

UNIVERSITE LIBRE DES PAYS DES GRANDS LACS

FACULTE DE SCIENCES ET TECHNOLOGIES

DEPARTEMENT DE GENIE ELECTRIQUE ET INFORMATIQUE



BP. 368 GOMA

www.ulpgl.net

**CONCEPTION ET REALISATION D'UN
ROBOT AUTONOME POUR
L'AUTOMATISATION DU TRANSPORT
INTERNE DE MEDICAMENTS EN MILIEU
HOSPITALIER**

Par **USHINDI GAHAMANYI ALFRED**

Travail présenté et défendu en vue de l'obtention du
Diplôme de Bachelor en Sciences de l'ingénieur

Option : Génie Electrique

**Directeur : Prof. Dr. Ir. BARAKA MUSHAGE
Olivier**

Encadreur : Ir. Johnson KISAMBA

ANNEE ACADEMIQUE 2024 - 2025

Epigraphe

« La technologie n'est pas une fin en soi, mais un moyen de servir l'humain et de soulager ses souffrances »

Alfred Nobel

Dédicace

A ma très chère mère Julienne BIFUKO

Remerciements

En premier lieu, nous rendons grâce à Dieu pour ses bienfaits de tous les jours.

Nous exprimons notre profonde gratitude à la Faculté des Sciences et Technologies pour l'excellente formation dont nous avons bénéficié tout au long de notre parcours académique.

Nous adressons nos sincères remerciements au Professeur BARAKA MUSHAGE Olivier, notre Directeur, pour ses orientations, ses précieux conseils et sa disponibilité qui ont grandement contribué au bon déroulement de ce travail. Nous remercions également l'Ingénieur Johnson KISAMBA, notre Encadreur, pour sa disponibilité, ses conseils avisés et son accompagnement constant.

Nos remerciements vont aussi au Réseau d'Innovation Organisationnel (RIO) pour son soutien et son accompagnement tout au long de notre parcours académique.

Nous exprimons notre gratitude à nos chers parents, Gilbert GAHAMANYI et Julienne BIFUKO, pour leur accompagnement et leur soutien constants à chaque étape de notre vie.

Nous remercions nos collègues et ami(e)s qui, de près ou de loin, ont contribué à la réalisation de ce travail. Nous citons notamment Séraphin, Chérubin, Gédéon, Celestin, Danny, Jonathan, Moise Wisdom, Jeremy, chacun ayant joué un rôle important dans ce projet.

Enfin, nous exprimons notre profonde reconnaissance à nos frères et sœurs Maurice, William, Prisca et Anne Marie pour leur soutien, ainsi qu'à toutes les autres personnes qui, de près ou de loin, nous ont aidés dans la réalisation de ce travail.

USHINDI GAHAMANYI Alfred

Résumé

L'évolution de la technologie permet aujourd'hui d'alléger certaines tâches répétitives et critiques, notamment dans le domaine de la santé où la logistique interne peut peser sur le personnel soignant. Pour notre travail, nous nous sommes intéressés à la distribution des médicaments au sein des hôpitaux, en concevant un robot autonome de livraison capable de se déplacer dans un environnement structuré mais dynamique comme les couloirs hospitaliers. Ces robots doivent pouvoir naviguer avec précision, éviter les obstacles imprévus et assurer une livraison fiable sans intervention humaine constante. Pour atteindre cet objectif, nous avons opté pour l'implémentation d'un robot à entraînement différentiel, équipé de capteurs à ultrasons pour la détection d'obstacles et utilisant l'algorithme A* pour la planification de trajectoire sur une carte discrète des couloirs. Une interface de contrôle via une application mobile et un module embarqué permet au personnel de superviser et de valider les livraisons. Étant donné que cette approche repose fortement sur une localisation par odométrie calibrée, sa précision peut être affectée sur de longues distances ou dans des environnements très dynamiques. Par conséquent, nous avons également envisagé l'intégration future de systèmes de localisation plus robustes, tels que le SLAM avec un capteur LiDAR économique, pour renforcer l'autonomie et la fiabilité du robot dans des espaces non préalablement modélisés.

Mots-clés

Robot autonome, livraison de médicaments, hôpital, planification de trajectoire, A*, SLAM, capteurs à ultrasons, application mobile



Abstract

Technological progress now allows us to ease certain repetitive and critical tasks, particularly in the healthcare sector where internal logistics can burden medical staff. For our work, we focused on medication distribution within hospitals, designing an autonomous delivery robot capable of moving through a structured yet dynamic environment like hospital corridors. These robots must navigate accurately, avoid unexpected obstacles, and ensure reliable delivery without constant human intervention. To achieve this goal, we implemented a differentially driven robot equipped with ultrasonic sensors for obstacle detection and used the A* algorithm for path planning on a discrete map of the corridors. A control interface via a mobile app and an onboard module allows staff to supervise and validate deliveries. Since this approach heavily relies on calibrated odometry for localization, its accuracy can be affected over long distances or in highly dynamic settings. Therefore, we have also considered the future integration of more robust localization systems, such as SLAM with an economical LiDAR sensor, to enhance the robot's autonomy and reliability in environments that are not pre-modeled.

Key words

Autonomous robot, medication delivery, hospital, path planning, A*, SLAM, ultrasonic sensors, mobile application

Table des matières

Epigraphe	i
Dédicace.....	ii
Remerciements.....	iii
Résumé.....	iv
Table des matières	vi
Liste des sigles et abréviations.....	x
Liste des figures	xi
0. INTRODUCTION GENERALE	1
0.1 Généralités sur le thème	1
0.2. Identification et formulation du problème	2
0.3. Questions de recherche	2
0.4. Formulation des hypothèses.....	3
0.5. Justification du choix du sujet et motivations.....	4
0.6. Énoncé des objectifs de recherche	4
0.6.1. L'objectif général.....	4
0.6.2. Les objectifs opérationnels	5
0.7. Méthodologie et délimitation du travail.....	5
0.8. Structure du mémoire.....	5
Chapitre 1 : Cadre théorique et généralités	7
1.1. Introduction.....	7
1.2 Concepts et typologies des robots mobiles	8
1.2.1 Définition d'un robot	8
1.2.2 Holonomie / non-holonomie	9

1.2.3 Différence entre un AGV et AMR.....	10
1.2.4 Exemples de robots de transport existants	11
1.3. Architecture du système de commande	13
1.3.1 Le niveau interface homme–machine (IHM).....	14
1.3.2. Le niveau de planification des tâches	14
1.3.3. Le niveau de contrôle d'exécution.....	14
1.3.4. Le niveau fonctionnel	14
1.4. Environnement dynamique et incertain	15
1.4.1. Notion d'environnement dynamique [34].....	15
1.4.2. Notion d'incertitude	16
1.4.3. Stratégies de gestion des environnements dynamiques et incertains.....	16
1.5 Navigation autonome : fondements et architectures.....	17
1.5.1 Navigation par suivi de ligne	18
1.5.1 L'espace de configuration : une formalisation mathématique.....	19
1.5.2 Architectures de navigation	20
1.5.3 Localisation et fusion de données	21
1.5.4 Problématique de la localisation	22
1.6 Perception et instrumentation sensorielle	23
1.6.1 Capteurs proprioceptifs.....	24
1.6.2 Capteurs extéroceptifs.....	24
1.6.3 Fusion de données.....	25
1.6.4 Défis en environnement réel et en milieu hospitalier	25
1.7 Exigences spécifiques liées au milieu hospitalier.....	26
1.8. Thématiques scientifiques de la robotique mobile.....	27
1.9. Conclusion partielle	28
Chapitre 2 : Modélisation et Simulation du Robot de Livraison de Médicaments	29
2.1 Introduction.....	29
2.2 Modélisation du robot.....	29

2.2.1 Modèle cinématique [51]	31
2.3 Localisation du robot [7].....	34
2.4 Configuration du robot et modélisation de l'environnement [51]	35
2.5 Planification de trajectoire	36
2.5.1 Algorithme de planification A^* [53]	36
2.5.2 Planification du Chemin d'aller	39
2.5.3 Planification du Chemin de Retour	40
2.6 Gestion des obstacles dynamiques	40
2.6.1 Priorité des obstacles et comportement adaptatif.....	40
2.6.2 Détection des obstacles mobiles	41
2.6.3 Replanification locale avec l'algorithme A^*	41
2.6.4 Replanification dynamique et planification prédictive	41
2.6.5 Exemple de scénario	42
2.7 Simulations et résultats	43
2.8 Conclusion partielle	45
Chapitre 3 : Réalisation du Robot.....	46
3.1 Introduction.....	46
3.2 Présentation générale du robot.....	46
3.3. Circuit du robot Hospitalier	46
3.4. Composants utilisés	48
3.4.1. Cartes de développement ESP32-WROOM-32.....	48
3.4.3. Capteur de distance HC-SRO4	49
3.4.4. Le module MPU9250.....	49
3.4.5. Le capteur de vitesse LM393 et disque d'encodeur.....	50
3. 4.6. Les moteurs à courant continu	50
3.4.7 Driver moteur L298N	51
3.4.8. Batteries Li-ion	52
3.4.9. Le régulateur de tension (LM2596)	52
3.4.10. Un clavier Matriciel	53

3.4.11. Un écran OLED	53
3.4.12. Une LED, Buzzer	53
3.4.13. Un module de gestion et de protection de batteries (BMS)	54
3.5. Approche utilisée pour la localisation du robot	54
3.6. Correcteur PID et Encodeurs	55
3.7. Localisation du robot	56
3.8. Construction de l'environnement.....	57
3.9. Les modes de fonctionnement robot	58
3.10. Le robot physique	59
3.11 Conclusion Partielle	61
Conclusion générale.....	62
Bibliographie.....	63
ANNEXES	70
ANNEXE 1	70
ANNEXE 2	71
ANNEXE 3	72

Liste des sigles et abréviations

AMR : *Autonomous Mobile Robot*

AOI : *Area of Interest (Zone d'intérêt)*

AVG : *Automated Guided Vehicle*

CPP : *Coverage Path Planning (Planification de trajectoire de couverture)*

DQN : *Deep Q-network (réseau Q profond)*

EKF : *le filtre de Kalman étendu*

IMU : *Unités de Mesure Inertielle*

MiR : *Mobile Industrial Robots*

PF : *le filtre à particules*

RL : *Reinforcement Learning (Apprentissage par renforcement)*

RRT : *Rapidly-exploring Random Trees*

SLAM : *Simultaneous Localization And Mapping*

Sense-Plan-Act : *Percevoir-Planifier-Agir*

TSP : *Traveling Salesman Problem (Problème du voyageur de commerce)*

Liste des figures

Figure 1 Robot holonome vs robot non-holonome [19]	10
Figure 2 Schéma comparatif entre un AGV et un AMR [24].....	11
Figure 3 Robot autonome Tug [28].	12
Figure 4 Robot autonome Relay [29].....	12
Figure 5 Robot autonome MiR250 [30].....	13
Figure 6 Schéma de principe d'un système de commande de robot mobile.....	15
Figure 7 Schéma de l'architecture de navigation autonome [39].	18
Figure 8 Illustration de l'espace de configuration (pose du robot et trajectoire libre) [40].....	20
Figure 9 Types d'architecture des navigations [43].....	21
Figure 10 Robot mobile différentiel REMO [48]	30
Figure 11 Mouvements d'un Robot à entraînement différentiel [49]	30
Figure 12 Roue Folle ou pivotante [50]	31
Figure 13 Cinématique d'un robot à entraînement différentielle [52]	32
Figure 14 Différence entre distance de Manhattan (couleur bleue et verte) et distance euclidienne (couleur rouge) [51].....	37
Figure 15: Exemple de chemin optimal calculé par l'algorithme A* dans un environnement comportant des obstacles [54].....	38
Figure 16: Algorithme A* [55]	39
Figure 17: Illustration de la gestion des obstacles dynamiques par le robot. Le robot détecte un obstacle temporaire (personnel, patient...) dans le couloir et adapte sa trajectoire en choisissant un chemin alternatif avant de revenir sur le chemin optimal [56].....	42
Figure 18. Présentation de l'environnement hospitalier (la vue générale des couloirs, chambres)	44
Figure 19. Navigation du robot dans les couloirs	44
Figure 20: Circuit de notre robot	47
Figure 21 : ESP32-DevKitC V4 avec module ESP32-WROOM-32E soudé [57]	48
Figure 22: Capteur ultrason HC-SRO4 [51]	49

Figure 23: Module MPU9250 [51]	49
Figure 24 Capteur de vitesse à LM393 [58]	50
Figure 25 Disques d'encodeur [59]	50
Figure 26: Moteur à courant continu du type TT [51]	51
Figure 27: Le driver L298N [51]	51
Figure 28: Batteries rechargeables en Lithium [51]	52
Figure 29: Le module LM2596 [51]	52
Figure 30: Clavier Matricielle 4X4 [60]	53
Figure 31: Ecran OLED Adafruit [61]	53
Figure 32: Diode Led [62]	54
Figure 33: Buzzer [63]	54
Figure 34: Module de gestion de batteries [64]	54
Figure 35: Notre environnement de test.....	58
Figure 36: Capture de l'application Mobile.....	59
Figure 37: Vue de dessus de notre robot.....	60
Figure 38: Vue de dessous de notre robot.....	60
Figure 39: Interface utilisateur de notre robot	61

0. INTRODUCTION GENERALE

0.1 Généralités sur le thème

Les nouvelles technologies, notamment la robotique, transforment peu à peu le domaine de la santé. Dans les hôpitaux, ces innovations ont pour objectif de rendre certaines tâches plus rapides et efficaces, tout en libérant du temps pour les soignants. L'une des applications intéressantes est l'utilisation des robots pour la gestion du transport de médicaments, afin d'optimiser les soins aux patients [1]. Dans de nombreux hôpitaux, les infirmiers doivent parcourir de grandes distances pour livrer les médicaments entre la pharmacie et les différents services. Cela leur prend un temps considérable, réduisant ainsi leur disponibilité pour d'autres soins essentiels aux patients. En plus, des retards ou des erreurs dans la livraison des médicaments peuvent perturber le bon déroulement des traitements.

Dans certaines régions comme celle des provinces de la République Démocratique du Congo, cette situation est encore plus compliquée. Les hôpitaux font face à plusieurs défis, comme : un seul infirmier qui doit s'occuper de 20 à 30 patients, ce qui est épuisant et diminue la qualité des soins [2] ; des retards dans l'administration des médicaments, souvent en raison du manque de temps ou d'organisation [3] ; un personnel obligé de gérer plusieurs tâches en même temps, entre les soins aux patients, la distribution des médicaments et la gestion des urgences [4]. À Goma, par exemple, l'Hôpital Général de Goma est confronté à ces difficultés au quotidien. Parfois, quand le personnel infirmier est en sous-effectif ou déjà occupé à prodiguer des soins aux patients, la livraison des médicaments peut prendre du retard. Dans ce cas, un robot autonome peut être utilisé pour faire la livraison sur demande. Cela permet de soulager les infirmiers et d'améliorer la régularité de la distribution des médicaments. Dans ce contexte, l'utilisation de robots mobiles autonomes apparaît comme une solution intéressante. Ces robots ne distribuent pas les médicaments aux patients, mais ils se chargent du transport des médicaments depuis la pharmacie jusqu'aux salles de service. Une fois arrivés à la destination, le personnel médical peut récupérer les médicaments pour les administrer aux patients. Ce système permet de gagner du temps, de réduire la fatigue des soignants, et d'améliorer l'efficacité des livraisons de médicaments [5]. De plus, le robot devra assurer la sécurité des

médicaments transportés grâce à un compartiment fermé et sécurisé, et éviter les accidents grâce à des capteurs qui lui permettent de détecter les obstacles.

