

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique  
جامعة سعد دحلب البليدة  
Université SAAD DAHLAB de BLIDA  
كلية التكنولوجيا  
Faculté de Technologie  
قسم الأوتوماتيك والهندسة الكهربائية  
Département d'Automatique et d'Electrotechnique



## Mémoire de Master

Filière : Automatique  
Spécialité : Automatique et Informatique Industrielle

*Présenté par :*

CHERIF Mohammed

&

MEKERKEB ABERANE Abdelkader

# Elaboration d'une stratégie de guidage pour un robot mobile d'intérieur dans un environnement mal connu

*Proposé et encadré par :* BENNILA Nour Eddine

*Co-encadré par :* BENSELAMA Zoubir Abdeslam

Année Universitaire 2023-2024

---

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

أَقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿١﴾ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ﴿٢﴾ أَقْرَأْ وَرَبُّكَ الْأَكْرَمُ ﴿٣﴾  
الَّذِي عَلَّمَ بِالْقَلَمِ ﴿٤﴾ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ ﴿٥﴾

---

سورة العلق الآية من 1 الى 5

## Dédicaces

---

*A mes parents, à mes frères et sœurs et à tous ceux qui m'aiment.*

*Mohammed Cherif*

*Je dédie ce travail à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet.*

*À mes parents, pour leur amour inconditionnel, leur soutien indéfectible et leurs encouragements constants. Vous avez toujours cru en moi, même dans les moments où j'avais des doutes, et votre confiance m'a porté jusqu'ici.*

*À mes frères et sœurs, pour leur soutien, leurs conseils, et les moments de partage qui ont été une source de réconfort et de motivation. Votre présence et vos encouragements ont été essentiels dans cette aventure.*

*À mes enseignants et professeurs, pour leur dévouement, leur patience et leur transmission de savoir. Vous avez su éveiller ma curiosité intellectuelle et m'accompagner tout au long de ce parcours académique.*

*À mes amis, pour leur soutien moral, leur écoute et les moments de détente partagés qui m'ont permis de garder le cap. Votre présence a été un véritable moteur dans cette aventure.*

*Enfin, à tous ceux qui, de près ou de loin, ont apporté leur pierre à l'édifice de cette réussite. Je vous en suis infiniment reconnaissant.*

***Abdelkader Mekerkeb Aberane***

## Remerciements

---

*Nous remercions en premier Allah le tout puissant qui a dit :*

*« Et Allah vous a fait sortir des ventres de vos mères, dénués de tout savoir, et vous a donné l'ouïe, les yeux et les cœurs (l'intelligence), afin que vous soyez reconnaissants»*

*(Al-Nahl-verset 78)*

*Nous le remercions de nous avoir donné le courage, la patience et la capacité de terminer ce travail.*

*Nos sincères remerciements et notre entière gratitude à notre encadreur Dr. BENNILA NOUR EDDINE, pour la qualité de son encadrement, ses précieux conseils, sa patience illimitée, sa rigueur professionnelle et son impressionnante disponibilité à nos différentes préoccupations.*

*Grand merci au Dr. BENSELAMA ZOUBIR ABDESLAM pour son Co-encadrement.*

*Nos sincères remerciements à Madame AIT MESSAOUD qui nous a honoré et a accepté de présider ce jury.*

*Merci également à Monsieur MADDI pour son acceptation d'examiner et de juger ce modeste travail.*

.

*Merci à tous ceux qui ont participé de près ou de loin à la réussite de cet humble travail.*

## Résumés

---

### ملخص:

في مجال الروبوتات، يمثل توجيه الروبوت تحديًا كبيرًا للملاحة الذاتية، خاصة في البيئات الداخلية غير المعروفة. تركز هذه الأطروحة على تطوير استراتيجيات التوجيه من خلال تصميم ومحاكاة نظام توجيه الروبوت باستخدام تقنية LiDAR ومنصات برمجية ROS و Gazebo. لقد اخترنا LiDAR لأدائه في إدراك البيئة، بالإضافة إلى ROS الذي يوفر نظامًا قويًا لإدارة الروبوت والتحكم فيه. لاختبار خوارزميات التنقل استخدمنا جهاز محاكاة ثلاثي الأبعاد Gazebo. ويهدف هذا المشروع بشكل أساسي إلى تزويد الروبوت بذكاء حركي يسمح له باتخاذ قرارات ديناميكية وقابلة للتكيف في مواجهة العوائق والتكوينات المكانية المتغيرة.

*LiDAR ; ROS ; Gazebo*

*الكلمات المفتاحية: توجيه الروبوت؛*

### Résumé :

Dans le domaine de la robotique, le guidage de robots représente un défi majeur pour la navigation autonome, particulièrement dans des environnements intérieurs mal connus. Ce mémoire se concentre sur l'élaboration d'une stratégie de guidage par la conception et la simulation d'un système de guidage de robot utilisant la technologie LiDAR et les plateformes logicielles ROS et Gazebo.

Nous avons choisi le LiDAR pour sa performance dans la perception de l'environnement, ainsi que ROS qui fournit un système robuste de gestion et de contrôle de robots. Pour tester les algorithmes de navigation nous avons utilisé le simulateur 3D Gazebo.

Ce projet vise principalement à doter le robot d'une intelligence de déplacement lui permettant de prendre des décisions dynamiques et adaptatives face aux obstacles et aux configurations spatiales changeantes.

**Mots clés :** *Guidage de robots ; LiDAR ; ROS ; Gazebo.*

### Abstract :

In the field of robotics, robot guidance represents a major challenge for autonomous navigation, particularly in poorly known indoor environments. This dissertation focuses on the development of a guidance strategy through the design and simulation of a robot guidance system using LiDAR technology and the ROS and Gazebo software platforms.

We chose LiDAR for its performance in environment perception, as well as ROS which provides a robust robot management and control system. To test the navigation algorithms we used the 3D Gazebo simulator.

This project mainly aims to provide the robot with movement intelligence allowing it to make dynamic and adaptive decisions in the face of obstacles and changing spatial configurations.

**Keywords:** *Robot guidance; LiDAR; ROS; Gazebo.*

---

# Table des matières

---

Dédicaces	i
Remerciements	iii
Résumés	iv
Table des matières	v
Liste des figures	viii
Liste des tableaux	ix
Liste des acronymes et abréviations	x
<b>Introduction générale</b>	<b>01</b>
<i>Chapitre I : Généralités</i>	
<b>I.1 Introduction à la robotique</b>	<b>03</b>
<b>I.1.1 Définition de la robotique</b>	<b>03</b>
<b>I.1.2 Historique de la robotique</b>	<b>03</b>
<b>I.1.3 Avancées récentes et tendances futures</b>	<b>05</b>
<b>I.1.3.1 Avancées récentes</b>	<b>05</b>
<b>I.1.3.2 Tendances futures</b>	<b>05</b>
<b>I.2 La navigation autonome</b>	<b>06</b>
<b>I.2.1 La robotique mobile</b>	<b>07</b>
<b>I.2.1.1 Définition</b>	<b>07</b>
<b>I.2.1.2 Types de déplacement</b>	<b>08</b>
<b>I.2.2 Principe de la navigation autonome</b>	<b>09</b>
<b>I.3 La technologie LiDAR</b>	<b>11</b>
<b>I.3.1 Principe de fonctionnement</b>	<b>11</b>
<b>I.3.2 Applications du LiDAR</b>	<b>12</b>
<b>I.3.2.1 Applications en navigation des robots</b>	<b>12</b>
<b>I.3.3 Avantages et limitations du LiDAR</b>	<b>12</b>
<b>I.4 Conclusion du chapitre</b>	<b>13</b>
<i>Chapitre II : Guidage de robots</i>	
<b>II.1 Introduction</b>	<b>14</b>
<b>II.1.1 Définition</b>	<b>14</b>
<b>II.2 Techniques de guidage de robots</b>	<b>14</b>
<b>II.2.1 Différentes techniques de guidage</b>	<b>15</b>

II.2.2	Méthodes d'évitement d'obstacles	16
II.2.3	Avantages et des limitations de chaque méthode	18
II.3	Robot Operating System (ROS) et simulateur Gazebo	19
II.3.1	Introduction à ROS	19
II.3.1.1	Architecture de ROS	20
II.3.1.2	Historique	21
II.3.1.3	Importance de ROS en robotique	22
II.3.2	Gazebo	23
II.3.2.1	Architecture Gazebo	23
II.3.2.2	Historique	24
II.3.2.3	Importance de Gazebo en robotique	25
II.3.3	Intégration de ROS – Gazebo	26
II.4	Conclusion du chapitre	27
 <i>Chapitre III : Conception et simulation</i>		
III.1	Introduction	28
III.2	Méthode de navigation	28
III.2.1	Contrôle des parois (Monitoring)	29
III.2.1.1	Contrôle des mouvements	29
III.2.1.2	Évitement des obstacles	30
III.2.1.3	Adaptation de la trajectoire	30
III.3	Méthode d'évitement d'obstacles	30
III.3.1	Intégration du compteur	31
III.3.1.1	Technique de fonctionnement	31
III.3.1.2	Rôle du compteur dans notre système de guidage	32
III.4	Architecture du système	33
III.4.1	Modèle de robot	33
III.4.1.1	Présentation matériel (Hardware)	33
III.4.1.2	Présentation logiciel (Software)	34
III.4.2	LiDAR 2D	35
III.4.3	Compteur	35
III.4.4	Communication	35
III.4.5	Interfaces	36
III.5	Logiciel de guidage	36



III.5.1	Installation des packages	37
III.5.2	Configuration du système	38
III.5.3	Développement des algorithmes de guidage	38
III.5.3.1	Algorithme du contrôle de paroi	38
III.5.3.2	Compteurs d'obstacles	40
III.5.4	Fonctionnement prévu	41
III.5.4.1	Contrôle de paroi	42
III.5.4.2	Comptage des obstacles	42
III.5.4.3	Evitement d'obstacles	43
III.6	Simulation	44
III.6.1	Environnement du guidage	44
III.6.2	Scénarios de simulation	44
III.6.2.1	Guidage en contrôle de parois	44
III.6.2.2	Test du compteur	46
III.6.2.3	Guidage avec exploration de l'environnement	47
III.6.3	Résultats et discussion	47
III.7	Conclusion du chapitre	48
	Conclusion générale	49
	Références bibliographiques	52

## Liste des figures

---

Figure I.1 : <i>Processus de navigation autonome</i>	06
Figure I.2 : <i>Interactions d'un robot mobile avec son environnement</i>	07
Figure I.3 : <i>Exemples de robots mobiles avec différents modes de déplacement</i>	09
Figure I.4 : <i>Principe de la méthode SLAM</i>	09
Figure I.5 : <i>Schéma de fonctionnement d'un système LiDAR</i>	12
Figure II.1 : <i>Exemple de fusion de capteurs de l'avion Eurofighter</i>	15
Figure II.2 : <i>Protocole de communication ROS</i>	20
Figure III.1 : <i>Capteurs et variables d'entrées en contrôle de paroi</i>	29
Figure III.2 : <i>Principe d'incrémentatation des obstacles</i>	31
Figure III.3 : <i>Rôle du compteur dans la planification de la trajectoire</i>	32
Figure III.4 : <i>Aperçu de TurtleBot3 Burger</i>	34
Figure III.5 : <i>Architecture Software de TurtleBot</i>	34
Figure III.6 : <i>Diagramme de communication des nœuds ROS</i>	35
Figure III.7 : <i>Architecture du système de guidage</i>	36
Figure III.8 : <i>Installation de ROS Noetic</i>	37
Figure III.9 : <i>Installation de Python3</i>	37
Figure III.10 : <i>Programmation du contrôle de parois</i>	39
Figure III.11 : <i>Schéma du diagramme contrôle de parois</i>	39
Figure III.12 : <i>Programmation du compteur d'obstacles</i>	40
Figure III.13 : <i>Exécution du compteur dans ROS</i>	40
Figure III.14 : <i>Schéma de l'algorithme compteur</i>	41
Figure III.15 : <i>Navigation prévu en contrôle de paroi</i>	42
Figure III.16 : <i>Fonctionnement prévu pour le compteur d'obstacles</i>	43
Figure III.17 : <i>Comportement prévu en cas d'obstacle</i>	43
Figure III.18 : <i>Croquette de vue-dessus du labo 201 PAV 26</i>	44
Figure III.19 : <i>Comportement du RMA en mode contrôle de paroi</i>	45
Figure III.20 : <i>Comportement du RMA en cas d'obstacles connus</i>	45
Figure III.21 : <i>Simulation test du compteur d'obstacle</i>	46
Figure III.22 : <i>Scénario avec un seul obstacle</i>	47
Figure III.23 : <i>Scénario avec plusieurs obstacles</i>	47

## Liste des tableaux

---

<b>Tableau I.1 : Jalons importants de l'histoire de la robotique</b>	<b>04</b>
<b>Tableau I.2 : Évolution de la technologie LiDAR</b>	<b>11</b>
<b>Tableau I.3 : Avantages et limitations du LiDAR</b>	<b>13</b>
<b>Tableau II.1 : Avantages et limitations des méthodes d'évitement d'obstacles</b>	<b>19</b>
<b>Tableau II.2 : Historique des releases de ROS 2 - REP 2000</b>	<b>21</b>
<b>Tableau II.3 : Historique de ROS et des releases – REP3</b>	<b>22</b>
<b>Tableau II.4 : Historique des développements de Gazebo</b>	<b>25</b>

## Listes des acronymes et abréviations

---

<b><i>AII</i></b>	Automatique et Informatique Industrielle
<b><i>AGV</i></b>	Automatic Guided Vehicle
<b><i>APF</i></b>	Artificial Potential Field
<b><i>API</i></b>	Application Program Interface
<b><i>DIAL</i></b>	Differential Absorption LiDAR
<b><i>DWA</i></b>	Dynamic Window Approach
<b><i>GPS</i></b>	Global Positioning System
<b><i>GUI</i></b>	Graphical User Interface
<b><i>GVD</i></b>	Generalized Voronoi Diagram
<b><i>IA</i></b>	Intelligence Artificielle
<b><i>IMU</i></b>	Inertial Measurement Units
<b><i>IoT</i></b>	Internet of Things
<b><i>LiDAR</i></b>	Light Detection and Ranging
<b><i>LTS</i></b>	Long Term Support
<b><i>OGRE</i></b>	Object-Oriented Graphics Rendering Engine
<b><i>OSRF</i></b>	Open Source Robotics Foundation
<b><i>RPC</i></b>	Remote Procedure Call
<b><i>PRM</i></b>	Probabilistic RoadMaps
<b><i>REP</i></b>	ROS Enhancement Proposal
<b><i>ROS</i></b>	Robot Operating System
<b><i>RPi</i></b>	Raspberry Pi
<b><i>RMA</i></b>	Robot Mobile Autonome
<b><i>RRT</i></b>	Rapidly-exploring Random Trees
<b><i>SLAM</i></b>	Simultaneous Localization And Mapping
<b><i>ToF</i></b>	Time-of-Flight
<b><i>URDF</i></b>	Unified Robot Description Format
<b><i>UWB</i></b>	Ultra-Wideband

# ***Introduction générale***

---

La robotique est un domaine en constante évolution, offrant des opportunités sans précédent pour l'automatisation et l'autonomie dans divers secteurs de la fabrication à l'exploitation. Au cœur de cette croissante évolution se trouve le défi complexe du guidage des robots, essentiel pour leur navigation autonome dans des environnements complexes et variés.

Pour répondre à cette demande croissante d'autonomie et motivé par la nécessité de développer des systèmes de guidage de robots précis et fiables, notre projet se concentre sur la conception et la simulation d'un système de guidage de robot mobile capable de naviguer efficacement dans des environnements intérieurs mal connus. L'objectif est de doter le robot d'une intelligence de déplacement lui permettant de prendre des décisions dynamiques et adaptatives en fonction des obstacles entrevus et des configurations spatiales changeantes.

Pour atteindre cet objectif, nous avons choisi d'utiliser la technologie LiDAR, comme méthode de télédétection pour la perception de l'environnement avec une précision remarquable. ROS (Robot Operating System) sera notre plateforme logicielle principale, fournissant un cadre robuste pour la gestion et le contrôle du robot, tandis que le simulateur 3D Gazebo nous permettra de tester et valider nos algorithmes de navigation dans un environnement virtuel réaliste.

En plus de développer un système de guidage de robots capable de se déplacer en toute sécurité et d'éviter les obstacles de manière autonome, notre but est d'appliquer les connaissances acquises en AII (Automatique et Informatique Industrielle) au cours de notre formation universitaire. Notre projet vise à proposer une solution pratique et innovante pour surmonter les défis du guidage autonome avec les objectifs suivants :

- Concevoir un système de guidage de robot basé sur la technologie LiDAR, reconnue pour sa capacité à fournir des données de perception précises et en temps réel ;
- Utiliser la plateforme logicielle ROS qui offre un ensemble d'outils puissants et une communauté active pour la conception, le contrôle et la simulation de robots ;
- Implémenter des algorithmes de navigation autonomes efficaces qui permettent au robot de prendre des décisions en temps réel en fonction des données LiDAR et des informations environnementales fournies par ROS ;
- Valider et tester le système de guidage de robot dans un environnement simulé avec le simulateur 3D Gazebo, permettant une évaluation approfondie des performances et des fonctionnalités du système dans divers scénarios ;

- Analyser les résultats obtenus et discuter des implications pratiques de la mise en œuvre de notre système de guidage.

Pour atteindre ces objectifs nous avons organisé notre mémoire selon trois chapitres, chacun étant dédié à une étape clé de la conception de notre système de guidage :

### **Chapitre I : Généralités**

Dans ce chapitre, nous procéderons à une introduction à la robotique mobile, explorant son évolution, ses avancées actuelles et ses tendances futures. Ensuite, nous ferons une revue exhaustive de la littérature sur la navigation autonome, incluant ses principes et les différents modes de déplacement des robots mobiles. Enfin, nous explorerons la technologie LiDAR dans ce même contexte de navigation autonome.

### **Chapitre II : Guidage de robots**

En cette partie, nous allons explorer en profondeur le guidage des robots, en mettant en lumière ses différents types et techniques, ainsi que les méthodes d'évitement d'obstacles. De plus, nous examinerons l'histoire, l'architecture et l'importance de ROS et Gazebo dans le domaine de la robotique. Nous discuterons également de l'intégration de ROS et Gazebo, en soulignant les avantages et les applications de cette combinaison puissante pour le développement, la simulation et le contrôle de robots.

### **Chapitre III : Conception et simulation du système de guidage de robot**

Sous cette section, nous détaillerons la conception de notre système de guidage, où nous discuterons des spécifications du système, du choix de la méthode utilisée pour éviter les obstacles, de son architecture, de l'algorithme de navigation que nous avons développé, ainsi que des composants matériels et logiciels utilisés. Cette partie mettra en lumière notre approche méthodique pour concevoir un système de guidage efficace.

Ensuite, nous présenterons la configuration de l'environnement de simulation que nous avons utilisé pour tester notre système de guidage de robot. Nous décrirons les scénarios de test que nous avons développés, la méthodologie de simulation que nous avons suivie, ainsi que l'analyse des résultats obtenus. Nous discuterons des performances du système, de sa comparaison avec nos attentes et des limitations identifiées, pour mettre en lumière d'éventuelles pistes d'amélioration qui optimisent davantage le système proposé.

En développant ce système de guidage de RMA qui combine les dernières avancées technologiques en matière de télédétection, de logiciels et de simulation, nous espérons que notre humble travail apportera une contribution même minime au domaine de la robotique.

# Chapitre I : Généralités

---

## I.1 Introduction à la robotique

La robotique est une science en constante évolution, qui englobe une grande diversité de robots, des automates et des humanoïdes capables de s'adapter à leur environnement et de prendre des décisions en conséquence [1]. Considérée comme la révolution technologique du XXIe siècle, les robots ne seront plus de simples machines, mais des compagnons intimes, capables d'apprendre et d'adapter leur comportement en fonction de leurs expériences [2].

Cette évolution promet de transformer profondément notre quotidien, avec l'émergence de robots omniprésents capables d'accomplir des tâches ingrates, précises et souvent irréalisables par les humains, ce qui pose de nombreux défis en raison de son impact croissant sur la société [3]. Parmi ces défis majeurs, nous trouvons la navigation autonome et la perception de l'environnement, ce qui souligne l'importance cruciale de cette revue de littérature pour la compréhension et le développement de notre système de guidage autonome.

### I.1.1 Définition de la robotique

La robotique est l'étude de toutes les technologies qui permettent la conception et la réalisation de machines automatiques ou de robots [4] qui sont des systèmes capables d'effectuer des tâches de manière autonome ou avec une intervention humaine minimale. Le domaine englobe diverses disciplines et compétences d'ingénierie, intégrant l'ingénierie mécanique, électrique et informatique pour développer des machines capables de reproduire ou d'assister les actions humaines [5].

