dans le repère c : $(\dot{x}_{ci}, \dot{y}_{ci})^T$

— V_{ci}^{B} : vecteur de vitesse linéaire du repère c_i dans le repère B

$V_{ci}^{B} = R(\theta_{bi}) \cdot V_{ci}$: $R(\theta_{bi})$ est la matrice de transformation de repère.

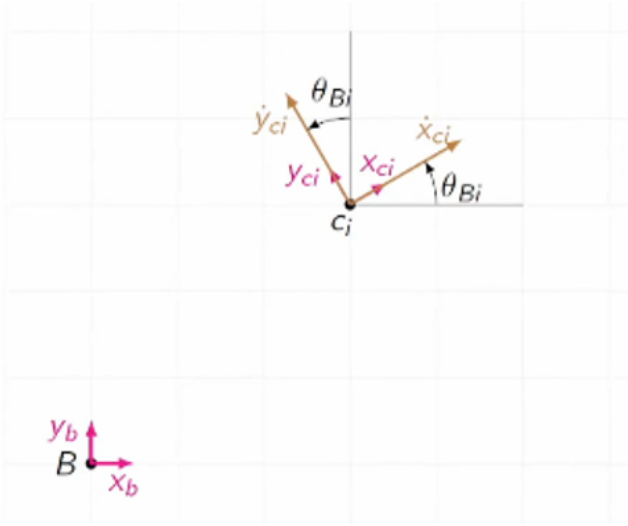

FIGURE 2.7 – Transformation du vecteur de vitesse entre le repère local du robot et celui de la roue

Par projection sur l'axe x_b :

$$\dot{x}_{ci/xb} = \dot{x}_{ci} \cdot \cos(\theta_{bi}) \tag{2.16}$$

$$\dot{y}_{ci/xb} = \dot{y}_{ci} \cdot \sin(\theta_{bi}) \tag{2.17}$$

Par projection sur l'axe y_b :

$$\dot{x}_{ci/yb} = \dot{x}_{ci} \cdot \sin(\theta_{bi}) \tag{2.18}$$

$$\dot{y}_{ci/yb} = \dot{y}_{ci} \cdot \cos(\theta_{bi}) \tag{2.19}$$

Donc

$$V_{ci}^{B} = \begin{pmatrix} \cos(\theta_{bi}) & -\sin(\theta_{bi}) \\ \sin(\theta_{bi}) & \cos(\theta_{bi}) \end{pmatrix} \cdot V_{ci} \tag{2.20}$$

Considérons que le corps et les roues forment un seul corps (les mêmes mouvements pour les deux). Supposons que le repère B a comme vitesse instantanée u, v, r, et la

distance entre le centre c_i et le centre du robot notée dx_i et dy_i suivant l'axe x et y respectivement, et θ_{bi} l'angle entre les deux repères. Mettons u, v, r dans le repère (c_i, x_{ci}, y_{ci}).

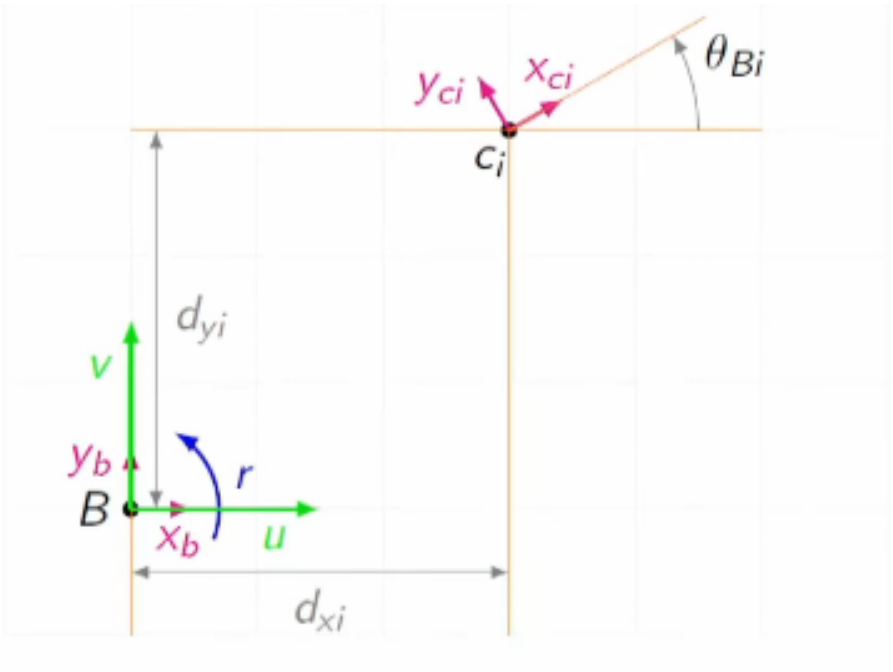

FIGURE 2.8 – Modèle de mouvement du robot utilisant les repères local et global

Puisque le même corps, le mouvement au niveau de repère B est le même au niveau de repère ci : on peut mettre u,v dans le repère du roue

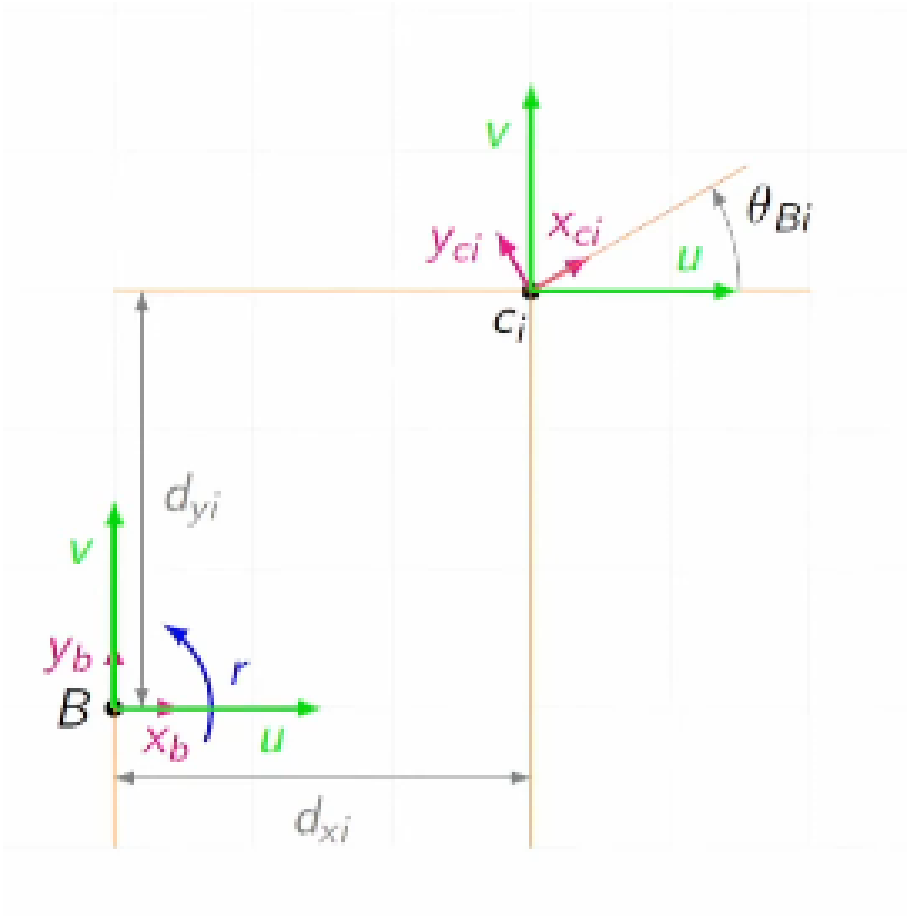

FIGURE 2.9 – Vitesses tangentielle et radiale dans le repère du robot

Pour la vitesse angulaire r , une vitesse tangentielle donne deux composants tangentielles rdxi et rdyi

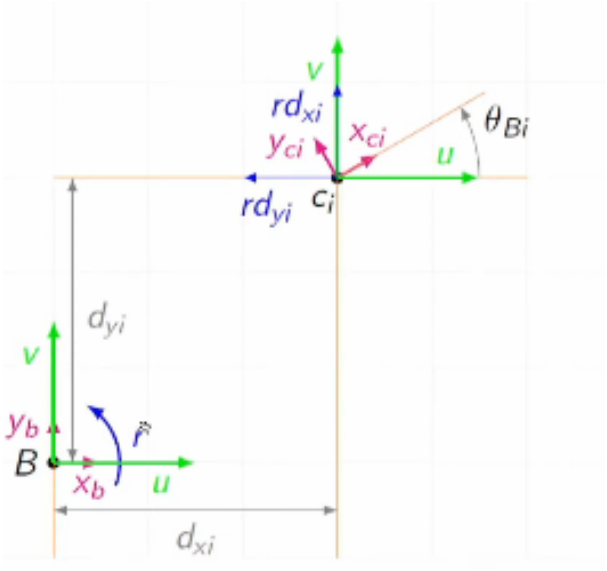

FIGURE 2.10 – Composantes de la vitesse angulaire dans le repère du robot pour un mouvement de rotation

$r \cdot dy_i$ devient sur l'axe x_b, mais avec un sens opposé car le sens de rotation r est dans le sens des aiguilles d'horloge.

$r \cdot dx_i$ devient sur l'axe y_b, mais dans le même sens que y_b.

Donc, la vitesse sur x_b au niveau de c_i est $u - r \cdot dy_i$ et sur y_b au niveau de c_i est $v + r \cdot dx_i$.

$$V^B_{ci/b} = \begin{pmatrix} 1 & 0 & -dy_i \\ 0 & 1 & dx_i \end{pmatrix} \begin{pmatrix} u \\ v \\ r \end{pmatrix} \tag{2.21}$$

D'après le résultat obtenu auparavant, (2.20) = (2.21) nous donne :

$$u - r \cdot dy_i = \dot{x}_{ci} \cdot \cos(\theta_{bi}) - \dot{y}_{ci} \cdot \sin(\theta_{bi})$$

$$v + r \cdot dx_i = \dot{x}_{ci} \cdot \sin(\theta_{bi}) + \dot{y}_{ci} \cdot \cos(\theta_{bi})$$

Cette relation est établie pour trouver la relation entre ω_i, u, v, et r.

La relation (2.15) sous forme matricielle est :

$$\omega_i = \begin{pmatrix} \frac{1}{a_i} & \frac{1}{a_i} \cdot \tan(\varphi_i) \end{pmatrix} \begin{pmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{pmatrix} \tag{2.22}$$

$$\begin{pmatrix} 1 & 0 & -dy_i \\ 0 & 1 & dx_i \end{pmatrix} \begin{pmatrix} u \\ v \\ r \end{pmatrix} = \begin{pmatrix} \cos(\theta_{bi}) & -\sin(\theta_{bi}) \\ \sin(\theta_{bi}) & \cos(\theta_{bi}) \end{pmatrix} \begin{pmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{pmatrix} \tag{2.23}$$

Remplaçons (2.23) dans (2.22) :

$$\omega_i = \begin{pmatrix} \frac{1}{a_i} & \frac{1}{a_i} \cdot \tan(\varphi_i) \end{pmatrix} \begin{pmatrix} 1 & 0 & -dy_i \\ 0 & 1 & dx_i \end{pmatrix} \begin{pmatrix} \cos(\theta_{bi}) & -\sin(\theta_{bi}) \\ \sin(\theta_{bi}) & \cos(\theta_{bi}) \end{pmatrix} \begin{pmatrix} u \\ v \\ r \end{pmatrix} \tag{2.24}$$

Du (2.11), on obtient la formule de $(u, v, r)^T$ et la remplaçons dans (2.24) :

$$\omega_i = \begin{pmatrix} \frac{1}{a_i} & \frac{1}{a_i} \cdot \tan(\varphi_i) \end{pmatrix} \begin{pmatrix} 1 & 0 & -dy_i \\ 0 & 1 & dx_i \end{pmatrix} \begin{pmatrix} \cos(\theta_{bi}) & -\sin(\theta_{bi}) \\ \sin(\theta_{bi}) & \cos(\theta_{bi}) \end{pmatrix} \begin{pmatrix} \cos\Psi & \sin\Psi & 0 \\ -\sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\Psi} \end{pmatrix} \tag{2.25}$$

[13]

2.3.5 Modélisation dynamique

La modélisation dynamique d'un robot mobile implique la description mathématique du mouvement du robot en fonction de ses caractéristiques physiques, de ses contraintes cinématiques et des forces appliquées. Cette modélisation permet de prédire le comportement du robot dans différentes conditions et d'aider à concevoir des algorithmes de contrôle pour guider le robot dans son environnement. Pour cela, il est recommandé d'utiliser la méthode de Lagrange.

La méthode de Lagrange est une approche mathématique qui permet de décrire le mouvement et le comportement dynamique du robot en termes de ses positions, vitesses, accélérations et forces appliquées.

La forme générale des équations de Newton-Euler Lagrange est donnée par :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = Q_i \tag{2.26}$$

où :

— L est la fonction de Lagrange, qui est définie comme la différence entre l'énergie cinétique KE et l'énergie potentielle PE du système : $L = KE - PE$

— q_i sont les coordonnées généralisées du système.

— $\dot{q}_i$ est la dérivée temporelle des coordonnées généralisées.

— Q_i représentent les forces généralisées agissant sur le système.

D'abord, on définit l'énergie cinétique KE et l'énergie potentielle PE de système d'état généralisé T :

$$T = f(\eta, \dot{\eta}, \dot{u}) : \text{réponse du système aux entrées} \tag{2.28}$$

L'énergie cinétique KE :

$$KE = T_{\text{translation}} + T_{\text{rotation}} \tag{2.29}$$

$$KE = \frac{1}{2} \cdot m \left(\dot{x}^2 + \dot{y}^2\right) + \frac{1}{2} I_z r^2 \tag{2.30}$$

Où :

— m : la masse du robot,

— $\dot{x}$ et $\dot{y}$: les vitesses de translation du robot selon les axes x et y,

— I_z : le moment d'inertie du robot par rapport à son centre de masse,

— r : la vitesse angulaire de rotation du robot.

L'énergie potentielle :

Souvent, la hauteur d'un système influence son énergie potentielle par rapport à une référence. L'énergie potentielle peut être considérée comme nulle si cette référence n'est pas définie ou si le système est considéré comme évoluant dans un plan horizontal où la hauteur n'est pas un facteur important.

En ce qui concerne un robot mobile qui n'a pas de référence en hauteur, cela implique que nous ne prenons pas en compte les variations de hauteur du robot par rapport à une surface orientée. Ainsi, l'énergie potentielle de gravitation, qui est fonction de la hauteur par rapport à une référence, est nulle.

Dans ce cas, l'équation de Newton-Euler Lagrange devient :

$$\frac{d}{dt}\left(\frac{\partial KE}{\partial \dot{q}}\right) - \frac{\partial KE}{\partial q} = Q_i \tag{2.31}$$

Soient B le centre de châssis du robot et C un autre point du châssis représenté dans la figure 11.