C'est dans cette optique que ce travail propose de développer un prototype simple de robot qui pourra : se déplacer seul dans les couloirs de l'hôpital ; transporter les médicaments ; s'arrêter aux bons endroits, facilitant ainsi la distribution des médicaments de manière plus rapide et plus efficace.

0.2. Identification et formulation du problème

Dans plusieurs hôpitaux de Goma, il arrive que des tâches simples, comme le transport des médicaments, prennent plus de temps que prévu, surtout lorsque le personnel est peu nombreux ou occupé à d'autres soins essentiels. Ce travail manuel, souvent confié aux infirmiers, limite leur efficacité et augmente les risques de retards dans les soins. Alors que la technologie progresse, très peu de solutions concrètes ont été adoptées localement pour améliorer ce processus. Il existe donc un manque de dispositifs automatisés adaptés aux réalités des hôpitaux de la région.

Ce mémoire vise à répondre à ce besoin en proposant un prototype de robot autonome capable de transporter des médicaments à l'intérieur d'un hôpital, afin d'illustrer une solution pratique à ce problème souvent ignoré.

0.3. Questions de recherche

Pour arriver à trouver une solution au problème que nous venons de soulever, Plusieurs questions se posent :

- Comment concevoir un robot capable de se déplacer de manière autonome dans les couloirs d'un hôpital sans utiliser de système de suivi de ligne ?
- Quelles sont les techniques les plus adaptées pour détecter et éviter les obstacles en temps réel dans un environnement hospitalier, où le passage est souvent encombré ?

- Comment le robot peut-il reconnaître les différentes zones de livraison (ex : pharmacie, salle de soins, blocs) pour déposer les médicaments au bon endroit ?
- Quelle serait la manière la plus simple pour le personnel médical d'interagir avec le robot (ex : réception des médicaments, validation de livraison) sans perturber leur travail ?
- Comment assurer la sécurisation du système, en garantissant à la fois la sécurité physique (patients, personnel, médicaments transportés) et la sécurité des données (communication et accès) dans un environnement hospitalier sensible ?

0.4. Formulation des hypothèses

À partir des questions de recherche évoquées précédemment, notre travail s'appuiera sur les fondements suivants :

- Les méthodes référencées de capteurs seraient les plus adaptées pour des environnements comme celui de l'hôpital où il y'a beaucoup plus de circulations car elles sont capables de détecter rapidement et avec précisions les obstacles. Un Robot équipé de capteurs de proximité (ultrason, infrarouge, etc.) et d'un système de navigation autonome peut se déplacer efficacement sans avoir besoin de suivre une ligne tracée au sol.
- Les techniques les plus adaptées dans le contexte hospitalier seraient l'utilisation combinée de capteurs à ultrasons, de capteurs infrarouges et d'un système de vision par ordinateur simplifié, permettant de détecter en temps réel les obstacles, d'éviter les collisions et d'améliorer la navigation ainsi que la reconnaissance des points de livraison.
- La navigation se fera par la programmation de points de livraison précis dans la mémoire du robot. À l'aide d'un algorithme de planification de trajectoire (A^*), le robot calculera le chemin optimal vers la destination et se déplacera de manière autonome tout en évitant les obstacles.
- Pour faciliter l'interaction entre le robot et le personnel médical, une interface simple comme un bouton, une alarme sonore ou un écran LCD, permettrait de signaler l'arrivée

du robot et d'assurer la confirmation que l'infirmier a bien récupéré les médicaments, sans perturber le rythme de travail du personnel.

- L'intégration de mécanismes de sécurisation (capteurs anticollision, verrouillage du compartiment de livraison, et sécurisation des communications) permettrait de garantir à la fois la sécurité physique des patients et du personnel, et la protection des médicaments transportés.

0.5. Justification du choix du sujet et motivations

Passionnés par la robotique, nous avons toujours souhaité mettre nos compétences au service de la résolution de problèmes concrets. L'un des moments les plus marquants fut une hospitalisation à l'Hôpital Général de Goma, durant laquelle nous avons constaté à quel point la livraison des médicaments pouvait être fastidieuse et peu efficace. Une infirmière, seule, devait pousser un chariot dans de longs couloirs tout en assurant la prise en charge d'un bloc hospitalier complet, incluant également la mise à jour des fiches des patients. Cette situation nous a amenés à réfléchir à une solution simple et innovante, telle que l'utilisation d'un robot autonome pour automatiser cette tâche. Ce sujet s'inscrit pleinement dans les recherches actuelles en robotique appliquée au domaine de la santé.

Il permet d'explorer comment des technologies simples peuvent répondre à des besoins réels dans les hôpitaux à faibles ressources.

Un tel robot pourrait aider à améliorer la qualité des soins en permettant aux infirmiers de se concentrer davantage sur les patients. Il s'agit donc d'une innovation à fort impact social, surtout dans un contexte où le personnel est souvent surchargé.

0.6. Énoncé des objectifs de recherche

0.6.1. L'objectif général

L'objectif principal de ce travail est de concevoir et développer un prototype de robot autonome capable de transporter des médicaments vers les différentes salles de soins de l'hôpital Général de Goma, afin d'illustrer un système de livraison automatisée fiable, économique et efficace.

0.6.2. Les objectifs opérationnels

Pour atteindre cet objectif général, nous visons les objectifs spécifiques suivants :

- Étudier les besoins liés à la distribution de médicaments en milieu hospitalier.
- Concevoir un système de navigation autonome pour le robot.
- Réaliser le prototype en assemblant les composants électroniques et mécaniques.
- Implémenter un système de commande simple pour le déplacement et l'arrêt du robot.
- Tester le prototype dans un environnement simulé ou réel (ex. : couloir d'hôpital).

0.7. Méthodologie et délimitation du travail

Pour réaliser ce travail, nous avons suivi une approche basée sur :

- La recherche documentaire, afin de comprendre les solutions similaires déjà proposées dans le domaine du transport médical automatisé.
- L'analyse technique, pour identifier les composants adaptés et modéliser la structure du robot.
- L'expérimentation, en concevant un prototype fonctionnel testé en conditions proches du réel.

Ce travail se limite à la réalisation d'un prototype simple, ne nécessitant pas de technologies complexes comme la vision par caméra ou les capteurs LIDAR. Le robot utilisera des capteurs de distance simples (ultrasons) pour détecter les obstacles, et sa navigation sera préprogrammée ou basée sur des repères simples.

0.8. Structure du mémoire

En dehors de l'introduction et de la conclusion, ce mémoire est structuré en trois chapitres :

- ✓ **Chapitre I : Cadre théorique et généralités**, présente les concepts clés liés à la robotique mobile, la navigation autonome, et les applications en milieu hospitalier.
- ✓ **Chapitre II : Conception et simulation du Robot**, expose la modélisation du robot, la présentation de l'approche utilisé dans ce travail, la présentation de différents résultats de simulations, choix techniques, la structure du robot, et le schéma de fonctionnement.

- ✓ **Chapitre III : Réalisation et tests du Robot**, détaille la mise en œuvre du prototype, les essais effectués, et les observations faites.

Chapitre 1 : Cadre théorique et généralités

1.1. Introduction

Dans ce chapitre nous présentons le cadre théorique et conceptuel relatif à la navigation des robots mobiles autonomes, en mettant en évidence les éléments essentiels qui permettront de mieux comprendre la suite du travail. L'objectif principal est de définir les notions fondamentales, d'expliquer les approches couramment utilisées, et de montrer leur pertinence dans un contexte hospitalier. Plus spécifiquement, il s'agit de : clarifier les concepts de base liés à la navigation, identifier les principales méthodes et technologies mises en œuvre, et analyser leur applicabilité dans le cadre d'un environnement hospitalier dynamique et contraint.

La navigation autonome constitue un champ de recherche multidisciplinaire, mobilisant la robotique, l'intelligence artificielle, la vision par ordinateur et l'automatique. La robotique mobile s'intéresse particulièrement aux problèmes de localisation, cartographie et planification de trajectoire, trois composantes fondamentales pour garantir le déplacement sûr et efficace d'un robot [6]. L'autonomie d'un robot dépend largement de sa capacité à percevoir son environnement et à s'y adapter en temps réel, ce qui constitue un enjeu majeur dans des milieux complexes comme les hôpitaux [7].

Les travaux de recherche soulignent également l'importance de prendre en compte les contraintes spécifiques du milieu d'application. Dans un contexte hospitalier, il s'agit non seulement de gérer la présence d'obstacles statiques et dynamiques (patients, personnels soignants, matériels roulants), mais aussi de respecter des exigences strictes de sécurité et de fiabilité [8]. Des recherches récentes mettent aussi en avant la nécessité d'intégrer des systèmes robustes de perception et de décision pour assurer la continuité du service robotisé dans des environnements critiques [9].

Ce chapitre est structuré autour de plusieurs points : la présentation générale de la navigation autonome, les différentes approches de navigation, la problématique de la localisation (qui

constitue un facteur clé de performance pour tout robot mobile) [10] et enfin les exigences spécifiques liées au milieu hospitalier, afin de situer clairement les enjeux de cette étude dans le cadre de notre projet.

1.2 Concepts et typologies des robots mobiles

1.2.1 Définition d'un robot

Un robot se définit comme un système mécanique ou électromécanique intégrant des capteurs, des actionneurs et une unité de commande. Conçu pour fonctionner de manière autonome ou semi-autonome, il constitue une machine programmable et reprogrammable, polyvalente et contrôlée automatiquement, capable d'exécuter diverses tâches en fonction des objectifs qui lui sont assignés [11].

Il s'agit d'une machine programmable qui interagit avec son environnement par la perception, la prise de décision et l'action [12]. Dans le cadre de la robotique mobile, le robot se déplace dans l'espace en fonction de ses objectifs et des contraintes de son environnement [13].

Dans un contexte hospitalier, un robot mobile est conçu pour assurer des tâches de logistique interne, telles que le transport de médicaments, d'échantillons biologiques ou de matériel médical [14]. Son rôle est de réduire la charge de travail du personnel soignant, d'optimiser la rapidité de service et de renforcer la fiabilité des opérations. Ces robots doivent être adaptés à un milieu sensible et dynamique, caractérisé par la présence constante de patients, de visiteurs et de professionnels de santé [15]. Ainsi, ils doivent répondre à des exigences strictes de sécurité, de précision et de fiabilité.

Le robot hospitalier se distingue donc par son intégration dans un environnement humain contraint et par la nécessité d'une navigation robuste et sécurisée. Il ne s'agit pas simplement d'une machine autonome générique, mais d'un outil technologique spécifiquement adapté aux besoins de la santé et de la logistique hospitalière.

1.2.2 Holonomie / non-holonomie

En robotique mobile, la notion d'holonomie caractérise la capacité d'une plateforme robotique à contrôler indépendamment tous ses degrés de liberté. Un robot évoluant dans un plan possède trois degrés de liberté : deux pour la translation (x et y) et un pour la rotation (θ) autour de l'axe vertical [16].

Une plateforme holonome est capable de se déplacer librement dans ces trois directions sans contrainte mécanique, ce qui simplifie grandement la planification de trajectoires et le contrôle du mouvement [17].

À l'inverse, une plateforme non-holonome présente des contraintes mécaniques qui limitent certains mouvements indépendants. Par exemple, un robot à roues différentielles, modèle classique utilisé dans ce travail, ne peut pas se déplacer latéralement de façon directe [18].

Mathématiquement, cette contrainte se traduit par une relation non-intégrable entre les vitesses du robot, exprimée par l'équation suivante [16] :

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (1)$$

Où : $\dot{y}$ et $\dot{x}$ sont les vitesses de translation du robot dans le repère global, θ est l'orientation du robot. Cette équation illustre que le robot doit combiner des mouvements de rotation et de translation pour atteindre une position donnée. Les détails de la modélisation cinématique de notre robot seront présentés au chapitre 2. La Figure 1 illustre la distinction entre un robot holonome et un robot non-holonome.

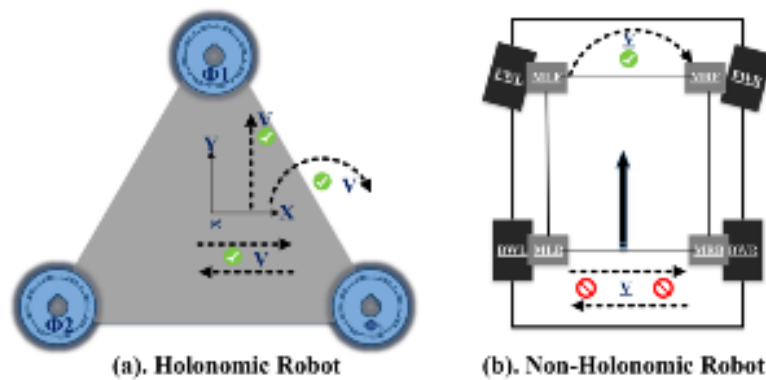


Figure 1 Robot holonome vs robot non-holonome [19]

1.2.3 Différence entre un AGV et AMR

Les véhicules à guidage automatique (AGV) et les robots mobiles autonomes (AMR) constituent deux catégories principales de robots mobiles, présentant des caractéristiques et des capacités distinctes [20].

Un AGV suit un itinéraire prédéfini à l'aide d'une infrastructure spécifique telle que des rails, des bandes magnétiques, des balises ou des capteurs au sol. Cette configuration entraîne des coûts d'installation relativement élevés et limite la flexibilité : toute modification du parcours nécessite une reprogrammation et, parfois, un ajustement physique de l'environnement [21]. L'AGV s'arrête automatiquement en présence d'obstacles et requiert une intervention humaine pour reprendre sa progression [22].

En revanche, un AMR repose sur des cartes numériques et des algorithmes de navigation avancés, utilisant divers capteurs tels que le lidar, les caméras et les ultrasons. Il peut analyser son environnement en temps réel et recalculer sa trajectoire de façon autonome afin de contourner les obstacles et optimiser ses déplacements [23].

Cette distinction est particulièrement pertinente dans les environnements hospitaliers : alors que les AGV conviennent aux parcours fixes et aux zones peu fréquentées, les AMR sont capables d'évoluer de manière autonome dans des couloirs occupés par le personnel, les

patients et le matériel, offrant ainsi une plus grande flexibilité opérationnelle. La Figure 2 illustre la différence entre un AVG et AMR face à un obstacle.



Figure 2 Schéma comparatif entre un AGV et un AMR [24].