### I.1.2 Historique de la robotique

Les ancêtres des robots sont les automates. Un automate très évolué fut présenté par Jacques de Vaucanson en 1738 [4]. Le terme « robotique » a été introduit pour la première fois par l'écrivain de science-fiction Isaac Asimov en 1942 [6], et le concept de robot qui tient son origine du mot slave, *Robota* qui signifie corvée a été popularisé par le dramaturge tchèque Karel Capek en 1921 dans sa pièce théâtrale : *Rossum's Universal Robots*. Depuis lors, les robots ont été utilisés dans divers domaines technologique [7].

Les premiers robots modernes ont été créés dans les années 1960, et depuis la robotique a pris son essor avec l'avènement de l'informatique et de la théorie du contrôle ce qui a permet

de diversifier les applications allant de l'automatisation industrielle à la chirurgie assistée par robot [8].

L'évolution de la robotique a été influencée par des concepts tels que la téléprésence, la réalité virtuelle et l'autonomie robotique, qui ont permis aux robots de fonctionner avec peu ou pas d'intervention humaine [9]. De plus, l'avènement de l'informatique ubiquitaire a donné naissance à la robotique ubiquitaire, où les services et interfaces fournis par les robots ne sont pas limités spatialement [10].

Année	Événement	Source
1738	Automate de Vaucanson	Heudin JC. (2008) <i>Les Créatures artificielles : des automates aux mondes virtuels</i> , Paris, Page 56.
1942	Terme "robotique" introduit par Isaac Asimov	Pickover, C. (2021). Les trois lois de la robotique d'Asimov. Dans : <i>La fabuleuse histoire de l'intelligence artificielle</i> (pp. 116-117). Paris : Dunod.
1956	Invention du 1 <sup>er</sup> bras manipulateur programmable	Gasparetto, A. and Scalera, L. (2019) A Brief History of Industrial Robotics in the 20th Century. <i>Advances in Historical Studies</i> , <b>8</b> , 24-35. Doi : 10.4236/ahs.2019.81002.
1961	Installation du 1 <sup>er</sup> robot industriel sur une chaîne de montage de General Motors	Braik M. et Soualem K. (2018). Conception et réalisation d'un robot <i>industriel</i> . Mémoire de master en AII. Université Ibn Khaldoun Tiaret. Page 12.
1980	Introduction des robots visionnaires	Lazaric Nathalie. (1992). Origines et développement de la robotique. <i>Revue d'économie industrielle</i> . DOI: 10.3406/rei.1992.1436
1990	Commercialisation des plates-formes pour le développement de la robotique mobile.	David Filliat. (2012) . Robotique mobile. École Nationale Supérieure de Techniques Avancées ParisTech. Page 12.
2000	Introduction d'Asimo, le robot humanoïde de Honda	Kupperberg Paul. (2007). <i>Careers In Robotics</i> . New York, Rosen Pub. ISBN 1427091250, 9781427091253
2000	Introduction des robots chirurgicaux comme le système de Vinci	Fatima Sakina. (2022). <a href="https://www.siasat.com/da-vinci-xi-surgical-robot-perform-kidney-surgery-in-dubai-2330469/">https://www.siasat.com/da-vinci-xi-surgical-robot-perform-kidney-surgery-in-dubai-2330469/</a>
2000	Application de l'intelligence artificielle (AI) dans la robotique	Asma Aimeur. <a href="https://www.iro.umontreal.ca/~aimeur/cours/ift6261/Ch1-Intro-IA-IFT6261-H-11.pdf">https://www.iro.umontreal.ca/~aimeur/cours/ift6261/Ch1-Intro-IA-IFT6261-H-11.pdf</a>
2010	Développement des robots collaboratifs	Hand <i>Sophie</i> . (2020). A brief history of collaborative robots. <a href="https://www.mhlnews.com/technology-automation/article/21124077/a-brief-history-of-collaborative-robots">https://www.mhlnews.com/technology-automation/article/21124077/a-brief-history-of-collaborative-robots</a>

**Tableau I.1 Jalons importants de l'histoire de la robotique**



### **I.1.3 Avancées récentes et tendances futures**

#### **I.1.3.1 Avancées récentes [11]**

Les avancées technologiques, tels que l'intégration de l'intelligence artificielle (IA) et de l'apprentissage automatique ont révolutionné les avancées de la robotique grâce à deux branches novatrices :

- **Robotique intelligente** : Les robots peuvent désormais percevoir et interpréter leur environnement, prendre des décisions éclairées et apprendre de leurs expériences ;
- **Robotique collaboratif (Cobots)** : Les progrès des technologies de détection et des algorithmes d'IA ont conduit au développement de robots collaboratifs, qui peuvent travailler aux côtés des humains, partageant même espace de travail et collaborer sur des tâches ;

Ces avancées ont permis de propulser la technologie robotique vers de nouveaux horizons d'efficacité et d'intégration dans divers domaines, dont la robotique mobile, médicale et de service.

#### **I.1.3.2 Tendances futures [11]**

L'avenir de la robotique est prometteur et riche en possibilités alors que les avancées technologiques se poursuivent. Voici quelques-unes des tendances majeures qui sont susceptibles de façonner l'avenir de la robotique.

- **Intégration de l'IA** : Avec des algorithmes d'IA de plus en plus sophistiqués les robots seront plus adaptables, intelligents et capables d'interagir de manière transparente avec les humains ;
- **Robots humanoïdes et robotique sociale** : Les robots humanoïdes auront des capacités de communication pour des tâches en environnements sociaux, permettant leur intégration dans les foyers intelligents. Ils combineront robotique, IA et Internet des objets (IoT) en tant qu'assistants personnels ;
- **Robotique en essaim** : Les robots en essaim collaboreront de manière coordonnée, démontrant une intelligence collective pour résoudre des problèmes complexes ;
- **Robotique douce** : Ces robots aux structures flexibles, seront conçus pour naviguer dans des environnements complexes et non structurés, interagir en toute sécurité avec les humains et manipuler des objets délicats.

- **Considérations éthiques et réglementation :** Des questions telles que les droits des robots, la confidentialité et la sécurité nécessiteront une attention particulière. Des cadres réglementaires seront mis en place pour un déploiement responsable et pour traiter les risques potentiels et les implications sociétales.

En conclusion, la synergie entre l'intelligence artificielle (IA) et la robotique n'est pas nouvelle, mais la collaboration interdisciplinaire, l'adaptabilité des robots, l'apprentissage automatique et les réseaux de neurones, suggèrent que nous sommes à la veille d'une transformation majeure, capable de redéfinir les paradigmes de l'automatisation, de l'efficacité et de l'innovation industrielle [12].

## I.2 La navigation autonome

La navigation autonome d'un robot mobile consiste à planifier un mouvement sans collision à travers l'espace de configuration, représentant la position et l'orientation du robot dans son environnement. Le robot évolue de sa configuration initiale vers une configuration finale désirée, en évitant les obstacles proches. Ce processus implique la mesure des distances aux obstacles et des positions actuelle et finale, la détermination des variables de commande appropriées (vitesse et angle de braquage), l'exécution des actions de déplacement, et la répétition itérative de ce processus jusqu'à atteindre la destination finale.

Le schéma suivant (figure I.1) présente une description de la tâche de navigation d'un robot mobile qui 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  $a(t)$ ) lui permettant de se mouvoir vers une nouvelle situation  $s(t+1)$  [13].

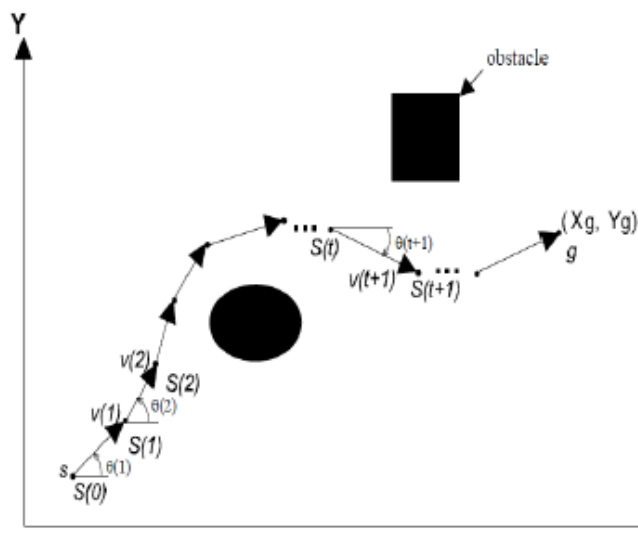


Figure I.1 Processus de navigation autonome ; Source : [13].

## I.2.1 La robotique mobile

### I.2.1.1 Définition

La robotique mobile est un domaine d'étude et de développement impliquant des systèmes mécaniques, électroniques et informatiques capables d'agir physiquement sur leur environnement pour atteindre des objectifs assignés. Ces robots sont polyvalents et peuvent s'adapter à des changements dans leurs conditions de travail.

Ils intègrent des fonctions de perception, de prise de décision et d'action (Figure I.2), ce qui leur permet d'accomplir diverses tâches de manière autonome, même dans des situations imprévues. Les robots mobiles sont équipés de dispositifs de déplacement dans leur espace de travail et peuvent être dotés de moyens de perception et de raisonnement selon leur degré d'autonomie ou d'intelligence. Ils sont souvent utilisés dans des environnements variés ou incertains, pleins d'obstacles, où ils doivent s'adapter aux conditions changeantes et accomplir des tâches spécifiques [4].

La nouvelle génération de systèmes de robots mobiles autonomes (RMA) haute performance est caractérisée par leur fonctionnement totalement autonome. Ces robots, ainsi que les véhicules à guidage automatique (AGV)\*, représentent un niveau avancé de collaboration l'homme-machine. Ils sont capables d'effectuer des tâches de manière entièrement automatisée ou de collaborer avec les humains si nécessaire. L'objectif de ces systèmes est de fournir des solutions flexibles et adaptables pour tous les secteurs d'activité [14].

*\* Différence entre RMA et AGV : Les AGV suivent des itinéraires fixes à l'intérieur des installations, tandis que les RMA naviguent de manière autonome en s'adaptant à des environnements dynamiques. Les AGV nécessitent des guides physiques, tandis que les RMA utilisent des capteurs et des algorithmes avancés. Les AGV sont moins flexibles et adaptatifs que les RMA, conçus pour des tâches répétitives le long de routes fixes.*

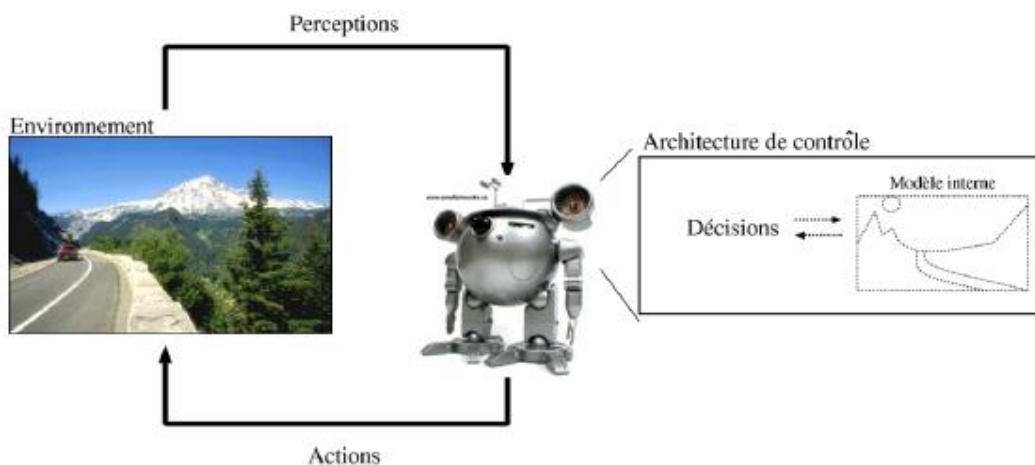


Figure I.2 Interactions d'un robot mobile avec son environnement. ; Source : [15]

### I.2.1.2 Types de déplacement

La caractéristique la plus distinctive d'un robot mobile réside sans aucun doute dans son mode de déplacement. Celui-ci est directement influencé par le contexte d'utilisation envisagé ainsi que par les caractéristiques du terrain sur lequel le robot évoluera (intérieur, extérieur, encombré d'obstacles, etc.). Les robots mobiles sont généralement classés en fonction du mode de locomotion utilisé selon deux types de contraintes :

- ✓ Les contraintes cinématiques, qui concernent la manière dont le robot peut se déplacer géométriquement dans son environnement ;
- ✓ Les contraintes dynamiques, qui découlent des effets du mouvement tels que les accélérations, les vitesses limitées, la présence de forces d'inertie ou de frottement. Ces facteurs ont une incidence sur la façon dont le mouvement est effectué.

En termes de cinématique, un robot est qualifié d'holonome s'il est capable de se déplacer instantanément dans toutes les directions possibles. Il est qualifié de non holonome si ses mouvements autorisés sont limités à des courbes dont la courbure est bornée [4]. Les contraintes cinématiques et dynamiques influent sur le mouvement du robot, notamment en limitant la géométrie et les effets du mouvement.

Voici les quatre moyens de déplacement (Figure I.3) :

- **Robots mobiles à roues** : Ce sont les plus répandus. Ils assurent un déplacement rapide sur des surfaces relativement planes, mais ont des difficultés à franchir des obstacles ;
- **Robots mobiles à chenilles** : Ils offrent une meilleure adhérence au sol et peuvent franchir des obstacles plus importants que les robots à roues ;
- **Robots mobiles à pattes** : Conçus pour se déplacer sur des terrains très accidentés. Cependant, leur conception et leur ;
- **Robots mobiles avec autres mode de locomotion** : Cette catégorie englobe les robots mobiles utilisant un moyen de locomotion différent des trois précédents. Par exemple, les robots bipèdes, monopèdes, marcheurs, sous-marins, volants entre autres.



Figure I.3 Exemples de robots mobiles avec différents modes de déplacement. ; Source : [15]

### I.2.2 Principe de la navigation autonome

Le principe de la navigation autonome repose sur la capacité d'un robot mobile à se déplacer de manière autonome dans un environnement donné en évitant les obstacles et en atteignant des destinations spécifiques. Voici quelques principes clés de la navigation autonome d'après Cédric Godin [16] :

- **Navigation basée sur la localisation** : Cette méthode utilise des capteurs tels que des caméras, des LiDARs et des encodeurs de roues pour permettre au robot de se localiser dans son environnement et de planifier sa trajectoire en conséquence ;
- **Localisation et cartographie simultanée (SLAM)** : Le robot doit être capable de créer une carte de son environnement tout en se localisant par rapport à cette carte. Le SLAM permet au robot de se situer et de se déplacer en tenant compte de sa position actuelle et des obstacles environnants ;

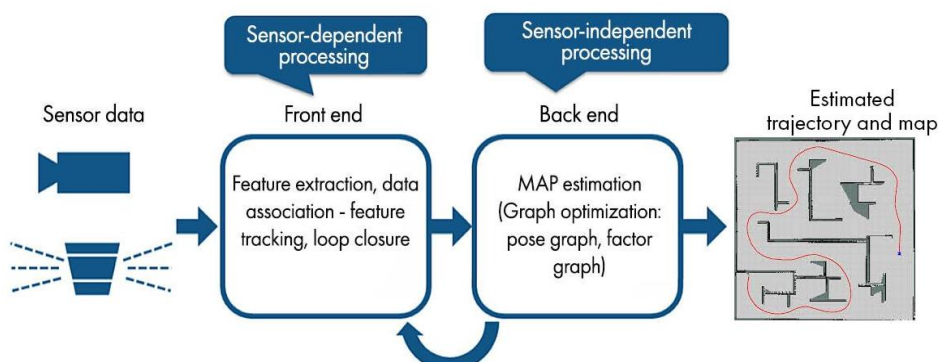


Figure I.4 Principe de la méthode SLAM ; Source : [16]

- **Planification de trajectoire** : Une fois que le robot a une carte de son environnement, il doit planifier une trajectoire sûre et efficace pour atteindre sa destination tout en évitant les obstacles. Cela implique de déterminer le chemin optimal à suivre en fonction de la configuration de l'environnement ;
- **Contrôle de mouvement et évitement d'obstacles** : Une fois la trajectoire planifiée, le robot doit contrôler ses mouvements pour suivre cette trajectoire de manière précise. Cela peut impliquer la régulation de la vitesse, de l'orientation et de la manœuvrabilité du robot pour éviter les collisions et se déplacer en toute sécurité ;
- **Interaction avec l'environnement** : Le robot doit être capable d'interagir avec son environnement en détectant les obstacles, en interprétant les informations sensorielles et en prenant des décisions en temps réel pour s'adapter aux changements dans l'environnement.

D'après Harbaoui N. l'interaction avec l'environnement englobe plusieurs aspects [17] :

- ✓ ***Correction d'erreur et recalibrage*** : C'est la capacité du RMA à détecter et à corriger les erreurs de localisation ou de cartographie, ainsi qu'à recalibrer ses capteurs ou ses systèmes en cas de besoin pour maintenir sa précision et son efficacité dans des environnements changeants ;
- ✓ ***Détection d'incertitude et prise de décision adaptative*** : Si l'environnement devient incertain ou imprévisible, le RMA doit être capable de détecter cette incertitude, d'évaluer les risques potentiels et de prendre des décisions adaptatives pour ajuster son comportement en conséquence ;
- ✓ ***Intégration de capteurs multiples*** : Ce qui permet au robot de recueillir des informations diverses sur son environnement, ce qui peut être essentiel pour s'adapter aux changements environnementaux tels que l'apparition d'obstacles imprévus ou les variations de luminosité.

En combinant ces principes, un robot autonome peut naviguer de manière efficace et sûre dans des environnements variés, que ce soit dans des entrepôts, des usines, des bureaux ou d'autres espaces.

### **I.3 La technologie LiDAR**

Le LiDAR représente aujourd'hui une technologie essentielle pour l'acquisition de données en trois dimensions sur l'environnement. Ses origines remontent aux années 1960 [18] et depuis il a connu de constantes évolutions (voir tableau I.3). Aujourd'hui, il s'est révélé être un outil polyvalent, trouvant des applications dans divers secteurs industriels et scientifiques grâce à sa capacité à fournir des données détaillées à haute résolution.

<b>Année</b>	<b>Événement</b>
1960	Invention du laser, ouvrant la voie au développement du LiDAR moderne
1962	Utilisation du LiDAR pour calculer la distance Terre-Lune dans le cadre du projet Luna See
1963	Application du LiDAR en météorologie par Goyer et Watson pour mesurer la hauteur des nuages
1971	Utilisation du MiDAR pour cartographier la Lune lors de la mission Apollo 15
1976	Publication du "Laser Monitoring of the Atmosphere" par ED. Hinkley, premier ouvrage dédié au LiDARE

*Tableau I.2 Évolution de la technologie LiDAR*

#### **I.3.1 Principe de fonctionnement**

Le LiDAR est une technologie essentielle qui utilise des impulsions laser pour mesurer les distances et créer des représentations précises du monde physique. Son fonctionnement repose sur l'émission de faisceaux laser courts qui se propagent dans l'environnement. Lorsque ces faisceaux rencontrent des objets, une partie de l'énergie lumineuse est réfléchiée ou dispersée vers le capteur LiDAR. En mesurant le temps mis par la lumière pour voyager jusqu'à l'objet et revenir (Time of Flight), le LiDAR peut calculer la distance entre le capteur et l'objet. En répétant ce processus pour plusieurs impulsions laser, une représentation tridimensionnelle de l'environnement peut être construite [18].

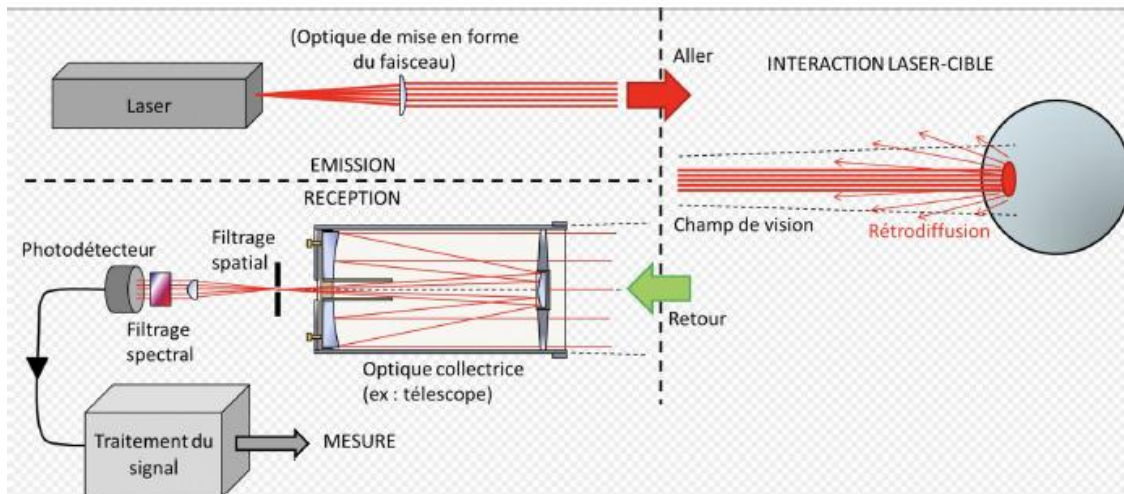


Figure I.5 Schéma de fonctionnement d'un système LiDAR ; Source : [18]

### I.3.2 Applications du LiDAR

Le LiDAR trouve des applications diverses dans différents domaines, ce qui démontre sa polyvalence et son importance croissante dans les domaines technologiques et la recherche scientifique [18]. En robotique, on le trouve dans la conception des véhicules autonomes, la surveillance environnementale, la cartographie géospatiale, l'arpentage et la recherche atmosphérique et climatique.