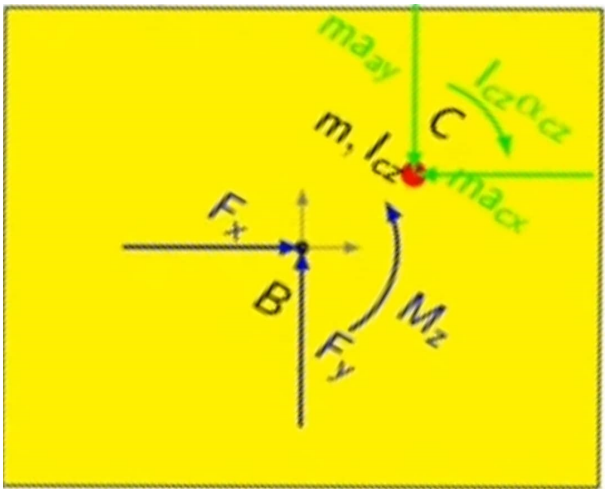

FIGURE 2.11 – Modèle de l'énergie cinétique du robot basé sur les coordonnées du point C

L'énergie cinétique du point C est :

$$KE = \frac{1}{2}m\left(\dot{x}_c^2 + \dot{y}_c^2\right) \tag{2.32}$$

Tel que :

$$\dot{x}_c = u - r \cdot y_{bc} \tag{2.33}$$

$$\dot{y}_c = v + r \cdot x_{bc} \tag{2.34}$$

Avec :

— x_{bc} : la distance entre les points B et C dans l'axe des x,

— y_{bc} : la distance entre les points B et C dans l'axe des y.

L'énergie totale L est donnée par :

$$L = KE = \frac{1}{2}m\left(u^2 + v^2 - 2u \cdot r \cdot y_{bc} + 2v \cdot r \cdot x_{bc} + r^2\left(y_{bc}^2 + x_{bc}^2\right)\right) + \frac{1}{2}I_z r^2 \tag{2.35}$$

Les dérivées partielles de L sont :

$$\frac{\partial L}{\partial u} = m \cdot u - m \cdot r \cdot y_{bc} \tag{2.36}$$

$$\frac{\partial L}{\partial v} = m \cdot v + m \cdot r \cdot x_{bc} \tag{2.37}$$

$$\frac{\partial L}{\partial r} = -m \cdot u \cdot y_{bc} + m \cdot v \cdot x_{bc} + m \cdot r \cdot \left(y_{bc}^2 + x_{bc}^2\right) + I_z \cdot r \tag{2.38}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial u}\right) = m \cdot \dot{u} - m \cdot \dot{r} \cdot y_{bc} - m \cdot r \cdot \dot{y}_{bc} \tag{2.39}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial v}\right) = m \cdot \dot{v} + m \cdot \dot{r} \cdot x_{bc} + m \cdot r \cdot \dot{x}_{bc} \tag{2.40}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\, r}\right) = -m{\cdot}\dot{u}{\cdot}y_{bc} - m{\cdot}u{\cdot}\dot{y_{bc}} + m{\cdot}\dot{v}{\cdot}x_{bc} + m{\cdot}v{\cdot}\dot{x_{bc}} + \dot{r}{\cdot}m{\cdot}\left(y_{bc}^2 + x_{bc}^2\right) + 2{\cdot}r{\cdot}m{\cdot}(2 \cdot y_{bc} \cdot \dot{y}_{bc} + 2 \cdot x_{bc} \cdot \dot{x}_{bc}) + I_z \tag{2.41}$$

Les équations de Newton-Euler-Lagrange :

$$\sum \mathbf{F} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = m{\cdot}a_c \quad \text{et} \quad \sum \mathbf{M} = \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} = I_z{\cdot}\alpha_c + \omega_c{\cdot}(I_z \cdot \omega_c) + m{\cdot}P_C^B{\cdot}(a_b + \omega_c{\cdot}v_b) \tag{2.42}$$

D'abord, on détermine les formules des accélérations :

a: accélération de robot

α : accélération angulaire du robot

P : est un vexteur de position tel que :

$$P_B^I = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \tag{2.43}$$

$$P_C^I = P_B^I + R(\Psi) \cdot P_C^B \tag{2.44}$$

$$\begin{pmatrix} x_{IC} \\ y_{IC} \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} + R(\Psi) \cdot \begin{pmatrix} x_{BC} \\ y_{BC} \\ 0 \end{pmatrix} \tag{2.45}$$

$$\dot{P}_C^I = \dot{P}_B^I + \dot{R}(\Psi) P_C^B + R(\Psi) \dot{P}_C^B \tag{2.46}$$

Pour un corps rigide $\dot{P}_C^B = 0$

$$\dot{R}(\psi) = \omega_B^I * R(\Psi) \tag{2.48}$$

$$\dot{P}_B^I = \dot{\eta} = J(\Psi) * (\xi) = R(\Psi) v_B \tag{2.49}$$

En fonction des vitesses de châssis fixes du robot

$$\dot{P}_C^I = \dot{P}_B^I + \omega_B^I * R(\Psi) P_C^B \tag{2.50}$$

$$\dot{P}_C^I = R(\Psi) v_B + \omega_B^I * R(\Psi) P_C^B \tag{2.51}$$

$$\dot{P}_C^I = R(\Psi)(v_B + \omega_B^I * P_C^B) \tag{2.52}$$

$$\ddot{P}_C^I = R(\psi)(\dot{v}_B + \dot{\omega}_B^I * P_C^B + \omega_B^I * \dot{P}_C^B) + \dot{R}(\psi)(v_B + \omega_B^I * P_C^B) \tag{2.53}$$

$$\ddot{P}_C^I = R(\Psi)(\dot{v}_B + \dot{\omega}_B^I * P_C^B) + \omega_B^I * R(\Psi)(v_B + \omega_B^I * P_C^B) \tag{2.54}$$

On note :

$$\alpha_B = \dot{\omega}_B^I$$

et

$$a_B = \dot{v}_B$$

$$\ddot{P}_C^I = R(\psi)(a_B + \alpha_B * P_C^B) + \omega_B^I * v_B + \omega_B^I * \omega_B^I * P_C^B) \tag{2.55}$$

$$R(\psi)^{-1} * \ddot{P}_C^I = (a_B + \alpha_B * P_C^B) + \omega_B^I * v_B + \omega_B^I * \omega_B^I * P_C^B) \tag{2.56}$$

$$a_C = a_B + \alpha_B * P_C^B + \omega_B^I * v_B + (\omega_B^I)^2 * P_C^B \tag{2.57}$$

$\alpha_B * P_C^B$: L'accélaration tangentionnelle
$\omega_B^I * v_B$:La force Coriolis
$\omega_B^{I\,2} * P_C^B$:L'accélération radiale
Tel que :

$$a_c = \begin{pmatrix} a_{cx} \\ a_{cy} \\ a_{cz} \end{pmatrix}, \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{\omega} \end{pmatrix}, \quad v_B = \begin{pmatrix} u \\ v \\ \omega \end{pmatrix}, \quad \alpha_B = \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix}, \quad \omega_B^I = \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \quad P_C^B = \begin{pmatrix} x_{BC} \\ y_{BC} \\ z_{BC} \end{pmatrix}$$

Pour un corps plan $z = 0$, $\omega = 0$, $\dot{\omega} = 0$, $a_c z = 0$, $r = 0$, $\dot{r} = 0$, $z_{BC} = 0$, $p = 0$, $q = 0$, $\dot{p} = 0$,$\dot{q} = 0$

Alors, (2.57) devient :

$$\begin{pmatrix} a_{cx} \\ a_{cy} \\ 0 \end{pmatrix} = \begin{pmatrix} \dot{u} \\ \dot{v} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \dot{r} \end{pmatrix} \begin{pmatrix} x_{BC} \\ y_{BC} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ r \end{pmatrix} \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ r \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ r \end{pmatrix} \begin{pmatrix} x_{BC} \\ y_{BC} \\ 0 \end{pmatrix} \tag{2.58}$$

Maintenant on détermine les forces et les moments agissant sur le robot :

$$F_x = m \cdot a_{cx} = m\left(\dot{u} - v \cdot r - x_{BC} r^2 - y_{BC}\dot{r}\right) \tag{2.59}$$

$$F_y = m \cdot a_{cy} = m\left(\dot{v} + u \cdot r - y_{BC} r^2 - x_{BC}\dot{r}\right) \tag{2.60}$$

$$M_z = I_{cz} \cdot \alpha_{cz} - m \cdot a_{cx} y_{BC} + m \cdot a_{cy} x_{BC} \tag{2.61}$$

$$M_z = I_{cz} \cdot \dot{r} - m \cdot y_{BC}\left(\dot{u} - v \cdot r\right) + m \cdot x_{BC}\left(\dot{v} + u \cdot r\right) + m \cdot \dot{r}\left(x_{BC}^2 + y_{BC}^2\right) \tag{2.62}$$

$$\begin{pmatrix} F_x \\ F_y \\ M_z \end{pmatrix} = \begin{pmatrix} m & 0 & -m \cdot y_{BC} \\ 0 & m & m \cdot x_{BC} \\ -m \cdot y_{BC} & m \cdot x_{BC} & I_{cz} + m\left(x_{BC}^2 + y_{BC}^2\right) \end{pmatrix} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{pmatrix} + \begin{pmatrix} -m \cdot r \cdot v - m \cdot x_{BC} r^2 \\ m \cdot r \cdot u - m \cdot y_{BC} r^2 \\ m \cdot x_{BC} u \cdot r + m \cdot y_{BC} v \cdot r \end{pmatrix} \tag{2.63}$$

$$\tau = D * \cdot\dot{\xi} + \eta(\zeta) \tag{2.64}$$

τ_η : vecteur des forces et des moments par rapport à la structure initiale

τ_ζ : vecteur des forces et des moments par rapport à la structure corporelle

$$\tau_\eta \dot{\eta} = \tau\zeta \tag{2.65}$$

$$\tau_\eta^t \dot{\eta} = \tau^t \zeta \tag{2.66}$$

$$\tau_\eta^t j(\eta)\zeta = \tau^t \zeta \tag{2.67}$$

$$\tau_\eta^t * j(\eta) = \tau^t \tag{2.68}$$

$$\tau_\eta * j(\eta)^t = \tau \tag{2.69}$$

On a :

$$\dot{\eta} = j(\eta)\zeta \tag{2.70}$$

$$\ddot{\eta} = j(\eta)\dot{\zeta} + \dot{j}(\eta)\zeta \tag{2.71}$$

$$\dot{\zeta} = j(\eta)^{-1}(\ddot{\eta} - \dot{j}(\eta)\zeta) \tag{2.72}$$

$$D * \dot{\zeta} + \eta(\zeta) = \tau \tag{2.73}$$

$$Dj(\eta)^{-1}(\ddot{\eta} - \dot{j}(\eta)\zeta) + \eta(\zeta) = \tau \tag{2.74}$$

$$Dj(\eta)^{-1}(\ddot{\eta} - \dot{j}(\eta)\zeta) + \eta(\zeta) = \tau_\eta j(\eta)^t \tag{2.75}$$

$$Dj(\eta)^{-t}(j(\eta)^{-1}(\ddot{\eta} - \dot{j}(\eta)\zeta)) + \eta(\zeta) = \tau_\eta \tag{2.76}$$

$$Dj(\eta)^{-t} * j(\eta)^{-1}\ddot{\eta} - Dj(\eta)^{-t}j(\eta)^{-1}\dot{j}(\eta)\zeta + \eta(\zeta)j(\eta)^{-t} = \tau_\eta \tag{2.77}$$

$$D_\eta * \ddot{\eta} + \eta_\eta = \tau_\eta \tag{2.78}$$

Avec :

$$D_\eta = Dj(\eta)^{-t} * j(\eta)^{-1} \tag{2.79}$$

et

$$\eta_\eta = -Dj(\eta)^{-t}j(\eta)^{-1}\dot{j}(\eta)\zeta + \eta(\zeta)j(\eta)^{-t} \tag{2.80}$$

2.4 Modèle dynamique d'un robot mobile avec la configuration des roues

Après avoir vu les forces qui agissent sur une roue, on peut maintenant faire un modèle de notre système représenté par la figure.

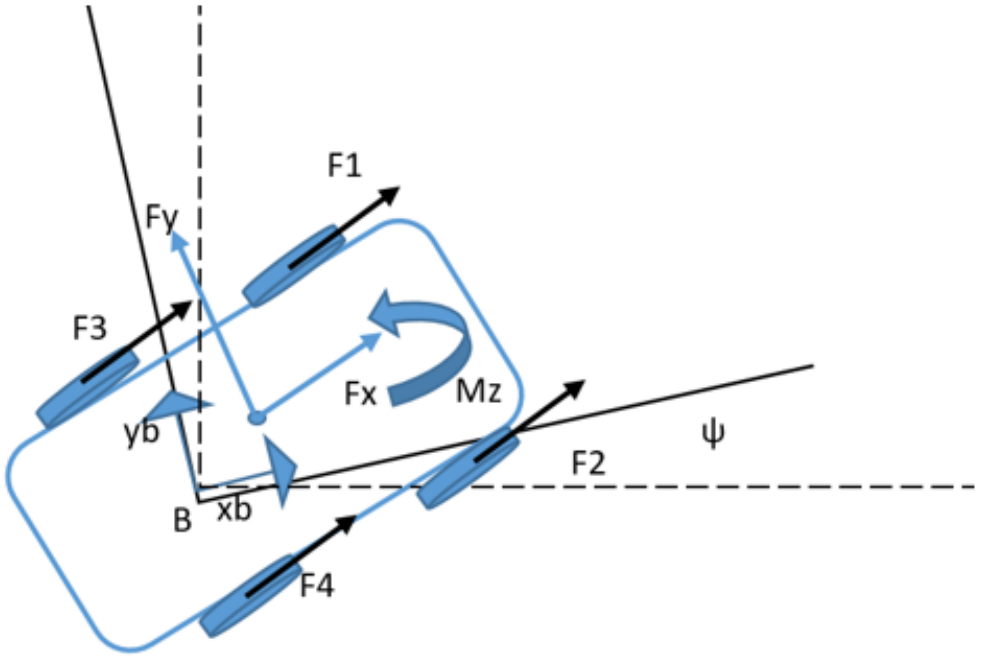

FIGURE 2.12 – Schéma des forces et moments agissant sur un robot mobile avec configuration des roues

Les forces F_y sont nulles car aucune force n'agit sur l'axe y et la résistance latérale est infinie.

$$F_x = F_1 + F_2 + F_3 + F_4 \tag{2.81}$$

$$M_z = -d \cdot F_1 + d \cdot F_2 - d \cdot F_3 + d \cdot F_4 \tag{2.82}$$

$$M_z = (-F_1 + F_2 - F_3 + F_4) \cdot d \tag{2.83}$$

Alors, $\tau = ik$

Tel que :

$$\begin{pmatrix} F_x \\ F_y \\ M_z \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ -d & d & -d & d \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} \tag{2.85}$$

[13]

2.5 Navigation autonome d'un robot mobile

2.5.1 La méthode classique

La figure 13 présente une description de la tâche de navigation d'un robot mobile. Comme montré, le robot démarre d'une situation initiale "s", il doit exécuter les actions de mouvement, qui sont généralement la vitesse et l'angle de braquage $v(t)$ et $\alpha(t)$ lui permettant de se mouvoir vers une nouvelle situation $s(t+1)$.