1.2.4 Exemples de robots de transport existants

Dans les hôpitaux modernes, la robotique mobile joue un rôle croissant dans la logistique interne, notamment pour la livraison de médicaments, de repas, de linge ou de documents médicaux. L'objectif est de réduire la charge de travail du personnel soignant, tout en optimisant la rapidité et la fiabilité des services. Ci-dessous sont présentés quelques exemples de robots de transport autonomes actuellement déployés dans le secteur hospitalier.

- **TUG** : Développé par l'entreprise américaine Aethon, TUG est capable de transporter automatiquement des médicaments et du linge. Il utilise un ensemble de capteurs (ultrasons, infrarouge, LIDAR) pour naviguer dans les couloirs, détecter les obstacles et interagir avec des équipements tels que les ascenseurs. Son compartiment sécurisé garantit la protection du contenu [25]. La figure 3 représente le robot TUG.



Figure 3 Robot autonome Tug [28].

- **Relay** : Conçu par Swisslog Healthcare, Relay est spécialisé dans le transport sécurisé de médicaments ou de documents entre la pharmacie et les services cliniques. Compact et entièrement fermé pour assurer l'hygiène, il dispose d'un système de navigation autonome [26]. La figure 4 représente le robot Relay.



Figure 4 Robot autonome Relay [29]

- **MiR** : Les robots MiR (Mobile Industrial Robots), fabriqués au Danemark, sont utilisés à la fois dans l'industrie et les établissements de santé. Ils peuvent être programmés pour livrer des médicaments ou du matériel médical dans différents services. Grâce aux technologies de localisation et cartographie simultanées (SLAM), au LIDAR et à la vision par ordinateur, ils se déplacent de manière autonome [27]. La figure 5 représente le robot MiR50.



Figure 5 Robot autonome MiR250 [30]

Cependant, la majorité de ces solutions sont conçues pour des hôpitaux modernes, disposant d'infrastructures adaptées (ascenseurs automatisés, couloirs larges, connectivité réseau performante, systèmes d'appel automatisés). Leur coût élevé et leur complexité technologique les rendent difficilement accessibles dans des contextes hospitaliers comme celui de notre ville.

C'est ainsi que nous avons conçu notre robot. L'objectif n'est pas de concurrencer ces systèmes sophistiqués, mais de proposer un prototype simple, économique et adapté aux réalités locales. Contrairement aux robots TUG, Relay ou MiR, notre robot : dispose d'une architecture simple avec des capteurs de proximité et un simple microcontrôleur, se déplace de manière autonome sans infrastructure coûteuse ni réseau avancé, intègre un compartiment sécurisé pour garantir la livraison fiable des médicaments, peut mémoriser les points de livraison et signaler son arrivée via un écran ou une alarme sonore, facilitant l'interaction avec le personnel.

Ainsi, notre projet adapte le concept de robot de livraison autonome à un hôpital à faibles ressources, en rendant cette technologie simple, accessible et opérationnelle dans un contexte local où les solutions industrielles existantes ne peuvent pas être utilisées directement.

1.3. Architecture du système de commande

L'architecture du système de commande d'un robot mobile désigne l'organisation hiérarchisée des différents modules assurant la réalisation autonome d'une mission. Elle repose sur une structure modulaire où chaque niveau joue un rôle spécifique, tout en communiquant avec les

autres par des flux d'informations descendantes (commandes) et ascendantes (retours d'état). Cette organisation garantit la cohérence des décisions et la sûreté de l'exécution des tâches [31].

1.3.1 Le niveau interface homme-machine (IHM)

Ce niveau constitue le point de contact entre le robot et son utilisateur. Il permet de définir les missions (par exemple, la livraison de médicaments dans un service hospitalier) et de recevoir des retours sur l'état du robot (messages textuels, alertes sonores, etc.). Ce dialogue bidirectionnel favorise la clarté des objectifs et la transparence de l'exécution.

1.3.2. Le niveau de planification des tâches

À partir des ordres reçus via l'IHM, ce niveau élabore un plan d'actions séquentiel en tenant compte de l'environnement et des contraintes de mission. Dans le cas hospitalier, cela correspond à la génération d'un itinéraire reliant la pharmacie aux services cliniques. Le planificateur doit aussi gérer les imprévus : obstacles soudains, perturbations ou nécessité de ré planification [32].

1.3.3. Le niveau de contrôle d'exécution

Il traduit le plan en commandes concrètes et adapte en continu le comportement du robot selon les conditions réelles. Par exemple, il ajuste la trajectoire ou la vitesse si un couloir est encombré. Il exploite les données issues des modules de perception et veille à la sûreté de l'exécution par des mécanismes de correction d'erreurs [33].

1.3.4. Le niveau fonctionnel

Il regroupe les primitives élémentaires du robot : déplacements de base, activation des effecteurs, gestion de l'énergie [33]. Ce niveau assure l'exécution physique des commandes en interaction directe avec les ressources matérielles (moteurs, roues, etc.). Cette hiérarchie assure une organisation claire : l'IHM définit la mission, la planification conçoit le scénario, le contrôle ajuste l'exécution et le niveau fonctionnel applique les actions. L'ensemble forme une

boucle fermée où les retours d'état permettent une correction dynamique et garantissent l'autonomie du robot. La figure 6 illustre schématiquement l'architecture d'un système de commande de robot mobile.

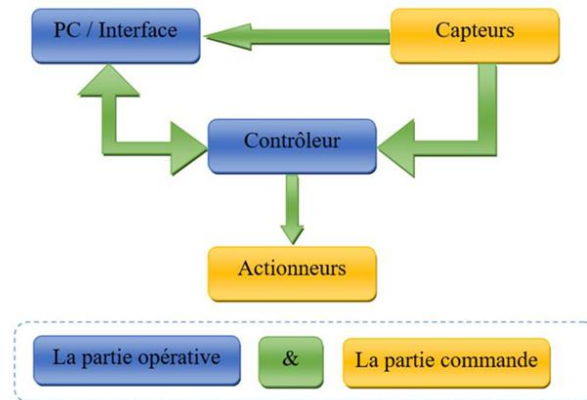


Figure 6 Schéma de principe d'un système de commande de robot mobile.

1.4. Environnement dynamique et incertain

1.4.1. Notion d'environnement dynamique [34]

Un environnement est dit dynamique lorsque la configuration des obstacles et des éléments qui le composent peut varier au cours du temps. Ces variations peuvent résulter de plusieurs facteurs :

- **Présence d'obstacles mobiles** : piétons (personnel médical, patients, visiteurs), véhicules, chariots de transport, autres robots, etc. Dans un hôpital, cela inclut le flux constant de personnel médical et de patients dans les couloirs.
- **Modifications de forme** : par exemple, un objet qui se replie, se déploie ou se déforme (comme un lit médical ajustable).
- **Apparitions ou disparitions soudaines** : ouverture et fermeture de portes, dépôts temporaires de matériel, déplacement d'équipements médicaux.

Dans ce contexte, le robot doit être capable de percevoir et identifier ces changements en temps réel, puis d'adapter sa trajectoire et son comportement afin de garantir la sécurité et l'efficacité de sa mission.

1.4.2. Notion d'incertitude

L'incertitude renvoie à toute limitation dans la fiabilité ou la précision des informations perçues ou estimées par le robot. Elle se manifeste principalement sous deux formes:

- **Incertitude aléatoire (ou stochastique)** : liée au bruit inhérent aux capteurs et aux conditions de détection. (Exemples : Bruit de mesure dû à un éclairage variable dans un couloir hospitalier ; Perturbations électromagnétiques causées par certains équipements médicaux ; Vibrations ou irrégularités du sol affectant les lectures des capteurs) [34].
- **Incertitude épistémique (ou incomplétude)** : due au manque d'informations sur l'environnement ou sur l'évolution future d'un phénomène. (Exemples : Portée limitée des capteurs empêchant de détecter des obstacles éloignés ; Absence de données sur la trajectoire future d'un piéton ; Erreurs liées aux modèles internes, comme les glissements de roues difficilement mesurables) [35].

1.4.3. Stratégies de gestion des environnements dynamiques et incertains

Dans un environnement hospitalier, où les situations sont à la fois dynamiques et incertaines, un robot doit s'appuyer sur des méthodes robustes de perception et de modélisation. Parmi les approches couramment utilisées, on retrouve :

- **Filtres de Kalman étendus (EKF)** : permettant de réduire les effets du bruit des capteurs et d'améliorer l'estimation de l'état du robot [36].
- **Grilles d'occupation dynamiques** : qui modélisent l'environnement en tenant compte du mouvement des obstacles et facilitent la prédiction de leurs trajectoires [37].
- **Méthodes de prédiction comportementale** : utilisées pour anticiper les déplacements des piétons et des chariots afin d'éviter les collisions dans les couloirs hospitaliers [38].

Ainsi, la combinaison du caractère dynamique et incertain représente un défi majeur pour la robotique mobile. La réussite d'un robot de livraison de médicaments en milieu hospitalier repose sur sa capacité à intégrer ces techniques afin de naviguer de manière sûre et fiable.

1.5 Navigation autonome : fondements et architectures

La navigation autonome constitue l'un des défis majeurs de la robotique mobile. Elle désigne la capacité d'un robot à se déplacer de manière sûre, efficace et intelligente dans un environnement, sans intervention humaine directe. Cette aptitude ne se limite pas à l'évitement d'obstacles ; elle repose sur une intégration cohérente de trois fonctions fondamentales :

- **Perception** : interprétation de l'environnement grâce aux données issues des capteurs (caméras, Lidar, radars, sonars). Elle permet de construire une représentation interne du monde et de détecter aussi bien les obstacles statiques que dynamiques [34].
- **Décision** : cœur cognitif du robot, impliquant la planification de trajectoire (choisir un chemin à suivre) et la gestion de comportements (ex. attendre l'ouverture d'une porte, éviter un piéton, retourner à une station de recharge) [33].
- **Action** : conversion des décisions en commandes motrices. Les contrôleurs de mouvement garantissent l'exécution fluide et précise de la trajectoire [17].

La robustesse d'un système de navigation dépend directement de sa capacité à gérer les incertitudes de l'environnement (statique ou dynamique, intérieur ou extérieur) et à exploiter efficacement l'information disponible (carte complète, partielle ou inexistante). La figure 7 illustre le processus de navigation autonome, montrant l'interaction entre la perception et la localisation, la planification de trajectoires et le contrôle du véhicule.

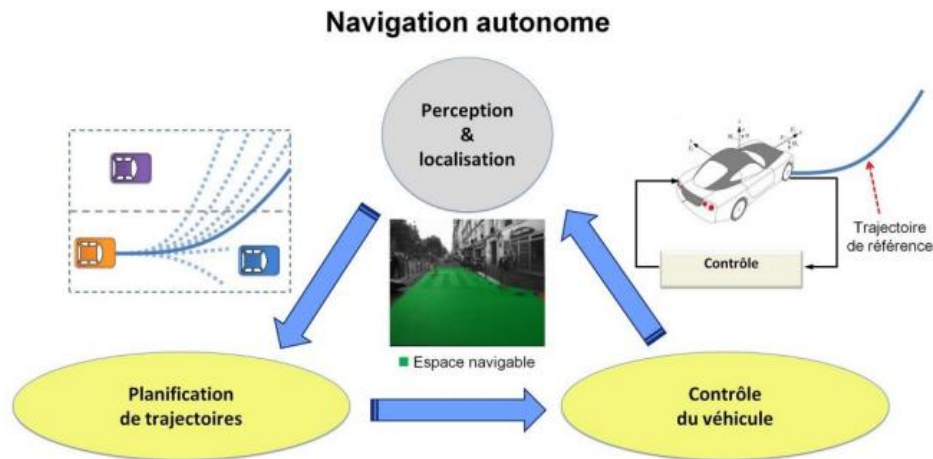


Figure 7 Schéma de l'architecture de navigation autonome [39].

1.5.1 Navigation par suivi de ligne

Parmi les méthodes classiques de navigation en robotique mobile, le suivi de ligne constitue une approche simple et largement utilisée, notamment dans les environnements industriels et pédagogiques. Cette technique repose sur l'utilisation de capteurs optiques, généralement infrarouges, permettant au robot de détecter une ligne tracée au sol et d'ajuster sa trajectoire en fonction de la position relative de cette ligne.

Le principe consiste à mesurer la différence de réflexion lumineuse entre une surface claire et une surface sombre. En fonction des variations détectées, un algorithme corrige la vitesse des roues afin de maintenir le robot aligné sur la trajectoire prédéfinie.

Cette méthode présente plusieurs avantages :

- Simplicité de conception et de programmation ;
- Coût matériel réduit ;
- Stabilité dans des environnements fixes et structurés.

Cependant, dans un environnement hospitalier réel, cette approche présente certaines limites importantes :

- Elle nécessite une modification de l'infrastructure (traçage de lignes au sol), ce qui peut être contraignant du point de vue organisationnel et hygiénique ;
- Elle offre peu de flexibilité en cas de changement d'itinéraire ;
- Elle ne permet pas une adaptation optimale face aux obstacles dynamiques tels que les patients, le personnel médical ou les équipements mobiles ;
- Elle limite l'autonomie réelle du robot à un parcours prédéfini.

Ainsi, bien que le suivi de ligne soit une solution efficace dans des environnements contrôlés, il apparaît moins adapté à un milieu hospitalier dynamique et évolutif.

Dans le cadre de ce travail, nous avons donc privilégié une approche de navigation autonome basée sur la planification de trajectoire et la détection d'obstacles, sans recours à un système de suivi de ligne. Cette orientation permet une plus grande flexibilité, une meilleure adaptation aux changements de l'environnement et une autonomie accrue du robot dans les couloirs hospitaliers.

1.5.1 L'espace de configuration : une formalisation mathématique

La navigation autonome peut être formalisée comme un problème de recherche de chemin dans un espace abstrait appelé espace de configuration. Chaque état du robot est représenté par un vecteur q , décrivant sa pose (position et orientation). Pour un robot mobile plan : $q = (x, y, \theta)$. L'espace de configuration se définit comme :

$$C = \mathbb{R}^2 \times S^1 \quad (2)$$

Où $\mathbb{R}^2$ représente la position cartésienne et S^1 la dimension de l'orientation. Le problème de navigation consiste à trouver une trajectoire continue : $p : [0, T] \rightarrow C_{libre}$, reliant une configuration initiale q_0 à une configuration finale q_{Finale} [17].

La figure 8 illustre visuellement le concept de l'espace de configuration, où le robot est modélisé par un point dans un espace de travail et les obstacles, les régions non accessibles sont agrandies pour inclure la taille du robot.