#### I.3.2.1 Applications en navigation des robots

Le Lidar est fréquemment employé dans la navigation des robots mobiles pour des applications cruciales, permettant des déplacements autonomes et optimisés [18] :

- **Perception de l'environnement** : En créant des cartes 3D détaillées, permettant aux robots mobiles de détecter les obstacles, d'estimer les distances et de modéliser l'environnement pour une navigation sûre ;
- **Localisation et cartographie simultanées (SLAM)** : En fusionnant les données Lidar avec d'autres capteurs et algorithmes, les robots peuvent se localiser, cartographier leur environnement et naviguer de manière autonome ;
- **Planification de trajectoire** : En évitant les obstacles et en optimisant les parcours, assurant une navigation fluide et sécurisée grâce aux données de distance et de forme qu'il fournit.

### I.3.3 Avantages et limitations du LiDAR

Le Lidar présente plusieurs avantages et limitations dans son utilisation comme nous le présentons dans le tableau I.3 :



<b>Avantages du LiDAR</b>	<b>Limitations du LiDAR</b>
Précision : Le LiDAR offre une précision élevée dans la capture de données 3D, ce qui le rend idéal pour la cartographie et la localisation précise.	Coût : les systèmes LiDAR peuvent être coûteux à entretenir, ce qui peut limiter leur accessibilité pour certains utilisateurs.
Rapidité : Il permet une acquisition rapide des données, ce qui est essentiel pour la cartographie aérienne ou la surveillance à temps réel.	Sensibilité aux conditions météorologiques : Les performances du LiDAR peuvent être affectées par les conditions météorologiques défavorables comme la pluie, la neige ou le brouillard, ce qui peut réduire sa fiabilité dans certains environnements.
Non dépendant de la lumière ambiante : Ce qui le rend efficace dans des environnements variés et à différentes heures de la journée.	Angle mort : Certains LiDAR ont des angles morts qui limitent leur capacité à détecter des objets dans certaines zones.
Capacité de pénétrer la végétation : Les LiDAR peuvent pénétrer la végétation dense pour cartographier le relief du sol.	Traitement des données : La quantité importante des données générées par le LiDAR nécessitent des capacités de traitement et de stockage adéquates, ce qui pose un défi pour des applications en temps réel.

**Tableau I.3** *Avantages et limitations du LiDAR*

Malgré ces limitations, le Lidar reste un outil précieux de la navigation autonome qui nécessite une perception détaillée de l'environnement [18].

## **I.4 Conclusion du chapitre**

Les généralités explorées dans ce chapitre, laisse suggérer que les techniques et les technologies examinées offrent un aperçu précieux des défis et des opportunités dans le développement de notre système de guidage et serviront de référence pour la conception et la simulation de notre robot autonome car l'analyse sert à la découverte et à la compréhension et a pour but d'élaborer les spécifications tandis que la conception sert à structurer et à développer une solution [19].

Avant de détailler la conception de notre système, nous devons d'abord acquérir une compréhension des systèmes de guidage de RMA, des différentes méthodes d'évitement d'obstacles, ainsi que de l'utilisation de ROS et Gazebo, ce que nous aborderons dans le prochain chapitre.

# **Chapitre II : *Guidage de robots***

---

## **II.1 Introduction**

Le guidage des robots est un aspect essentiel pour leur performance. Il s'appuie sur des technologies variées pour permettre aux robots de s'orienter et naviguer de manière autonome et efficace dans des environnements complexes.

Grâce au guidage les RMA et les VGA peuvent accomplir leurs missions de manière optimale en sélectionnant les itinéraires les plus courts, en évitant les obstacles et en optimisant leurs mouvements pour une productivité maximale.

Les systèmes de guidage sont essentiels pour améliorer la prise de décision et l'adaptabilité en temps réel dans une multitude d'applications, allant de la robotique industrielle et logistique à la robotique de service et mobile.

### **II.1.1 Définition**

Le guidage des robots est le processus par lequel un robot est dirigé pour se déplacer de manière autonome ou semi-autonome dans un environnement donné. Ce processus comprend plusieurs étapes clés : la perception de l'environnement par divers capteurs, la localisation du robot par rapport à son environnement, la planification du trajet optimal pour atteindre une destination tout en évitant les obstacles, et le contrôle des mouvements en suivant le trajet planifié [20].

Il existe deux types de guidage : le guidage réactif et le guidage orienté. La première catégorie regroupe les solutions de guidage local pour des engins naviguant sans destination précise, tandis que la seconde regroupe les solutions de guidage visant à atteindre une destination spécifique [20].

## **II.2 Techniques de guidage de robots**

Les techniques de guidage des robots sont des méthodes et des stratégies utilisées pour diriger et contrôler les mouvements des robots dans leur environnement.

Grace à ces techniques les robots peuvent être capables de s'adapter à des environnements changeants, de prendre des décisions en temps réel et d'interagir de manière intelligente avec leur environnement. Ce qui fait qu'ils sont essentiels pour permettre aux robots de naviguer de manière autonome, d'éviter les obstacles et d'atteindre leurs destinations de manière efficace et sûre [21].

## II.2.1 Différentes techniques de guidage

- **Interaction multimodale homme-robot** (Multimodal Human-Robot Interaction) : Qui implique l'utilisation de plusieurs modalités telles que l'audio et le visuel pour une interaction homme-robot efficace [22] ;
- **Guidage basé sur la réactivité** : Les robots utilisent des algorithmes de contrôle réactif pour réagir directement aux informations sensorielles sans nécessiter de planification à long terme. Par exemple, les robots peuvent éviter les obstacles en ajustant leur trajectoire en temps réel en fonction des données du guidage gestuel [23] ;
- **Guidage basé sur la planification de trajectoire** : Cette technique implique la génération préalable d'une trajectoire sûre et optimale vers une destination donnée, en tenant compte des obstacles et des contraintes environnementales. Les robots suivent ensuite cette trajectoire planifiée tout en ajustant leur mouvement pour éviter les obstacles imprévus [24] ;
- **Guidage basé sur les réseaux de neurones** : Les réseaux de neurones artificiels sont utilisés pour apprendre à partir de données d'entraînement afin de prendre des décisions de navigation autonomes. Ces réseaux peuvent être entraînés à reconnaître les schémas dans les données sensorielles et à prédire les actions appropriées pour naviguer dans l'environnement [25] ;
- **Guidage basé sur la fusion de données** : Cette technique combine les informations provenant de plusieurs capteurs pour améliorer la perception de l'environnement et la prise de décision. Les données provenant de capteurs visuels, LiDAR et ultrasoniques peuvent être fusionnées pour obtenir une représentation plus complète de l'environnement et prendre des décisions de navigation plus précises [26] ;

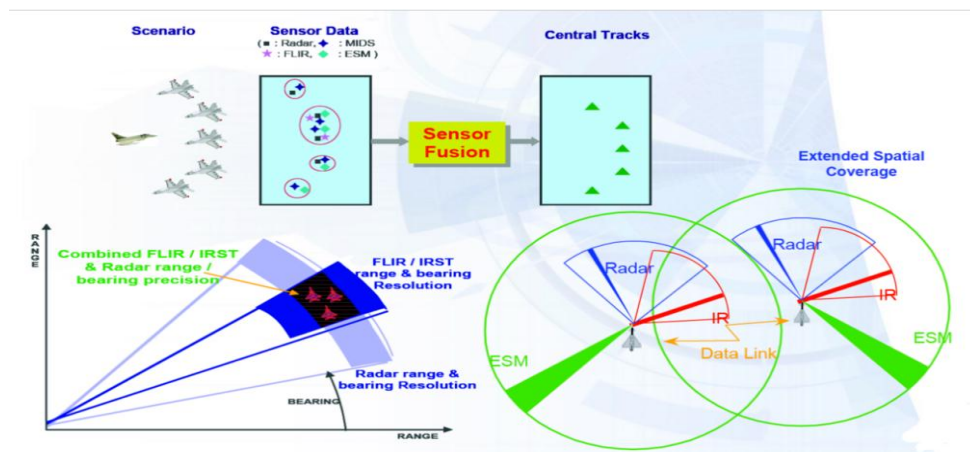


Figure II.1 Exemple de fusion de capteurs de l'avion Eurofighter ; Source : [27]

- **Guidage basé sur l'apprentissage par renforcement** : Les robots utilisent des techniques d'apprentissage par renforcement pour apprendre à naviguer efficacement dans leur environnement en interagissant avec lui. Les actions du robot sont récompensées ou pénalisées en fonction de leur efficacité, ce qui permet au robot d'optimiser son comportement au fil du temps [28].

## II.2.2 Méthodes d'évitement d'obstacles

Nous avons observé que diverses méthodes sont employées dans le domaine de la robotique pour permettre aux robots de se déplacer de façon autonome tout en évitant les obstacles. Voici quelques-unes des approches les plus fréquemment rencontrées.

- **Dynamic Window Approach (DWA)** : Cette méthode est utilisée pour la navigation des robots mobiles dans des environnements dynamiques. Elle consiste à générer une fenêtre dynamique de vitesses possibles pour le robot en fonction de sa position actuelle et des obstacles détectés. Le robot évalue ensuite chaque combinaison de vitesses possibles pour sélectionner celle qui minimise le risque de collision avec les obstacles et atteint la destination souhaitée [29] ;
- **Méthode Follow the Gap** : Cette méthode consiste à suivre un chemin sûr à travers les obstacles en utilisant les intervalles entre eux, appelés "gaps". Le robot détecte les gaps dans son environnement et suit celui qui offre la trajectoire la plus sûre vers sa destination, en évitant les collisions avec les obstacles [30] ;
- **Fuzzy Inference** : La logique floue est une méthode utilisée pour modéliser le raisonnement humain dans des situations d'incertitude ou de complexité. Dans le contexte de l'évitement d'obstacles, la logique floue peut être utilisée pour prendre des décisions de navigation en évaluant les degrés d'appartenance à différentes catégories (par exemple, "proche de l'obstacle", "loin de l'obstacle") et en déterminant la trajectoire optimale en fonction de ces évaluations [31,32] ;
- **Artificial Potential Field (APF)** : Cette méthode crée un champ de forces attractives et répulsives autour des obstacles et de la destination. Le robot navigue en suivant le gradient du potentiel, évitant ainsi les obstacles en les contournant [33] ;
- **Algorithmes de planification de trajectoire** : Ces algorithmes génèrent une trajectoire sûre pour le robot en tenant compte de la disposition des obstacles dans l'environnement. Des techniques telles que les algorithmes RRT (Rapidly-exploring Random Trees) et PRM (Probabilistic RoadMaps) [34] ou l'algorithme D et L'algorithme de Dijkstra [35] sont souvent utilisées ;

- **Navigation réactive** : Les techniques de navigation réactive permettent au robot de réagir directement aux données des capteurs sans nécessiter de planification à long terme. Cela peut inclure des méthodes telles que la navigation basée sur la réactivité et les réseaux de neurones pour la prise de décision en temps réel [36] ;
- **Inflation Layer (Expansion d'obstacles)** : Lors de la planification de trajectoires, les obstacles de la carte sont gonflés pour correspondre au rayon du cercle dans lequel le périmètre du robot peut être inscrit. Cette méthode simplifie la représentation du robot en un point lors de la planification de trajectoires pour plus de simplicité et d'efficacité en évitement d'obstacle [35] ;
- **Generalized Voronoi Diagram (GVD)** : Les diagrammes de Voronoi sont utilisés pour accélérer la recherche de trajectoires sûres en suivant les contours des régions du diagramme. Ces trajectoires permettent de planifier des chemins sécurisés éloignés des obstacles [35].

Comme il existe d'autres méthodes moins courantes et sont plutôt des approches plus générales pour la planification de mouvement et le contrôle des robots dans des environnements dynamiques et encombrés. Parmi ces méthodes :

- **Bug1 et Bug2** : Sont des algorithmes proposés comme une solution simple pour les robots mobiles afin d'atteindre une destination tout en évitant les obstacles. Bug1 est basée sur la recherche de chemins. Lorsque le robot rencontre un obstacle, il suit le contour de l'obstacle jusqu'à ce qu'il puisse atteindre sa destination. Bug2 est une amélioration de la méthode Bug1. Plutôt que de suivre le contour de l'obstacle, le robot suit un chemin direct vers sa destination jusqu'à ce qu'il rencontre un obstacle, il le contourne l'obstacle en calculant une nouvelle trajectoire pour atteindre sa destination [37] ;
- **Méthode de fonction de navigation** : Cette méthode consiste à utiliser une fonction de coût ou de valeur pour évaluer les différentes options de déplacement du robot en fonction de son objectif et de l'environnement. La fonction de navigation peut être basée sur divers critères tels que la distance à la destination, la sécurité par rapport aux obstacles, la vitesse, etc. Le robot utilise cette fonction pour sélectionner la meilleure action à entreprendre à chaque étape de son déplacement [38] ;
- **Régulateur optimal** : Cette méthode implique la conception d'un régulateur (ou contrôleur) optimal pour stabiliser et contrôler le mouvement du robot tout en

minimisant un certain critère de performance. Elle peut être utilisée pour générer des trajectoires sûres et efficaces en présence d'obstacles [39].

Les méthodes peuvent être combinées comme le recours à Algorithmes de planification de trajectoire et Capteurs de proximité ou la combinaison Fuzzy Inference, capteurs de proximité et vision par ordinateur [32].

### II.2.3 Avantages et des limitations de chaque méthode

Le tableau suivant (Tableau II.1) présente une comparaison des avantages et des limitations de chaque méthode d'évitement d'obstacles.

Méthode	Avantages	Limitations
Dynamic Window Approach (DWA)	<ul style="list-style-type: none"> <li>- Adaptabilité aux environnements dynamiques</li> <li>- Efficace pour la navigation en temps réel</li> </ul>	<ul style="list-style-type: none"> <li>- Complexité élevée</li> <li>- Peut nécessiter des ajustements constants des paramètres</li> </ul>
Follow the Gap	<ul style="list-style-type: none"> <li>- Utilisation efficace des espaces entre les obstacles</li> <li>- Bonne performance dans des environnements structurés</li> </ul>	<ul style="list-style-type: none"> <li>- Peut rencontrer des difficultés dans des environnements encombrés</li> <li>- Sensible aux erreurs de détection des obstacles</li> </ul>
Fuzzy Inference	<ul style="list-style-type: none"> <li>- Modélisation flexible des décisions de navigation</li> <li>- Robuste face à l'incertitude dans l'environnement</li> </ul>	<ul style="list-style-type: none"> <li>- Complexité de mise en œuvre</li> <li>- Besoin d'une base de règles bien définie</li> </ul>
Artificial Potential Field (APF)	<ul style="list-style-type: none"> <li>- Concept intuitif</li> <li>- Adaptabilité à différents environnements</li> </ul>	<ul style="list-style-type: none"> <li>- Sensible aux minima locaux</li> <li>- Risque de convergence vers des configurations indésirables</li> </ul>
Algorithmes de planification de trajectoire	<ul style="list-style-type: none"> <li>- Génération de trajectoires sûres et efficaces</li> <li>- Adaptabilité à divers types d'environnements</li> </ul>	<ul style="list-style-type: none"> <li>- Complexité élevée pour les environnements complexes</li> <li>- Temps de calcul important</li> </ul>
Capteurs de proximité	<ul style="list-style-type: none"> <li>- Réponse rapide</li> <li>- Faible coût</li> <li>- Utilisation dans diverses conditions environnementales</li> </ul>	<ul style="list-style-type: none"> <li>- Portée limitée</li> <li>- Sensibilité aux conditions environnementales</li> <li>- Risque de fausses détections</li> </ul>
Vision par ordinateur	<ul style="list-style-type: none"> <li>- Capacité à identifier des obstacles complexes</li> <li>- Flexibilité dans les types d'obstacles détectables</li> </ul>	<ul style="list-style-type: none"> <li>- Exige des ressources computationnelles importantes</li> <li>- Sensible aux variations d'éclairage et aux obstructions</li> </ul>

Apprentissage par renforcement	<ul style="list-style-type: none"> <li>- Capacité à apprendre des comportements complexes</li> <li>- Adaptabilité à l'environnement</li> </ul>	<ul style="list-style-type: none"> <li>- Nécessite une phase d'apprentissage initiale</li> <li>- Sensibilité aux erreurs dans les récompenses</li> </ul>
Navigation réactive	<ul style="list-style-type: none"> <li>- Temps de réponse rapide</li> <li>- Pas besoin de planification à long terme</li> <li>- Adaptabilité aux environnements dynamiques</li> </ul>	<ul style="list-style-type: none"> <li>- Risque de mouvements suboptimaux</li> <li>- Sensibilité aux erreurs de capteurs</li> </ul>
Inflation Layer	<ul style="list-style-type: none"> <li>- Simple à mettre en œuvre et à comprendre</li> <li>- Permet de garantir une distance de sécurité en évitement d'obstacles</li> </ul>	<ul style="list-style-type: none"> <li>- Peut ne pas prendre en compte les contraintes d'orientation</li> <li>- Peut conduire à des trajectoires sans orientation</li> </ul>
Diagrammes de Voronoi	<ul style="list-style-type: none"> <li>- Recherche de trajectoires sûres et rapides</li> <li>- Produit des trajectoires éloignées des obstacles</li> </ul>	<ul style="list-style-type: none"> <li>- Nécessite une étape d'optimisation pour ajuster le trajet</li> <li>- Risque d'échec de la planification</li> </ul>
Bug1 et Bug2	<ul style="list-style-type: none"> <li>- Simplicité de mise en œuvre</li> <li>- Efficace dans des environnements simples</li> </ul>	<ul style="list-style-type: none"> <li>- Limitée aux environnements sans obstacles complexes</li> <li>- Risque de chemins non optimaux</li> </ul>
Fonction de navigation	<ul style="list-style-type: none"> <li>- Utilisation d'une approche basée sur des critères définis par l'utilisateur</li> <li>- Flexibilité dans l'évaluation des actions</li> </ul>	<ul style="list-style-type: none"> <li>- Nécessite une définition précise des critères</li> <li>- Complexité de l'implémentation</li> </ul>
Régulateur optimal	<ul style="list-style-type: none"> <li>- Stabilisation efficace du mouvement du robot</li> <li>- Contrôle précis en présence d'incertitudes</li> </ul>	<ul style="list-style-type: none"> <li>- Nécessite une modélisation précise du système</li> <li>- Complexité de réglage des paramètres</li> </ul>

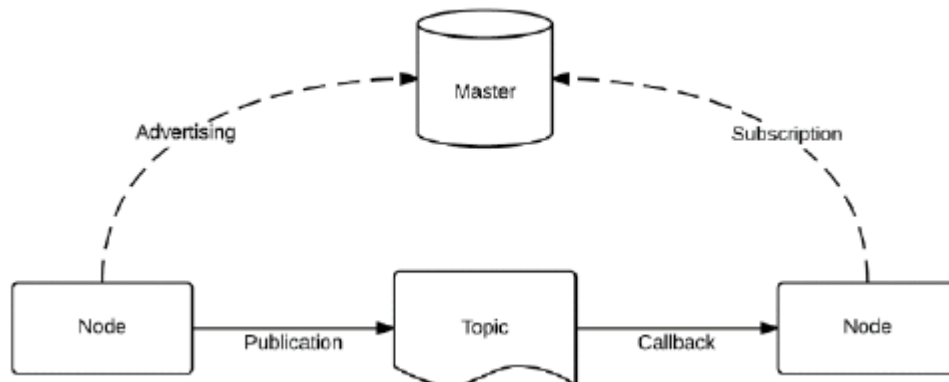
*Tableau II.1 Avantages et limitations des méthodes d'évitement d'obstacles*

## **II.3 Robot Operating System (ROS) et simulateur 3D Gazebo**

### **II.3.1 Introduction à ROS**

ROS (Robot Operating System) est un cadre de développement open-source essentiel pour le développement de logiciels en robotique. Contrairement à un système d'exploitation traditionnel, ROS ne gère pas directement les processus et l'ordonnancement, mais il fournit plutôt un ensemble complet de bibliothèques, de conventions et d'outils pour simplifier le développement de logiciels destinés à la paramétrisation des robots [40].