La navigation est obtenue à travers un processus itératif comme suit :

1. À chaque instant t, avec t =0,1, ..., k,. . ., le robot doit mesurer les distances aux obstacles de l'environnement di et les positions : courante et finale :$(x_r(t),y_r(t),\theta(t))^T$, $P_g(x_g, y_g)$
2. Le système de contrôle détermine les variables de commande adéquate $v_r(t+1)$ et $\alpha(t+1)$.
3. Le robot exécute ces actions en déplaçant vers les nouvelles coordonnées.
4. Répéter le même processus (les étapes : 1, 2 et 3) de détection de la situation et génération des actions jusqu'à la destination finale appelée **but**.[5]

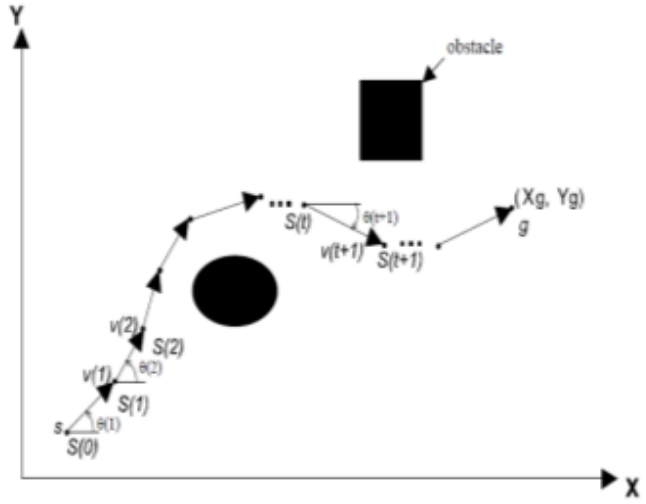

FIGURE 2.13 – Représentation de la méthode classique de la navigation autonome

Planification de la trajectoire

Pour atteindre son but final, l'un des principaux objectifs du robot mobile est de pouvoir évoluer dans un environnement complexe rempli d'obstacles. La séquence de déplacement sans collision avec ces obstacles entre la position initiale (point de démarrage) et le point but ou cible est nécessaire. Figure (2.13). La planification de trajectoire dans sa

formulation classique est le problème du calcul de ce chemin, dans un modèle géométrique de l'environnement cela est fait en introduisant le concept d'espace des configurations qui permet de transformer le problème de la recherche d'un chemin pour un système à n degrés de liberté dans l'espace euclidien en celui du mouvement d'un point dans un espace à n dimensions où le robot est représenté par un point. Plusieurs approches sont proposées pour la planification de trajectoire. Cependant, les plus utilisées sont la planification globale et locale

a- La planification globale de trajectoire

Est la modélisation de l'espace de l'environnement par un graphe, ou la recherche de trajectoire est basée sur l'utilisation d'algorithmes de graphe, comme le graphe de visibilité, la décomposition cellulaire... etc.

b- Planification de trajectoire locale

En général, l'environnement du robot mobile est inconnu et le robot ne dispose aucune information sur l'environnement à priori. Ainsi, une planification locale de type réflexe est requise. Le robot mobile doit analyser son environnement pendant le déplacement et prendre des décisions en fonction de cette analyse. Les stratégies d'intelligence artificielle réactives sont considérées comme des approches de planification locale[5].

Localisation

L'exactitude du déplacement d'un robot autonome est intimement liée à sa faculté de situer précisément sa position dans son environnement. Cette faculté est principalement alimentée par ses systèmes de localisation internes, qui fournissent les données critiques permettant au robot de planifier des trajectoires précises et efficientes vers sa destination.

Le robot doit se localiser dans l'environnement afin d'exécuter le mouvement prévu. L'estimation de la position du robot par rapport à un repère fixe de l'environnement est appelée localisation.

Une mesure des déplacements du robot ou sa position absolue dans l'environnement peuvent être utilisées pour effectuer cette estimation de la position.

Une liste des méthodes et capteurs disponibles est présentée [Borenstein 1996].L'estimation de la position absolue du robot dans l'environnement ne peut être effectuée indépendamment de la création d'une carte d'amers de l'environnement. Le problème de localisation est généralement modélisé dans un cadre probabiliste en raison des incertitudes sur les mesures utiles de localisation[5].

Les méthodes de localisation sont ainsi divisées en deux catégories :

a- Localisation estimée ou relative :

Obtenue par des informations des capteurs proprioceptifs et consiste à calculer la variation des coordonnées de position lors d'un déplacement en mesurant simplement les

distances parcourues et les directions empruntées depuis sa position initiale.

b- Localisation absolue : obtenue par des informations de capteurs extéroceptifs, le robot doit toujours connaître sa situation pour se déplacer d'un point à un autre en identifiant des repères artificiels, la méthode des balises est la plus utilisée.

Suivi de trajectoire

Le guidage désigne la manière dont le robot est orienté vers sa trajectoire cible, en ajustant constamment sa position et son orientation en fonction de la trajectoire de référence. Le suivi de trajectoire, quant à lui, s'occupe de calculer les commandes des actionneurs du système pour réaliser le mouvement prévu, en tenant compte des caractéristiques dynamiques du robot.

Ainsi, le guidage sert à définir l'orientation et la direction dans lesquelles le robot doit se déplacer, tandis que le suivi de trajectoire veille à ce que le robot suive fidèlement cette orientation tout en respectant les contraintes dynamiques du système. Les méthodes de commande par retour d'état sont souvent employées pour assurer ce suivi, car elles permettent d'asservir un robot sur une trajectoire de référence en ajustant ses paramètres internes.

Évitement d'obstacles

Le suivi de la trajectoire planifiée ne garantit pas l'absence de collisions avec les objets statiques ou dynamiques qui existent déjà. Le comportement d'évitement des obstacles est présent dans presque tous les mouvements des robots mobiles. Lors de l'exécution de la trajectoire, ces collisions peuvent se produire en raison d'une localisation imparfaite, d'un plan imprécis ou d'obstacles qui n'étaient pas inclus dans le modèle de l'environnement utilisé pour la planification de la trajectoire. Le robot mobile autonome doit être capable d'éviter efficacement les obstacles[5].

Panneaux de signalisation

L'objectif d'une mission de navigation est souvent d'atteindre une configuration finale spécifique. La réalisation de cet objectif est essentielle au succès de cette mission. Par ailleurs, les panneaux de signalisation vont exiger au robot à suivre un chemin précis même s'il n'est pas optimal.

2.5.2 La méthode SOFT COMPUTING

Le Soft computing a été proposé par Dr.L.A.Zadeh pour construire une nouvelle génération de la machine intelligente, ce n'est pas une méthodologie simple, mais une fusion

de plusieurs méthodologies, tel que les réseaux de neurones, la logique floue et les algorithmes génétiques, afin de couvrir les inconvénients d'une méthode en tirant profit de l'avantage de l'autre.

Définition

Le soft computing ayant un aspect plus complémentaire que compétitif, plus précisément, on tire un grand avantage en employant les réseaux de neurones, la logique floue et les algorithmes génétiques plus en combinaison qu'exclusivement. La Figure suivante illustre les différents fusionnements possibles entre les trois méthodes

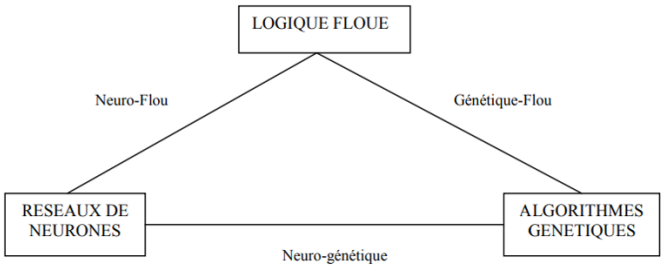

FIGURE 2.14 – Relations entre la logique floue, les réseaux de neurones et les algorithmes génétiques

a- Les réseaux neuronaux RNA : Les réseaux de neurones peuvent être utilisés pour une variété d'applications, notamment la reconnaissance des formes, le traitement du signal et l'apprentissage. Les réseaux de neurones artificiels ont été développés à partir de l'hypothèse que la structure et le comportement des éléments de base du cerveau sont la source du comportement intelligent. Les composants, leurs variables descriptives et leurs interactions peuvent être utilisés pour décrire les réseaux de neurones artificiels car ils sont des modèles[5].

b- Logique Flou : Cette technique permet de représenter et raisonner avec des informations imprécises ou incomplètes, ce qui est courant dans la perception robot. Elle peut être utilisée pour prendre des décisions de navigation en tenant compte de facteurs incertains, comme la distance estimée d'un obstacle.

c- Les Algorithmes Génétiques : sont basés sur la théorie de l'évolution de Darwin. Ils impliquent la croissance d'une variété de gadgets en utilisant diverses techniques telles que la sélection, les croisements et les mutations.Les Algorithmes Génétiques sont basés sur la théorie de l'évolution de Darwin.Ils impliquent la croissance d'une variété de gadgets en utilisant diverses techniques telles que la sélection, les croisements et les mutations [12].

2.6 Conclusion

La navigation d'un robot mobile représente un défi passionnant dans le domaine de la robotique, où les avancées technologiques permettent aux robots d'évoluer de manière autonome dans des environnements variés et complexes. L'exactitude du déplacement d'un robot autonome est intimement liée à sa faculté de situer précisément sa position dans son environnement. Cette faculté est principalement alimentée par ses systèmes de localisation internes, qui fournissent les données critiques permettant au robot de planifier des trajectoires précises et efficientes vers sa destination.

Le robot doit se localiser dans l'environnement afin d'exécuter le mouvement prévu. L'estimation de la position du robot par rapport à un repère fixe de l'environnement est appelée localisation.

Une mesure des déplacements du robot ou sa position absolue dans l'environnement peuvent être utilisées pour effectuer cette estimation de la position. Une liste des méthodes et capteurs disponibles est présentée [Borenstein 1996]. L'estimation de la position absolue du robot dans l'environnement ne peut être effectuée indépendamment de la création d'une carte d'amers de l'environnement. Le problème de localisation est généralement modélisé dans un cadre probabiliste en raison des incertitudes sur les mesures utiles de localisation[9].

Les méthodes de localisation sont ainsi divisées en deux catégories : L'importance de l'architecture du robot, tant structurelle que fonctionnelle, est cruciale pour assurer son bon fonctionnement, en organisant efficacement ses composants électriques, mécaniques et informatiques.

La navigation autonome, qui englobe la planification de trajectoire, la localisation, le suivi de trajectoire, l'évitement d'obstacles et la reconnaissance de panneaux de signalisation, nécessite des méthodes et des algorithmes sophistiqués pour permettre au robot de prendre des décisions rapides et précises en fonction de son environnement en constante évolution.

De plus, l'utilisation du soft computing, une approche combinatoire de techniques telles que les réseaux de neurones, la logique floue et les algorithmes génétiques, offre de nouvelles possibilités pour améliorer la navigation des robots mobiles en tenant compte de l'incertitude et de la complexité des environnements réels.

En somme, la navigation d'un robot mobile représente un domaine de recherche en constante évolution, avec des implications importantes dans de nombreux domaines tels que la logistique, la surveillance, et l'exploration.

Chapitre 3

Initiation à l'apprentissage automatique (Machine-Learning)

3.1 Introduction

L'intelligence artificielle (IA) est un domaine de l'informatique qui vise à développer des systèmes capables d'imiter l'intelligence humaine. Son histoire remonte aux années 1950, avec des chercheurs comme Alan Turing et John McCarthy qui ont posé les bases de ce domaine, dans les années 1970, l'IA a encore progressé dans les années 1980 avec des projets comme Prodigy, un service de messagerie électronique, avec une intelligence virtuelle intégrée à iOS. Depuis les années 2000, l'IA continue à progresser grâce à des technologies telles que la reconnaissance vocale, la reconnaissance faciale et la reconnaissance d'images. Elle a continué à faire de la progression avec des projets tels que Deep Blue, un ordinateur capable de jouer au jeu d'échecs.

L'intelligence artificielle comprend l'apprentissage automatique et l'apprentissage profond, le premier étant une méthode essentielle pour enseigner aux machines à apprendre à partir des données, tandis que l'apprentissage profond utilise des réseaux neuronaux profonds pour des analyses avancées.

Au fil des décennies, l'IA a connu des avancées significatives grâce au développement du Deep Learning, une approche basée sur des réseaux de neurones artificiels profonds.

3.2 Définition de l'apprentissage automatique

L'apprentissage automatique est un domaine de l'intelligence artificielle qui vise à donner aux machines la capacité d'apprendre des données à l'aide de modèles mathématiques. Il s'agit plus précisément du processus par lequel les informations pertinentes sont extraites d'un ensemble de données d'entraînement. Cette étape vise à obtenir les paramètres d'un modèle qui fourniront les meilleures performances, en particulier lors de la réalisation de la tâche confiée au modèle. Le modèle pourra être utilisé en production une fois l'apprentissage terminé [6].

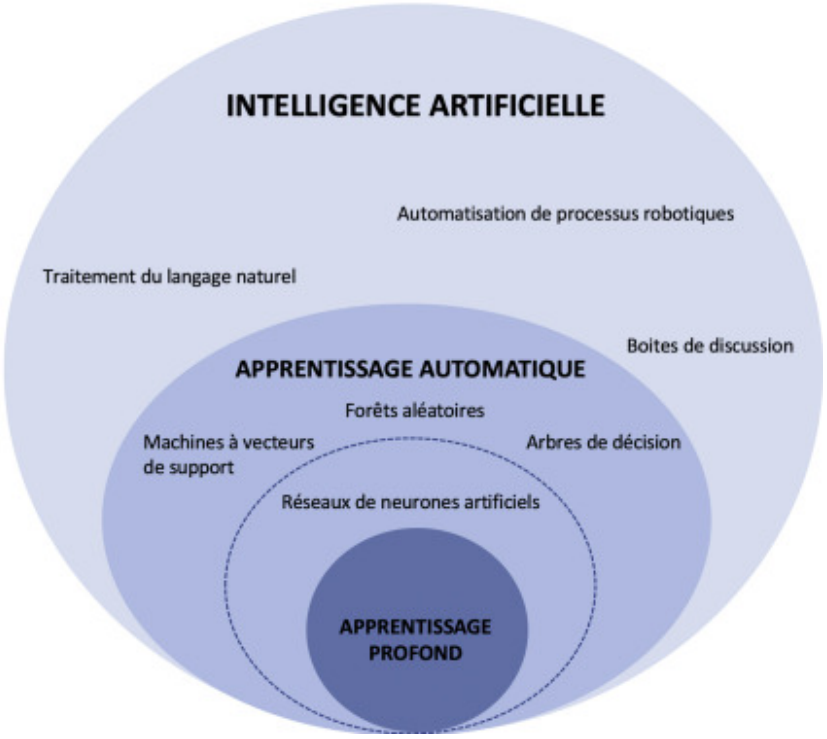

FIGURE 3.1 – Schéma des algorithmes de l'apprentissage automatique

3.2.1 Les phases d'apprentissage automatique

Les algorithmes d'apprentissage automatique sont généralement divisés en plusieurs étapes :

— **Phase d'entraînement (ou d'apprentissage) :** Un grand nombre d'exemples significatifs sont utilisés pour tester le modèle choisi. Les données d'entraînement sont utilisées par le système pour apprendre des règles implicites. Cette phase d'apprentissage précède généralement l'utilisation du modèle, bien que certains systèmes puissent apprendre indéfiniment s'ils disposent d'un retour sur les résultats.