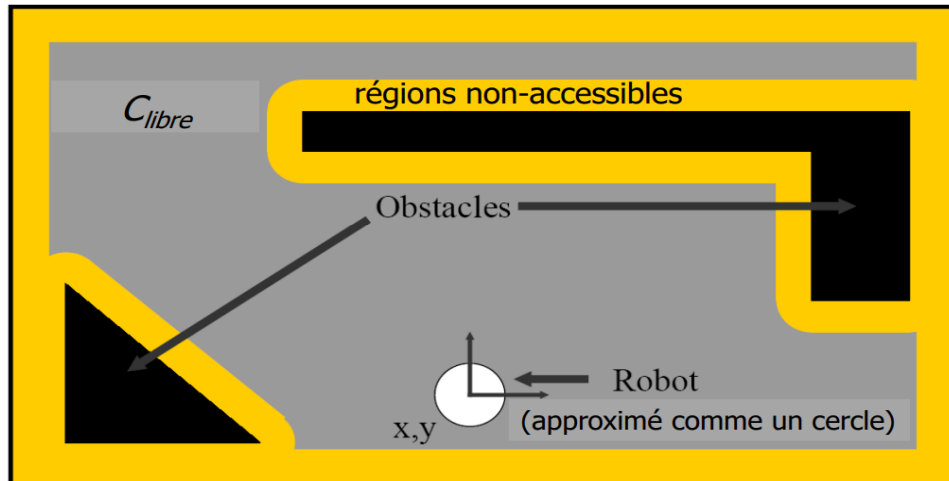


Figure 8 Illustration de l'espace de configuration (pose du robot et trajectoire libre) [40].

1.5.2 Architectures de navigation

Les architectures de navigation reposent sur deux grandes approches, souvent combinées dans des systèmes hybrides.

1.5.2.1 Approche délibérative

Cette approche suit le paradigme *Sense–Plan–Act* :

- **Modélisation** : localisation du robot dans une carte préalablement connue.
- **Planification globale** : calcul d'un chemin optimal via des algorithmes tels que A* ou les Rapidly-exploring Random Trees [17].
- **Exécution** : suivi de la trajectoire à l'aide de contrôleurs d'asservissement.

Cette approche est performante en environnements statiques et connus, mais sa rigidité limite sa réactivité face aux changements soudains.

1.5.2.2 Approche locale et réactive

Ici, la priorité est l'adaptation en temps réel : le robot réagit directement aux capteurs, sans planification exhaustive. Parmi les méthodes :

- **Champs de potentiels artificiels** : la cible exerce une force attractive et les obstacles des forces répulsives [41].
- **Dynamic Window Approach (DWA)** : optimisation en temps réel de la vitesse et de la direction en fonction des contraintes dynamiques du robot [42].

Ces approches sont rapides et réactives, mais peuvent rester piégées dans des minima locaux.

1.5.2.3 Architectures hybrides

Les systèmes modernes combinent planification globale et réactivité locale pour tirer profit des deux mondes : un plan global est généré, mais constamment ajusté par des modules réactifs face aux imprévus (piétons, portes, obstacles mobiles) [33]. La figure 9 résume les trois grandes catégories d'architectures de navigation : cognitive(délibérative), réactive, et hybride.

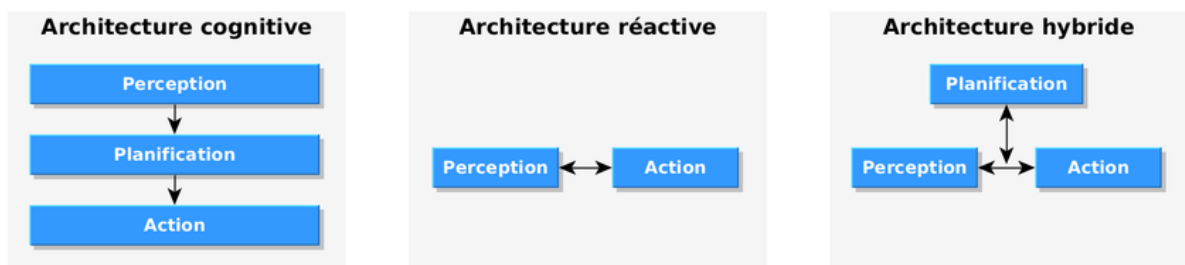


Figure 9 Types d'architecture des navigations [43].

1.5.3 Localisation et fusion de données

La localisation est une fonction critique : le robot doit estimer en continu sa pose pour naviguer efficacement. Trois composantes principales se complètent :

- **Localisation relative (odométrie, IMU)** : estimation à partir des capteurs proprioceptifs. Simple mais sujette à la dérive cumulée.
- **Localisation absolue (GPS, balises, repères visuels)** : positionnement direct par rapport à l'environnement, précis mais dépendant des conditions et infrastructures.
- **Fusion de données et SLAM** : combinaison des deux précédents via des algorithmes tels que le filtre de Kalman étendu, le filtre à particules, ou encore le SLAM, qui permet au robot de se localiser tout en construisant la carte de l'environnement [44]

Le SLAM moderne inclut des techniques de loop-closure (détection de retour sur un lieu déjà visité) afin de corriger les dérives et d'améliorer la précision globale.

1.5.4 Problématique de la localisation

La localisation constitue l'un des défis majeurs de la navigation autonome, car elle conditionne directement la capacité du robot à planifier et exécuter correctement ses déplacements. En effet, un robot mobile doit en permanence estimer sa position et son orientation par rapport à son environnement afin d'éviter les obstacles, atteindre une cible et garantir un transport fiable des médicaments [33]. Plusieurs difficultés rendent la localisation complexe, en particulier dans un milieu hospitalier :

- **Présence de nombreux obstacles dynamiques** : les couloirs sont fréquentés par des patients, du personnel soignant et divers équipements médicaux en mouvement, ce qui génère des incertitudes dans la mesure des positions [44].
- **Environnements homogènes et peu distinctifs** : les couloirs hospitaliers présentent souvent une architecture répétitive (portes identiques, longues allées, mobilier standardisé), ce qui limite la disponibilité de points de repère visuels fiables [45].
- **Perturbations des capteurs** : les interférences électromagnétiques, les variations de luminosité, ou encore le bruit affectant les capteurs (caméras, LiDAR, IMU) peuvent introduire des erreurs dans l'estimation de position [36].
- **Accumulation d'erreurs (dérive)** : les méthodes basées sur l'odométrie ou les capteurs inertiels dérivent au fil du temps, nécessitant des mécanismes de correction par fusion de données ou recalage sur l'environnement [44] [33].

Face à ces difficultés, plusieurs approches ont été développées :

- **Les méthodes basées sur la cartographie** (localisation relative à une carte connue, par exemple avec SLAM) [44].
- **Les méthodes basées sur des repères naturels ou artificiels** (codes QR, balises RFID, marqueurs visuels) [33].
- **Les approches hybrides**, combinant odométrie, vision par ordinateur et capteurs de télémétrie pour une localisation robuste [36].

Ainsi, la localisation n'est pas un sous-problème technique, mais un facteur clé de performance dans l'efficacité globale d'un robot autonome en milieu hospitalier. Un système de localisation fiable doit à la fois compenser les erreurs de mesure, s'adapter aux environnements dynamiques et rester suffisamment rapide pour être utilisé en temps réel.

1.6 Perception et instrumentation sensorielle

La perception est la capacité d'un robot mobile à collecter, traiter et interpréter les informations issues de son environnement afin de prendre des décisions adaptées. Elle constitue l'une des composantes fondamentales de l'autonomie robotique, permettant au robot d'appréhender son environnement pour interagir de manière sûre et intelligente [33]. Contrairement aux robots manipulateurs opérant dans des environnements structurés, les robots mobiles doivent évoluer dans des environnements ouverts, dynamiques et souvent inconnus, ce qui rend la perception encore plus cruciale.

La perception est ainsi au cœur de plusieurs fonctions essentielles :

- **Sécurité** : détection et évitement d'obstacles statiques et dynamiques [45].
- **Modélisation de l'environnement** : construction de cartes et représentation de l'espace pour la planification [17].
- **Localisation** : estimation de la position et de l'orientation du robot par rapport à une référence [44].

Pour accomplir ces tâches, les robots utilisent un ensemble de capteurs, classés en deux grandes catégories : les capteurs proprioceptifs (mesure de l'état interne) et les capteurs extéroceptifs (mesure de l'environnement) [33].

1.6.1 Capteurs proprioceptifs

Les capteurs proprioceptifs fournissent des informations sur l'état interne du robot, en mesurant notamment les vitesses, accélérations et angles de giration. Ils sont indispensables pour estimer la localisation relative du robot.

1.6.1.1 L'odométrie

L'odométrie repose sur les mesures de codeurs incrémentaux placés sur les roues, permettant de calculer la distance parcourue et les changements d'orientation. La pose du robot est ainsi mise à jour de manière itérative. Bien que simple et économique, l'odométrie souffre d'une dérive cumulative due au glissement des roues et aux irrégularités du sol, rendant l'estimation de position de plus en plus imprécise [17].

1.6.1.2 Les centrales inertielles (IMU)

Les centrales inertielles (IMU) combinent des accéléromètres (mesure de l'accélération linéaire) et des gyroscopes (mesure de la vitesse angulaire). L'intégration de ces mesures permet d'estimer orientation et déplacements, même sur des terrains accidentés où l'odométrie perd en fiabilité. Les IMU sont ainsi particulièrement utiles pour la navigation en extérieur [33].

1.6.2 Capteurs extéroceptifs

Les capteurs extéroceptifs renseignent sur l'environnement externe du robot. Ils complètent les mesures proprioceptives et permettent au robot de se localiser de manière absolue et d'éviter les obstacles.

1.6.2.1 La télémétrie et la détection par laser

Les télémètres mesurent la distance aux objets environnants :

- **Ultrasons** : économiques mais peu précis, sensibles aux interférences.
- **Infrarouges** : simples mais dépendants de la luminosité.
- **Laser (LiDAR)** : extrêmement précis, capables de générer un nuage de points 2D ou 3D pour construire une carte détaillée de l'environnement [33]. Les LiDAR 2D sont souvent utilisés pour la détection d'obstacles sur un plan horizontal, tandis que les LiDAR 3D offrent un modèle volumétrique complet.

1.6.2.2 Caméras et vision par ordinateur

Les caméras, capteurs passifs, fournissent des informations visuelles interprétées par des algorithmes de vision par ordinateur. Elles permettent :

- La reconnaissance d'objets (balises, personnes, équipements hospitaliers).
- La reconstruction 3D (caméras stéréo ou capteurs RGB-D).
- Le suivi visuel de trajectoires et points de repère.

1.6.3 Fusion de données

Aucun capteur ne donne une vision complète et fiable de l'environnement. Par exemple, l'odométrie dérive, le GPS est imprécis et inopérant en intérieur, et le LiDAR reste coûteux. La solution repose sur la fusion de données, qui combine les mesures de différents capteurs pour réduire incertitudes et erreurs. Les approches les plus courantes incluent le filtre de Kalman étendu et le filtre à particules, utilisé pour estimer la pose à l'aide d'un ensemble de particules représentant des hypothèses de position [44].

1.6.4 Défis en environnement réel et en milieu hospitalier

Même équipés de capteurs avancés, les robots rencontrent des limites en conditions réelles :

- **Bruit de mesure** : erreurs aléatoires inévitables.
- **Occlusions** : obstacles masquant des zones entières, fréquents dans les couloirs hospitaliers.
- **Conditions environnementales** : luminosité variable et surfaces réfléchissantes perturbent caméras et lasers.

En milieu hospitalier, d'autres contraintes apparaissent :

- **Portée et angle limité** des LiDAR 2D, créant des angles morts.
- **Normes d'hygiène et de sécurité** : les capteurs doivent être facilement nettoyables et garantir une détection fiable des patients et du personnel.
- **Variabilité de l'environnement** : gestion d'obstacles dynamiques (patients, brancards en mouvement) et statiques (lits stationnés).

1.7 Exigences spécifiques liées au milieu hospitalier

L'intégration d'un robot autonome dans un environnement hospitalier soulève des exigences particulières qui dépassent les simples considérations techniques. En effet, contrairement à un environnement industriel ou domestique, l'hôpital est un lieu hautement sensible où la sécurité, la fiabilité et l'ergonomie du robot sont primordiales [33].

Les principales exigences se déclinent en plusieurs dimensions :

- **Sécurité des patients et du personnel** : le robot doit être capable d'éviter les collisions, même en présence de personnes vulnérables ou de mouvements imprévisibles dans les couloirs. L'implémentation de systèmes de détection avancée (LiDAR, caméras, capteurs ultrasoniques) est donc essentielle [44].
- **Fiabilité et continuité du service** : un robot de livraison doit pouvoir fonctionner en continu, avec une tolérance minimale aux pannes, afin d'assurer le transport régulier de médicaments ou de matériel médical sans interruption [29].
- **Hygiène et stérilité** : l'hôpital impose des standards stricts de propreté. Le robot doit être conçu avec des matériaux faciles à désinfecter et limiter les contacts physiques

superflus. Certains robots intègrent déjà des surfaces antibactériennes et des protocoles de nettoyage automatisés [46].

- **Autonomie et adaptabilité** : l'environnement hospitalier étant dynamique et complexe (ascenseurs, couloirs étroits, portes automatiques), le robot doit disposer d'algorithmes de navigation robustes et flexibles pour s'adapter à des conditions changeantes [33].
- **Confidentialité et protection des données** : dans un contexte médical, la transmission des informations (par exemple, lorsqu'un robot transporte des médicaments prescrits à un patient particulier) doit respecter les normes de sécurité des données, telles que le RGPD ou les directives locales en matière de santé [46].

Bref, les robots mobiles destinés au milieu hospitalier doivent comporter la sécurité, hygiène, fiabilité et intelligence adaptative, afin de s'intégrer harmonieusement dans l'organisation des soins et soutenir le personnel médical dans ses missions.

1.8. Thématiques scientifiques de la robotique mobile

La robotique mobile autonome est un domaine en pleine expansion, avec des applications variées allant de la logistique à la médecine. Pour un robot de livraison point à point, les axes de recherche les plus pertinents sont :

- **Techniques de localisation et de cartographie** : La navigation précise nécessite l'intégration de capteurs multiples pour déterminer la position du robot et générer des cartes fiables. Les méthodes SLAM sont largement utilisées [33] et permettent de créer des cartes en temps réel. L'utilisation de cartes 2D et 3D ainsi que de Systèmes d'Information Géographique permet d'optimiser les trajets dans des environnements complexes [44].
- **Contrôle et commande des véhicules** : La planification de trajectoires et la commande dynamique sont essentielles pour atteindre les destinations avec précision. Les recherches se concentrent sur l'intégration des contraintes cinématiques et dynamiques du robot ainsi que la gestion des incertitudes de mesure pour garantir une navigation fiable [45].

- **Communication inter-véhicule et coordination** : Dans les systèmes multirobots, la coopération permet l'échange d'informations et la coordination des actions pour accomplir des missions communes [21]. Même pour un robot de livraison unique, ces concepts peuvent servir pour la mise à jour de cartes partagées ou l'optimisation des itinéraires dans un environnement dynamique.
- **Interprétation de scènes et perception avancée** : Pour une navigation sécurisée, le robot doit détecter et interpréter les obstacles mobiles et les objets dans son environnement. Les approches combinant capteurs multiples et intelligence artificielle permettent de prédire le comportement des objets et d'adapter le trajet [20] [36].