En tant que middleware fonctionnant exclusivement dans l'environnement Linux Ubuntu, ROS repose sur une plomberie efficace qui facilite la communication entre les composants logiciels, quelle que soit leur répartition au sein de l'architecture distribuée. Ces mécanismes de communication, illustrés sur la figure I.10, permettent aux différents nœuds de ROS de s'échanger des messages via des topics, créant ainsi un réseau robuste et modulaire de traitement des données pour les applications robotiques [41].



*Figure II.2 Protocole de communication ROS ; Source : [41]*

### II.3.1.1 Architecture de ROS

ROS est basé sur 4 grands types de mécanismes permettant de construire une application robotique pour que le robot puisse interagir avec son environnement en fonction de sa perception. Ces mécanismes sont [42] :

- **La plomberie** : C'est la connection des composants logiciels entre eux, indépendamment de leur répartition dans un réseau. Elle a pour rôle de faciliter la communication entre les composants logiciels, favorisant ainsi la modularité et l'interopérabilité au sein de l'architecture distribuée de ROS. Elle repose sur plusieurs niveaux d'organisation et de communication :
  - ✓ Niveau du système de fichiers : ROS est organisé en packages qui sont des unités de base contenant des fichiers, des nœuds, des bibliothèques, des paramètres et des dépendances et en stacks qui regroupent des packages liés par un objectif commun, comme la perception, la planification ou le contrôle.
  - ✓ Niveau du graphe de calcul : Le cœur de ROS est le graphe de calcul, un réseau de nœuds interconnectés qui communiquent via des topics, des services et des actions. Chaque nœud est un processus exécutable effectuant une tâche spécifique.
- **Les outils** : Sont indispensables pour assurer le bon fonctionnement et la performance des applications robotiques. C'est un ensemble de logiciels qui permettent d'analyser,



d'afficher et de déboguer les applications réparties développées avec ROS. Ils sont utilisés pour surveiller le comportement des nœuds, visualiser les données échangées via les topics, diagnostiquer les problèmes de communication, etc.

- **Les capacités :** Elles correspondent aux bibliothèques qui implémentent les fonctionnalités nécessaires à la réalisation de différentes tâches dans les applications robotiques. Elles comprennent des outils et des algorithmes pour la planification de tâches, la planification de mouvements, la construction de modèles d'environnement, etc. Elles sont essentielles pour permettre aux robots d'exécuter diverses tâches de manière efficace et précise dans des environnements complexes.
- **L'écosystème :** L'écosystème de ROS se compose d'un nombre important d'utilisateurs qui contribuent à la plateforme en partageant activement des packages, des tutoriels, des exemples et des ressources, en signalant des bugs et en participant aux discussions. Ce qui favorise l'innovation et contribue à l'amélioration continue de ROS et à son adaptation aux besoins changeants de la robotique.

### II.3.1.2 Historique

ROS1 a été développé à partir de 2008 par la société Willow dont les contributions ont été cruciales pour l'adoption et le développement de ROS-1, atteignant une masse critique d'utilisateurs sans précédent. Cependant, en 2013, Willow Garage a fermé ses portes, laissant place à l'Open Source Robotics Foundation (OSRF), financée par plusieurs sociétés, notamment Toyota.

ROS2 a été développé pour répondre aux défis posés par les systèmes robotiques modernes. Il a été repensé pour inclure des fonctionnalités et des algorithmes de qualité production absents dans ROS1. Actuellement on vise à aligner les versions de ROS1 et ROS2 avec celles des versions LTS (Long Term Support) d'Ubuntu.

Le tableau II.2 résume l'historique de la version ROS-2 et le tableau II.3 résume celui de la version ROS-1.

2017 - Dec	Ardent Apalone (EOL - Dec 2018)
2018 - June	Bouncy Bolson (EOL - June 2019)
2018 - Dec	Crystal Clemminys (EOL - Dec 2019)
2019 - Mai	Dashing Diademata (EOL - Mai 2021)
2019 - Nov	Eloquent Elusor (EOL - Mai 2020)
2020 - May	Foxy Fitzroy (EOL - Mai 2023)
2021 - May	Galactic Geochelone (EOL - Nov 2022)
2022 - May	Humble Hawksbill (EOL - Mai 2027)

*Tableau II.2 Historique des releases de ROS 2 - REP 2000 ; Source : [42]*

2008	Démarrage de ROS par Willow Garage
2010 - Janvier	ROS 1.0
2010 - Mars	Box Turtle
2010 - Aout	C Turtle
2011 - Mars	Diamondback
2011 - Aout	Electric Emys
2012 - Avril	Fuerte
2012 - Décembre	Groovy Galapagos
2013 - Février	Open Source Robotics Foundation poursuit la gestion de ROS
2013 - Aout	Willow Garage est absorbé par Suitable Technologies
2013 - Aout	Le support de PR-2 est repris par Clearpath Robotics
2013 - Septembre	Hydro Medusa (prévu)
2014 - Juillet	Indigo Igloo (EOL - Avril 2019)
2015 - Mai	Jade Turtle (EOL - Mai 2017)
2016 - Mai	Kinetic Kame (EOL - Mai 2021)
2017 - Mai	Lunar Loggerhead (EOL - Mai 2019)
2018 - Mai	Melodic Morenia (EOL - Mai 2023)
2020 - Mai	Noetic Ninjemys (EOL - Mai 2025)

*Tableau II.3 Historique de ROS1 et des releases - REP 3 ; Source : [42]*

### II.3.1.3 Importance de ROS en robotique

ROS s'est imposé comme la référence dans le monde robotique aussi bien au niveau académique, industriel et amateurs [39,41] par son importance qui réside dans plusieurs aspects :

- **Démocratisation de la robotique** : En utilisant des langages de programmation modernes tels que Python, Java et C++, ROS réduit les obstacles à la création de programmes robotiques et ouvre la robotique à une utilisation plus large [39,41] ;
- **Standardisation** : ROS permet la réutilisation des programmes et facilite le transfert de connaissances entre différents types de robots, ce qui accélère le développement et réduit les coûts [41,42] ;
- **Collaboration interdisciplinaire** : En abaissant le niveau technique requis pour travailler sur des projets robotiques, ROS permet à des experts de différentes disciplines de contribuer plus facilement à la robotique [39,42] ;
- **Innovation** : ROS stimule l'innovation en permettant aux chercheurs et ingénieurs de se concentrer sur des solutions robotiques avancées plutôt que sur des problèmes de programmation bas niveau [49]. Il offre un environnement flexible et adaptatif pour le développement de solutions robotiques [39,41] ;
- **Architecture modulaire** : Ce qui favorise la réutilisabilité du code et le travail indépendant sur les composants du système robotique [39, 41,42,43] ;

- **Interopérabilité** : Avec une variété de capteurs, d'actionneurs et de plates-formes matérielles, permettant la création de solutions robotiques adaptées à divers environnements [41, 42,43] ;
- **Communauté active et diversifiée** : Stimulant l'innovation, le partage de connaissances et le développement de nouveaux packages logiciels. Aussi ROS peut se mettre à jour automatiquement grâce des améliorations continues des programmes externes portés par de la communauté [39, 41, 42,43].

### II.3.2 Gazebo

Gazebo est un simulateur 3D largement utilisé dans le domaine de la robotique pour la simulation de robots et d'environnements robotiques. Il offre un environnement de simulation réaliste et flexible pour tester et développer des systèmes robotiques complexes.

Souvent utilisé en conjonction avec ROS, Gazebo offre une plateforme complète pour la simulation robotique, permettant aux développeurs de créer, tester et valider des systèmes robotiques avancés dans des environnements virtuels réalistes et contrôlés [44].

#### II.3.2.1 Architecture Gazebo

Pour offrir une simulation réaliste et flexible pour les robots et les environnements virtuels, Gazebo utilise une architecture distribuée avec des bibliothèques distinctes. Il fournit deux programmes exécutables pour exécuter des simulations, l'un *gzserver* destiné au serveur pour simuler la physique, le rendu et les capteurs, l'autre *gzclient* conçu pour les clients et fournit une interface graphique pour visualiser et interagir avec la simulation.

Le client et le serveur communiquent à l'aide de la bibliothèque de communication open source Gazebo [45].

Les principaux composants de l'architecture Gazebo sont [44,45] :

- **Le Gazebo Master** : Un serveur de noms de sujets dans le contexte de la communication entre les processus de Gazebo. Il fournit des services de recherche de noms et de gestion des sujets. Un seul maître Gazebo peut gérer plusieurs simulations physiques, générateurs de capteurs et interfaces graphiques ;
- **La bibliothèque de communications** : Utilisée par presque toutes les bibliothèques, elle agit comme un mécanisme de communication et de transport essentiel pour Gazebo. Ce qui permet l'échange de données entre les différents composants de la simulation. Les nœuds ROS peuvent communiquer avec Gazebo via des topics pour contrôler les robots et collecter des données sensorielles.

- **La bibliothèque physique** : Elle fournit une interface simple et générique pour les composants fondamentaux de simulation, des interactions physiques entre les objets, tels que la gravité, les collisions, les forces et les mouvements des corps rigides et souples.
- **La bibliothèque de rendu** : Sert d'interface fluide permettant de visualiser des scènes 3D à la fois dans l'interface graphique et dans les bibliothèques de capteurs. Elle englobe des fonctionnalités telles que l'éclairage, les textures et même la simulation du ciel grâce au moteur de rendu 3D. Dans cette bibliothèque Gazebo exploite le potentiel d'OGRE pour élaborer des environnements virtuels complexes et d'une grande réalité.
- **Le générateur de capteurs** : La bibliothèque de génération de capteurs implémente tous les différents types de capteurs tels que des caméras, des lidars, des radars, des capteurs de force, etc., pour simuler des données sensorielles réalistes et permettre aux robots de percevoir leur environnement.
- **Les interfaces** : Gazebo offre des interfaces pour interagir avec les simulations, telles que des API de programmation pour contrôler les robots, des outils de visualisation pour observer le comportement des robots, des interfaces graphiques permettant aux utilisateurs d'interagir avec la simulation et des outils de débogage pour analyser les données de simulation.
- **Les plugins** : Les bibliothèques de physique, de rendu et le générateur de capteurs prennent en charge les plugins. Ces plugins permettent aux utilisateurs d'accéder aux bibliothèques respectives sans utiliser le système de communication. Ces plugins qui permettent d'étendre les fonctionnalités comme ajouter des capteurs, des contrôleurs, des actionneurs et d'autres fonctionnalités personnalisées aux modèles de robots.
- **Modèles de robots** : Les robots et les objets sont décrits à l'aide de modèles URDF qui spécifient la cinématique, la dynamique, la géométrie et d'autres propriétés des entités simulées. Ce qui permet créer des simulations rapidement et efficacement.

Finalement, l'architecture de Gazebo intègre ROS permettant une communication transparente entre la simulation et d'autres composants logiciels, tels que des contrôleurs de robot ou des algorithmes de planification, facilitant ainsi le développement et le test de systèmes robotiques complets.

### II.3.2.2 Historique

Les dates importantes du développement du simulateur Gazebo sont énoncées dans le tableau II.4 suivant :

Année	Évènement
2002	Développement initial de Gazebo come projet de recherche à l'USC
2004	Version 0.5 de Gazebo, première version publique
2008	Développement continu de Gazebo par la communauté Open Source
2011	Intégration de Gazebo dans ROS
2012	Gazebo devient un projet officiel de la fondation ROS
2012	Version 1.0 de Gazebo avec une refonte majeure de l'architecture
2015	Début du développement de Gazebo 2
2018	Lancement de Gazebo 9.0 avec de nouvelles fonctionnalités et améliorations
2020	Introduction de Gazebo 11.0 avec améliorations et nouvelles fonctionnalités
2021	Développement continu de Gazebo avec des mises à jour régulières

*Tableau II.4 Historique des développements de Gazebo*

### II.3.2.3 Importance de Gazebo en robotique

Gazebo joue un rôle crucial dans le domaine de la robotique en offrant un environnement de simulation avancé et polyvalent. Son importance réside dans les points clés suivants [44] :

- **Développement et test de robots** : Par la permission aux développeurs de robots de créer, tester et valider des algorithmes de contrôle, des stratégies de planification de mouvement et des comportements robotiques dans un environnement virtuel réaliste. Cela réduit le besoin de tests sur des robots physiques, ce qui peut être coûteux et risqué.
- **Sécurité** : En utilisant Gazebo, les chercheurs et les ingénieurs peuvent effectuer des expériences et des tests en toute sécurité sans risquer d'endommager des robots réels ou de mettre en danger des opérateurs humains. Cela permet de garantir la sécurité des personnes et des équipements.
- **Optimisation des performances** : Gazebo permet d'optimiser les performances des robots en simulant différents scénarios et en ajustant les paramètres de contrôle pour obtenir des résultats optimaux. Cela permet d'améliorer l'efficacité et la fiabilité des systèmes robotiques.
- **Validation d'algorithmes** : Les algorithmes de perception, de planification de trajectoire, de navigation et d'interaction avec l'environnement peuvent être validés et testés efficacement dans Gazebo. Cela permet de s'assurer que les algorithmes fonctionnent correctement avant de les déployer sur des robots réels.
- **Formation et apprentissage** : Gazebo est un outil précieux pour la formation et l'apprentissage en robotique. Les étudiants, les chercheurs et les professionnels peuvent

acquérir des compétences en programmation robotique, en contrôle de robots et en vision par ordinateur en utilisant Gazebo comme plateforme d'apprentissage.

- **Intégration avec ROS :** Gazebo s'intègre parfaitement avec Robot Operating System (ROS), une plateforme logicielle largement utilisée en robotique. Cette intégration facilite le contrôle des robots simulés, la collecte de données sensorielles et la communication entre les différents composants d'un système robotique.

### II.3.3 Intégration de ROS - Gazebo

Gazebo s'intègre étroitement avec ROS, offrant une interface transparente pour contrôler et surveiller les robots simulés à l'aide de ROS, facilitant le développement de systèmes robotiques basés sur ROS. En combinant ROS et Gazebo, les chercheurs en robotique peuvent bénéficier d'un environnement de développement complet pour la conception, la simulation et la validation de systèmes robotiques avancés, ce qui contribue à l'avancement de la recherche en robotique. Cette intégration se concrétise comme suit [44] :

- **Communication :** ROS et Gazebo peuvent communiquer entre eux en utilisant des ponts de communication tels que les nœuds ROS et les plugins Gazebo. Les nœuds ROS peuvent publier et souscrire à des messages sur des topics, tandis que les plugins Gazebo peuvent envoyer des données de capteurs simulés à des nœuds ROS pour le traitement.
- **Contrôle de robots :** ROS permet de contrôler des robots simulés dans Gazebo en utilisant des packages et des contrôleurs spécifiques. Les développeurs peuvent définir des trajectoires, des commandes de vitesse et des stratégies de contrôle dans ROS, qui sont ensuite transmises aux robots simulés dans Gazebo.
- **Visualisation :** ROS et Gazebo offrent des outils de visualisation avancés pour observer et analyser le comportement des robots simulés. Les données de capteurs, les trajectoires planifiées et d'autres informations peuvent être affichées en temps réel pour faciliter le débogage et l'optimisation des systèmes robotiques.
- **Simulation de capteurs :** ROS et Gazebo permettent de simuler une grande variété de capteurs tels que des caméras, des lidars, des capteurs inertiels, etc. Les données de ces capteurs simulés peuvent être traitées dans ROS pour la perception, la localisation et d'autres tâches.
- **Intégration de packages ROS :** De nombreux packages ROS sont compatibles avec Gazebo, ce qui facilite l'intégration de fonctionnalités supplémentaires dans les simulations

- **Développement modulaire :** En combinant ROS et Gazebo, les développeurs peuvent créer des systèmes robotiques modulaires et flexibles. Différents composants logiciels peuvent être développés et testés indépendamment dans ROS, puis intégrés dans des simulations Gazebo pour évaluer le système dans son ensemble.

En somme, l'intégration de ROS et Gazebo offre une plateforme puissante et flexible pour le développement, la simulation et l'optimisation des systèmes robotiques en fournissant des outils complets pour développer des solutions avancées et robustes.

Cette synergie justifie notre choix de l'utiliser pour la conception et la validation de notre système de guidage de RMA.

## **II.4 Conclusion du chapitre**

Le guidage de robot constitue un pilier fondamental de la robotique moderne. En alliant des technologies avancées basées sur des capteurs et la planification de trajectoires, le guidage permet aux robots de se déplacer de manière autonome et précise dans des environnements complexes. Cela garantit l'efficacité du déplacement, l'évitement d'obstacles et l'accomplissement de diverses missions, tout en améliorant la productivité, la flexibilité, la sécurité, la rapidité et la précision.

Aussi nous avons constaté que l'association de ROS et Gazebo offre une plateforme de simulation et de développement complète qui soutient de manière significative l'innovation en robotique, permettant de tester et valider des algorithmes dans un environnement sûr et économique.

En résumé, la conception et la simulation de notre système de guidage de robot autonome avec ROS et Gazebo offre un environnement de simulation complet, flexible et puissant pour le développement, le test et la validation de notre système robotique.

Conception et simulation que nous allons réaliser dans le chapitre suivant.

# **Chapitre III : *Conception et simulation***

---

## **III.1 Introduction**

Après avoir exploré les concepts fondamentaux de la robotique et les différentes méthodes de navigation autonome dans les chapitres précédents, nous allons maintenant appliquer ces connaissances pour détailler la conception et la simulation d'un système de guidage pour un RMA dans des espaces intérieurs mal connus. Ce système utilisera un LiDAR 2D et sera testé dans un environnement virtuel sous ROS et Gazebo.

Comme nous l'avons constaté précédemment, le guidage RMA repose sur la capacité à percevoir et interpréter l'environnement, à planifier des trajectoires sûres et à éviter les obstacles en temps réel. C'est pourquoi nous avons choisi les outils et technologies développés dans la suite, pour leur compatibilité et leur capacité à fournir une plateforme de simulation complète et flexible, essentielle pour le développement, le test et la validation de notre système de guidage de RMA.

## **III.2 Méthode de navigation**

Notre choix s'est porté sur l'utilisation de la navigation réactive en combinaison avec un LiDAR 2D parce qu'elle permet au robot de réagir instantanément aux informations sensorielles fournies par le LiDAR 2D, tout en étant simple à implémenter et dans des environnements mal connus.

En effet, en combinant la méthode de navigation réactive avec les données LiDAR précises sur la proximité des obstacles dans un plan horizontal, le robot peut ajuster sa trajectoire en fonction des obstacles détectés, ce qui lui permet de naviguer de manière fluide et efficace dans des environnements mal connus où les obstacles peuvent apparaître de manière imprévisible.

En outre, la méthode de navigation réactive sera concrétisée par la stratégie de contrôle des parois, qui consiste à maintenir une distance constante par rapport aux parois environnantes. Cette approche permettra au robot de naviguer de manière sûre et efficace en longeant les parois, en utilisant le LiDAR 2D pour détecter et contourner les obstacles tout en maintenant une trajectoire stable et prévisible.

Nous partons de l'hypothèse qu'en combinant ces deux techniques, notre système de guidage de robot autonome sera en mesure de surmonter les défis posés par des environnements



intérieurs mal connus, offrant ainsi une solution simple et adaptable pour la navigation autonome.

### III.2.1 Contrôle des parois (Wall monitoring)

La méthode du contrôle des parois est une stratégie de navigation simple et robuste qui s'avère efficace dans les environnements intérieurs. Elle offre l'avantage de ne pas nécessiter de connaissance préalable de la disposition ou de la forme de l'environnement intérieur, ce qui signifie qu'il n'est pas nécessaire d'avoir un modèle de l'environnement à l'avance. Cela permet d'appliquer facilement cette méthode à divers environnements complexes avec une préparation simple [46].

Le LiDAR jouera un rôle crucial dans le contrôle des mouvements du robot le long des parois et dans l'évitement des obstacles de la manière suivante :

#### III.2.1.1 Contrôle des mouvements

Le système de contrôle embarqué du robot est utilisé pour définir des règles de contrôle basées sur les données des capteurs infrarouges et ultrasoniques. Ces règles permettent au RMA de maintenir une trajectoire parallèle à la paroi en ajustant sa vitesse et son angle de virage en fonction des distances mesurées par les capteurs [47]. Cela permet au robot de suivre de manière précise les contours des parois tout en explorant l'environnement inconnu.