— **Phase d'inférence :** La phase d'inférence consiste à utiliser le modèle entraîné sur de nouvelles entrées. Même si les entrées fournies lors de la phase d'inférence n'ont pas été vues par le modèle lors de la phase d'apprentissage, elles peuvent être traitées. En effet, le modèle peut se généraliser à des entrées inconnues en extrayant des règles implicites. [6]

3.3 Les types d'apprentissage automatique

L'apprentissage automatique permet aux ordinateurs d'apprendre et de s'améliorer par eux-mêmes à partir de données. Il existe plusieurs types d'apprentissage automatique, chacun ayant ses propres caractéristiques et ses propres applications.

3.3.1 Apprentissage supervisé

Lorsque les données d'entraînement sont étiquetées, c'est-à-dire que la sortie souhaitée est connue, on parle d'apprentissage supervisé. En enregistrant les N entrées et les sorties cibles correspondantes, on obtient l'ensemble de données

$D \equiv x_i, y_i, i[1, N]$.

L'objectif est d'entraîner le modèle choisi pour qu'il puisse prédire la sortie correctement pour des entrées non étiquetées.

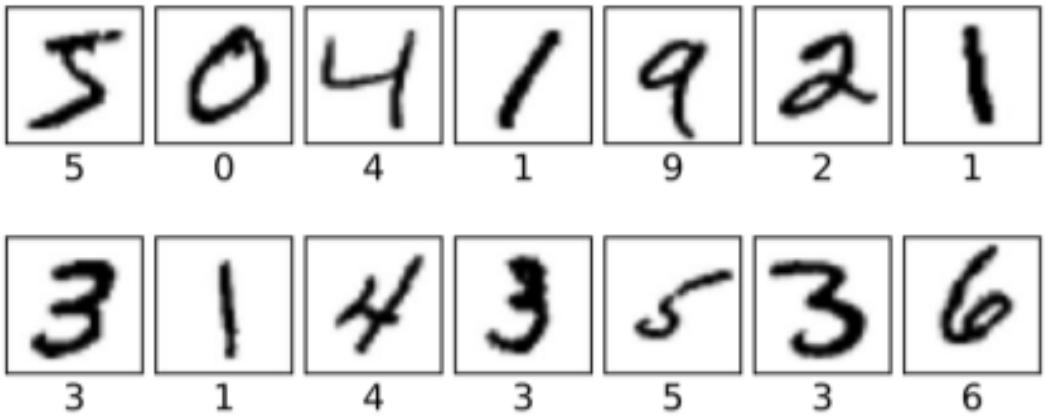

FIGURE 3.2 – L'apprentissage supervisé

L'apprentissage supervisé est généralement utilisé pour de la régression ou de la classification :

- La régression :

Lorsque la sortie à prédire prend des valeurs continues, il s'agit d'une variable réelle, qu'on appelle la régression.

- La classification :

Une tâche consistant à choisir une classe (valeur) parmi toutes celles possibles. Exemple : Un algorithme prédisant le chiffre manuscrit sur l'image d'entrée ou un algorithme classifiant une tumeur comme bénigne ou maligne . Il est important de noter que ces deux familles, classification et régression, ne sont pas exhaustives et qu'il existe des algorithmes réalisant des prédictions parmi des vecteurs de grande dimension, entre régression et classification. Les exemples les plus communs de modèles entre régression et classification se retrouvent dans le traitement automatique du langage où les mots peuvent être modélisés comme des combinaisons de lettres, donnant alors lieu à des vecteurs de très grande dimension. Parmi les algorithmes d'apprentissage supervisé classiques, on peut citer l'algorithme des k plus proches voisins, la régression linéaire, la régression logistique et les modèles tels que les réseaux de neurones artificiels, les arbres de décision, les forêts aléatoires et les machines à support de vecteur[6].

3.3.2 Apprentissage non supervisé

Les algorithmes d'apprentissage automatique non supervisés sont utilisés lorsque l'information utilisée pour entraîner le modèle n'est ni classifiée ni étiquetée. Le modèle en question étudie ses données d'entraînement dans le but de déduire une fonction pour décrire une structure cachée à partir des données.

À aucun moment le système ne connaît la sortie correcte avec certitude, Au lieu de cela, il tire des inférences des ensembles de données quant à ce que la sortie devrait être.

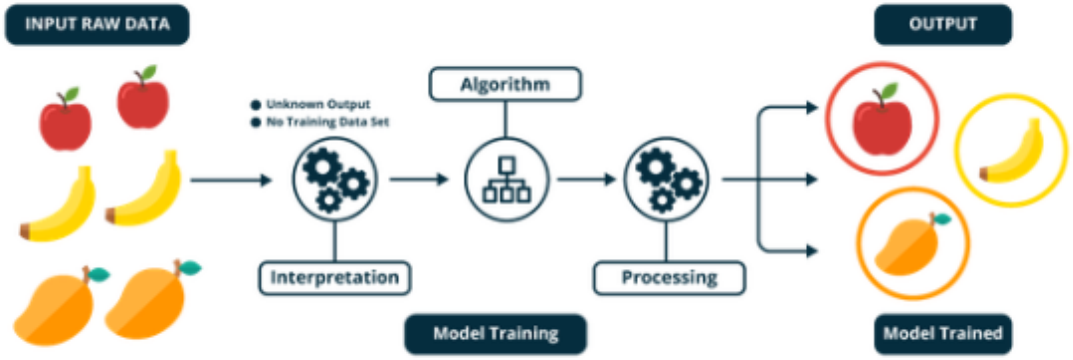

FIGURE 3.3 – Apprentissage non supervisé

Les algorithmes de ce type d'apprentissage peuvent être utilisés pour trois types de problèmes.

- Association :

Un problème où on désire découvrir des règles qui décrivent de grandes portions de ses données. L'apprentissage non supervisé peut être utilisé pour établir des règles de comportement basées sur les interactions entre les robots dans des environnements où plusieurs robots interagissent. Cela aide les robots à naviguer ensemble en évitant les collisions et en optimisant leur mouvement dans l'espace partagé.

- Regroupement :

Un problème où on veut découvrir les groupements inhérents aux données. En robotique, les données provenant de différents capteurs embarqués sur un robot peuvent être combinées pour identifier des modèles d'observations similaires. Cela peut être utilisé pour surveiller l'état du robot, détecter des anomalies et planifier des actions de maintenance.

- La réduction de dimension :

On vise à réduire le nombre de variables à prendre en compte dans l'analyse. Par exemple en robotique mobile, les données des capteurs peuvent être simplifiées en utilisant la réduction de dimensionnalité pour aider les robots mobiles à prendre des décisions

de navigation en se basant sur des caractéristiques importantes tout en réduisant la complexité du traitement[4].

3.3.3 Apprentissage par renforcement

Le renforcement Learning (RL) est une technique d'apprentissage automatique (ML) qui aide les logiciels à prendre des décisions pour obtenir les meilleurs résultats. Elle imite la méthode d'apprentissage par tâtonnements que les gens utilisent pour atteindre leurs objectifs. Les actions logicielles qui aident à atteindre votre objectif sont renforcées, tandis que les actions logicielles qui l'empêchent sont ignorées. Lorsqu'ils traitent des données, les algorithmes de RL utilisent un paradigme de récompense et de punition. Ils apprennent du retour d'information de chaque action et décident par eux-mêmes les meilleures méthodes de traitement pour atteindre les résultats finaux. Les algorithmes peuvent également différer les récompenses. La meilleure stratégie globale peut nécessiter des sacrifices à court terme ; par conséquent, ils peuvent déterminer que la méthode la plus efficace consiste à infliger des punitions ou des retours en arrière. Le RL est un moyen efficace d'aider l'IA à obtenir les meilleurs résultats dans des environnements invisibles. L'apprentissage par renforcement (RL) utilise divers algorithmes, notamment l'apprentissage par Q, les méthodes de gradient politique, les méthodes de Monte Carlo et l'apprentissage par différence temporelle. Les deux principales catégories comprennent tous ces algorithmes[1].

Applications

L'apprentissage automatique peut s'étendre à des problèmes complexes dans de nombreux domaines tels que :

- Robotique : Les problèmes de navigation autonome des véhicules.

- Médecine : L'apprentissage automatique est utilisé dans le domaine médical pour détecter les maladies tôt, analyser les images médicales et personnaliser les traitements.

- Reconnaissance d'images et de vidéos : Les problèmes de détection et de traitement d'images, et de reconnaissance faciale.

- Énergies renouvelables : l'apprentissage automatique est utilisé pour optimiser l'utilisation des sources d'énergie renouvelables.

- Cartographie et localisation simultanée (SLAM) : Les problèmes d'amélioration de la cartographie et de la localisation dans des environnements inconnus ou dynamiques.

3.4 Apprentissage profond

Le Deep Learning est une classe de techniques d'apprentissage automatique. Il est devenu une avancée majeure dans la vision informatique après que l'Alex Net (Krizhevsky, Sutskever, and Hinton, 2012a) ait réalisé une performance record sur ImageNet. (Rahmani and Mian, 2016). En général, les méthodes d'apprentissage profond sont des méthodes de machine Learning, utilisées pour modéliser des abstractions de haut niveau dans les données grâce à l'utilisation de réseaux neuronaux artificiels, qui sont composés de plusieurs transformations non linéaires.

3.5 Conclusion

Ce chapitre nous a permis de fournir un aperçu essentiel d'un domaine d'intelligence artificielle en plein essor qui est l'apprentissage automatique une forme d'IA qui permet aux machines d'apprendre à partir de données à l'aide de modèles mathématiques, même si l'objectif ultime est de créer des entités intelligentes capables de résoudre des problèmes. Cette ressource démontre l'ampleur des possibilités offertes par cette discipline en mettant en lumière les phases d'apprentissage, les types d'apprentissage (supervisé, non supervisé et par renforcement) et leurs applications dans divers domaines tels que la robotique, la médecine, la reconnaissance d'images et de vidéos, les énergies renouvelables. De plus, elle fournit une base solide pour comprendre et exploiter le potentiel de l'apprentissage automatique dans la création de solutions innovantes et la résolution de problèmes complexes. Cette ressource met l'accent sur quelques points importants à garder à l'esprit, tels que les différents types d'apprentissage, leurs limites et l'importance de choisir les méthodes appropriées pour des problèmes particuliers, par conséquent, elle encourage une compréhension approfondie des méthodes utilisées pour s'assurer qu'elles sont pertinentes dans divers contextes d'application.

Chapitre 4

Chapitre 04 :Implémentation, Test et Résultats

4.1 Introduction

Dans le cadre du développement de robots autonomes, la simulation joue un rôle essentiel pour tester et affiner les comportements avant une mise en œuvre dans le monde réel. Ce chapitre explore la simulation d'un robot mobile capable de détecter les panneaux de signalisation routiers et d'agir en conséquence en utilisant la plateforme ROS2 (Robot Operating System 2) couplée avec un simulateur de robots 3D (Gazebo).

Cette simulation va nous permettre de créer un environnement de commande où le robot peut interagir avec divers types de panneaux de signalisation, tels que les panneaux de stop, de limitation de vitesse, ou de sens interdit. À l'aide d'un modèle de détection basé sur l'algorithme YOLOv8n et d'un système de contrôle géré par ROS2, le robot doit être en mesure de reconnaître ces panneaux à partir de données capturées par une caméra, de traiter ces informations et de réagir en ajustant sa trajectoire ou sa vitesse, tout en gardant les mêmes conditions réelles dans le cas pratique.

Ce chapitre détaillera les différentes étapes de la mise en place de cette simulation, incluant la configuration des capteurs, l'intégration des algorithmes de détection, la gestion des contrôleurs pour les actions du robot, ainsi que l'analyse des résultats obtenus. Nous aborderons également comment la simulation peut être utilisée pour affiner les algorithmes de reconnaissance de panneaux et les stratégies de contrôle avant de les appliquer dans des scénarios réels.

4.2 Présentation de l'environnement de simulation

L'environnement de simulation est un composant crucial du processus de développement de systèmes robotiques, offrant une plateforme virtuelle où les comportements et les interactions du robot peuvent être testés sans les risques et les coûts associés aux essais en conditions réelles. Dans le cadre de notre projet, l'environnement de simulation est conçu pour recréer un scénario routier réaliste dans lequel un robot mobile doit détecter et réagir aux panneaux de signalisation.

4.2.1 Logiciel Blender

Définition

Blender est un logiciel open-source de création de contenu 3D qui permet de modéliser, texturer, animer, et rendre des objets et des scènes tridimensionnels. Utilisé largement dans les domaines de l'animation, du design de jeux vidéo, de la réalité virtuelle, et du développement de simulations.

Blender offre une suite complète d'outils pour la modélisation polygonale, la sculpture, le texturing, le shading, la simulation de fluides et de particules, et bien plus encore. En robotique, Blender est souvent utilisé pour créer des modèles 3D détaillés d'environnements et d'objets qui sont ensuite importés dans des simulateurs comme Gazebo.

FIGURE 4.1 – Logo de logiciel blender

Blender : La Modélisation 3D

Blender a été utilisé pour créer les modèles 3D des éléments de l'environnement, tels que les panneaux de signalisation, les routes, et les infrastructures environnantes. Grâce à ses capacités avancées de modélisation et de texturisation, Blender permet de concevoir des objets avec un niveau de détail élevé, ce qui renforce le réalisme de la simulation. Une fois ces modèles créés, ils sont exportés et intégrés dans Gazebo pour être utilisés dans la simulation.

L'utilisation de Blender permet de personnaliser les éléments du décor et d'adapter l'environnement à des scénarios spécifiques que le robot devra affronter. Cette flexibilité est particulièrement utile pour tester le robot dans diverses situations et pour évaluer sa capacité à naviguer et à réagir dans des contextes variés[3].