1.9. Conclusion partielle

Dans cette partie de notre travail, il était question de présenter le cadre théorique et conceptuel relatif à la navigation des robots mobiles autonomes. Nous avons commencé par définir les notions de base de la robotique mobile et les différents types de robots utilisés en milieu hospitalier. Ensuite, nous avons présenté l'architecture fonctionnelle, les défis liés à la perception et à la localisation, la problématique de la localisation, les exigences spécifiques liées au milieu hospitalier, ainsi que les principales thématiques scientifiques de la robotique mobile. Dans le chapitre suivant, nous allons modéliser notre robot et aborder les principes cinématiques et dynamiques qui permettront de concevoir un prototype adapté au transport de médicaments en milieu hospitalier.

Chapitre 2 : Modélisation et Simulation du Robot de Livraison de Médicaments

2.1 Introduction

Dans ce chapitre, nous allons d'abord modéliser notre robot, c'est-à-dire présenter son architecture générale et son modèle cinématique. Ensuite, nous montrerons comment sera assurée la localisation du robot dans l'environnement, ainsi que la représentation de l'environnement hospitalier et les algorithmes utilisés pour planifier et optimiser ses trajets. Enfin, nous présenterons quelques résultats issus des simulations.

2.2 Modélisation du robot

Le choix d'un robot à entraînement différentiel se justifie par les contraintes de l'environnement hospitalier. Cette architecture offre un excellent compromis : elle est plus maniable qu'un robot de type voiture (elle peut tourner sur place) et beaucoup plus simple et robuste qu'un robot omnidirectionnel, tout en étant parfaitement adaptée à la navigation dans les couloirs étroits et les intersections.

Notre robot de livraison sera modélisé comme un robot différentiel. Un robot à entraînement différentiel est un robot mobile dont le mouvement est basé sur deux roues motrices entraînés indépendamment. Ces deux roues ont généralement le même rayon et sont placées de chaque côté du corps du robot [47]. La figure 10 montre un exemple de robot différentiel :



Figure 10 Robot mobile différentiel REMO [48]

La locomotion de ce type de robot est simplifiée car il est capable de tourner autour de son centre de rotation situé au milieu de l'axe reliant les deux roues ; ce qui n'est pas le cas pour les robots de type voiture. De ce fait, le déplacement d'un robot mobile à entraînement différentiel peut être réalisé en exécutant seulement des mouvements de rotation autour de lui-même et des mouvements de translation ; Ce qui facilite sa commande par rapport à la plupart des robots mobiles. Le terme « différentiel » signifie que la vitesse du véhicule, qui est en train d'exécuter un virage, est déterminée par la différence de vitesse entre ses deux roues. Par exemple, pour effectuer une translation, il faut actionner les deux roues à la même vitesse et dans un même sens. Alors que pour une rotation, il faut actionner les deux roues à la même vitesse, mais dans des sens opposés. La figure 11 illustre les différents mouvements que le robot peut réaliser en fonction des vitesses de ses deux roues.

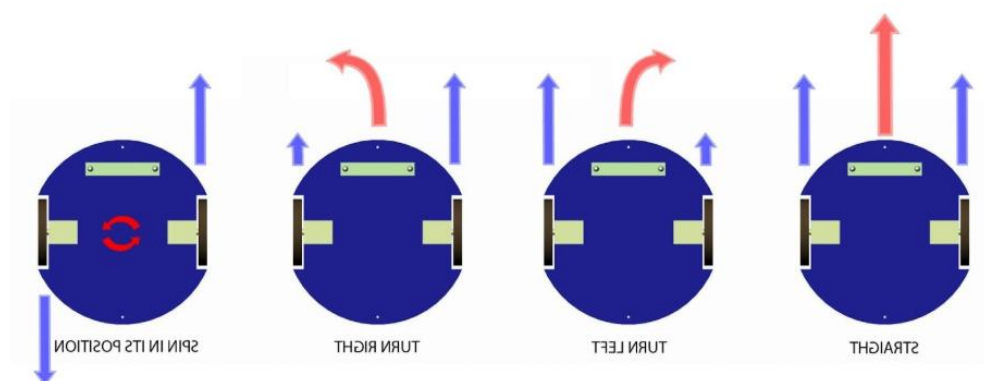


Figure 11 Mouvements d'un Robot à entraînement différentiel [49]

Contrairement à un robot de type voiture, ce robot est considéré moins stable puisqu'il contient deux appuis seulement. Pour cela, une ou plusieurs roues folles (pivotantes) peuvent être ajoutées afin d'assurer son équilibre et sa stabilité. Dans ce travail nous avons utilisé une seule roue folle. La figure 12 représente une roue folle ou pivotante.



Figure 12 Roue Folle ou pivotante [50]

Ainsi, dans la suite de ce travail, nous présentons le modèle cinématique de notre robot, qui permettra de déterminer sa position à tout moment et de faciliter sa localisation.

2.2.1 Modèle cinématique [51]

Comme dit précédemment, notre robot est un robot à entraînement différentielle, c'est-à-dire qu'il possède deux roues placées de chaque côté du robot et se trouvant sur un même axe. Comme illustré à la figure 13, les variables d'entrée sont la vitesse de la roue droite vd et la vitesse de la roue gauche vg . La signification des autres variables est la suivante : CIR est le centre instantané du robot (le point autour duquel le robot est en train de tourner à un instant donné), $2L$ la distance qui sépare les deux roues A et B, r le rayon des roues (elles sont supposées avoir le même rayon), Ω la vitesse angulaire du robot par rapport au CIR et ρ le rayon de courbure de la trajectoire (la distance entre le point CIR et le point C).

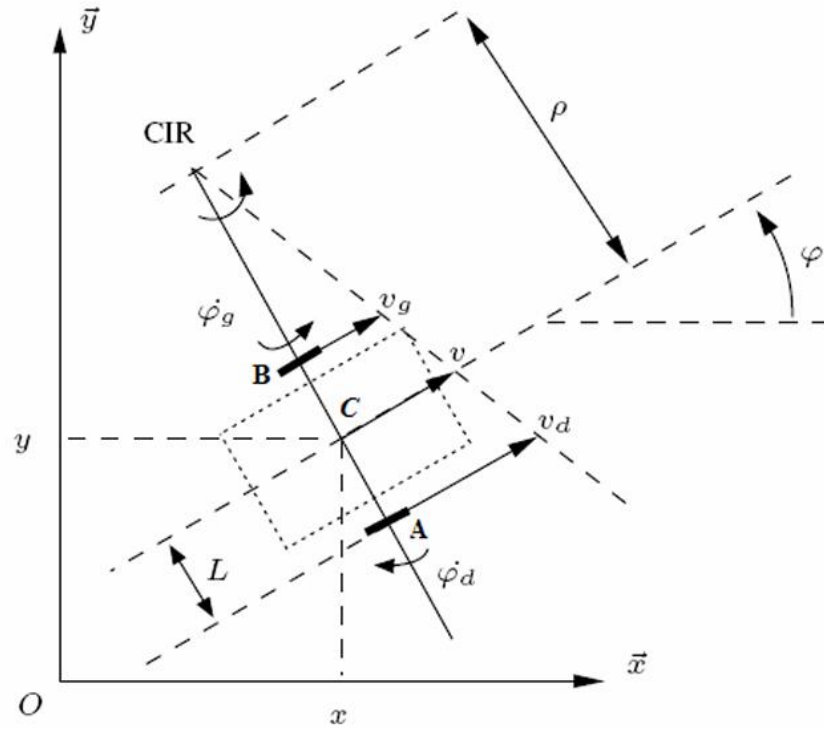


Figure 13 Cinématique d'un robot à entraînement différentielle [52]

La vitesse angulaire a pour expression :

$$\Omega(t) = \frac{vd(t)}{\rho(t) + L} \quad (3)$$

$$\Omega(t) = \frac{vg(t)}{\rho(t) - L} \quad (4)$$

De ces deux expressions précédentes on peut écrire :

$$\rho(t) = L \frac{vd(t) + vg(t)}{vd(t) - vg(t)} \quad (5)$$

$$\Omega(t) = \frac{vd(t) - vg(t)}{2L} \quad (6)$$

La Vitesse linéaire du robot sera alors calculée comme suit :

$$v(t) = \rho(t)\Omega(t) = \frac{vd(t) + vg(t)}{2} \quad (7)$$

Les vitesses linéaires des roues sont :

$$vd(t) = r\dot{\phi}_d \text{ Et } vg(t) = r\dot{\phi}_g$$

Où $\dot{\phi}_d$ et $\dot{\phi}_g$ sont respectivement les vitesses angulaires des roues droites et gauches autour de leurs axes.

L'équation (3) permet de situer le CIR sur l'axe des roues. Par ailleurs, ces équations expliquent deux propriétés particulières du mouvement des robots différentiels :

- Si $vd = vg$ la vitesse angulaire Ω sera nulle et le rayon de courbure ρ est infini. Le robot se déplace donc en ligne droite.
- Si $vd = -vg, \Omega \neq 0$ et ρ est nul, alors le robot effectue une rotation sur lui-même.

Cependant, dans le cas où $vd \neq -vg$, le déplacement du robot est un virage à gauche ou à droite et ceci en fonction du signe de vd par rapport à vg (le virage est dans une direction qui correspond à la vitesse inférieure).

Ainsi, pour analyser la cinématique du robot différentiel, il est nécessaire de définir deux systèmes de coordonnées distincts :

- Un repère local de (Xm, Ym) : Ce repère est solidaire au châssis du robot. Il a comme origine le centre de l'axe reliant les deux-roues donc le point C. L'axe Xm est orienté vers l'avant du robot, Ym est orienté vers sa gauche.
- Un repère global (X, Y) : Ce repère est lié à l'environnement. Il permet de décrire la position et l'orientation du robot dans l'espace par rapport à un référentiel fixe.

En prenant en compte les relations ci-dessus, la cinématique du robot (en coordonnées locales) peut être exprimée comme suit :

$$\begin{bmatrix} \dot{X}_m(t) \\ \dot{Y}_m(t) \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} V_{xm}(t) \\ V_{ym}(t) \\ \Omega(t) \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ -\frac{r}{2L} & \frac{r}{2L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_g(t) \\ \dot{\phi}_d(t) \end{bmatrix} \quad (8)$$

La cinématique externe du robot (en coordonnées globales) est donnée par :

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} \cos\varphi(t) & 0 \\ \sin\varphi(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \Omega(t) \end{bmatrix} \quad (9)$$

Où $v(t)$ et $\Omega(t)$ sont les variables de contrôle.

L'utilisation de ce mode de locomotion fournit une solution simple pour déplacer le robot d'une position à une autre. C'est, sans doute l'une des raisons du succès de ce type de robot.

2.3 Localisation du robot [7]

Le modèle que nous avons présenté précédemment ne permet pas seulement de déterminer la vitesse instantanée du robot. En intégrant ces équations dans le temps, il devient possible d'estimer la position et l'orientation du robot à partir d'une position initiale.

$$\begin{cases} x(t) = x(0) + \int_0^t v(t) \cos\varphi(t) dt \\ y(t) = y(0) + \int_0^t v(t) \sin\varphi(t) dt \\ \varphi(t) = \varphi(0) + \int_0^t \Omega(t) dt \end{cases} \quad (10)$$

Cette méthode est appelée odométrie. Elle permet de localiser le robot en estimant sa position et son orientation à partir des vitesses mesurées par les capteurs.

Cependant, elle a un défaut majeur : la position calculée devient de plus en plus inexacte avec la distance parcourue. En effet, de petites erreurs (comme le glissement des roues ou l'imprécision des capteurs) s'accumulent. Pour une utilisation réelle à l'hôpital, l'odométrie devrait donc être combinée avec d'autres systèmes de localisation plus fiables.

2.4 Configuration du robot et modélisation de l'environnement [51]

Comme vu précédemment, la position de notre robot dans l'environnement est représentée par son abscisse, son ordonnée ainsi que son orientation par rapport à un repère fixe. La configuration du robot dans l'espace de travail est donc représentée par $q = [x, y, \varphi]^T$. Notre robot est modélisé par un cercle de rayon α , qui correspond à sa dimension maximale.

Partant de cette configuration, si le robot se déplace d'une distance d , Sa nouvelle configuration $q' = [x', y', \varphi']^T$ est déterminée par :

$$q' = \begin{bmatrix} x' \\ y' \\ \varphi' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \varphi \end{bmatrix} + \begin{bmatrix} d \cos \varphi \\ d \sin \varphi \\ 0 \end{bmatrix} \quad (11)$$

De même, si le robot tourne d'un angle θ sa nouvelle configuration $q' = [x', y', \varphi']^T$ est déterminée par :

$$q' = \begin{bmatrix} x' \\ y' \\ \varphi' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \varphi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta \end{bmatrix} \quad (12)$$

Si $\theta > 0$, Le robot tourne dans le sens anti horlogique, sinon il tourne dans le sens contraire.

L'ensemble de toutes les configurations possibles du robot définit son espace de configuration, noté C . Pour notre cas, cet espace est donné par : $C = \{(x, y, \varphi) | x, y \in \mathbb{R}, \varphi \in [0, 2\pi[\}$.

Dans l'environnement hospitalier, le robot ne peut pas accéder à toutes les configurations de C en raison de la présence d'obstacles. Soit $O = \cup_i O_i$ la réunion de tous les obstacles fixes (murs, mobilier, etc.) dans l'espace de travail $W \subset \mathbb{R}^2$. Une configuration q est dite valide si le robot $A(q)$ n'intersecte aucun obstacle. Inversement, une configuration est dite invalide si le robot entre en collision avec au moins un obstacle.

L'espace libre C_{libre} est alors défini comme l'ensemble des configurations valides :

$$C_{libre} = \{q \in C | A(q) \cap O = \emptyset\} \quad (13)$$

Pour la navigation pratique dans l'environnement hospitalier, nous adoptons une représentation discrète basée sur une carte topologique sous forme de graphe $G = (V, E)$ où les sommets $V \subset C_{libre}$ représentent des waypoints stratégiques (intersection, portes, points de livraison) et les arêtes E connectent les waypoints entre eux, pondérées par la distance réelle et la difficulté de navigation.

Cette approche hybride combine la rigueur théorique de la modélisation continue avec l'efficacité calculatoire des algorithmes discrets.

La modélisation de l'environnement hospitalier tiendra compte des caractéristiques spécifiques suivantes : les couloirs comme contraintes géométriques linéaires, les portes comme goulots d'étranglement nécessitant une précision de positionnement accrue, et les zones de circulation comme régions pouvant contenir des obstacles dynamiques. Cette modélisation servira de base à l'implémentation des algorithmes de planification de trajectoire qui seront détaillés dans la section suivante.

2.5 Planification de trajectoire

Cette partie constitue l'élément important pour l'autonomie de notre robot. Elle lui permet de déterminer le chemin optimal à suivre dans l'environnement. Cette section décrit l'algorithme A^* , retenu pour son efficacité et son optimalité, et son application aux trajets aller et retour.