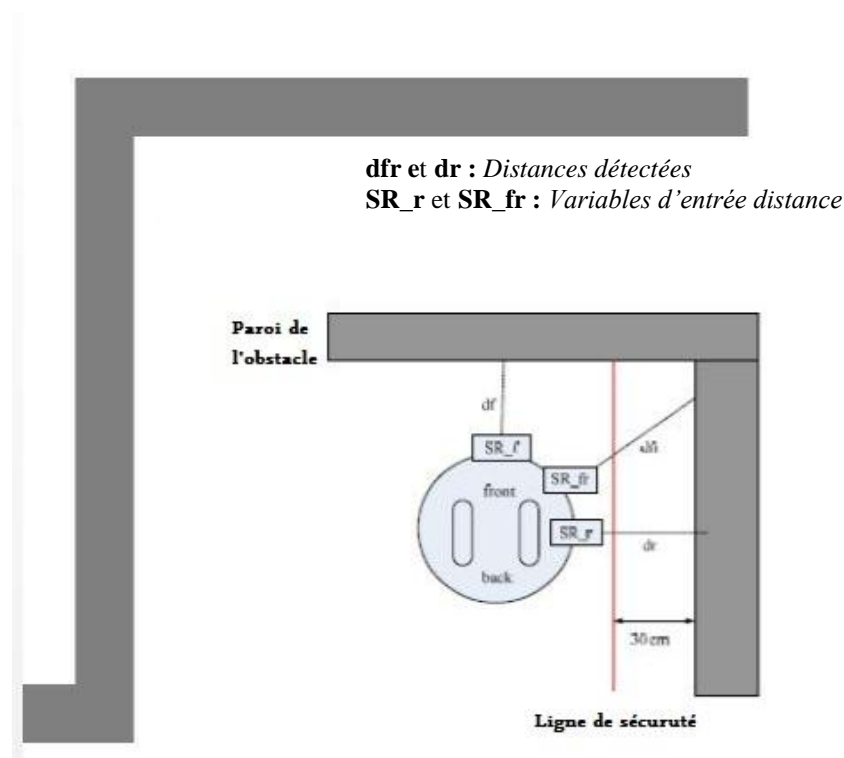


Figure III.1 Capteurs et variables d'entrées en contrôle de paroi ; Source : [47]

- **Règles de contrôle** : La méthode de contrôle des parois est basée sur les règles de contrôle du robot mobile, qui déterminent ses actions parmi quatre cas : "Avancer", "Tourner à gauche", "Tourner à droite" et "Faire demi-tour". Pour la plupart des règles de contrôle, le robot mobile avance en suivant une paroi à une distance décalée suivant les données capteurs. La technique de linéarisation de rétroaction est utilisée pour mettre en œuvre le contrôleur de navigation afin de guider le robot mobile le long de la ligne de base de paroi en maintenant une distance constante par rapport à elle [46].

### **III.2.1.2 Évitement des obstacles**

Les données laser du LiDAR 2D permettent de détecter les obstacles sur le chemin du robot. Le RMA utilise ces données pour prendre des décisions en temps réel sur la manière d'éviter les obstacles tout en se maintenant sur sa trajectoire le long des parois. En ajustant la vitesse et l'angle de virage en fonction des données des capteurs, le robot sera capable d'éviter les obstacles de manière autonome et efficace [47].

### **III.2.1.3 Adaptation de la trajectoire**

Le LiDAR effectue d'abord la détection des obstacles et des parois autour du RMA. Ensuite, les données collectées sur les distances et les angles par rapport aux obstacles et aux parois sont enregistrées pour chaque position du robot. À partir de ces données, une trajectoire de déplacement est calculée pour le RMA, tenant compte de sa position actuelle et des obstacles détectés. Cette trajectoire calculée permet de générer une carte de l'environnement inconnu.

## **III.3 Méthode d'évitement d'obstacles**

Le contrôle de paroi est une méthode efficace de navigation dans des environnements structurés et contraints, comme les intérieurs de bâtiments. Cependant, bien qu'il inclue des mécanismes pour gérer les obstacles rencontrés en cours de route, ce n'est pas exclusivement une méthode d'évitement d'obstacles.

Pour améliorer notre système de guidage du RMA, nous allons intégrer un compteur dans la conception de notre robot. Ce compteur servira de mécanisme de comptage des obstacles rencontrés et des changements de direction effectués par le robot. Il jouera un rôle spécifique dans la gestion de l'évitement d'obstacles en fournissant une indication claire du nombre d'obstacles rencontrés et des actions correctives prises.

Nous pensons qu'en intégrant un compteur dans la stratégie de contrôle de parois, nous pourrions non seulement améliorer l'évitement d'obstacles, mais aussi obtenir des informations précieuses sur les interactions du robot avec son environnement, permettant une optimisation continue de l'algorithme de navigation.

### III.3.1 Intégration du compteur

Pour améliorer la gestion des obstacles et le guidage de notre RMA dans des environnements inconnus, nous allons intégrer un compteur d'obstacle à notre système. Son rôle sera d'ajouter les obstacles détectés dans chaque secteur angulaire.

En fournissant un mécanisme de comptage des obstacles et des changements de direction, le compteur permet un contrôle précis des interactions du robot avec son environnement, ce qui facilite l'optimisation continue des algorithmes de navigation.

Nous suggérons que ce type de compteur sera un outil précieux pour suivre et gérer les réactions du RMA face aux obstacles, contribuant à un guidage autonome plus intelligent et plus efficace dans un environnement inconnu.

#### III.3.1.1 Technique de fonctionnement

Le principe de fonctionnement est basique :

- **Détection d'obstacles** : Lorsque le robot détecte un obstacle à l'aide de ses capteurs (le LiDAR dans notre cas), le compteur incrémente sa valeur et enregistre le nombre d'obstacles rencontrés ;
- **Changement de direction** : Chaque fois qu'un obstacle est détecté, le robot change de direction pour éviter la collision. Le compteur affiche alors la valeur actuelle (1, 2, 3, etc.) correspondant au nombre d'obstacles rencontrés jusqu'à présent ;
- **Suivi du chemin parcouru** : Le compteur aide à suivre le nombre de déviations du chemin initial dues aux obstacles. Cela permet de savoir combien de fois le robot a dû ajuster sa trajectoire pour continuer sa navigation ;
- **Retour à la navigation normale** : Si le robot avance sans rencontrer de nouveaux obstacles, le compteur décrémente une valeur jusqu'à être réinitialisé à 0, indiquant qu'il suit maintenant une trajectoire dégagée et sans obstacle imminent.

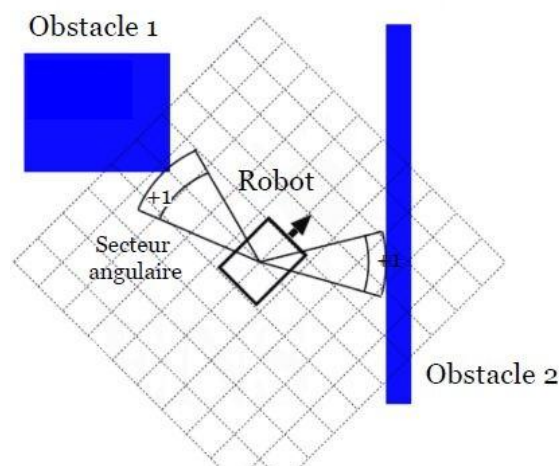


Figure III.2 Principe d'incrément des obstacles ; Source : [48]

### III.3.1.2 Rôle du compteur dans notre système de guidage

Ce mécanisme de comptage des obstacles rencontrés et les changements de direction effectués par le robot offre plusieurs avantages :

- **Suivi des interactions avec l'environnement** : Le compteur enregistre chaque rencontre avec un obstacle, permettant au système de navigation de suivre combien de fois le robot a dû ajuster sa trajectoire. Cela fournit des informations précieuses sur la densité et la distribution des obstacles dans l'environnement [49] ;
- **Amélioration de la robustesse de la navigation** : En comptant les obstacles et les changements de direction, le robot peut adapter ses stratégies de navigation pour devenir plus efficace. Par exemple, si un nombre élevé de changements de direction est détecté, le robot peut ajuster ses paramètres de navigation pour mieux gérer des environnements [48] ;
- **Feedback et Diagnostic** : Le compteur permet de diagnostiquer et de comprendre les comportements de navigation du robot. En analysant les données du compteur, nous pouvons identifier les zones problématiques et optimiser les algorithmes [50].

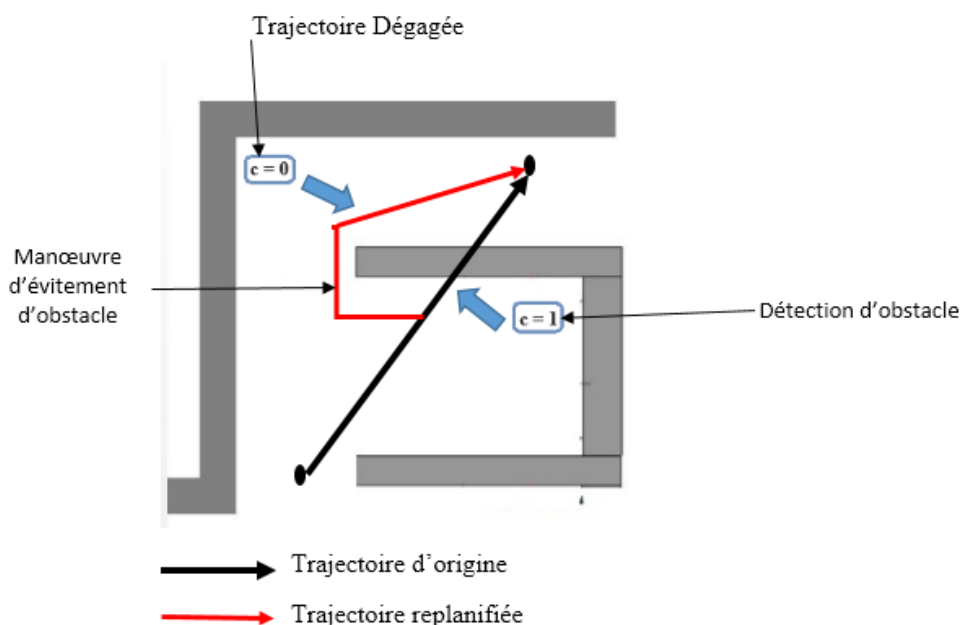


Figure III.3 Rôle du compteur dans la planification de la trajectoire

## III.4 Architecture du système

Les principes fondamentaux et les éléments clés de l'architecture de notre système de guidage sont :

### III.4.1 Modèle de robot

Nous avons choisi le robot *Turtlebot3* Burger parce que les packages ROS pour ce robot fournissent une variété de fonctionnalités pour contrôler et interagir avec le robot et qu'il présente plusieurs avantages qui vont faciliter la conception de notre système de guidage [51] :

- ✓ **Capteurs avancés** : Turtlebot3 est équipé de capteurs avancés tels que LIDAR, IMU, RGB-D, et encodeurs ;
- ✓ **Cadre logiciel standard** : Ce qui nous donne accès à une multitude de possibilités offertes par la plateforme ROS comme la variété de logiciels tiers dont Python3 ;
- ✓ **Facilité d'utilisation** : L'utilisation de Turtlebot3 est plus conviviale grâce aux améliorations continues ;
- ✓ **Expériences pratiques** : L'usage de Turtlebot3 renforcera nos compréhension des concepts théoriques servira d'expériences pratiques pour des carrières futures dans le domaine de la robotique.

#### III.4.1.1 Présentation matériel (Hardware) [51]

Les principaux composants matériels de la plateforme Turtlebot3 Burger sont :

- **LIDAR** : Un scanner laser à 360 degrés est monté sur le dessus du robot. Ce capteur émet un laser rotatif qui se reflète sur les obstacles proches ;
- **Raspberry Pi** : Une carte monocarte Raspberry Pi (RPi) est située sous le LIDAR. Elle lit les données des capteurs et communique avec un ordinateur principal qui effectue les calculs réels ;
- **OpenCR** : Sous le RPi, se trouve la carte de contrôle matérielle. Cette carte contient une IMU avec un accéléromètre à trois axes, un gyroscope et un magnétomètre. Sa principale fonction est de connecter le Raspberry Pi aux moteurs et aux capteurs, et de fournir des connexions d'alimentation pour tous les composants ;
- **Moteurs et batterie** : Le modèle "Burger" de Turtlebot3 a une configuration d'entraînement différentielle, avec 2 moteurs entraînés indépendamment et 1 roue folle pour la stabilité.

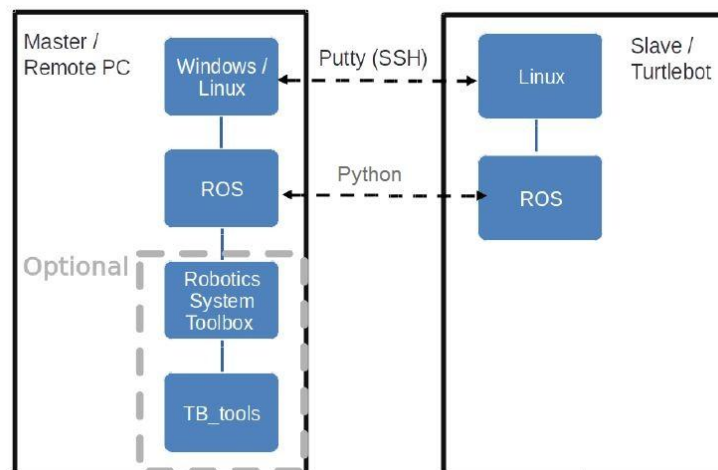


*Figure III.4 Aperçu de TurtleBot3 Burger ; Source [51]*

### III.4.1.2 Présentation logiciel (Software) [51]

Turtlebot3 utilise un ensemble de logiciels spécifiques pour son fonctionnement :

- **Système d'exploitation** : Le Raspberry Pi de Turtlebot3 fonctionne sous Ubuntu de la distribution Linux. Il est mieux adapté aux ordinateurs monocartes en raison de ses exigences matérielles réduites ;
- **Cadre logiciel ROS** : Turtlebot3 utilise ROS qui offre une architecture flexible pour le développement de logiciels robotiques, facilitant la programmation, la simulation et le contrôle des robots ;
- **Améliorations logicielles** : Des modifications peuvent être apportées à la configuration de Turtlebot3 pour améliorer la convivialité, la stabilité et la simplification d'utilisation.



*Figure III.5 Architecture Software de TurtleBot ; Source [51]*

### III.4.2 LiDAR 2D

Le modèle de LIDAR utilisé sur le TurtleBot3 Burger est le LDS-01 [52]. Voici ces spécifications :

Modèle : LDS-01  
Plage de Détection : 0.12m à 3.5m  
Angle de Scan : 360 degrés  
Précision :  $\pm 30$ mm

### III.4.3 Compteur

Le compteur des obstacles fait partie du programme exécuté sur la Raspberry Pi qui agit comme le cerveau du robot, traitant les données des capteurs et exécutant les algorithmes de navigation et de contrôle. Voici les spécifications de la Raspberry Pi :

Modèle : Raspberry Pi 3 Model B+  
Processeur : Quad-core ARM Cortex-A53  
RAM : 1GB  
Système d'Exploitation : Ubuntu avec ROS

### III.4.4 Communication

Dans notre programme de contrôle du robot TurtleBot3, la *communication inter-processus* est gérée par ROS, qui facilite l'échange de données entre différents nœuds via un système de messagerie.

Les principaux nœuds de notre programme (comme nous allons le voir la partie logiciel) incluent la publication des commandes de vitesse (`/cmd_vel`), la souscription aux données du capteur laser (`/scan`), et la souscription aux données de l'odométrie (`/odom`).

Les messages LaserScan reçus permettent d'éviter les obstacles, tandis que les données d'odométrie sont utilisées pour la navigation et le suivi de la position du robot. La communication asynchrone via ces topics assure une coordination fluide entre les capteurs et les commandes de mouvement, permettant au robot de naviguer efficacement dans son environnement.

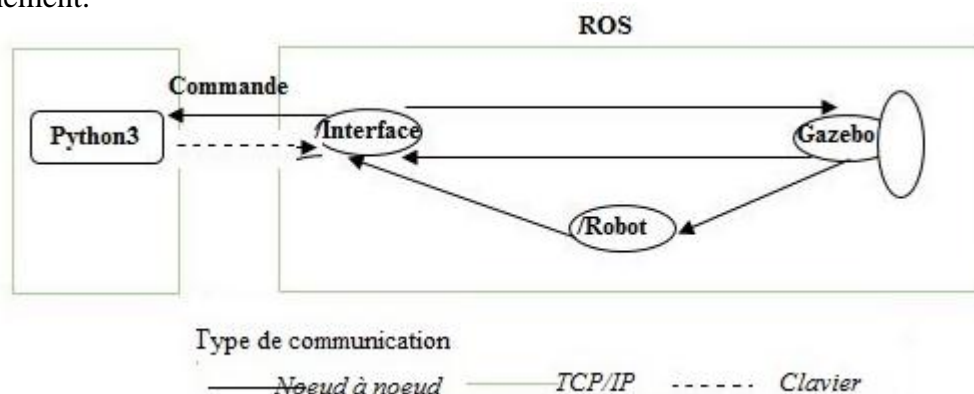


Figure III.6 Diagramme de communication des nœuds ROS ; [53]

### III.4.5 Interfaces

- **Interfaces utilisateur** : Dans notre programme les interfaces utilisateur sont principalement implémentées à l'aide de *RViz* et *rqt*, deux outils fournis par ROS.
  - ✓ *RViz* est utilisé pour la visualisation des données de capteurs, nous permettant de voir en temps réel les informations du LIDAR, la position du robot, et d'autres données essentielles.
  - ✓ *Rqt*, une suite d'outils GUI (Graphical User Interface) pour ROS, permet de contrôler et configurer le robot via des interfaces graphiques conviviales. Ces outils facilitent l'interaction et le diagnostic, offrant une vue d'ensemble du fonctionnement et de l'état du robot pendant les opérations.
- **Interfaces de programmation** : Les interfaces de programmation dans notre programme utilisent les API ROS en Python3, permettant le développement de modules personnalisés pour le **contrôle** du robot et le traitement des données. Les principales interfaces de programmation incluent la gestion des topics pour la communication inter-processus, les services pour les requêtes synchrones, et les actions pour les tâches complexes nécessitant des retours d'état. Les API ROS facilitent l'implémentation de fonctionnalités avancées comme l'évitement d'obstacles, la navigation autonome, et l'intégration de capteurs, tout en assurant la modularité et la réutilisabilité du code.

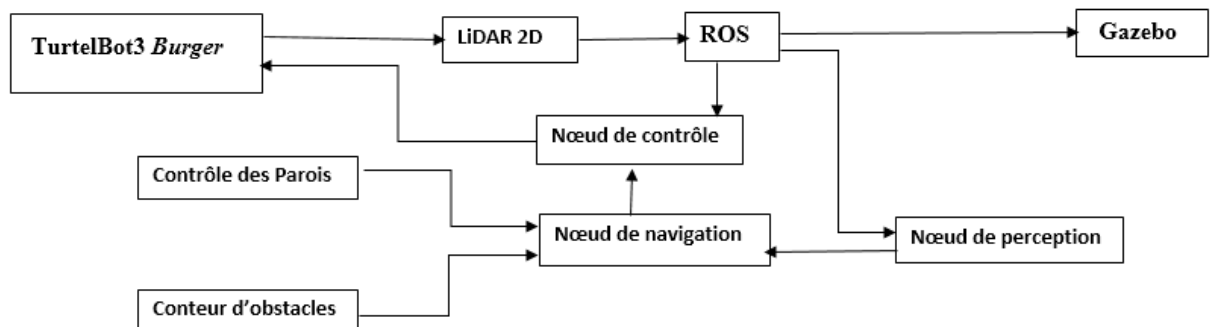


Figure III.7 Architecture du système de guidage

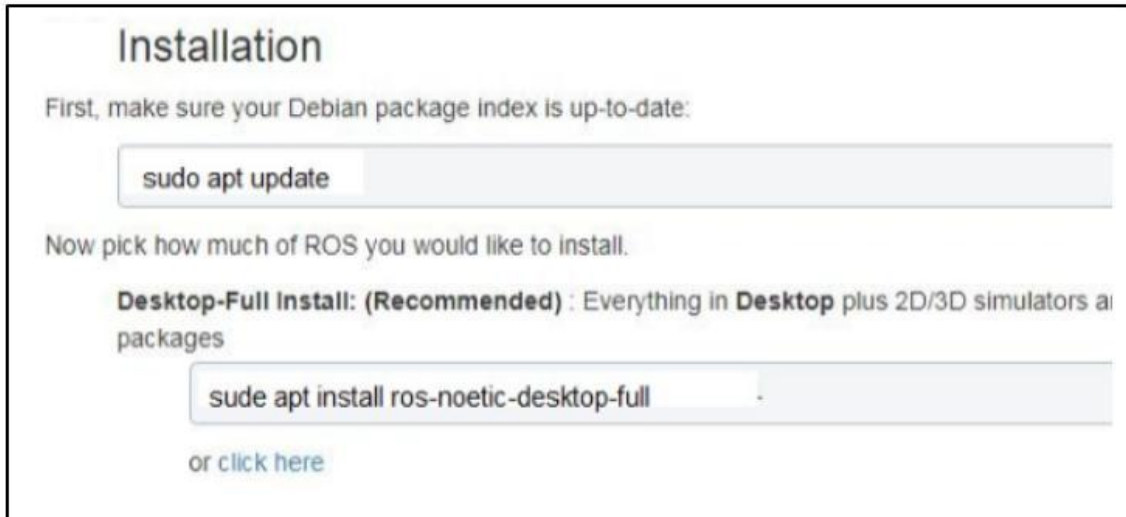
## III.5 Logiciel de guidage

Avant le développement du système de guidage nous devons procéder à l'installation des packages requis pour ROS et Gazebo. Comme nous l'avons déjà mentionné, Nous utilisons ROS Noetic, qui est compatible avec Ubuntu 20.04.