4.2.2 Gazebo : Le Simulateur 3D

Définition

Gazebo est un simulateur de robotique 3D open-source qui offre des capacités de simulation physique avancées pour tester et développer des systèmes robotiques dans un environnement virtuel. Gazebo permet de simuler des robots, des capteurs, et des environnements entiers avec une grande précision, en tenant compte de la physique réelle, y compris la gravité, la friction, et les collisions. Grâce à ses fonctionnalités d'intégration avec ROS (Robot Operating System), Gazebo est largement utilisé pour le prototypage et le test de robots avant leur déploiement dans le monde réel.

FIGURE 4.2 – Logo de Gazebo

Gazebo est utilisé comme simulateur 3D pour modéliser l'environnement et les interactions du robot. Ce simulateur offre une modélisation physique détaillée, permettant au robot de se déplacer et d'interagir de manière réaliste avec les objets, y compris les panneaux de signalisation. L'environnement simulé comprend des routes, des intersections et une variété de panneaux placés stratégiquement pour tester les capacités du robot en matière de navigation autonome.

La version utilisée dans notre simulation est "Gazebo 11.11" car elle est une version LTS (Long-Term Support), ce qui signifie qu'elle bénéficie d'un support et de mises à jour à long terme, stable et largement utilisée, offrant une excellente compatibilité avec ROS2 Foxy, ce qui en fait un choix fiable pour la simulation de robots. Elle bénéficie également d'une communauté active et d'une large gamme de fonctionnalités éprouvées pour la modélisation physique et les environnements complexes[7].

4.2.3 ROS2 : Le Cadre de Contrôle et de Communication

Définition

ROS2 (Robot Operating System 2) est une plateforme logicielle open-source qui fournit un ensemble d'outils et de bibliothèques pour le développement de logiciels robotiques. ROS2 facilite la création de systèmes robotiques modulaires et distribués, où différents composants du robot, tels que les capteurs, les algorithmes de contrôle, et les interfaces utilisateur, peuvent communiquer et coopérer de manière efficace. Évolution de ROS, ROS 2 introduit des améliorations en termes de performance, de sécurité, et de support pour les systèmes embarqués, rendant la plateforme plus adaptée aux applications robotiques modernes, y compris les environnements de production industrielle et les systèmes embarqués.

FIGURE 4.3 – Logo de logicielle ROS

ROS2 (Robot Operating System 2) est utilisé pour gérer la communication entre les différents composants du robot et pour contrôler ses actions en réponse aux stimuli de l'environnement simulé. Grâce à ROS2, le système est organisé en nœuds qui interagissent via des topics et des services. Cela inclut la capture des images depuis une caméra virtuelle, la détection des panneaux, et la commande des moteurs en fonction des résultats de la détection.

4.3 Les notions de base de ROS

Les notions de bases suivantes doivent être acquissent pour utiliser ROS :

— **Ros Node :** c'est une instance d'un exécutable. Un nœud peut correspondre à un capteur, un moteur, un algorithme de traitement ou de surveillance. Chaque nœud qui se lance se déclare au Master.

— **Ros Topics :** l'échange de l'information s'effectue soit de manière asynchrone via un topic ou de manière synchrone via un service. Un topic est un système de transport de l'information basé sur le système de l'abonnement / publication (subscribe / publish). Un ou plusieurs nœuds pourront publier de l'information sur un topic et un ou plusieurs nœuds pourront lire l'information. Le topic est en quelque sorte un bus d'information asynchrone. Il est typé, c'est-à-dire que le type d'information qui est publiée (le message) est toujours structuré de la même manière. Les nœuds envoient ou reçoivent des messages sur des topics.

— **Ros Topics :** Un paquet une unité de base du ROS. L'application ROS est développée sur la base de paquets, et le paquet contient soit un chier de con guration pour lancer d'autres paquets ou nœuds. Le paquet contient également tous les chiers nécessaires au fonctionnement du paquet, y compris les bibliothèques de dépendances ROS pour l'exécution de divers processus, les ensembles de données et le chier de con guration. Le nombre de paquets au ciels est d'environ 2 500 pour ROS Indigo en juillet 2017 . En outre, bien qu'il puisse y avoir quelques redondances, il existe environ 4 600 paquets développés et difusés par les utilisateurs

— **Ros Messages :** un message est une structure de données définissant le type d'un sujet, elle est composée d'une structure imbriquée de « entiers, booléens, flottants,

chaîne de caractères,.. »

— **Topic :** Ce mode est asynchrone, il est basé sur la notion du "publisher" et "subscriber". Comme les noms en Anglais l'indiquent, ce mode fonctionne comme suivant : Si les topics sont les routes permettant le transfert, la récupération, ou encore l'envoi de messages aux nœuds, alors les publishers et subscribers sont les bus transportant ces messages. C'est par leur biais que communiqueront les topics et nœuds. Ainsi, pour envoyer un message, un nœud utilisera un publisher pour transmettre son information au topic associé, s'il est possible de recevoir une information du composant relatif au nœud. Dans la même idée, un subscriber sera utilisé pour recevoir des informations venant d'un topic. De plus, plusieurs n÷uds peuvent être des subscribers sur le même topic, voir la figure 4.4 Une fois que la connexion est faite, le Topic transmit et reçoit les messages en continue, c'est pour cela que ce mode de communication est idéal pour les applications de mesure continue comme les radars.

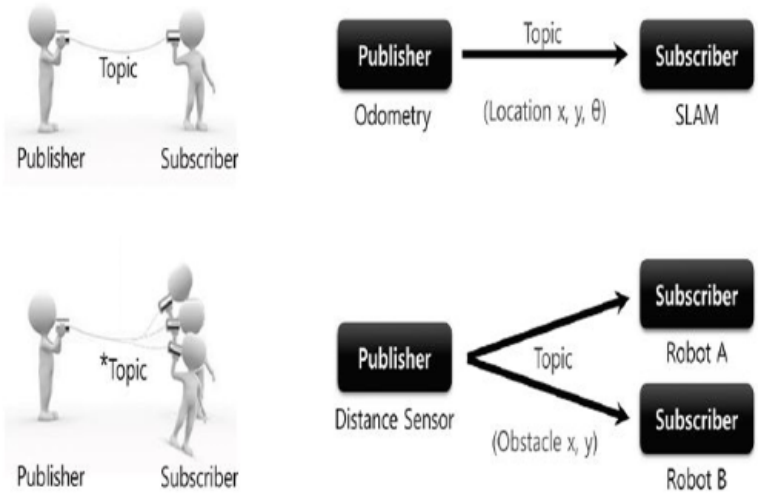

FIGURE 4.4 – Exemple de communication en mode Topic

— **Service :** dans ce mode illustré par la figure 4.5, la communication est synchrone entre le client "client" qui demande un service concernant une tâche particulière et le serveur "server" qui est chargé de répondre aux demandes. Le serveur ne répond qu'après la demande arrivant d'un client, le client attends la réponse, c'est la raison pour laquelle ce mode est dit synchrone. Une fois le client reçoit le message demandé la connexion sera interrompue, grâce à cette caractéristique le mode service est très important pour réduire l'encombrement du réseau

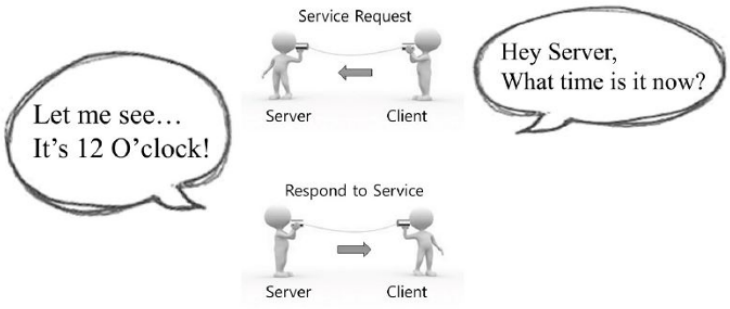

FIGURE 4.5 – Interaction client-serveur : requête et réponse de service

— **Les bags :** Les données des messages ROS peuvent être enregistrées. Le format de fichier utilisé est appelé bag, et ".bag" est utilisé comme extension de chier. Dans le ROS, bag peut être utilisé pour enregistrer les messages et les lire si nécessaire pour reproduire l'environnement dans lequel les messages sont enregistrés. Par exemple, lors d'une expérience de robot utilisant un capteur, les valeurs du capteur sont stockées dans le formulaire de message à l'aide du bag. Ce message enregistré peut être chargé à plusieurs reprises sans effectuer le même test en lisant le fichier enregistré dans le sac. Les fonctions d'enregistrement et de lecture de rosbag sont particulièrement utiles lors du développement d'un algorithme avec des modifications fréquentes du programme

4.4 Les outils essentiels dans ROS

ROS est un ensemble d'outils, il en existe de nombreux qui sont largement utilisés dans la programmation, la simulation ou l'exécution des comportements des robots. Citons quelques outils ou algorithmes fréquemment utilisés par le programmeur de ROS.

— **Rviz :** C'est une interface graphique permettant d'afficher les modèles des robots, des cartes de navigation reconstruites par des algorithmes de SLAM (Simultaneous Localization and Mapping), d'interagir avec le robot, d'afficher des images, des points 3D fournis par des caméras 3D.

— **Colcon :** est un outil de build utilisé dans ROS 2 (et d'autres environnements de développement multi-paquets) pour compiler et organiser des projets complexes composés de plusieurs packages. Colcon a été conçu pour remplacer Catkin, offrant une meilleure gestion des dépendances, des builds parallèles, et une plus grande flexibilité pour différents langages de programmation

— **Rqt graph :** C'est un plugin fourni par ROS qui est très utile. Il s'agit d'une interface qui permet de visualiser les nœuds actifs ainsi que les transferts d'information entre ceux-ci. C'est un excellent outil pour déboguer un programme ROS et valider son bon fonctionnement[7] [8].

4.5 Les Commandes essentiel de ROS 2

Fonctionnalité	Commande ROS 2
Initialiser un workspace	mkdir -p /ros2$_w s/src$ cd /ros2$_w s$ colcon build
Compiler des packages	colcon build
Source du workspace	source install/setup.bash
Lancer un nœud	ros2 run <package> <node>
Lancer un fichier de lancement	ros2 launch <package> <file>.launch.py
Lister les packages installés	ros2 pkg list
Afficher les informations d'un package	ros2 pkg info <package>
Lister les nœuds actifs	ros2 node list
Obtenir des infos sur un nœud	ros2 node info <node>
Lister les topics	ros2 topic list
Écouter un topic	ros2 topic echo <topic>
Publier un message sur un topic	ros2 topic pub <topic> <msg$_t ype >< msg >$
Lister les services	ros2 service list
Appeler un service	ros2 service call <service> <args>
Lister les actions	ros2 action list
Appeler une action	ros2 action send$_g oal < action >< goal >$
Lister les paramètres d'un nœud	ros2 param list
Modifier un paramètre d'un nœud	ros2 param set <node> <param> <value>
Voir les informations sur le système	ros2 doctor
Tester la fréquence des topics	ros2 topic hz <topic>
Voir les informations sur les types de messages	ros2 msg list

TABLE 4.1 – Liste des commande de ROS2

4.5.1 Pourquoi ROS2 et non pas ROS1 ?

Dans le domaine de la détection de panneaux de signalisation et la réaction en conséquence, ROS2 est bien plus adapté que ROS1 pour plusieurs raisons clés. Tout d'abord, ROS2 offre une meilleure gestion des communications en temps réel, ce qui est essentiel pour traiter rapidement les images capturées par la caméra et pour réagir sans délai aux panneaux détectés. Grâce à son middleware DDS(Data Distribution Service), ROS2 permet une communication fiable et décentralisée entre les différents nœuds, ce qui le rend plus robuste pour les systèmes critiques comme les robots autonomes dans un environnement complexe. Contrairement à ROS1, où le maître central (roscore) peut devenir un point de défaillance et introduire des latences, ROS2 élimine ce problème avec une architecture décentralisée. De plus, ROS2 prend en charge des systèmes embarqués plus facilement, comme les Raspberry Pi, largement utilisés pour la détection de panneaux grâce à des modèles de vision par ordinateur comme YOLO. Cela permet une meilleure

intégration entre les capteurs, le traitement des images, et la prise de décision en temps réel, essentiels pour que le robot puisse réagir rapidement et de manière fiable après la détection d'un panneau. Ainsi, ROS2 est plus robuste, performant et évolutif que ROS1 pour des applications critiques comme la reconnaissance de panneaux et la prise de décision en conséquence.

4.6 Détection des Panneaux dans Gazebo

Contrairement à l'application réelle, où la détection des panneaux serait effectuée par un modèle YOLO v8n fonctionnant sur un Raspberry Pi 5, dans cette simulation, le processus de détection est directement intégré au sein de Gazebo. Cette approche permet de simuler la reconnaissance des panneaux et de générer les réponses appropriées du robot sans dépendre d'un matériel physique spécifique. Les images capturées par la caméra virtuelle sont traitées pour identifier les panneaux, et les commandes de contrôle du robot sont ajustées en conséquence.

4.7 Interaction et Contrôle

Le robot simulé est équipé de deux roues motrices, et ses mouvements sont contrôlés via un système de commande' gazebo ros diff drive ' implémenté en ROS2 foxy. Lorsqu'un panneau de signalisation est détecté, le robot ajuste sa vitesse et sa direction en fonction des informations reçues. Par exemple, un panneau de stop oblige le robot à s'arrêter pendant un délai, tandis qu'un panneau de limitation de vitesse régule sa vitesse maximale.

4.8 Les étapes de simulation

La première étape de la simulation consiste à créer l'environnement virtuel dans Blender, un logiciel de modélisation 3D. Dans ce projet, l'environnement est composé d'une route, des intersections, de panneaux de signalisation distribués .

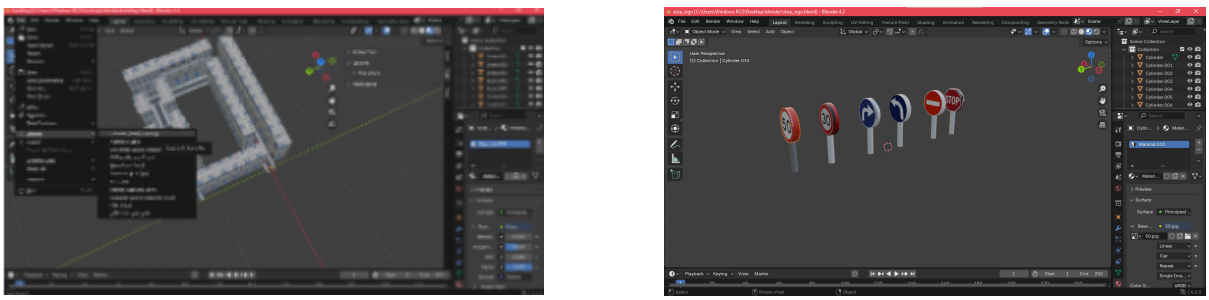

FIGURE 4.6 – Exportation du .blend vers .dae sur blender 4.2.0

Chaque élément est modélisé avec précision pour refléter les conditions réelles auxquelles le robot sera confronté. Une fois l'environnement complété dans Blender, le fichier .blend est exporté au format .dae et converti en un fichier .world (changement d'extension), qui est le format utilisé par le simulateur Gazebo.