2.5.1 Algorithme de planification A^* [53]

Dans le cadre de notre système de navigation robotique, nous utilisons l'algorithme A^* pour permettre au robot de trouver le chemin optimal entre son point de départ et sa destination. Le principe fondamental de cet algorithme repose sur la fonction d'évaluation :

$$f(n) = g(n) + h(n) \quad (14)$$

Où $g(n)$ représente le coût réel depuis le point de départ jusqu'au nœud actuel, et $h(n)$ est l'estimation du coût restant jusqu'à la destination. Cette combinaison permet à l'algorithme de prendre des décisions éclairées tout en garantissant l'optimalité du chemin final.

Pour guider efficacement le robot dans l'environnement hospitalier, nous avons choisi la distance de Manhattan comme fonction heuristique $h(n)$. Cette distance est donnée par :

$$h(n) = |x_1 - x_2| + |y_1 - y_2| \quad (15)$$

Où (x_1, y_1) et (x_2, y_2) représentent respectivement les coordonnées actuelles du robot et de sa destination. Contrairement à la distance euclidienne qui calcule la ligne droite, la distance de Manhattan correspond parfaitement aux déplacements (couloirs rectilignes dans notre cas) et permet au robot d'éviter les obstacles en suivant des trajectoires adaptées à la structure de l'environnement, tandis que la distance euclidienne, basée sur une trajectoire directe en ligne droite, ne permet pas d'éviter les obstacles. L'adéquation de cette heuristique est confirmée par deux propriétés théoriques. Elle est admissible : elle ne surestime jamais la distance réelle dans un environnement quadrillé, garantissant ainsi que l'algorithme trouvera le chemin optimal. Elle est également consistante : pour tout nœud et son voisin, l'estimation vérifie $h(n) \leq \text{coût}(n, \text{voisin}) + h(\text{voisin})$, ce qui rend la recherche plus efficace

La figure 14 représente la différence qui existe entre la distance de Manhattan et la distance euclidienne.

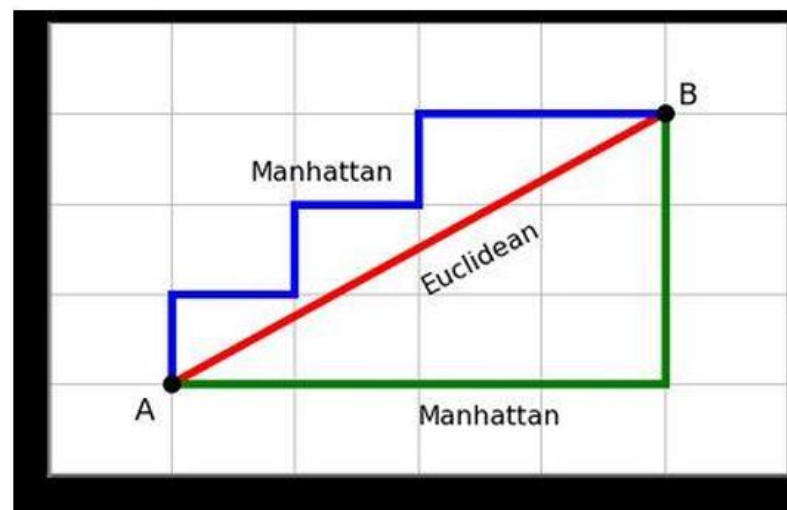


Figure 14 Différence entre distance de Manhattan (couleur bleue et verte) et distance euclidienne (couleur rouge) [51]

Notre choix de la distance de Manhattan repose sur plusieurs avantages déterminants. Premièrement, son adéquation parfaite avec l'architecture de notre environnement composée de couloirs orthogonaux. Deuxièmement, son admissibilité mathématique qui garantit que

l'heuristique ne surestime jamais la distance réelle, assurant ainsi l'optimalité du chemin trouvé. Troisièmement, sa simplicité de calcul qui permet des décisions rapides essentielle pour la navigation en temps réel. La figure 15 illustre un exemple de chemin optimal calculée par l'algorithme A* dans un environnement comportant des obstacle.

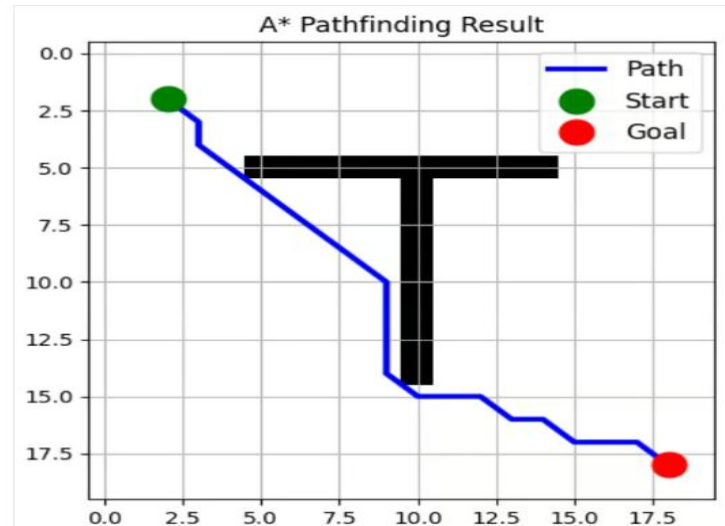


Figure 15: Exemple de chemin optimal calculé par l'algorithme A dans un environnement comportant des obstacles [54].*

Où la forme en T noire représente l'obstacle, le point vert correspond au point de départ (Start), le point rouge indique la destination (Goal) et la trajectoire bleue illustre le chemin optimal obtenu.

Cette manière permet que notre système de navigation combine d'une manière efficace la précision et la performance, permettant à notre robot de se déplacer de manière optimale tout en respectant les contraintes liées à notre environnement.

L'implémentation de l'algorithme A-star, utilisant deux listes pour gérer l'exploitation des nœuds : une liste ouverte pour les nœuds à examiner et une liste fermée pour ceux déjà traités, se présente comme sur la figure 16 :

Algorithme A * Pseudo code

```

Fonction A *(départ, destination)
Initialise Liste ouverte et Liste fermée à liste vide
Ajoute départ à Liste ouverte
Coût  $g(\text{départ}) = 0$ 
Coût  $f = \text{Coût } g(\text{départ}) + \text{Coût } h(\text{départ}, \text{destination})$ 

Tant que la liste ouverte n'est pas vide
    NoeudActuel = Nœud de la liste ouverte avec Coût  $f$  minimum
    Si NoeudActuel = destination
        Retourne chemin

Supprime NoeudActuel de la liste ouverte
Ajoute NoeudActuel à la liste fermée
Pour chaque Voisin dans noeudsvoisins (NoeudActuel)
    Si le Voisin est dans la liste fermée
        Continue
    Coût  $g$  temporaire = cout  $g$  (NoeudActuel) + distance
    (NoeudActuel,
Voisin)
        Si Voisin n'est pas dans la liste ouverte ou si le coût  $g$ 
temporaire < coût
     $g(\text{voisin})$ 
        Si voisin n'est pas dans la liste ouverte
            Ajoute voisin à la liste ouverte
        Coût  $g(\text{voisin}) = \text{Coût } g \text{ temporaire}$ 
        Coût  $f(\text{voisin}) = \text{Coût } g(\text{voisin}) ++ \text{Coût } h(\text{voisin}, \text{destination})$ 
retourne "impossible de trouver le chemin"

```

Figure 16: Algorithme A* [55]

2.5.2 Planification du Chemin d'aller

Pour atteindre efficacement chaque chambre depuis la pharmacie, notre robot utilise une méthode de calcul à la fois simple et performante. Il applique la fonction $f(n)$ présentée précédemment, où $g(n)$ représente la distance exacte déjà parcourue depuis la pharmacie, et $h(n)$ correspond à l'estimation de la distance restante jusqu'à la chambre de destination. Grâce à cette approche, le robot est capable de déterminer en permanence la meilleure direction à suivre afin d'arriver à destination de manière rapide et optimale

2.5.3 Planification du Chemin de Retour

Après chaque livraison, le robot recalcule automatiquement son chemin soit pour retourner à la pharmacie, soit pour se rendre à la chambre suivante. Cette fonction est très utile car elle permet de s'adapter aux imprévus : par exemple, si un couloir devient temporairement bloqué, le robot trouvera un autre chemin.

Pour le retour, il utilise la même formule $f(n)$, où $g(n)$ représente cette fois la distance parcourue depuis la chambre et $h(n)$ estime la distance restante jusqu'à la pharmacie ou la prochaine chambre.

Ainsi notre système de navigation est conçu que pour les environnements hospitaliers. Il privilégie les couloirs où la circulation est simple, et cela en évitant les zones encombrées. Le robot se déplace exclusivement dans les couloirs et suit le chemin optimal calculé pour atteindre une chambre spécifique. Une fois arrivé à sa destination, il s'arrête automatiquement, garantissant ainsi un arrêt précis devant la chambre pour effectuer sa livraison.

2.6 Gestion des obstacles dynamiques

Dans un environnement hospitalier, les obstacles rencontrés par le robot ne sont pas uniquement statiques (murs, mobilier), mais peuvent également être mobiles, tels que le personnel, les patients, les brancards ou d'autres robots. La capacité du robot à détecter et à réagir à ces obstacles est essentielle pour assurer une navigation sûre et efficace.

2.6.1 Priorité des obstacles et comportement adaptatif

Pour améliorer la sécurité et l'efficacité, le robot applique un système de priorité lorsqu'il rencontre simultanément plusieurs obstacles dynamiques. Les obstacles humains (personnel, patients) sont prioritairement évités, tandis que les obstacles mobiles non critiques, tels que les chariots ou petits objets, peuvent être contournés par de légers ajustements de trajectoire. Cette hiérarchisation permet au robot de maintenir une navigation fluide tout en minimisant les arrêts inutiles et en respectant les contraintes de sécurité de l'environnement hospitalier.

2.6.2 Détection des obstacles mobiles

La détection des obstacles mobiles repose sur les capteurs embarqués du robot, notamment les capteurs à ultrasons et infrarouges, qui permettent de mesurer en temps réel la distance aux objets environnants. Lorsqu'un obstacle mobile est identifié sur la trajectoire du robot, le système déclenche automatiquement une alerte interne indiquant la présence de l'obstacle. Cette information est utilisée pour adapter le comportement de navigation sans interrompre la mission de livraison.

2.6.3 Replanification locale avec l'algorithme A*

Pour gérer les obstacles temporaires, le robot applique une stratégie de replanification locale. Concrètement, lorsqu'un obstacle bloque partiellement ou totalement un couloir, le robot : Identifie la zone occupée et la considère temporairement comme inaccessible, Applique l'algorithme A* sur le graphe de waypoints afin de déterminer un chemin alternatif permettant de contourner l'obstacle. Une fois l'obstacle contourné, le robot réintègre automatiquement le chemin optimal initial dès que celui-ci redevient accessible.

Cette approche garantit la continuité de la mission du robot tout en minimisant les risques de collision avec les éléments dynamiques de l'environnement hospitalier.

2.6.4 Replanification dynamique et planification prédictive

En complément de la replanification locale, le robot intègre une planification prédictive basée sur les données de capteurs et la carte topologique de l'hôpital. Cette approche permet d'anticiper les zones à forte probabilité de présence d'obstacles (intersections, portes, zones de passage fréquent) et d'ajuster la vitesse et la trajectoire avant même de rencontrer l'obstacle. Ainsi, le robot peut réduire les ralentissements inutiles et améliorer l'efficacité globale de ses missions de livraison.

2.6.5 Exemple de scénario

Par exemple, si un couloir menant à une chambre spécifique est temporairement occupé par un brancard, le robot détecte cet obstacle grâce à ses capteurs. Il active alors la replanification locale et choisit un couloir alternatif pour atteindre sa destination tout en maintenant une distance de sécurité avec l'obstacle. Lorsque le couloir initial redevient libre, le robot reprend le chemin optimal prévu pour poursuivre ses livraisons. Cette stratégie permet une navigation adaptative et fluide, même dans un environnement hospitalier en constante évolution.

Cette stratégie adaptative permet également de gérer plusieurs obstacles mobiles simultanément. Par exemple, si deux couloirs sont temporairement encombrés, le robot choisira prioritairement la voie la moins critique et la plus sûre en fonction de la hiérarchisation des obstacles, garantissant ainsi la continuité de la mission et la sécurité des personnes dans son environnement.

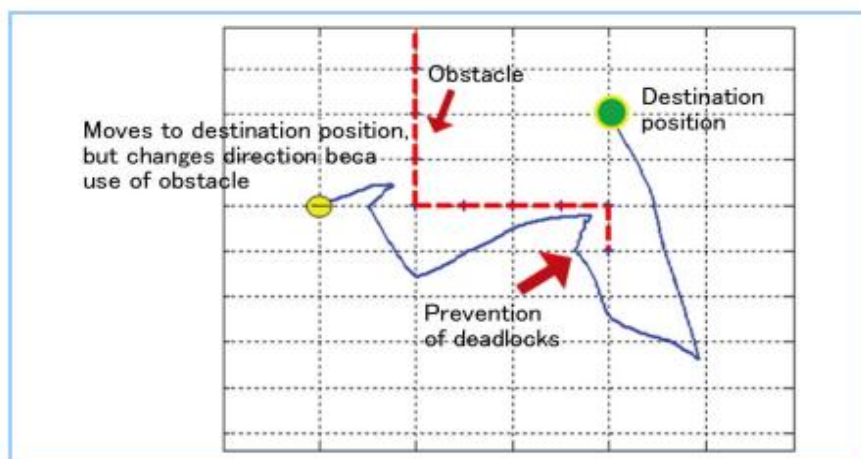


Figure 17: Illustration de la gestion des obstacles dynamiques par le robot. Le robot détecte un obstacle temporaire (personnel, patient...) dans le couloir et adapte sa trajectoire en choisissant un chemin alternatif avant de revenir sur le chemin optimal [56]

2.7 Simulations et résultats

Pour montrer l'efficacité de l'approche proposée, une simulation a été réalisée en employant un environnement hospitalier construit contenant des obstacles. La position de tous les obstacles est inconnue, le robot ne connaît que les positions de départ (pharmacie) et d'arrivée(chambres). La simulation a été fait dans l'environnement Webots en utilisant le modèle cinématique du robot présenté précédemment.

Le comportement normal dans la navigation autonome du robot mobile est de le guider vers sa cible. Sur son chemin immédiat, le robot essaie de naviguer vers la cible et lorsqu'il s'approche ou rencontre des obstacles, le comportement concerné est exécuté avec un degré convenable. Une fois le robot est en dehors de la zone de collision, il continue sa tâche en se dirigeant vers la cible et en suivant le chemin le plus court. La Figure 17 montre un exemple de navigation autonome du robot mobile dans l'environnement. Au départ, le robot essaye de se déplacer en ligne droite se basant sur l'algorithme d'odométrie, mais lorsqu'il détecte la présence des obstacles dans son chemin, le comportement d'évitement d'obstacles est activé. Pour cet effet, nous avons réalisé une simulation en 3D sur le logiciel Webots qui est spécialisé pour la robotique.

La navigation du robot est réalisée en adaptant dynamiquement sa vitesse en fonction de la distance à la cible et des contraintes de l'environnement. Dans certaines zones, notamment à l'approche des portes et des chambres, des waypoints sont utilisés afin de faciliter les manœuvres d'entrée et de sortie et de réduire les risques de collision avec les obstacles. La détection des obstacles est assurée par les capteurs embarqués, ce qui permet l'activation automatique du comportement d'évitement lorsque cela est nécessaire.