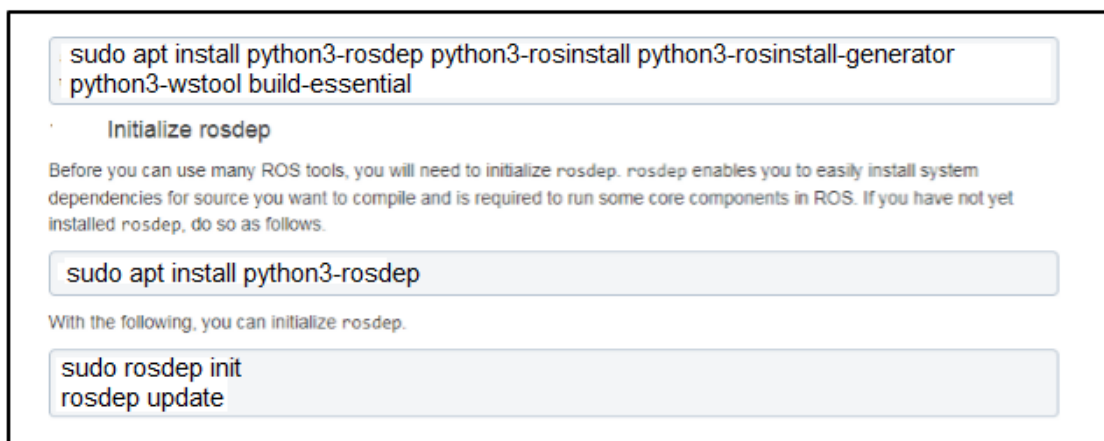
### III.5.1 Installation des packages

- **Installation de ROS Noetic** : L'installation se fait en suivant les instructions de [wiki.ros.org](http://wiki.ros.org), comme démontré dans la figure IV.5 suivante.



*Figure III.8 Installation de ROS Noetic ; [54]*

- **Installation des dépendances de ROS** : ROS Noetic est conçu pour fonctionner avec Python 3 par défaut car les outils et bibliothèques de ROS sont installés avec. Cependant, il est conseillé comme bonne pratique de vérifier et d'installer les packages Python 3 nécessaires pour s'assurer que toutes les dépendances sont satisfaites, comme indiqué dans la figure qui suit :



*Figure III.9 Installation de Python3 ; [54]*

- **Installation des packages TurtleBot3 :**

```
user@ubuntu:~$ sudo apt install
ros-noetic-turtlebot3
ros-noetic-turtlebot3-simulations
ros-noetic-turtlebot3-navigation
```

- **Installation de Gazebo :** Gazebo est inclus dans les installations de ROS Noetic. Pour s'assurer que nous avons les dernières versions des packages nécessaires :

```
user@ubuntu:~$ sudo apt update
sudo apt install
ros-noetic-gazebo-ros-pkgs
ros-noetic-gazebo-ros-control
```

### III.5.2 Configuration du système

- **Configuration de l'environnement :** Pour définir le modèle Burger de TurtleBot3 :

```
user@ubuntu:~$ echo "export TURTLEBOT3_MODEL=burger">>
~/.bashrc
source ~/.bashrc
```

- **Création de l'espace de travail Catkin :**

```
user@ubuntu:~$ mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/
catkin_make
source devel/setup.bash
```

### III.5.3 Développement des algorithmes de guidage

Le développement du logiciel de guidage comprend deux parties principales : l'algorithme de contrôle des parois et le compteur d'obstacles.

#### III.5.3.1 Algorithme du contrôle de paroi

Pour permettre au RMA de se déplacer en suivant les parois tout en évitant les obstacles nous avons développé l'algorithme en créant un fichier Python `wall_following.py` dans le dossier `~/catkin_ws/src/turtlebot3_navigation/src/`. La figure III.10 représente une partie du programme.

```

142 self.turn_angle += delta_angle
143 turn_angle += delta_angle
144 last_angle = rotation
145 self.stop_robot()
146
147 def follow_wall(self):
148     ranges = self.laser_msg.ranges
149
150     if not ranges or len(ranges) < 271:
151         rospy.logwarn("Laser scan data is not sufficient")
152         self.stop_robot()
153         return
154
155     front_distance = min(ranges[:10] + ranges[-10:])
156     left_distance = min(ranges[80:100])
157     right_distance = ranges[270]
158     front_max_distance = max(ranges[:10] + ranges[-10:])
159
160     rospy.loginfo(f"Front max distance: {front_max_distance}, Left distance: {left_distance}, Right distance: {right_distance}")
161
162     if all(d == float('inf') for d in ranges):
163         self.stop_robot()
164         rospy.signal_shutdown("Robot is out of the map")
165         return
166
167     if front_max_distance < self.safe_distance:
168         self.stop_robot()
169         self.rotate_degrees(-85)
170         self Obst.Cnt += 1

```

Figure III.10 Programmation du contrôle de parois

Fonctionnement de l’algorithme contrôle de paroi :

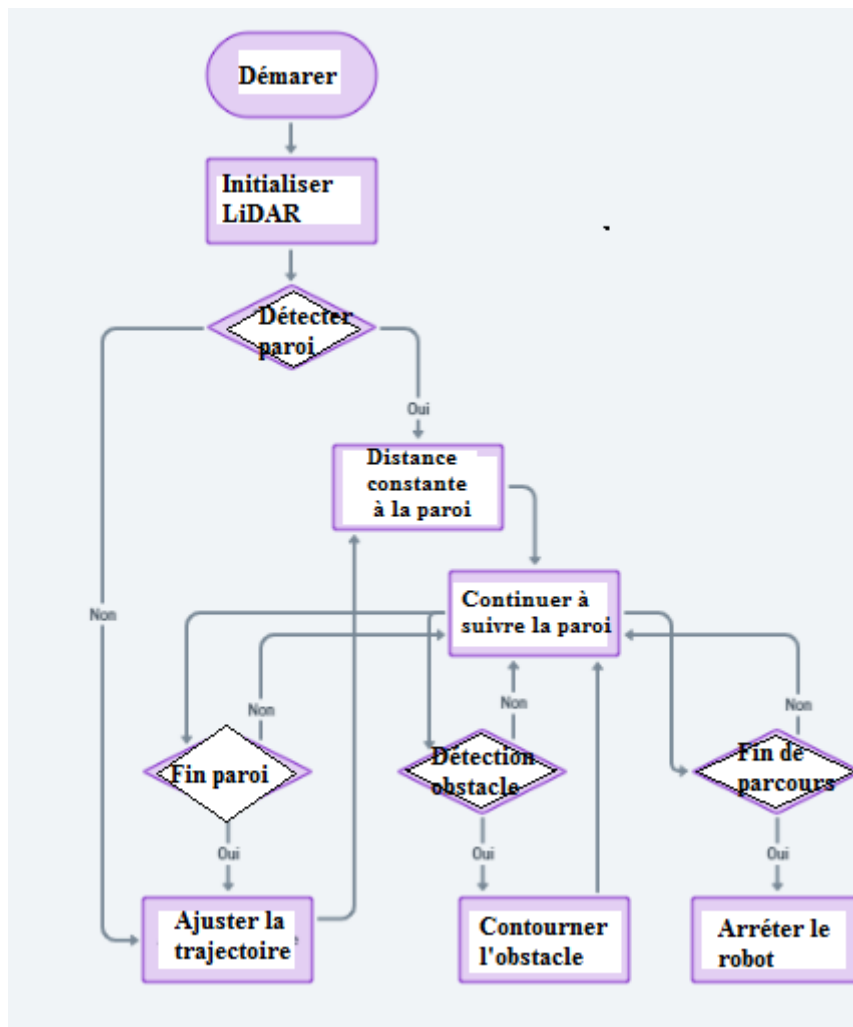


Figure III.11 Schéma du diagramme contrôle de parois

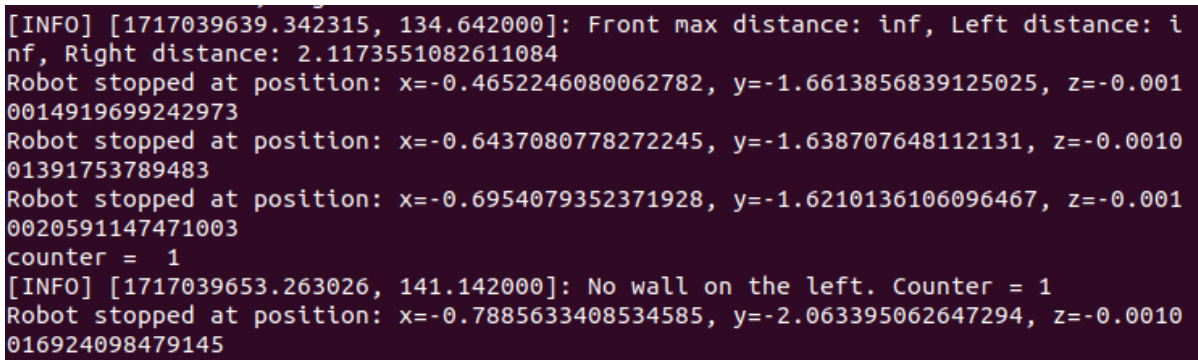
### III.5.3.2 Compteurs d'obstacles

L'algorithme du compteur d'obstacles doit incrémenter chaque fois que le robot détecte un obstacle et décrémenter à chaque fois que la voie est libre jusqu'à ce que le compteur soit à la valeur zéro pour continuer son parcours. On crée un fichier Python `obstacle_counter.py` dans le même dossier :



```
163 self.stop_robot()
164 rospy.signal_shutdown("Robot is out of the map")
165 return
166
167 if front_max_distance < self.safe_distance:
168     self.stop_robot()
169     self.rotate_degrees(-85)
170     self.Obst_Cntr += 1 ← Incréméntation du compteur "C"
171     print("counter = ",self.Obst_Cntr)
172     rospy.logInfo(f"Obstacle detected in front. Counter = {self.Obst_Cntr}")
173
174 elif left_distance > 3:
175     self.stop_robot()
176     self.move_straight_time("forward", 0.3, 0.5)
177     self.rotate_degrees(91)
178     self.Obst_Cntr -= 1 ← Décréméntation du compteur "C"
179     print("counter = ",self.Obst_Cntr)
180     rospy.logInfo(f"No wall on the left. Counter = {self.Obst_Cntr}")
181     self.move_straight_time("forward", 0.3, 2)
182     if self.Obst_Cntr == 0:
183         print("The robot stopped, counter = 0")
184         self.stop_robot()
185         rospy.signal_shutdown("Obstacle counter is zero. Stopping the robot and simulation.")
186     return
187
```

Figure III.12 Programmation du compteur d'obstacles



```
[INFO] [1717039639.342315, 134.642000]: Front max distance: inf, Left distance: i
nf, Right distance: 2.1173551082611084
Robot stopped at position: x=-0.4652246080062782, y=-1.6613856839125025, z=-0.001
0014919699242973
Robot stopped at position: x=-0.6437080778272245, y=-1.638707648112131, z=-0.0010
01391753789483
Robot stopped at position: x=-0.6954079352371928, y=-1.6210136106096467, z=-0.001
0020591147471003
counter = 1
[INFO] [1717039653.263026, 141.142000]: No wall on the left. Counter = 1
Robot stopped at position: x=-0.7885633408534585, y=-2.063395062647294, z=-0.0010
016924098479145
```

Figure III.13 Exécution du compteur dans ROS

### Fonctionnement de l'algorithme du compteur :

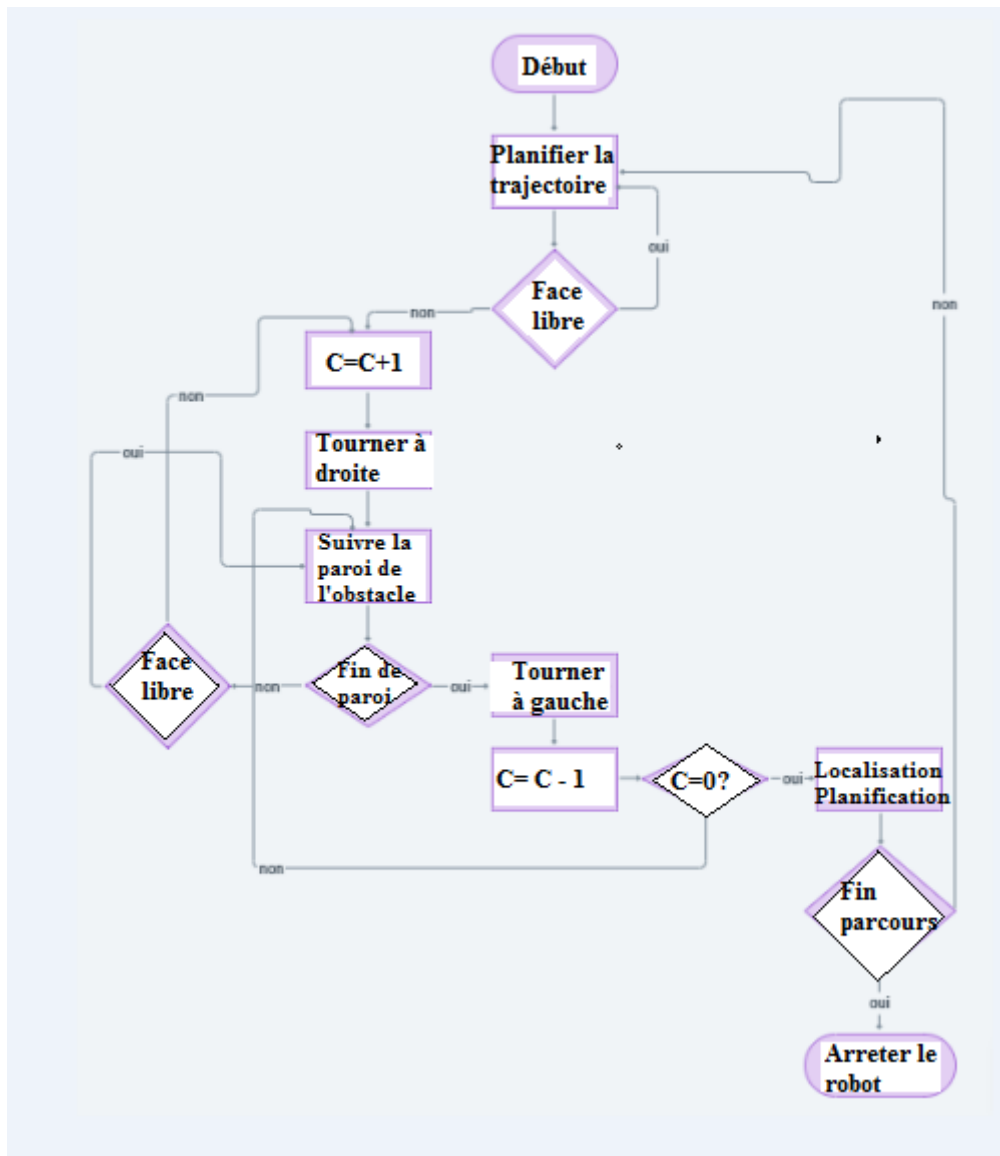


Figure III.14 Schéma de l'algorithme compteur

#### III.5.4 Fonctionnement prévu

Dans cette section, nous détaillons les attentes pour le fonctionnement de notre système de guidage pour le RMA dans des environnements intérieurs mal connus.

Comme nous l'avons mentionné notre système repose sur deux éléments clés : le contrôle de paroi et le comptage des obstacles, avec une capacité d'évitement d'obstacles intégrée.

### III. 5.4.1 Contrôle de paroi

Le RMA se déplacera en maintenant une distance constante par rapport aux parois environnantes. Voici les comportements attendus :

- **Détection de la paroi** : Utilisation du LiDAR 2D pour détecter les parois environnantes et maintenir une distance constante définie ;
- **Ajustement de la trajectoire** : Si le robot s'écarte de la paroi, il ajuste sa trajectoire en modifiant l'angle de virage pour revenir à la distance souhaitée.
- **Navigation fluide** : Le robot doit naviguer de manière fluide le long des parois, en évitant les obstacles tout en maintenant une trajectoire stable et prévisible.

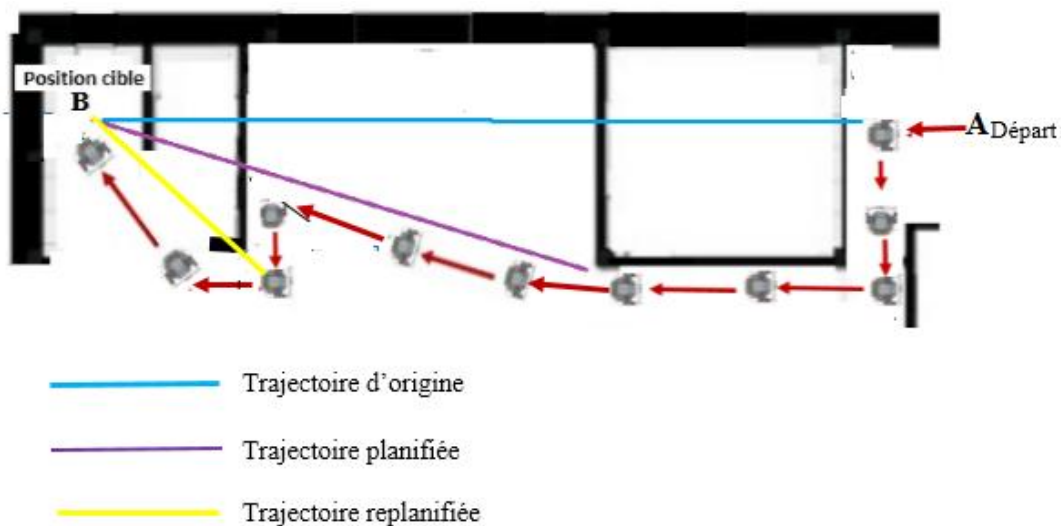
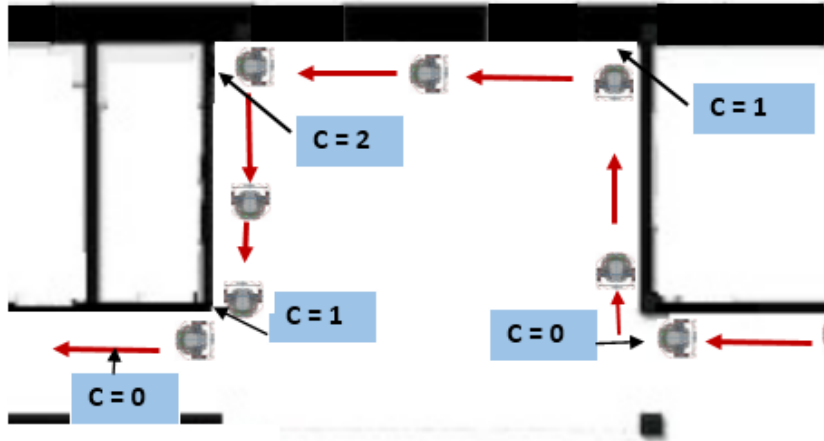


Figure III.15 Navigation prévu en contrôle de parois

### III.5.4.2 Comptage des obstacles

Par ce mécanisme le robot sera en mesure de suivre le nombre d'obstacles rencontrés pendant la navigation. Voici les comportements attendus :

- **Incrémentation du compteur** : Chaque fois qu'un obstacle est détecté à une distance critique, le compteur incrémente sa valeur.
- **Affichage des obstacles** : Le robot affiche le nombre actuel d'obstacles rencontrés, fournissant un feedback sur la densité des obstacles dans l'environnement.
- **Suivi des changements de direction** : En plus du comptage des obstacles, le système suit les changements de direction nécessaires pour éviter les obstacles.