Le fichier .world est ensuite intégré dans le package yolobotgazebo au sein du workspace yolobot de ROS2. Cela permet à Gazebo de charger l'environnement simulé et d'activer les différents capteurs et contrôleurs du robot dans un cadre virtuel réaliste.

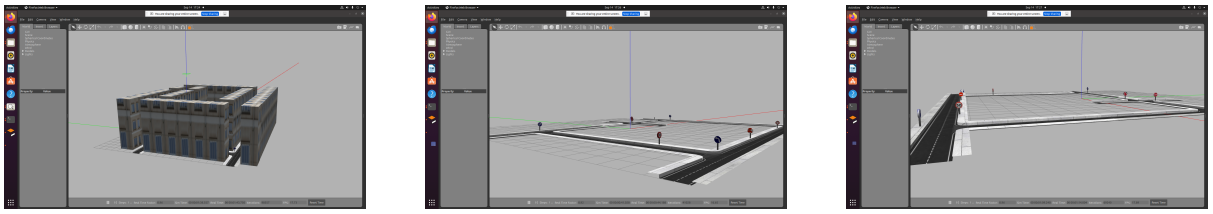

FIGURE 4.7 – Le terrain sur gazebo

La configuration de ROS2 implique la création de nœuds pour gérer la communication entre le robot et l'environnement simulé. Des nœuds sont configurés pour capturer les données des capteurs (comme la caméra) et pour traiter ces informations via des algorithmes de détection de panneaux de signalisation. Ces nœuds contrôlent également les actions du robot en réponse aux signaux détectés, tels que l'ajustement de sa trajectoire ou de sa vitesse en fonction des panneaux identifiés. Ainsi, en passant par ces étapes, de la création de l'environnement dans Blender à la configuration de ROS2, le robot peut être testé dans un environnement virtuel proche de la réalité, permettant d'affiner et de valider ses capacités avant de passer à une mise en œuvre physique.

4.9 Visualisation et Simulation

4.9.1 Installation du ros2 foxy

L'installation du ROS2 foxy se fait en suivant les instructions d'installation sur Installation de ROS Foxy.

4.9.2 Installation du dependance nécessaire

Le fonctionnement correcte de notre simulation nécessite l'installation des dépendances suivantes :

pip

est un gestionnaire de paquets pour Python, utilisé pour installer, mettre à jour et gérer des bibliothèques ou des paquets Python. Il est largement utilisé pour installer

des packages à partir du Python Package Index (PyPI), qui est un dépôt officiel contenant des milliers de bibliothèques Python.

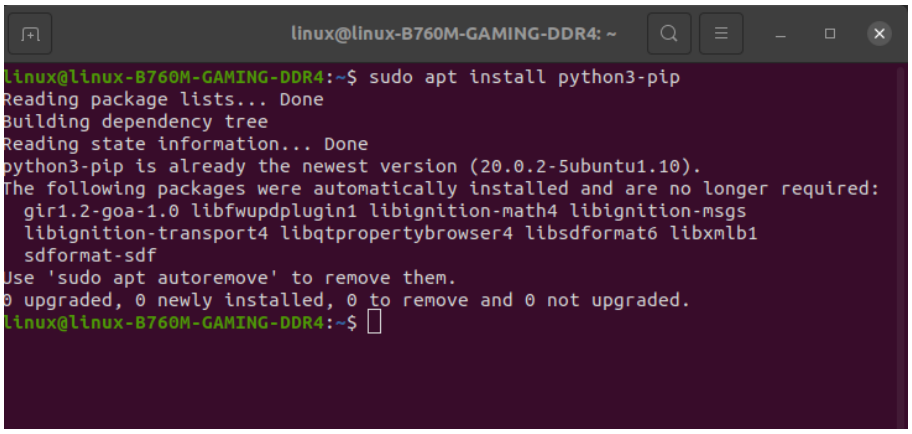

FIGURE 4.8 – Commande d'installation de pip

Ultralytics

est une entreprise et une plateforme dédiée à l'intelligence artificielle (IA), principalement connue pour avoir développé et maintenu la famille des modèles YOLOv5 et plus récemment YOLOv8, des algorithmes puissants pour la détection d'objets en temps réel. Ces modèles sont largement utilisés dans la vision par ordinateur pour des tâches telles que la détection d'objets, la segmentation d'images et la classification.

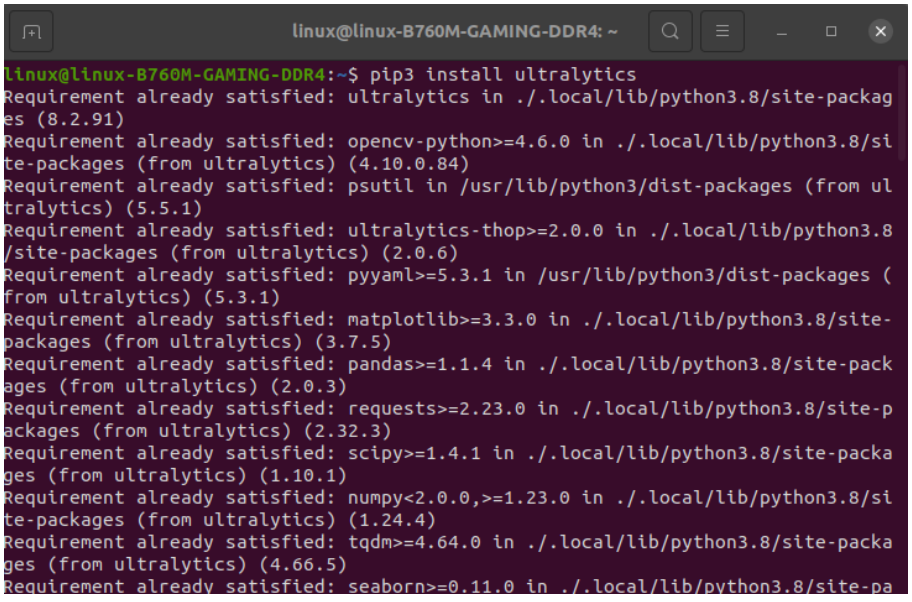

FIGURE 4.9 – Commande d'installation d'ultralytics

4.9.3 Installation du Gazebo

L'installation du gazebo se fait par la commande suivante :

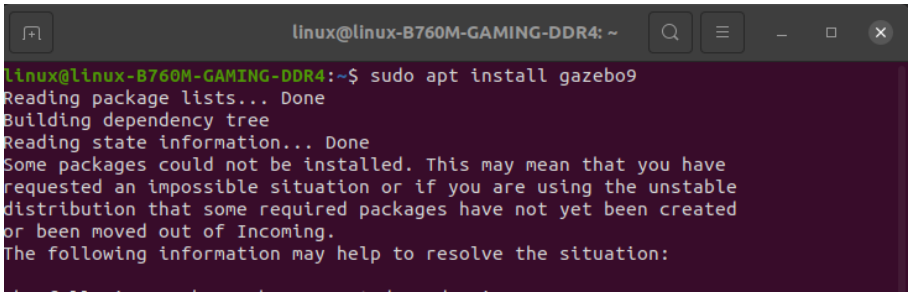

FIGURE 4.10 – Commande d'installation de gazebo

4.9.4 Installation du python3-colcon-common-extensions

Le paquet python 3-colcon-common-extensions est une collection d'extensions pour l'outil Colcon, qui est utilisé pour construire, tester et installer des packages dans des environnements comme ROS2. Colcon est l'outil recommandé pour la gestion des workspaces dans ROS2, le téléchargement et la compilation des "packages" nécessaires.

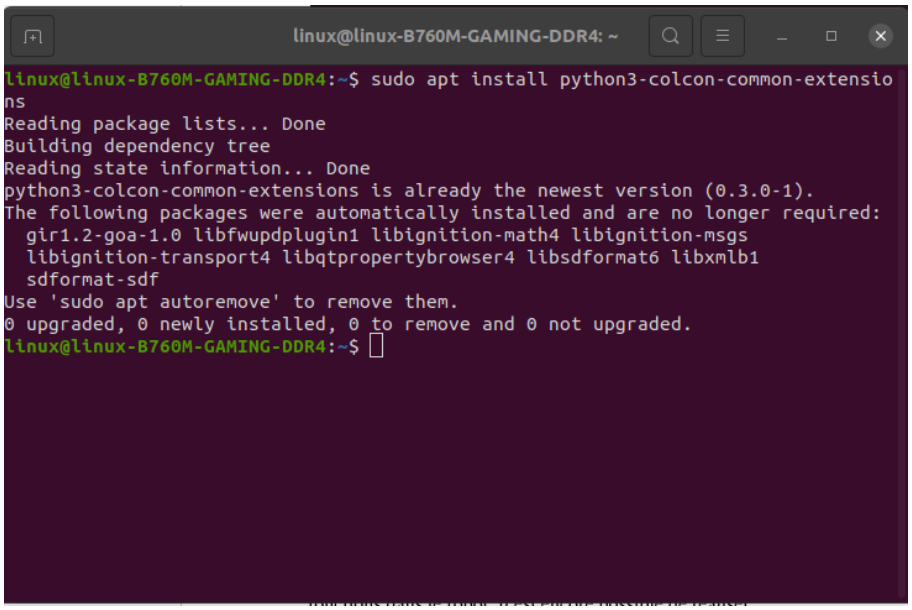

FIGURE 4.11 – Commande d'installation de python3-colcon-common-extensions

4.9.5 L'installation du paquet ros-foxy-gazebo-ros

Le paquet ros-foxy-gazebo-ros est une collection de modules permettant l'intégration entre ROS2 Foxy et le simulateur Gazebo. Il fournit les outils et interfaces nécessaires pour permettre la communication entre les nœuds ROS2 et Gazebo, facilitant ainsi le développement et la simulation de robots dans un environnement virtuel.

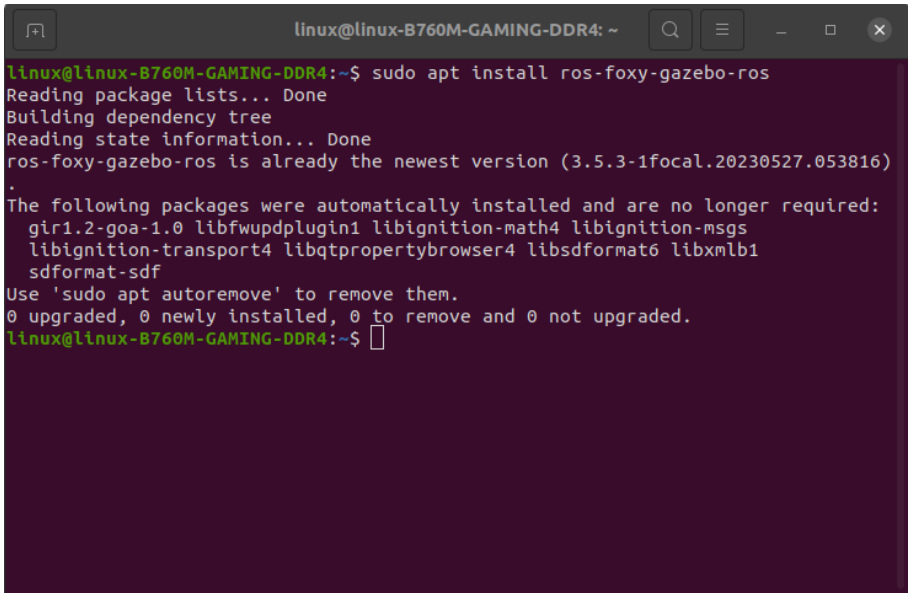

FIGURE 4.12 – Commande d'installation du package ros-foxy-gazebo-ros

4.9.6 L'installation de ros-foxy-gazebo-ros-pkgs

Le paquet ros-foxy-gazebo-ros-pkgs est un méta-paquet dans ROS2 Foxy qui regroupe plusieurs paquets permettant l'intégration entre ROS2 et Gazebo. Il facilite la simulation des robots dans Gazebo tout en interagissant avec ROS2 pour tester et valider des algorithmes robotiques, des systèmes de contrôle et des capteurs dans un environnement simulé.

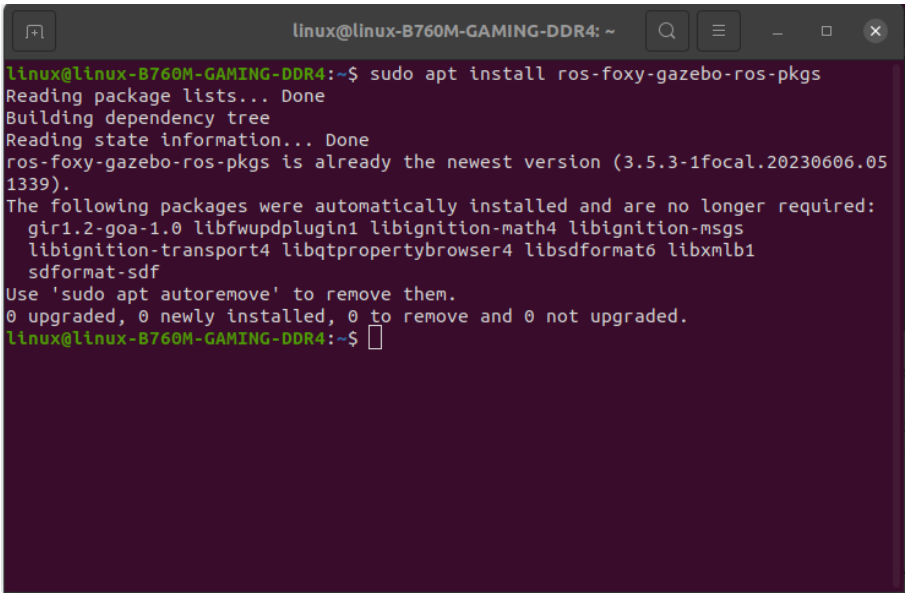

FIGURE 4.13 – Commande d'installation du package ros-foxy-gazebo-ros-pkgs

4.9.7 Construction des packages

La construction des packages se fait avec la commande colcon build qui est utilisée dans ROS2 pour compiler et construire les packages dans un workspace. C'est une étape essentielle pour transformer le code source en exécutables et en bibliothèques que ROS2 peut utiliser.