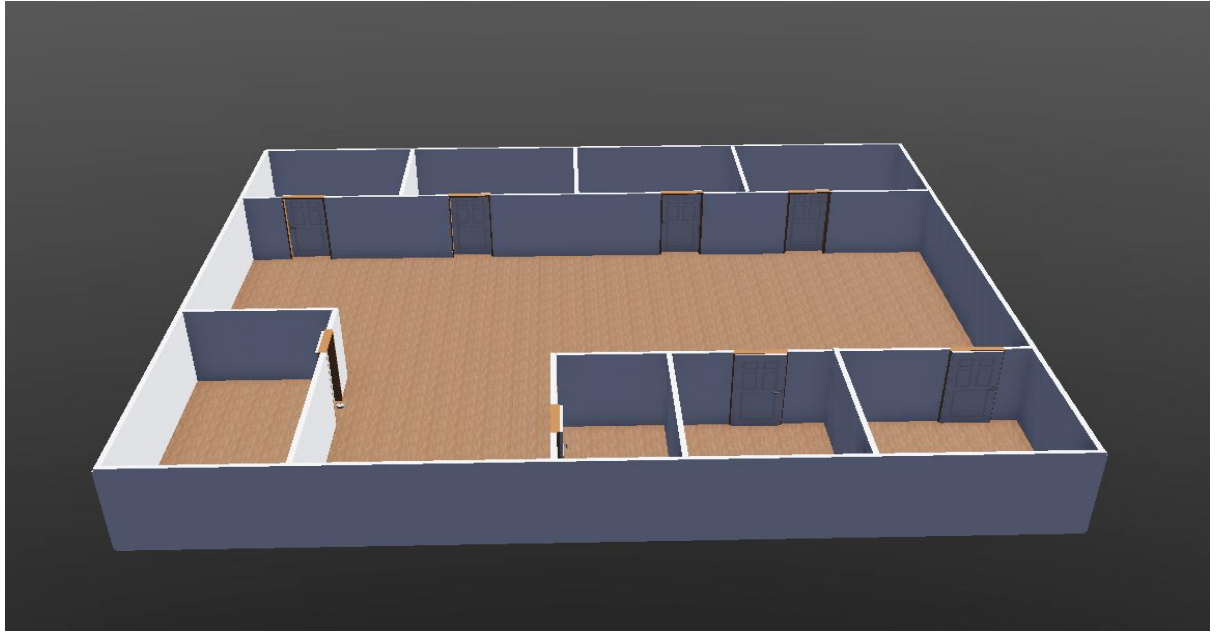


Figure 18. Présentation de l'environnement hospitalier (la vue générale des couloirs, chambres)

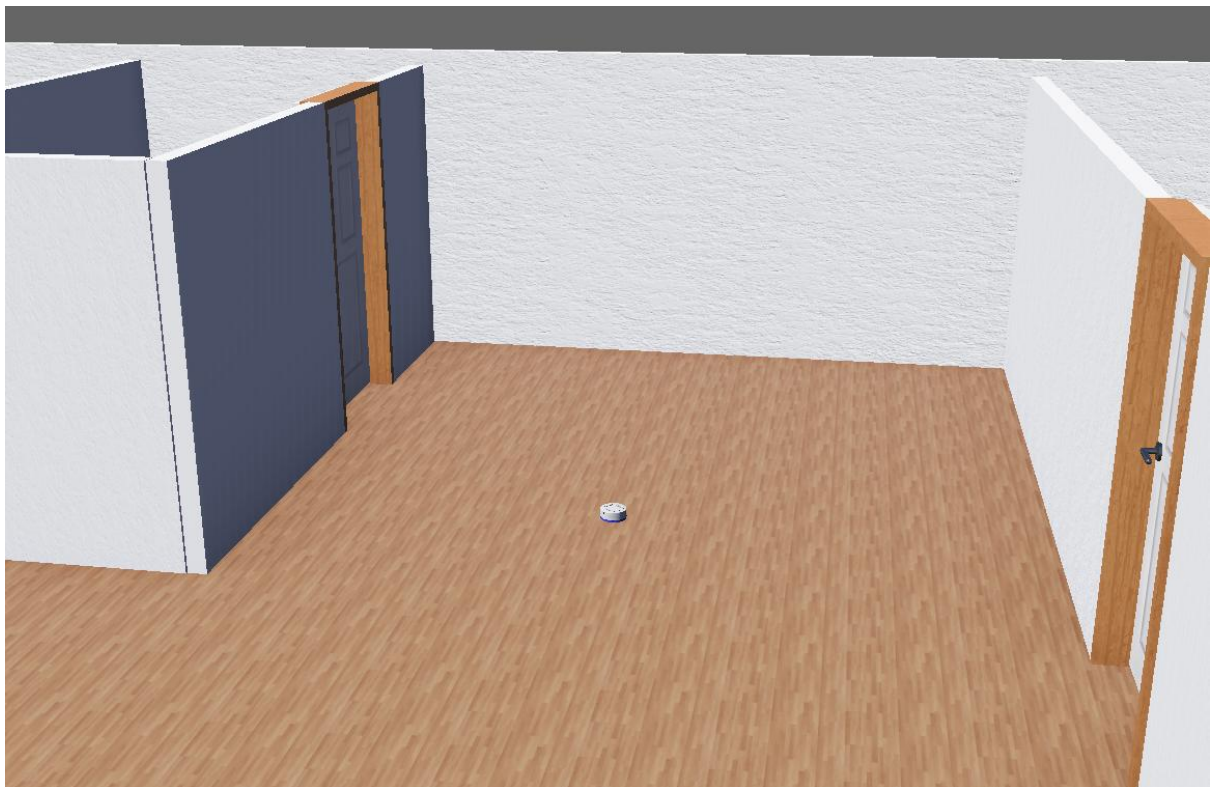


Figure 19. Navigation du robot dans les couloirs

Les figures 18 et 19 représentent la disposition de l'environnement et le comportement de navigation autonome du robot lors de la simulation (Un extrait du code de simulation est présenté en ANNEXE 1).

L'analyse des simulations démontre que le système de navigation proposé, combinant l'algorithme A* et la replanification locale, permet au robot de se déplacer de manière autonome et sûre dans un environnement hospitalier complexe. La navigation prédictive et l'anticipation des obstacles contribuent à réduire les arrêts inutiles et à maintenir un chemin proche de l'optimal. L'observation détaillée des simulations permet de tirer deux constat principaux. D'une part, le système a démontré une fiabilité opérationnelle en accomplissant ses missions sans collision. D'autre part, on observe que les trajectoires réelles présentent parfois des écarts par rapport au chemin théorique optimal, notamment dans les virages serrés ou lors des manœuvres d'évitement. Ces écarts, inhérents à la dynamique du robot et aux impératifs de sécurité, n'ont pas compromis les livraisons mais identifient une limite du système actuel : pour des applications exigeant un positionnement millimétrique (comme l'alignement parfait devant un distributeur), un contrôleur de trajectoire plus précis serait nécessaire. Ainsi, ces résultats valident l'approche globale tout en circonscrivant le domaine d'application optimal.

2.8 Conclusion partielle

Dans ce chapitre, nous avons tout d'abord exposé le modèle cinématique du robot différentiel, qui sera fondamental pour sa commande et sa localisation. Ensuite, nous avons détaillé la méthode employée pour la navigation autonome dans l'environnement hospitalier, et enfin, nous avons présenté certains des résultats obtenus à travers des simulations. Ainsi le chapitre suivant sera consacré à la mise en œuvre du prototype.

Chapitre 3 : Réalisation du Robot

3.1 Introduction

Dans ce chapitre, nous présentons la réalisation pratique de notre robot de livraison. Tout d'abord, nous expliquons l'approche utilisée pour la localisation du robot. Ensuite, nous présentons les différents composants utilisés pour la réalisation du système, ainsi que les difficultés rencontrées au cours du développement. Par la suite, nous décrivons le fonctionnement général du robot. Enfin, nous présentons le prototype final réalisé.

3.2 Présentation générale du robot

Notre robot a pour rôle principal de transporter les médicaments d'une chambre à une autre, pour réduire les déplacements du personnel soignant. Il est destiné à fonctionner dans un hôpital, où il se déplace dans les couloirs et devant les chambres. Pendant son déplacement, il peut rencontrer des obstacles. Pour cela, il est équipé de capteurs qui lui permettent de détecter les obstacles et de se déplacer en toute sécurité vers sa destination.

De manière générale, notre robot est composé d'une plateforme mobile avec des moteurs, des capteurs pour la perception de l'environnement et des cartes ESP32 qui assurent le contrôle du robot. Une interface utilisateur permet aux infirmières de communiquer avec notre robot. Tous ces éléments travaillent ensemble pour permettre à notre robot d'exécuter correctement sa mission de livraison.

3.3. Circuit du robot Hospitalier

Le schéma présenté sur la figure 20, est une illustration de comment sont connectés les composants. Il a été réalisé avec le logiciel Fritzing.

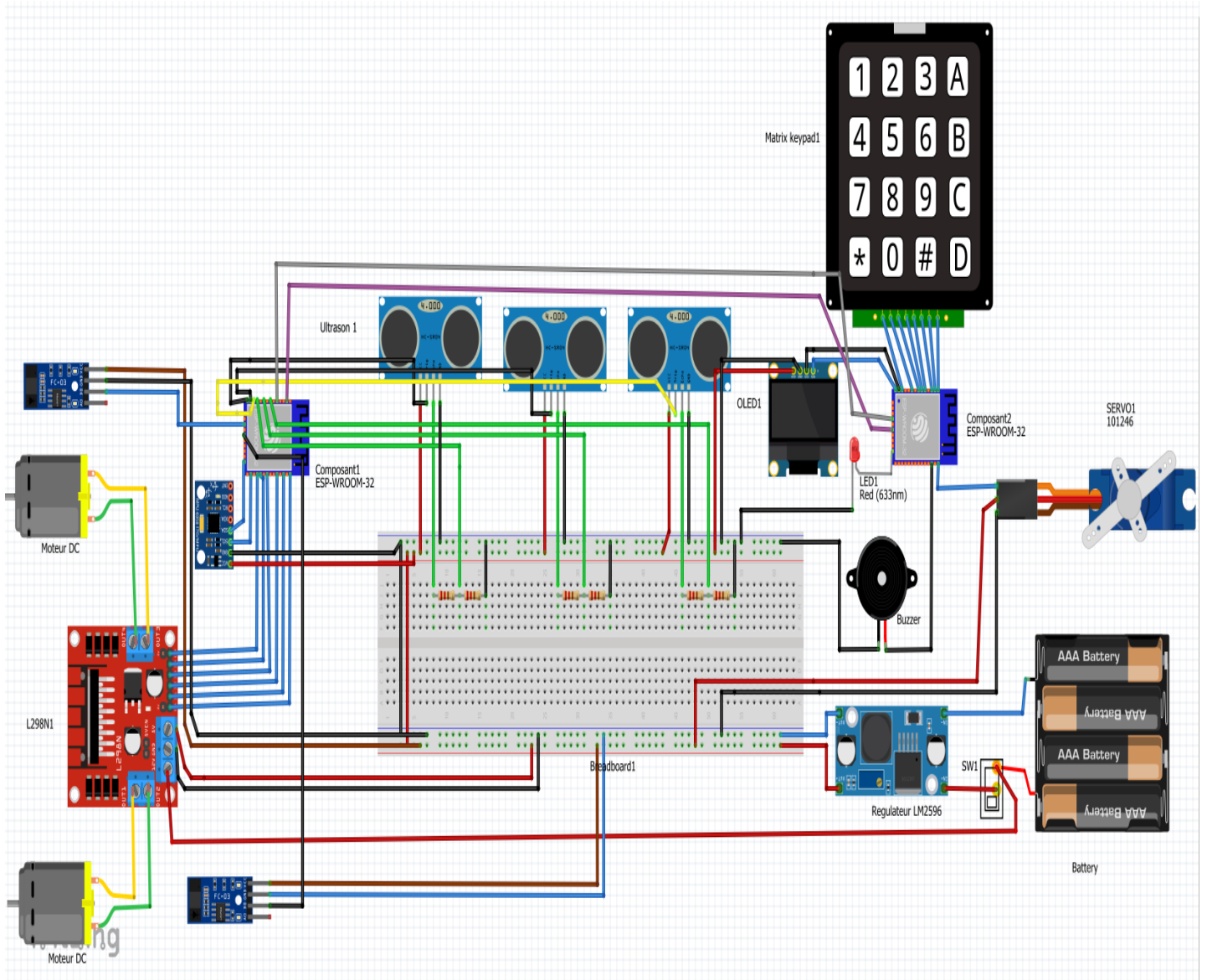


Figure 20: Circuit de notre robot

3.4. Composants utilisés

Pour parvenir à réaliser notre robot, voici les composants que nous avons utilisés :

3.4.1. Cartes de développement ESP32-WROOM-32

Nous avons utilisé deux cartes ESP32 comme éléments centraux de notre robot. La première carte ESP32 a été utilisée pour la commande des moteurs, des capteurs ultrasoniques, du MPU, des encodeurs, ainsi que pour la communication avec l'application mobile. Cette carte a été utilisée comme le cerveau de notre système, car elle assure le contrôle et la prise de décision du robot. La seconde carte ESP32 a été utilisée pour la commande de la porte du compartiment sécurisé et pour l'interface utilisateur. Elle est connectée à un clavier, un écran OLED, une LED verte, un servo-moteur et un buzzer. Les deux cartes ESP32 utilisent une communication UART bidirectionnelle, ce qui permet l'échange des informations et le bon fonctionnement de notre robot.

L'ESP32 est une carte de développement basée sur un microcontrôleur double cœur intégrant le Wi-Fi et le Bluetooth, et elle est programmable via l'IDE Arduino. Grâce à ses entrées/sorties, elle permet de recevoir les signaux des capteurs et de commander différents actionneurs [57].

La figure 21 présente une carte de développement ESP32-WROOM-32 utilisée dans notre robot.