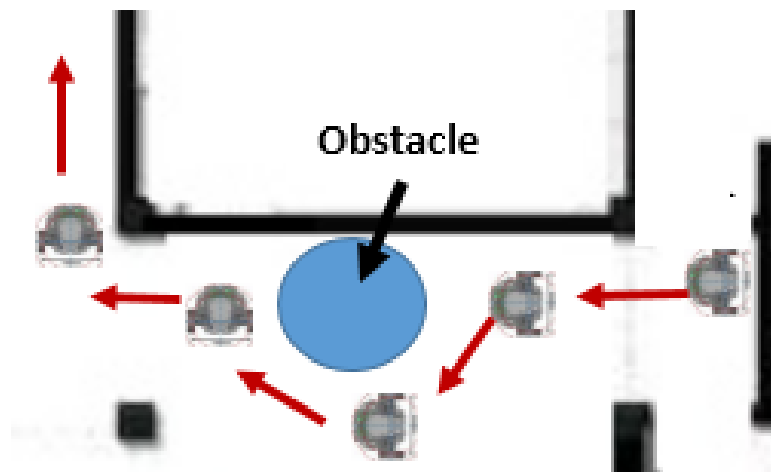


*Figure III.16 Fonctionnement prévu pour le compteur d'obstacles*

### III.5.4.3 Evitement d'obstacles

L'évitement d'obstacle est crucial pour garantir la sécurité et l'efficacité de la navigation. Voici les comportements attendus :

- **Détection en temps réel** : Utilisation du LiDAR 2D pour détecter les obstacles en temps réel.
- **Prise de décision rapide** : Lorsque le robot détecte un obstacle, il prend des décisions rapides pour ajuster sa vitesse et son angle de virage afin d'éviter la collision.
- **Reprise de la trajectoire** : Après avoir évité un obstacle, le robot reprend sa trajectoire initiale le long de la paroi, en ajustant si nécessaire.



*Figure III.17 Comportement prévu en cas d'obstacle*

## III.6 Simulation

Après avoir conçu et implémenté notre système de guidage pour le RMA, nous allons valider son fonctionnement à travers une série de tests en environnement simulé. Dans cette section, nous recréerons des environnements intérieurs variés et complexes.

Ces simulations nous permettront de vérifier l'efficacité de notre stratégie de guidage pour le RMA d'intérieur dans un environnement mal connu, et d'identifier d'éventuelles améliorations nécessaires.

Les résultats obtenus fourniront des preuves concrètes de la robustesse et de la fiabilité de notre système de guidage dans des conditions réalistes.

### III.6.1 Environnement du guidage

Pour tester notre système de guidage, nous avons créé une carte d'environnement de simulation sous Gazebo, représentant le laboratoire 201, pavillon 26 de notre département qui est un espace intérieur complexe avec divers obstacles et couloirs, permettant de reproduire des conditions réalistes pour le RMA.

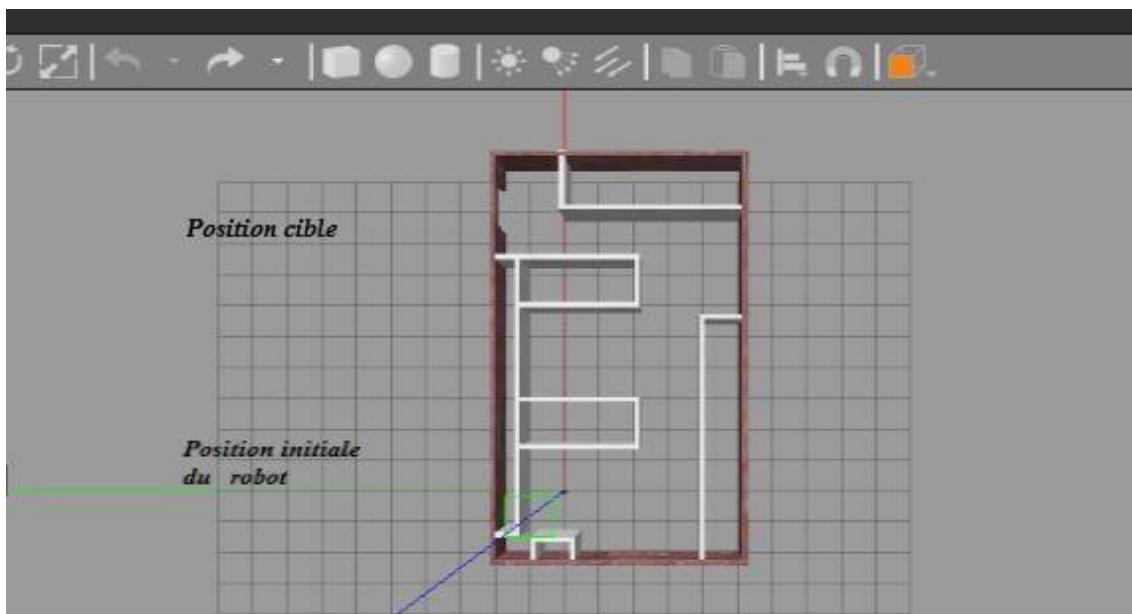


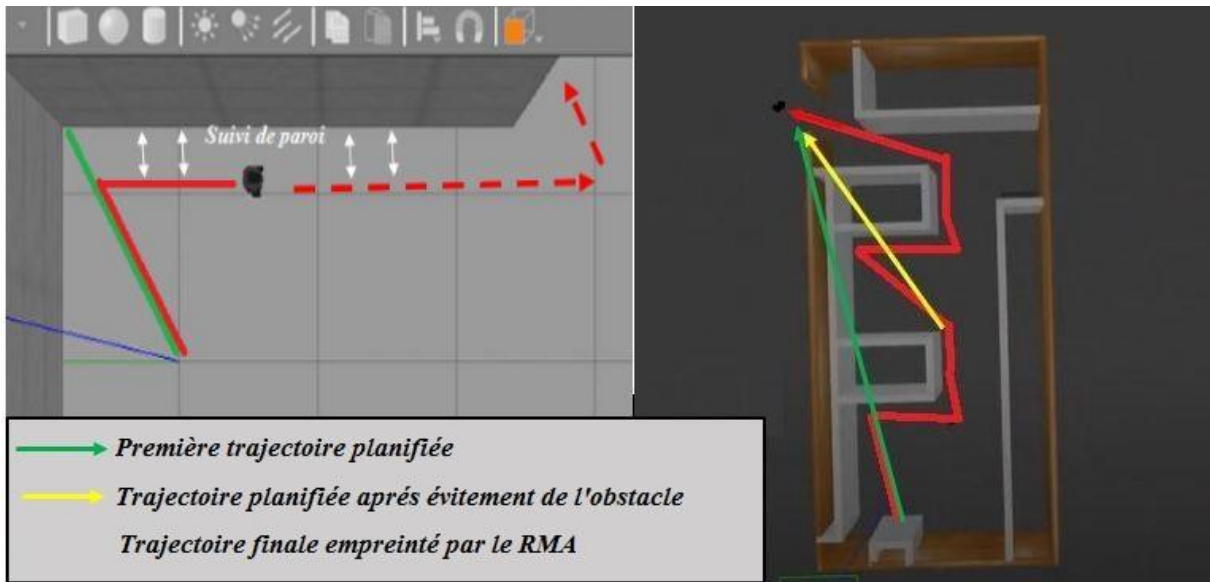
Figure III.18 Croquette de vue-dessus du labo 201 PAV 26

### III.6.2 Scénarios de simulation

#### III.6.2.1 Guidage en contrôle de parois

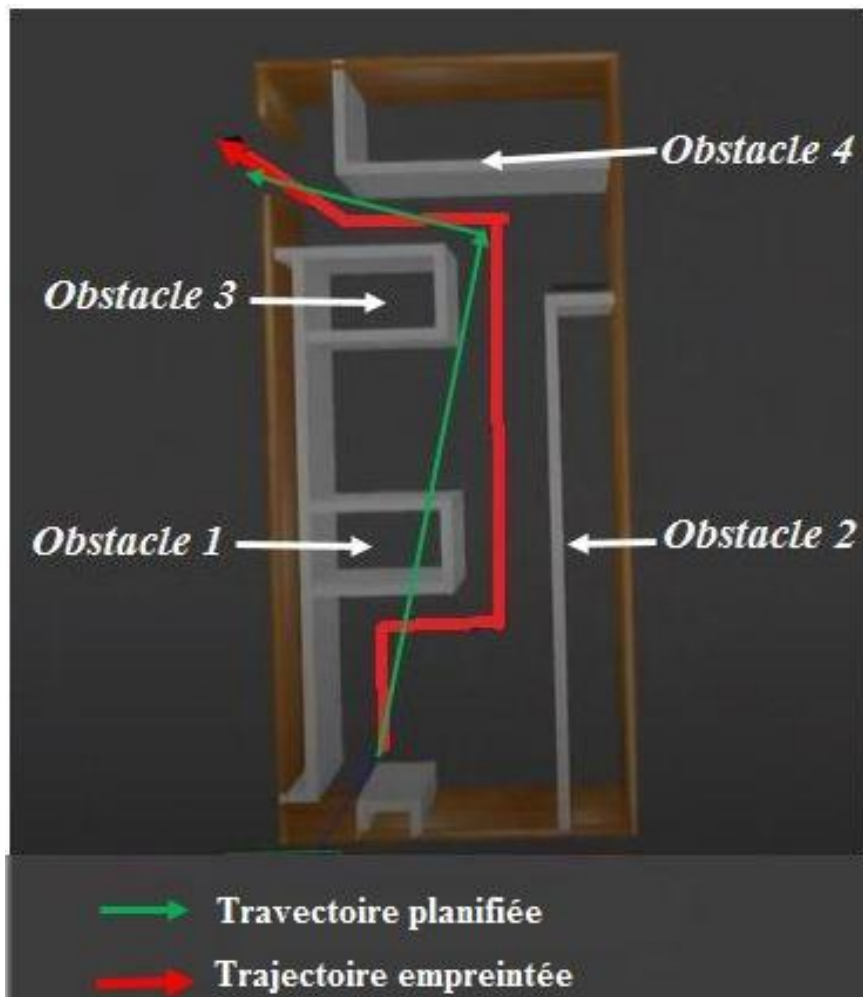
**A-** En premier nous avons testé notre système de guidage dans une carte sans obstacles (à part les parois) afin de vérifier l'efficacité de notre méthode de contrôle de paroi et de s'assurer qu'elle permet au robot de naviguer avec fluidité le long des parois.





*Figure III.19 Comportement du RMA en mode contrôle de paroi*

**B-** Nous avons également évalué la précision de notre algorithme en modifiant le programme afin de définir explicitement la position des obstacles, à l'exception de l'obstacle 1 (voir figure III.20) que nous avons gardé inconnu. Le robot a ensuite établi une trajectoire en tenant compte des obstacles et a navigué avec fluidité et précision vers la destination cible.



*Figure III.20 Comportement du RMA en cas d'obstacles connus*

### III.6.2.2 Test du compteur

Pour évaluer la performance du compteur d'obstacles, nous avons modifié notre environnement de simulation afin de vérifier la crédibilité du compteur dans l'évitement d'obstacles. Cela permet de tester comment le système détecte et comptabilise les obstacles rencontrés tout en naviguant dans un environnement plus réaliste.

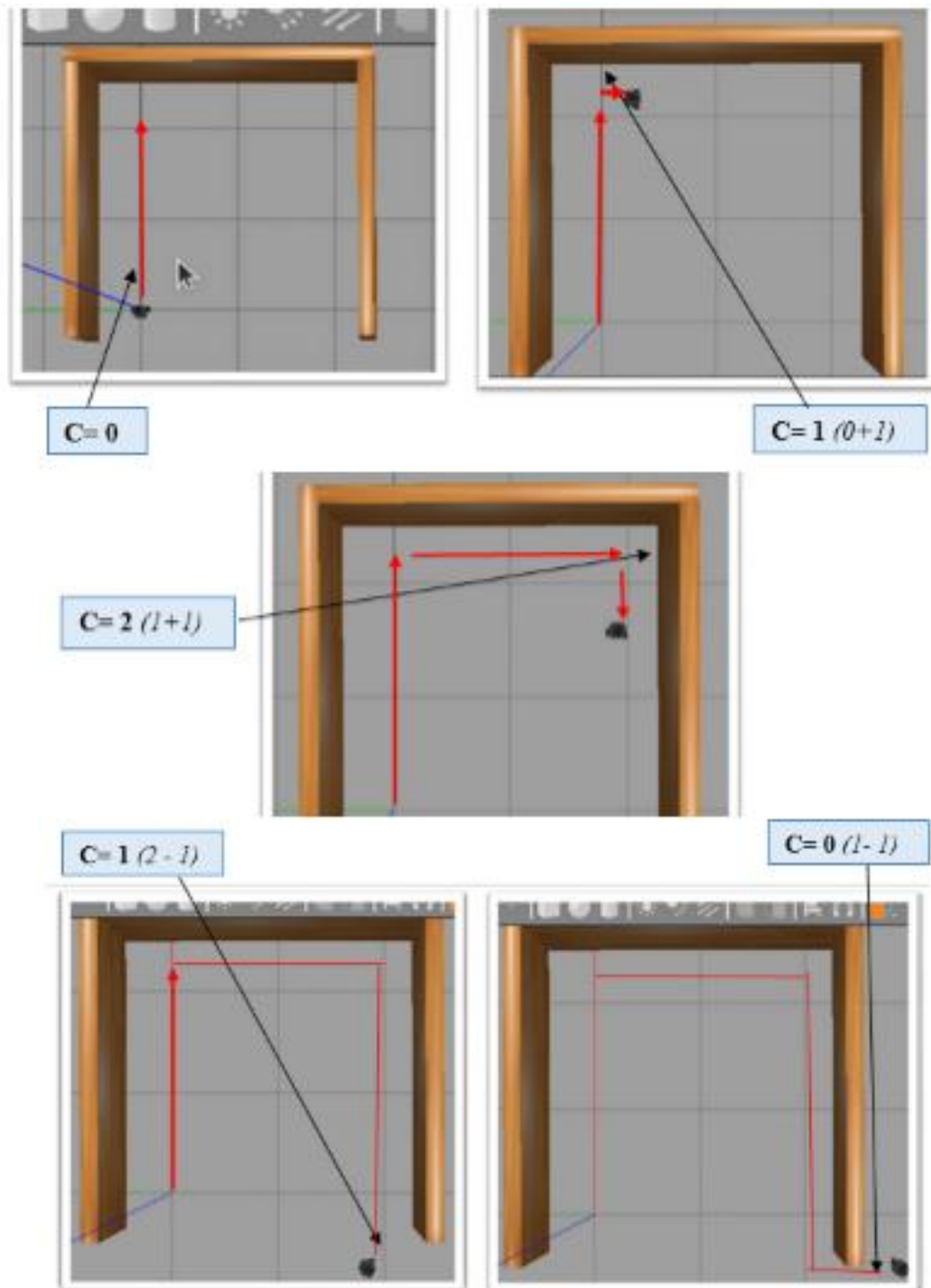
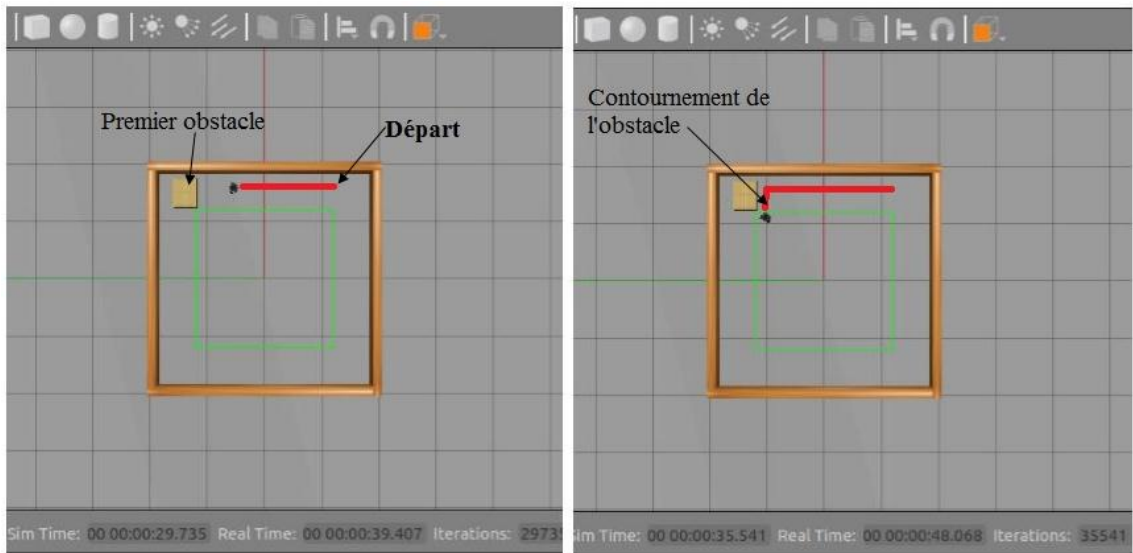


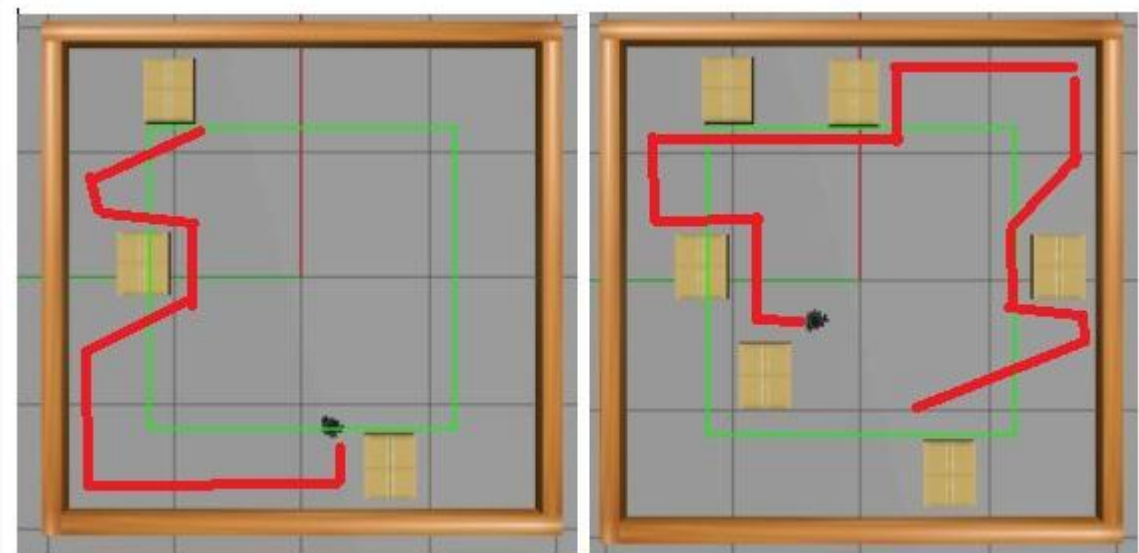
Figure III.21 Simulation test du compteur d'obstacle

### III.6.2.3 Guidage avec exploration de l'environnement

Dans les scénarios précédents, nous avons testé notre stratégie de guidage dans un environnement partiellement connu en guidant le robot vers un point cible. Maintenant, nous allons évaluer notre système dans un environnement totalement inconnu. Pour cela, nous avons simulé un intérieur fermé sans point d'arrêt et surveillé le comportement du RMA face à des obstacles.



*Figure III.22 Scénario avec un seul obstacle*



*Figure III.23 Scénario avec plusieurs obstacles*

### III.6.3 Résultats et discussion

Les résultats des simulations montrent que le système de guidage a réussi à naviguer efficacement dans tous les scénarios testés :

- Premièrement, dans des conditions où aucun obstacle n'était présent, le RMA a pu naviguer le long des parois avec fluidité, validant ainsi l'efficacité de notre méthode de contrôle de paroi. Cette validation préliminaire a confirmé que notre algorithme de guidage fonctionne comme prévu dans des situations dépourvues d'obstacles autres que les parois.
- Ensuite, nous avons évalué la capacité du système à détecter et à éviter des obstacles tout en maintenant une trajectoire cohérente grâce au compteur. Nous avons constaté que le compteur d'obstacles intégré a bien fonctionné, permettant au robot de naviguer de manière sécurisée en comptabilisant les obstacles rencontrés et en ajustant sa trajectoire en conséquence.
- Dans la phase de test de notre système de guidage avec exploration de l'environnement, nous avons observé des résultats encourageants :
  - ✓ Lorsque le robot est confronté à un environnement totalement inconnu, il a démontré une capacité remarquable à éviter les obstacles et à s'orienter vers la paroi la plus proche.
  - ✓ Plus important encore, lorsque plusieurs obstacles étaient présents, le comportement du RMA a démontré une capacité d'adaptation impressionnante. La trajectoire du robot a évolué de manière dynamique pour contourner efficacement les obstacles tout en restant orienté vers la paroi la plus proche, comme prévu dans notre stratégie de guidage.

Ces résultats soulignent la validité de notre système de guidage dans des environnements variés et inconnus. La capacité RMA à éviter les obstacles et à naviguer de manière autonome ouvre la voie à des applications pratiques dans des contextes intérieurs nécessitant un guidage précis dans des environnements confinés.