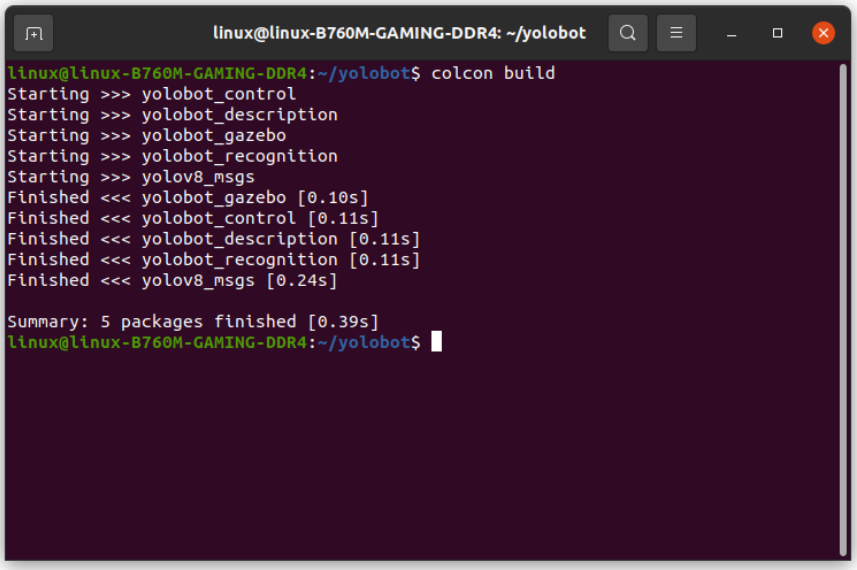

FIGURE 4.14 – Commande de construction des packages

4.9.8 Sourcer le workspace dans ROS2

Sourcer le workspace dans ROS2 est une étape essentielle pour permettre à votre environnement de terminal d'accéder aux packages et ressources compilés dans le workspace. En exécutant la commande source <le nom de workspace>/install/setup.bash, vous configurez votre session de terminal pour qu'elle puisse utiliser les éléments construits dans votre workspace

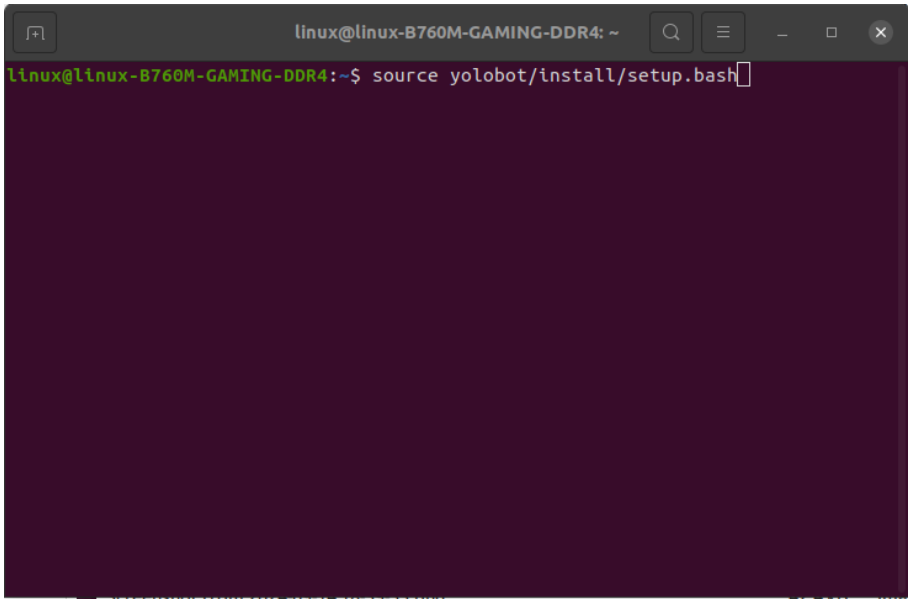

FIGURE 4.15 – Commande de sourçage du workspace

4.9.9 Lancer la simulation

Pour commencer la simulation ,on doit lancer le fichier principale qui exécute les autres fichiers de démarrage pour chaque package en utilisant la commande ros2 launch

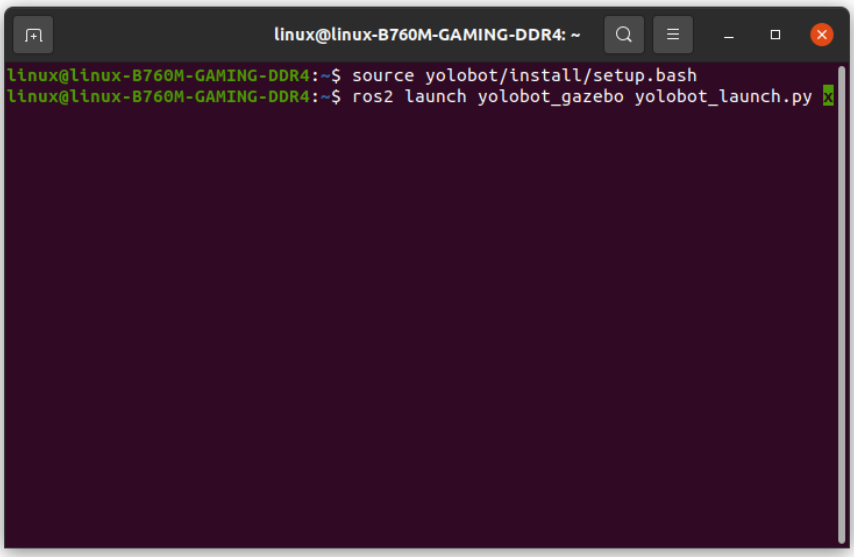

FIGURE 4.16 – Commande de lancement du simulation

4.10 Résultats

Après la finalisation du simulation, les résultats sont les suivants :

1. Pour le panneau de “turn left”

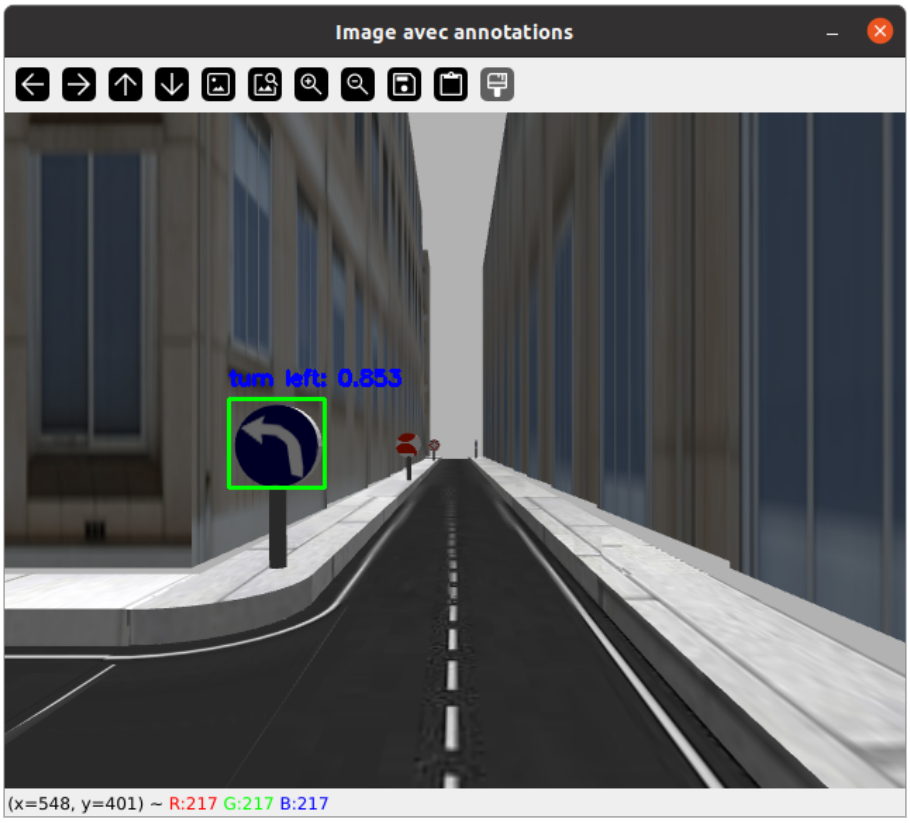

FIGURE 4.17 – Détection du panneau Turn Left

Le robot détecte le panneau “turn left” avec un degré de précision de 0.853 comme le montre la figure au-dessus .

```
[yolov8_ros2_pt.py-6] 0: 480x640 1 turn left, 51.0ms
[yolov8_ros2_pt.py-6] Speed: 0.9ms preprocess, 51.0ms inference, 0.9ms postprocess per image at shape (1, 3, 480, 640)
[robot_control.py-5] [INFO] [1726321479.901371678] [commander]: Le robot avance à 0.1 m/s.
[yolov8_ros2_pt.py-6] [INFO] [1726321479.901572633] [camera_subscriber]: Objet détecté: turn left, Précision: 0.719, Distance: 0.002 m
[robot_control.py-5] [INFO] [1726321479.904558706] [inference_subscriber]: Panneau 'Turn Left' détecté à 0.00 m!
[yolov8_ros2_pt.py-6]
[robot_control.py-5] [INFO] [1726321479.920950273] [commander]: Le robot avance à 0.1 m/s.
[robot_control.py-5] [INFO] [1726321479.921171588] [commander]: Panneau 'Turn Left' détecté, le robot commence à tourner à gauche.
[robot_control.py-5] [INFO] [1726321479.941224709] [commander]: Le robot tourne à gauche avec une vitesse angulaire de -0.1 rad/s.
```

FIGURE 4.18 – Détection du panneau Turn left et l’action approprié dans le terminal

Quand le robot détecte le panneau turn left à la distance est de 0.002m, il tourne à gauche avec une vitesse angulaire de -0.1ras/s comme le montre la figure ci-dessus (la dernière ligne)

2. Pour le panneau de “speed limit 30 ”

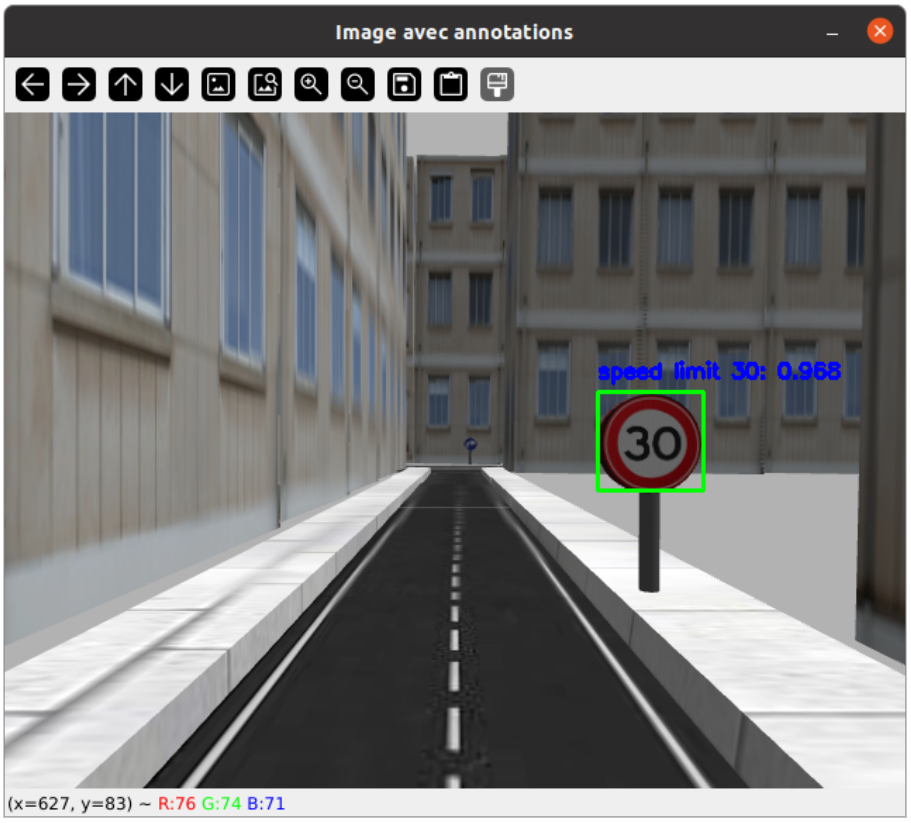

FIGURE 4.19 – Détection du panneau speed limit 30

Le robot détecte le panneau “speed limit 30” avec un degré de précision de 0.968 comme le montre la figure au-dessus .

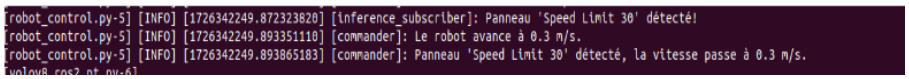

FIGURE 4.20 – Détection du panneau speed limit 30 et l’action approprié dans le terminal

Quand le robot détecte le panneau “speed limit 30”, il ajuste sa vitesse de 0.1m/s vers 0.3m/s comme le montre la figure ci-dessus (la dernière ligne)

3. Pour le panneau de “turn right”

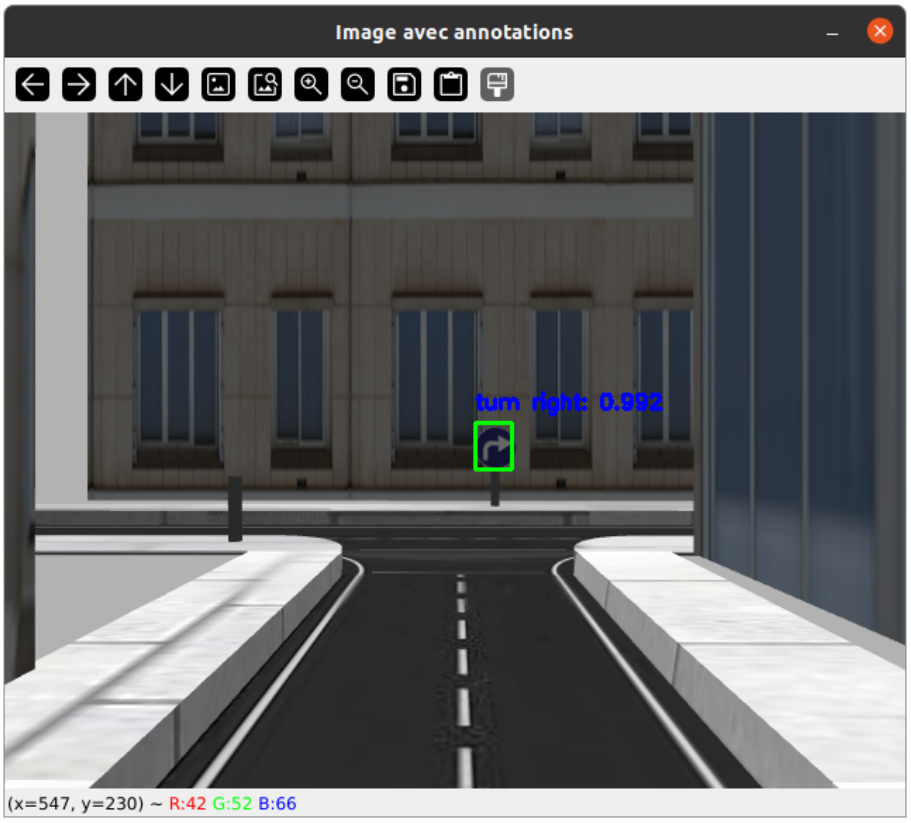

FIGURE 4.21 – Détection du panneau turn right

Le robot détecte le panneau “turn right” avec un degré de précision de 0.981 comme le montre la figure ci-dessus .

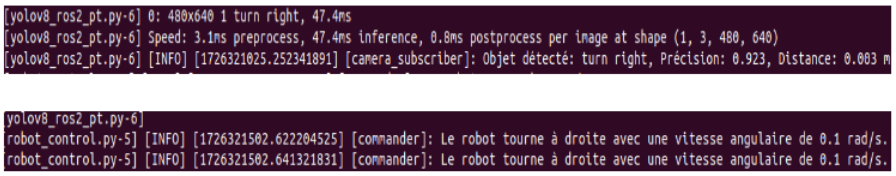

FIGURE 4.22 – Détection du panneau Turn right et l’action appropriée dans le terminal

quand le robot détecte le panneau “turn right” avec une distance est de 0.003 m, il tourne à droite avec une vitesse angulaire de 0.1 rad/s comme le montre la figure ci-dessus (les deux dernières lignes)

4. Pour le panneau de “stop”

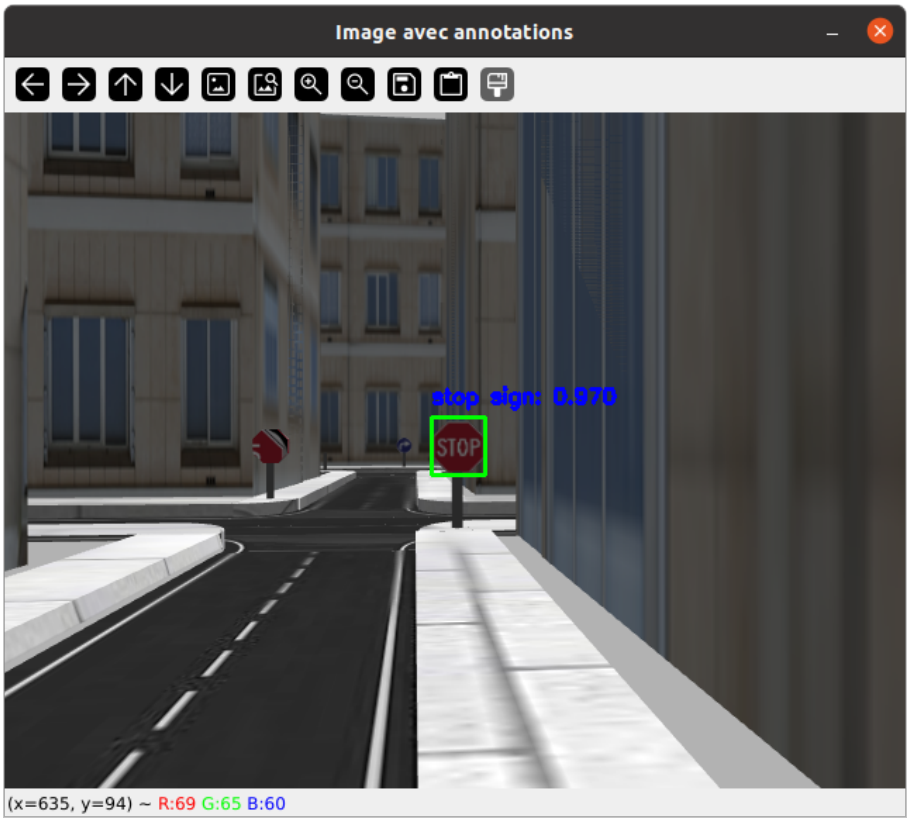

FIGURE 4.23 – Détection du panneau Stop

Le robot détecte le panneau “stop” avec un degré de précision de 0.993 comme le montre la figure ci-dessus.

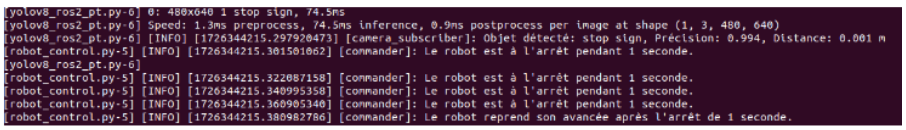

FIGURE 4.24 – Détection du panneau Stop et l’action approprié dans le terminal

Quand le robot détecte le panneau “Stop” avec une distance est de 0.002 m, il s’arrête pendant une seconde puis reprend sa trajectoire comme le montre la figure ci-dessus (la dernière ligne).

5. Pour le panneau de “speed limit 50”

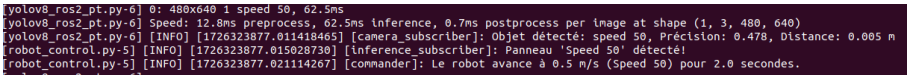

FIGURE 4.26 – Détection du panneau speed 50 et l'action approprié dans le terminal

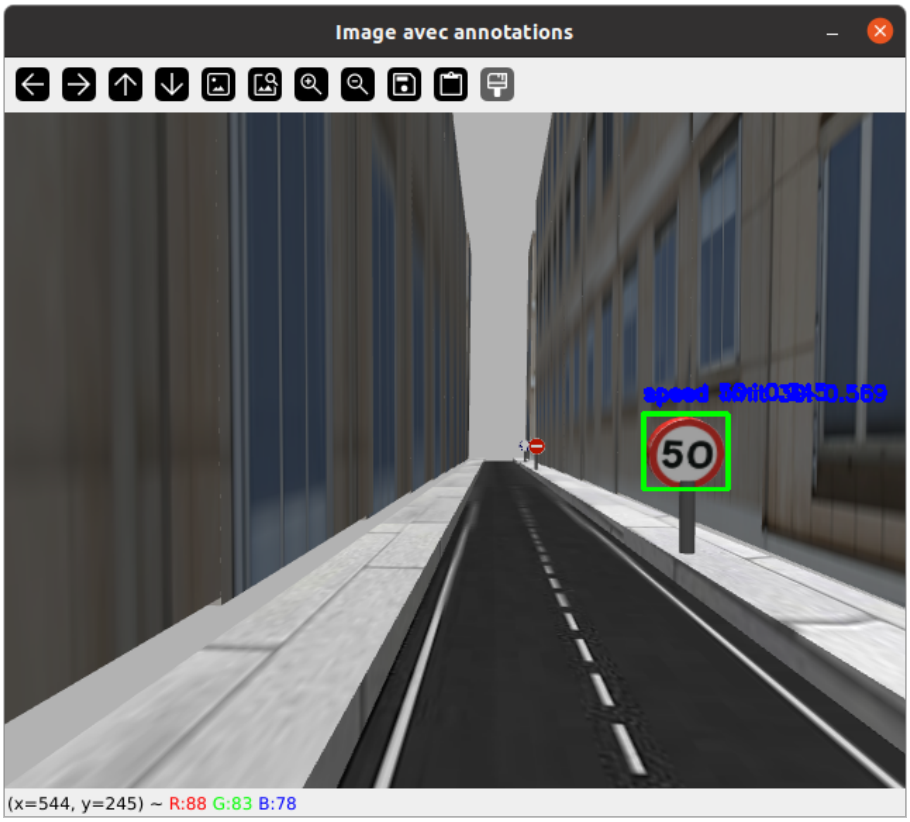

FIGURE 4.25 – Détection du panneau speed limit 50

Le robot détecte le panneau "speed 50" avec un degré de précision de 0.667 comme la figure au-dessus montre Quand le robot détecte le panneau "speed 50", il ajuste sa vitesse de 0.3m/s vers 0.5m/s pendant 2 secondes comme le montre la figure ci-dessus (la dernière ligne)

4.10.1 Interprétation des résultats

Les résultats de la simulation montrent que le robot est capable de détecter avec précision les différents panneaux de signalisation et d'adopter les actions appropriées en fonction de chaque panneau. Les degrés de précision pour la détection varient de 66,7 pourcent à 99,3 pourcent, ce qui démontre une bonne performance du système de reconnaissance des panneaux de signalisation.

4.11 Conclusion

ROS est une plateforme de développement logiciel très répandue dans les applications de la robotique, il est utilisé dans de nombreux robots : drones, voitures autonomes, robots humanoïdes, bras robotisés, et bien d'autres. ROS est un ensemble d'outils dont l'architecture de base permet de réaliser des applications avec une plus grande abstraction, grâce à des outils intégrés à ROS il est possible de faire de la planification de mouvement, reconnaissance d'objets, navigation 2D, cartographie 3D.

Ce chapitre a démontré l'efficacité d'un système de robot mobile capable de détecter des panneaux de signalisation et de réagir en conséquence dans un environnement simulé. Grâce à l'utilisation conjointe de ROS2, Gazebo, et l'algorithme de détection YOLOv8n, nous avons pu recréer des conditions réalistes de circulation routière dans lesquelles le robot interprète et répond aux informations fournies par différents panneaux de signalisation. Les résultats obtenus indiquent que le robot est en mesure de détecter les panneaux avec un degré de précision allant de 66,7 pourcent à 99,3 pourcent, et qu'il ajuste de manière appropriée sa vitesse ou sa direction selon le panneau détecté, confirmant ainsi la performance et la robustesse de l'algorithme de détection et du système de contrôle intégré.

Cette simulation nous a permis aussi de valider les capacités du robot à réagir efficacement dans un environnement contrôlé avant son application dans des scénarios réels. Les tests effectués dans un environnement virtuel, confirme l'affinement des algorithmes de reconnaissance et de contrôle, tout en réduisant les risques et les coûts associés aux tests en conditions réelles.

De plus, l'intégration de ROS2 a offert une infrastructure flexible, évolutive et robuste, garantissant une communication en temps réel et une gestion efficace des actions du robot. Les résultats obtenus sont prometteurs pour une transition future vers des tests en conditions réelles et ouvrent la voie à de nouvelles améliorations pour optimiser encore d'avantage le comportement du robot dans des environnements plus complexes et dynamiques.

Conclusion générale

Notre projet de développement du robot mobile autonome capable de détecter et de réagir aux panneaux de signalisation dans un environnement simulé a démontré des résultats prometteurs. Grâce à l'intégration de ROS2, de Gazebo et de l'algorithme YOLOv8n, le robot a pu reconnaître les différents types de panneaux avec une précision élevée afin d'adapter son comportement. La simulation a joué un rôle essentiel en permettant de tester et d'affiner les algorithmes avant de les appliquer dans des scénarios réels, réduisant ainsi les risques associés aux tests pratiques.

Cette simulation nous a permet d'évaluer les performances du robot dans un environnement virtuel en réduisant ainsi les risques et les coûts liés aux tests pratiques. Les résultats montrent que notre robot est capable de réagir efficacement aux panneaux de signalisation avec un haut degré de précision, démontrant la fiabilité du système de reconnaissance et de navigation.

En conclusion, notre projet ouvre la voie à de nombreuses opportunités pour la recherche et le développement de robots autonomes. En combinant la puissance des simulations avec des systèmes d'intelligence artificielle de plus en plus sophistiqués, les robots pourront bientôt naviguer de manière autonome dans des environnements réels tout aussi complexes que variés.

Perspective

L'avenir de la navigation autonome repose sur la capacité des robots à évoluer dans des environnements de plus en plus imprévisibles et dynamiques. Les améliorations possibles pour ce projet incluent l'intégration de techniques d'intelligence artificielle plus avancées, comme les algorithmes d'apprentissage par renforcement, qui pourraient permettre au robot d'apprendre de ses erreurs et de s'adapter à de nouvelles situations sans intervention humaine.

Cependant, malgré ces résultats encourageants, des défis persistent, notamment en ce qui concerne l'optimisation de la précision de détection et l'amélioration de la robustesse des algorithmes dans des environnements plus complexes. Par exemple, dans des environnements où les conditions d'éclairage ou la densité d'obstacles varient, il sera nécessaire d'adapter le modèle de détection pour maintenir un haut niveau de performance

Une autre perspective intéressante est l'ajout de la cartographie en temps réel avec la localisation simultanée (SLAM). Cela permettrait au robot de créer une carte de son environnement tout en se localisant dans cet espace, améliorant ainsi sa capacité à naviguer dans des environnements non structurés ou inconnus. Enfin, les avancées en matière de communication entre véhicules et infrastructures (V2X) pourraient permettre au robot de recevoir des informations supplémentaires depuis l'environnement (comme des données de trafic en temps réel), augmentant ainsi la sécurité et l'efficacité de la navigation.

Bibliographie

[1] Amazon Web Services. *Qu'est-ce que l'apprentissage par renforcement ?* Accessed: 2024-09-15. 2024. URL : https://aws.amazon.com/fr/what-is/reinforcement-learning/#:~:text=L'apprentissage%20par%20renforcement%20(RL,humains%20pour%20atteindre%20leurs%20objectifs..

[2] Atelier Canopé 95. *50 robots et leurs capteurs*. Accédé le: 16 septembre 2024. 2024. URL : https://atelier-canope-95.canoprof.fr/eleve/Automates%20et%20robots/res/robot.dossierHtml/co/1914_50robotsCapteurs.html.

[3] Blendamator. *Logiciel de modélisation 3D gratuit : Blender*. Accessed: 2024-09-15. 2024. URL : https://blendamator.com/logiciel-de-modelisation-3d-gratuit-blender/.

[4] Blent AI. *Apprentissage supervisé : Définition et explication*. Accessed: 2024-09-15. 2024. URL : https://blent.ai/blog/a/apprentissage-supervise-definiti.

[5] Lakhmissi Cherroun. « Navigation Autonome d'un Robot Mobile par des Techniques Neuro-Floues ». Thèse de doct. Faculté des sciences et de la technologie UMKBiskra, 2014.

[6] DE MATTEIS, Ludovic and Janny, S and Nathan, S and Shu-Quartier, W. *Introduction*
l'apprentissage automatique. 2022.

[7] Fabien Grzeskowiak. « Simulation et expérimentations pour l'évaluation de la navigation de robots dans la foule ». Thèse de doct. Université de Rennes 1 (UR1), 2021.

[8] Imed Eddine Guebli. « Navigation et commande d'un robot mobile à 4 roues sous ROS ». Mém. de mast. XXXX, 2020.

[9] DIB NOUR EL-ISLEM. « Domaine: Mathématique et Informatique ». In : ().

[10] Gilles Mourioux. « Architecture contribuant à l'autonomie d'un robot mobile ». Mém. de mast. xxxx.

[11] Noureddine Slimane. « Systeme de localisation pour robots mobiles ». Thèse de doct. Université de Batna 2, 2008.

[12] Université de Tebessa. *Chapitre 1_ 2 Master 2 Electrotechnique Tech Int 2021*. Accessed: 2024-09-15. 2024. URL : http://e-learning.univ-tebessa.dz/moodle/pluginfile.php/28706/mod_resource/content/1/Chapitre1_2_master2_Electrotechnique_Tech_Int_2021.pdf.

[13] YouTube. *Wheeled mobile robots*. Accessed: 2024-09-15. 2024. URL : https://www.youtube.com/watch?v=k5Er0-HD-qw&list=PLyqSpQzTE6M9CXsZljkH_lCxRSiaXF566.