### **III.7 Conclusion du chapitre**

Dans ce chapitre, nous avons présenté la conception et la simulation d'un système de guidage pour un RMA utilisant un LiDAR 2D. Nous avons détaillé notre approche de navigation basée sur le suivi de paroi et l'évitement d'obstacles, ainsi que l'intégration d'un compteur d'obstacles pour améliorer la gestion des interactions du robot avec son environnement.

Les simulations réalisées ont confirmé la validité de notre approche. Le robot a démontré sa capacité à naviguer efficacement dans différents scénarios, de l'évitement des parois à la navigation dans un environnement inconnu.

## *Conclusion générale*

---

Pour répondre à la demande croissante de systèmes autonomes capables de naviguer efficacement dans des environnements intérieurs complexes, nous avons entrepris ce projet visant à élaborer une stratégie de guidage pour un robot mobile d'intérieur dans un environnement mal connu. Pour réaliser ce projet, nous sommes partis de deux hypothèses fondamentales :

- ✓ Premièrement, nous avons supposé que l'utilisation de la technologie LiDAR pour la perception de l'environnement fournirait des données précises et en temps réel pour une navigation autonome ;
- ✓ Deuxièmement, nous avons envisagé que la plateforme logicielle ROS offrirait un cadre robuste pour la gestion et le contrôle du robot, tandis que le simulateur 3D Gazebo permettrait de tester et de valider les algorithmes de navigation dans un environnement virtuel réaliste.

Pour confirmer ou infirmer ces hypothèses, nous avons en premier exploré les concepts clés de la robotique moderne, notamment l'importance de l'intelligence artificielle et de l'apprentissage automatique dans l'évolution de la navigation autonome et des technologies de perception, avec un accent particulier sur le LiDAR comme outil fondamental.

L'approfondissement de l'exploration du guidage de robot, nous a permis de souligner son rôle crucial dans la navigation autonome, ainsi que les différentes approches d'évitement d'obstacles. En détaillant l'importance de ROS et Gazebo dans le développement de systèmes robotiques robustes et efficaces. Ces outils offrent une infrastructure logicielle et une plateforme de simulation complètes.

Nous avons ensuite conçu et simulé notre système de guidage de RMA en nous appuyant sur une méthode de suivi de paroi et d'évitement d'obstacles. Cette approche, faisant partie de la navigation réactive, est particulièrement adaptée aux environnements intérieurs complexes ou mal connus. Combinée à l'utilisation du LiDAR 2D, cette méthode permet au RMA de maintenir une distance constante par rapport aux parois environnantes, assurant une navigation stable et efficace même en présence d'obstacles imprévus.

Pour améliorer la gestion des interactions du RMA avec son environnement, nous avons intégré à l'algorithme de guidage un compteur d'obstacles. Grâce à cette approche, le RMA peut non seulement éviter les obstacles en temps réel, mais aussi ajuster sa stratégie de navigation en fonction de la densité des obstacles détectés.

Les résultats des simulations montrent que le système de guidage a réussi à naviguer efficacement dans tous les scénarios testés :

- En absence totale d'obstacles, le RMA a navigué le long des parois de manière fluide, démontrant ainsi l'efficacité de notre méthode de contrôle de paroi. Cette validation initiale a confirmé que notre algorithme de guidage fonctionne comme prévu dans des situations où seuls les murs sont présents ;
- Le compteur d'obstacles intégré s'est avéré efficace pour détecter et éviter les obstacles tout en maintenant une trajectoire cohérente. Le RMA a navigué en toute sécurité en prenant en compte les obstacles rencontrés et en ajustant sa trajectoire en conséquence ;
- Confronté à un environnement totalement inconnu, le robot a montré une capacité remarquable à éviter les obstacles ;
- Confronté à plusieurs obstacles, le RMA a montré une remarquable capacité à s'adapter. Sa trajectoire a été ajustée de façon dynamique pour naviguer efficacement autour des obstacles conformément à notre stratégie de guidage.

Les simulations effectuées sous ROS et Gazebo ont ainsi validé l'efficacité de notre stratégie de guidage. Le RMA a démontré sa capacité à naviguer de manière fluide le long des parois, à éviter les obstacles et à explorer un environnement totalement inconnu.

Les résultats obtenus montrent que notre approche est non seulement viable, mais également adaptable à des environnements variés et complexes, ouvrant ainsi la voie à des applications pratiques dans des contextes intérieurs nécessitant une navigation autonome précise et fiable.

Avec ce projet, nous espérons avoir contribué modestement au développement de la robotique, en démontrant le potentiel des RMA dans des applications concrètes et en ouvrant de nouvelles perspectives pour leur utilisation dans des environnements intérieurs variés, complexes et inconnu comme :

- ✓ Le nettoyage automatique : Grâce à la capacité de suivre les parois, le robot peut assurer un nettoyage complet et uniforme de l'espace intérieur ;
- ✓ La surveillance et la sécurité : En patrouillant le long des parois le RMA peut surveiller l'environnement domestique et détecter les intrusions ;
- ✓ Assistance aux personnes âgées ou handicapées : Les RMA d'assistance peuvent accompagner les personnes vulnérables dans leur maison. En offrant un soutien physique et une compagnie constante.

Enfin nous espérons que ces résultats ouvrent la voie à des recherches futures visant à perfectionner davantage les capacités de navigation autonome des RMA d'intérieur.

Des recherches futures pourraient traiter l'intégration de capteurs supplémentaires pour une perception plus fine de l'environnement, l'exploration de techniques de planification de trajectoire plus avancées, ou encore l'adaptation de ces systèmes à des environnements en constante évolution.

## Références bibliographiques

---

- [1] Espiau, Bernard & Oudeyer, Pierre-Yves. (2008). Robotique : de l'automate à l'humanoïde par Bernard Espiau. Un robot très curieux, entretien avec Pierre-Yves Oudeyer, propos recueillis par Dominique Chouchan.. La Recherche. Les Cahiers de l'Inria.  
[https://www.researchgate.net/publication/47783626\\_Robotique\\_de\\_l'automate\\_a\\_l'humanoïde\\_par\\_Bernard\\_Espiau\\_Un\\_robot\\_tres\\_curieux\\_entretien\\_avec\\_Pierre-Yves\\_Oudeyer\\_propos\\_recueillis\\_par\\_Dominique\\_Chouchan](https://www.researchgate.net/publication/47783626_Robotique_de_l'automate_a_l'humanoïde_par_Bernard_Espiau_Un_robot_tres_curieux_entretien_avec_Pierre-Yves_Oudeyer_propos_recueillis_par_Dominique_Chouchan)
- [2] UNESCO. (2017). Rapport de la COMEST sur l'éthique de la robotique, SHS/YES/COMEST-10/17/2 REV.
- [3] Jean-Paul Laumond, Joanna Jongwane. (2012). À propos de robotique humanoïde. Interstices. (hal-01350246).
- [4] Snani Haifaa. (2021). Navigation d'un robot mobile dans un environnement dynamique. Mémoire présenté en vue de l'obtention du Diplôme de Master en informatique. Université Badji Mokhtar – Annaba. Page12.
- [5] Sirohi, Himanshu. (2020). A Review on Research Output Analysis on Robotic Technology. SSRN Electronic Journal. 10.2139/ssrn.3573258.
- [6] Pickover, C. (2021). Les trois lois de la robotique d'Asimov. Dans : , La fabuleuse histoire de l'intelligence artificielle (pp. 116-117). Paris: Dunod.
- [7] Conditt, Michael. (2011). History of Robots in Orthopedics. Surgical Robotics: Systems Applications and Visions. 661-677. DOI : 10.1007/978-1-4419-1126-1\_27.
- [8] Liu C, Liu Y, Xie R, Li Z, Bai S, Zhao Y. (2024). The evolution of robotics: research and application progress of dental implant robotic systems. Int J Oral Sci. DOI: 10.1038/s41368-024-00296-x. PMID: 38584185; PMCID: PMC10999443.
- [9] Jayakrishna Kandasamy, Kamalakanta Muduli, V. P. Kommula, Purushottam L. Meena. (2023). Smart Manufacturing Technologies for Industry 4.0. CRC Press. Doi: doi 10.1201/9781003186670-2.  
[https://books.google.dz/books?id=OvGaEAAAQBAJ&pg=PA9&lpg=PA9&dq=doi+10.1201/9781003186670-2&source=bl&ots=baK\\_T8VmCa&sig=ACfU3U3GP1mmqkm04eUoSNjR-08Z4Ealwg&hl=fr&sa=X&ved=2ahUKEwiOlfvA0u6FAxXZVKQEHcG8BG0Q6AF6BAgJEAM#v=onepage&q=doi%2010.1201%2F9781003186670-2&f=false](https://books.google.dz/books?id=OvGaEAAAQBAJ&pg=PA9&lpg=PA9&dq=doi+10.1201/9781003186670-2&source=bl&ots=baK_T8VmCa&sig=ACfU3U3GP1mmqkm04eUoSNjR-08Z4Ealwg&hl=fr&sa=X&ved=2ahUKEwiOlfvA0u6FAxXZVKQEHcG8BG0Q6AF6BAgJEAM#v=onepage&q=doi%2010.1201%2F9781003186670-2&f=false)
- [10] Kim, JH., Zaheer, S.A., Ryu, SJ. (2015). Intelligence Technology for Ubiquitous Robots. In: Mohammed, S., Moreno, J., Kong, K., Amirat, Y. (eds) Intelligent Assistive Robots. Springer Tracts in Advanced Robotics, vol 106. Springer, Cham. Doi : 10.1007/978-3-319-12922-8\_10.
- [11] Alahi, A. H. M Monjur & Potter, Kaledio. (2024). The Future of Robotics: Advancements and Implications. The Computer Journal.  
[https://www.researchgate.net/publication/378435947\\_The\\_Future\\_of\\_Robotics\\_Advancements\\_and\\_Implications](https://www.researchgate.net/publication/378435947_The_Future_of_Robotics_Advancements_and_Implications)
- [12] Wani, Sagar. (2023). The Synergy between Artificial Intelligence and Robotics. 10.5281/zenodo.10676937. Doi: 10.528/zenodo.10676937.



- [13] Lakhmissi, Cherroun. (2014). Navigation Autonome d'un Robot Mobile par des Techniques Neuro-Floues. 10.13140/RG.2.2.30956.03208.
- [14] Stäubli. (2021). Le Guide de la Robotique. <https://fr.scribd.com/document/677556722/2021-guide-de-la-robotique-staubli>
- [15] David Filliat. (2012) . Robotique mobile. École Nationale Supérieure de Techniques Avancées ParisTech. [https://www.gdr-robotique.org/cours\\_de\\_robotique/online/Filliat\\_Rob-Mobile/Filliat\\_Rob-Mobile.pdf](https://www.gdr-robotique.org/cours_de_robotique/online/Filliat_Rob-Mobile/Filliat_Rob-Mobile.pdf)
- [16] Godin, C. (2021, avril). Planification de trajets pour un véhicule à guidage automatique holonome à parcours libre (Mémoire de maîtrise). Université de Sherbrooke, Québec, Canada.
- [17] Nesrine Harbaoui. (2022). Diagnostic adaptatif à l'environnement de navigation : apport de l'apprentissage profond pour une localisation sûre et précise. Apprentissage [cs.LG]. Université de Lille. Français. ffNNT : 2022ULILB041ff. fftel-04136855f
- [18] Waykar, Y. A. (2022). Lidar Technology: A Comprehensive Review and Future Prospects. Journal of Emerging Technologies and Innovative Research, 9(7). Retrieved from <https://www.jetir.org/papers/JETIR2207728.pdf>
- [19] Issa Baldé. (2008) Mise en place d'une plateforme de cartographie dynamique. Mémoire pour un diplôme d'ingénieur de conception en génie informatique. Ecole supérieur polytechnique de Dakar.
- [20] Duperré, A. (2020). Guidage et planification réactive de trajectoire d'un drone monoculaire contrôlé par intelligence artificielle [Mémoire de maîtrise, Polytechnique Montréal]. <https://publications.polymtl.ca/5399/>
- [21] Raj, R., & Kos, A. (2022). A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives. Applied Sciences, 12(14), 6951. <https://doi.org/10.3390/app12146951>
- [22] Hang Su, Wen Qi, Jiahao Chen, Chenguang Yang, Juan Sébastian Sandoval Arevalo, et al. (2023). Recent advancements in multimodal human–robot interaction. Frontiers in Neurobotics, , 17, pp.1084000. ff10.3389/fnbot.2023.1084000ff. ffhal-04439174
- [23] Florent Taralle. (2016). Guidage Gestuel pour des Robots Mobiles. Interface homme-machine [cs.HC]. Université Paris sciences et lettres,. Français. ffNNT : 2016PSLEM096ff. fftel-01814263f
- [24] Hayajneh, M.; Al Mahasneh, A. (2022). Guidance, Navigation and Control System for Multi-Robot Network in Monitoring and Inspection Operations. Drones, 6,332. <https://doi.org/10.3390/drones6110332>
- [25] Emami, Seyyed Ali & Banazadeh, Afshin & Hajipourzadeh, Pedram & Castaldi, Paolo & Fazelzadeh, S. (2023). Disturbance observer-based adaptive neural guidance and control of an aircraft using composite learning. Control Engineering Practice. 134. 105463. DOI: 10.1016/j.conengprac.2023.105463.
- [26] Gao, Guomin & Guo, Hui & Zhou, Wei & Luo, Dan & Zhang, Jing. (2023). DESIGN OF A CONTROL SYSTEM FOR A SAFFLOWER PICKING ROBOT AND RESEARCH ON MULTISENSOR FUSION POSITIONING. Engenharia Agrícola. <https://doi.org/10.1590/1809-4430-Eng.Agric.v43n2e20210238/2023>
- [27] [https://upload.wikimedia.org/wikipedia/commons/thumb/f/f9/Eurofighter\\_sensor\\_fusion.png/1024px-Eurofighter\\_sensor\\_fusion.png](https://upload.wikimedia.org/wikipedia/commons/thumb/f/f9/Eurofighter_sensor_fusion.png/1024px-Eurofighter_sensor_fusion.png)

- [28] Kim, Myeongseop & Kim, Jung-Su & Park, Jae-Han. (2023). Automated Hyperparameter Tuning in Reinforcement Learning for Quadrupedal Robot Locomotion. *Electronics*. 13. 116. DOI: 10.3390/electronics13010116
- [29] Nguyen, A.-T., & Vu, C.-T. (2021). Obstacle Avoidance for Autonomous Mobile Robots Based on Mapping Method. <https://arxiv.org/abs/2109.06773>
- [30] Yucel, B., Yilmaz, A., Ervan, O., & Temeltas, H. (2021). Fuzzy Controlled Adaptive Follow the Gap Obstacle Avoidance Algorithm. In *Proceedings of the 7th International Conference on Robotics and Artificial Intelligence (ICRAI '21)* (pp. 93–98). <https://doi.org/10.1145/3505688.3505704>
- [31] L. Zhang *et al.* (2020). "WiFi-Based Indoor Robot Positioning Using Deep Fuzzy Forests," in *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10773-10781, , DOI:10.1109/JIOT.2020.2986685
- [32] Liu, Y., Wang, S., Xie, Y., Xiong, T., & Wu, M. (2024). A Review of Sensing Technologies for Indoor Autonomous Mobile Robots. *Sensors*, 24(4), 1222. <https://doi.org/10.3390/s24041222>
- [33] Rostami, S.M.H., Sangaiah, A.K., Wang, J. *et al.*(2019). Obstacle avoidance of mobile robots using modified artificial potential field algorithm. *J Wireless Com Network* , 70. <https://doi.org/10.1186/s13638-019-1396-2>
- [34] Karaman, S., & Frazzoli, E. (2011). Sampling-based Algorithms for Optimal Motion Planning. *International Journal of Robotics Research*. <https://doi.org/10.48550/arXiv.1105.1186>
- [35] Arias, Monica & Nechyporenko, Nataliya. (2016). Generalized Voronoi Diagram application to Turtlebot Navigation in ROS. DOI : 10.13140/RG.2.2.22959.43688.
- [36] Lindqvist, B. (2023). Reactive Navigation Methods for Autonomous Robots: Safety, Coordination, and Field Deployment [Doctoral thesis, Luleå University of Technology]. Department of Computer Science and Electrical Engineering.
- [37] Mcguire, Kimberly & Croon, Guido & Tuyls, Karl. (2019). A comparative study of bug algorithms for robot navigation. *Robotics and Autonomous Systems*. 121. 103261. DOI: 10.1016/j.robot.2019.103261.
- [38] Klančar, G.; Zdešar, A.; Krishnan, M. (2022). Robot Navigation Based on Potential Field and Gradient Obtained by Bilinear Interpolation and a Grid-Based Search. *Sensors* **2022**, 22, 3295. <https://doi.org/10.3390/s22093295>
- [39] Mohammed, Ibrahim & Noaman, Mohanad. (2022). Optimal Control Approach for Robot System Using LQG Technique. 55. 671-677. DOI :10.18280/jesa.550513. <https://www.researchgate.net/publication/366146468> [Optimal Control Approach for Robot System Using LQG Technique](https://www.researchgate.net/publication/366146468)
- [40] Quigley, Morgan & Conley, Ken & Gerkey, Brian & Faust, Josh & Foote, Tully & Leibs, Jeremy & Wheeler, Rob & Ng, Andrew. (2009). ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software*. <http://www.robotics.stanford.edu/~ang/papers/icraoss09-ROS.pdf>
- [41] Cyril Briand, Olivier Stasse, Michel Taïx, Thierry Germa, Fabien Marco, et al.(2021). ROS : une solution middleware adaptée pour l'industrie du futur ?. 17ème colloque national S-mart AIP-PRIMECA, Université Polytechnique Hauts-de-France [UPHF], , LAVAL VIRTUAL WORLD, France. hal-03296139

- [42] Stasse, O. (2022). Robot Operating System Introduction. PUBLISHED BY LAAS-CNRS. Retrieved from <https://homepages.laas.fr/ostasse/Teaching/ROS/poly-rosintro.pdf>
- [43] Hax, Vinicius & Duarte Filho, Nelson & Botelho, Silvia & Mendizabal, Odorico. (2013). ROS as a middleware to Internet of Things. Journal of Applied Computing Research. 2. 91-97. 10.4013/jacr.2012.22.05.
- [44] H. Jung, M. Kim, Y. Chen, H. G.( 2020 ). Min and T. Park, "Implementation of a unified simulation for robot arm control with object detection based on ROS and Gazebo," 17th International Conference on Ubiquitous Robots (UR), Kyoto, Japan, 2020, pp. 368-372, Doi: 10.1109/UR49135.2020.9144984
- [45] [https://classic.gazebosim.org/tutorials?cat=get\\_started&tut=architecture](https://classic.gazebosim.org/tutorials?cat=get_started&tut=architecture)
- [46] M. B. Holder, M. M. Trivedi and S. B. Marapane, (1996,). Mobile robot navigation by wall following using a rotating ultrasonic scanner. Proceedings of 13th International Conference on Pattern Recognition, Vienna, Austria, pp. 298-302 vol.3, doi: 10.1109/ICPR.1996.546959.
- [47] Juang, J.G., & Yang, Y.C. (2015). Fuzzy Sensor Fusion and Curve Approximation for Indoor Map Building. In International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2015). doi: 10.2991/aiie-15.2015.148
- [48] Morbidi, F. (s.d.). Localisation et navigation de robots [PDF]. Récupéré sur [https://home.mis.u-picardie.fr/~fabio/Eng/documenti/Teaching/LNR18-19/LNR\\_p1.pdf](https://home.mis.u-picardie.fr/~fabio/Eng/documenti/Teaching/LNR18-19/LNR_p1.pdf)
- [49] Kumar, A., Mann, J., Nehra, M., Jangra, N., & Vashsith, S. (2013). Un simple compteur d'obstacles amélioré pour fonctionner comme un système de détection d'intrusion. Revue internationale de recherche scientifique et technique, 4(7), 2518-2521. ISSN 2229-5518.
- [50] Thao Dang, Alexandre Donzé, Inzemamul Haque, Nikolaos Kekatos, Indranil Saha. (2023). Counter-Example Guided Imitation Learning of Feedback Controllers from Temporal Logic Specifications. 62nd IEEE Conference on Decision and Control, Dec 2023, Singapore, Singapore. Hal-04295795
- [51] Amsters, Robin & Slaets, Peter. (2020). Turtlebot 3 as a Robotics Education Platform. DOI :10.1007/978-3-030-26945-6\_16.
- [52] Gurel, Canberk Suat. (2018). REAL-TIME 2D AND 3D SLAM USING RTAB-MAP, GMAPPING, AND CARTOGRAPHER PACKAGES. DOI:10.13140/RG.2.2.14901.99049.
- [53] Duperré, A. (2020). Guidage et planification réactive de trajectoire d'un drone monoculaire contrôlé par intelligence artificielle [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. <https://publications.polymtl.ca/5399/>
- [54] <http://wiki.ros.org/noetic/Installation/Ubuntu>