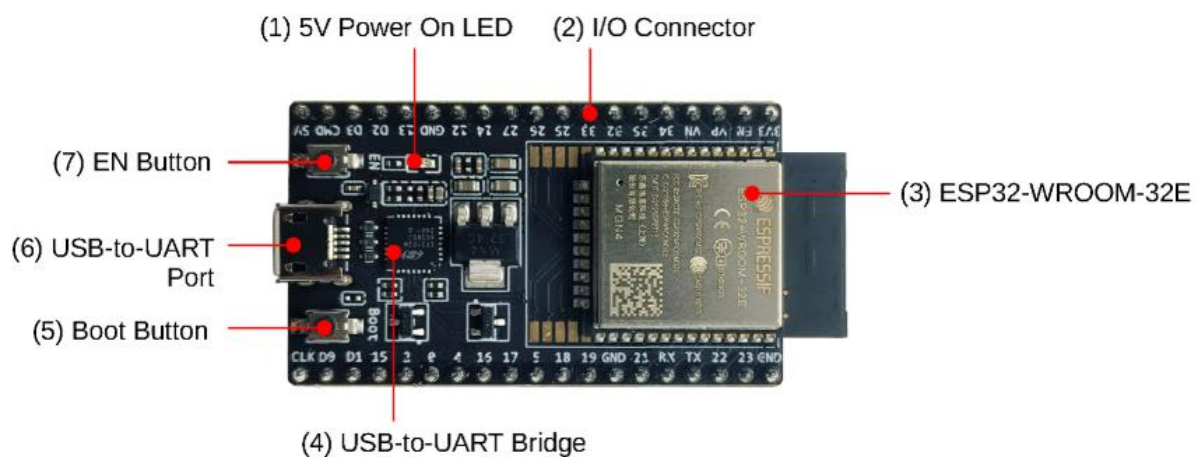


Figure 21 : ESP32-DevKitC V4 avec module ESP32-WROOM-32E soudé [57]

3.4.3. Capteur de distance HC-SR04

Nous avons utilisé les capteurs ultrasoniques pour éviter que notre robot entre en collision avec les obstacles. Cependant, l'écho de ces capteurs envoie un signal de 5 V, alors que la carte ESP32 ne peut supporter que 3,3 V. Pour protéger la carte, nous avons conçu pour chaque ultrason un pont diviseur de tension, voir la figure 20.

La figure 22 représente le capteur ultrason HC-SR04



Figure 22: Capteur ultrason HC-SR04 [51]

3.4.4. Le module MPU9250

Le MPU9250 est un module intégrant trois capteurs principaux : un gyroscope (mesure des vitesses de rotation selon les trois axes), un accéléromètre (mesure des accélérations linéaires selon les trois axes) et un magnétomètre 3 axes. Il inclut également un capteur de température. Dans notre projet, nous l'avons utilisé pour déterminer l'orientation du robot en mesurant l'angle de rotation autour de l'axe Z (le cap). La figure 51 présente le module MPU9250 que nous avons utilisé.



Figure 23: Module MPU9250 [51]

3.4.5. Le capteur de vitesse LM393 et disque d'encodeur

Le capteur de vitesse LM393 est un module à infrarouge qui mesure la rotation des roues. Il détecte les impulsions d'un disque codeur et transmet ces signaux à notre carte ESP32 pour calculer la vitesse de chaque roue. Dans notre travail, nous avons utilisé deux modules, un pour chaque roue, afin de maintenir la direction du robot droite et de déterminer la distance parcourue. Cette rectification de la trajectoire est réalisée par l'exploitation différentielle des impulsions issues des deux encodeurs : la distance parcourue par chaque roue est comparée, puis une correction est appliquée aux commandes des moteurs afin de compenser toute différence de rotation. Par ailleurs, les impulsions sont intégrées dans l'algorithme d'odométrie pour estimer la position et l'orientation du robot au cours de son déplacement. C'est grâce à ce composant que nous avons pu confier à notre robot une mission spécifique. Nous l'avons également utilisé pour l'adaptation de vitesse. Si, le MPU9250 faisait avancer le robot droit, il présentait cependant un peu de bruit. La combinaison de ces deux capteurs nous a permis d'obtenir une navigation optimale pour notre robot. Les figures 24 et 25 représentent respectivement un capteur de vitesse à LM393 et un disque encodeur.

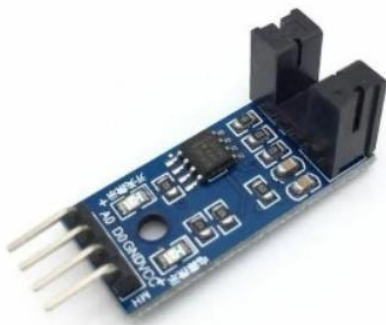


Figure 24 Capteur de vitesse à LM393 [58]



Figure 25 Disques d'encodeur [59]

3. 4.6. Les moteurs à courant continu

Pour notre projet, nous avons utilisé deux moteur7070s à courant continu 3V-6V (qui peuvent absorber un courant max de 220mA) pour donner la vitesse voulue au robot. La figure 26 montre l'image d'un moteur à courant continu



Figure 26: Moteur à courant continu du type TT [51]

3.4.7 Driver moteur L298N

La carte ESP32 ne peut pas commander directement certains moteurs, à cause des limites de ses sorties en termes de tension et de courant. De plus, pour faire fonctionner un moteur en marche avant ou en marche arrière, il est nécessaire de contrôler le sens du courant appliqué au moteur, ce qui demande un circuit de commande adapté. Pour notre robot, nous avons utilisé le driver moteur L298N un module à double pont H, qui permet de commander les moteurs à partir des signaux envoyés par l'ESP32. Ce driver facilite le contrôle du sens de rotation et de la vitesse des moteurs, tout en protégeant la carte ESP32. Il peut supporter une tension d'alimentation comprise entre 5V à 35V et un courant de 2A par canal. Le L298N a été choisi car il est facile à utiliser, accessible et bien adapté à notre application.

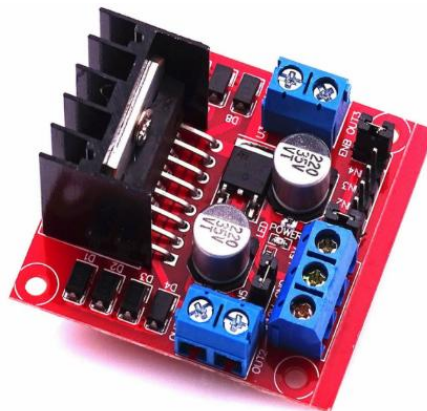


Figure 27: Le driver L298N [51]

3.4.8. Batteries Li-ion

Nous avons utilisé 3 batteries couplés en série qui nous ont servis pour l'alimentation de l'ensemble de notre robot. Leurs caractéristiques sont : 2500mAh ; 4,4 A et 3,7V. La figure 28 représente une image de deux de ces batteries



Figure 28: Batteries rechargeables en Lithium [51]

3.4.9. Le régulateur de tension (LM2596)

C'est un module basé sur le LM2596. Il fournit toutes les fonctions d'un régulateur abaisseur à découpage, capable de piloter une charge de 3A avec une excellente régulation de la charge. Pour une tension d'entrée comprise entre 3V et 40V en sortie on peut avoir entre 1.5V et 37V. Pour notre robot, nous l'avons alimenté avec une tension Constante de 11,8V et en sortie on a réglé à 5V pour l'alimentation de tous nos capteurs. La figure 29 représente une image de ce module :

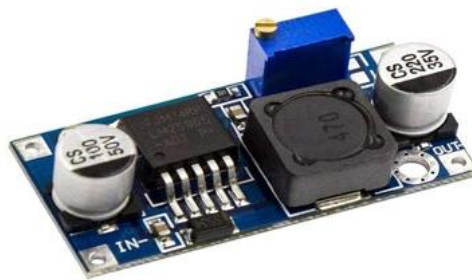


Figure 29: Le module LM2596 [51]

3.4.10. Un clavier Matriciel

Ce composant a été utilisé dans notre système pour ouvrir et fermer le compartiment sécurisé où sont stockés les médicaments, en fonction du code saisi.

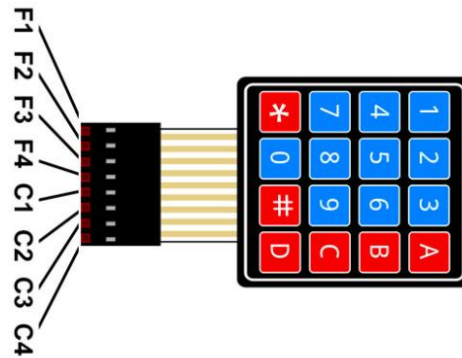


Figure 30: Clavier Matricielle 4X4 [60]

3.4.11. Un écran OLED

L'écran OLED a été utilisé dans notre système pour afficher les informations en temps réel de ce que fait notre robot. La figure 31 représente l'image d'un écran OLED

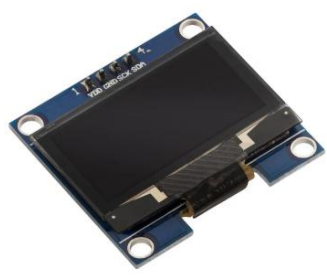


Figure 31: Ecran OLED Adafruit [61]

3.4.12. Une LED, Buzzer

Nous avons utilisé une LED verte et un buzzer pour la signalisation. Quand le robot arrive devant une chambre, ils permettent d'indiquer à l'utilisateur(infirmière) que le robot est arrivé à la destination.



Figure 32: Diode Led [62]



Figure 33: Buzzer [63]

3.4.13. Un module de gestion et de protection de batteries (BMS)

Dans le cadre de l'alimentation de notre système nous avons utilisé un module des gestions de batterie (Battery Management system), de 12V/10A pour la charge et protéger notre système contre la décharge excessive, les courts-Circuits.

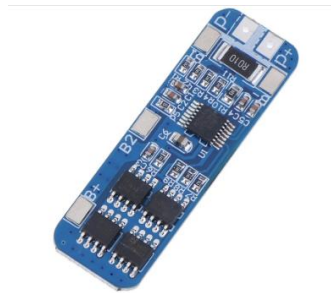


Figure 34: Module de gestion de batteries [64]

3.5. Approche utilisée pour la localisation du robot

Le grand défi avant même de penser à la localisation de notre robot était de le faire avancer en ligne droite car il ne serait pas possible de pouvoir donner à notre robot une mission spécifique sans que le robot puisse avancer en ligne droite pour s'orienter vers sa cible. Théoriquement, si les deux roues motrices reçoivent exactement la même vitesse de rotation, le robot devrait se déplacer en ligne droite. Cependant, les essais expérimentaux ont montré qu'il dévie de sa trajectoire. Cette déviation peut s'expliquer par plusieurs facteurs, notamment une répartition non homogène des masses sur le châssis, de légères différences de diamètre effectif des roues, des tolérances mécaniques, ou encore des variations dans les performances des moteurs. Le

robot finissait par dévier de sa trajectoire, ce qui a été pour nous un défi majeur de pouvoir donner au robot une mission précise. Ainsi, nous avons mis en œuvre une boucle de rétroaction, dans laquelle la sortie du système est réinjectée en entrée afin de corriger l'erreur de trajectoire. Cette régulation repose sur un contrôleur PID (Proportionnel–Intégral–Dérivé), associé aux capteurs encodeurs des roues. Les paramètres du correcteur ont été ajustés expérimentalement afin d'assurer une correction efficace et une stabilisation de la trajectoire du robot. Dans la section suivante nous expliquons comment cela a été effectué.

3.6. Correcteur PID et Encodeurs

Afin de garantir un déplacement rectiligne précis et stable, notre robot mobile différentiel utilise une régulation en boucle fermée basée sur un correcteur PID (Proportionnel–Intégral–Dérivé) associé aux encodeurs incrémentaux montés sur chaque roue. Cette approche permet de corriger automatiquement les écarts entre la trajectoire réelle du robot et la trajectoire de référence, assurant ainsi une trajectoire droite et une bonne précision de positionnement.

Les encodeurs mesurent la rotation des roues en générant un nombre d'impulsions proportionnel à l'angle parcouru. À partir du nombre d'impulsions comptées durant une période d'échantillonnage donnée, il est possible d'estimer le déplacement linéaire de chaque roue selon la relation [65] :

$$d = \frac{2\pi r N}{P} \quad (16)$$

Où r est le rayon de la roue, N le nombre d'impulsions comptées pendant la période T , et P le nombre total d'impulsions par tour de l'encodeur. Cette relation découle directement de la cinématique du mouvement circulaire et constitue une méthode standard d'estimation du déplacement en robotique mobile [65].

Le correcteur PID calcule ensuite un signal de commande à partir de l'erreur d'orientation du robot par rapport à la direction désirée :

$$e(t) = \theta_{\text{réf}} - \theta_{\text{mesuré}} \quad (17)$$

La loi de commande continue s'écrit :

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (18)$$

Dans notre implémentation sur microcontrôleur (ESP32), une forme discrète est utilisée (Un extrait du programme embarqué est présenté en ANNEXE 2) :

$$u(k) = K_p e(k) + K_i \sum_{i=0}^k e(i) \Delta t + K_d \frac{e(k) - e(k-1)}{\Delta t} \quad (19)$$

Où K_p , K_i et K_d représentent respectivement les gains proportionnel, intégral et dérivé, et Δt le pas d'échantillonnage. Dans notre travail, nous avons retenus expérimentalement les valeurs suivantes : $K_p = 2.7$, $K_i = 0.10$, $K_d = 0.30$, $\Delta t = 20ms$

Pour maintenir une trajectoire rectiligne, la correction du PID est appliquée de manière différentielle aux moteurs gauche et droit, ce qui permet d'ajuster automatiquement les vitesses de rotation afin de compenser les écarts d'orientation du robot.

La distance totale parcourue est estimée par :

$$d = \frac{d_g + d_d}{2} \quad (20)$$

Ce qui correspond au modèle cinématique standard des robots différentiels [66].

Les paramètres du PID ont été réglés expérimentalement, afin d'obtenir une réponse rapide, une erreur statique minimale et une trajectoire stable sans oscillations excessives. Grâce à cette combinaison encodeurs-PID, le robot est capable de maintenir une vitesse constante, de corriger automatiquement les déviations et d'exécuter avec précision les missions programmées.

3.7. Localisation du robot

Pour localiser notre robot, dans le chapitre 2, nous avons montré que théoriquement avec l'odométrie il était possible de localiser le robot grâce à son modèle cinématique. Mais avec

notre robot, nous avons essayé cette méthode et les résultats n'étaient pas assez précis. Les mesures des encodeurs, utilisées pour mesurer la distance parcourue par chaque roue, contenaient des bruits et la distance parcourue variait énormément. Par exemple, si nous avons programmé que le robot s'arrête à 1 m, parfois le robot faisait moins d'un mètre et s'arrêtait à 75 cm. Nous avons pensé à utiliser des méthodes statistiques en essayant de prendre une médiane ou une moyenne sur un ensemble fixé de valeurs, mais cela n'a pas beaucoup amélioré la précision du robot. Étant donné que les encodeurs produisaient des signaux instables, nous avons également envisagé de recourir aux valeurs d'accélération du MPU9250. Cependant, une fois de plus, nous avons constaté une importante présence de bruit.

Nous avons donc décidé de ne pas nous appuyer uniquement sur les impulsions des encodeurs, mais de trouver la valeur de la distance parcourue par des expérimentations. En maintenant une vitesse constante pour les roues avec un signal PWM de 160 (sur une échelle de 0 à 255), nous avons effectué des tests sur une distance de référence de 1 mètre. Au début de nos essais, nous avons remarqué que le robot s'arrêtait à 75 cm au lieu de parcourir le mètre complet. C'est au fur et à mesure des essais que nous avons réglé de manière expérimentale les paramètres de temps et de vitesse pour que notre robot parcoure réellement 1 mètre. À partir de cette vitesse, nous avons ensuite utilisé les équations discrètes du chapitre 2 pour calculer la position du robot en temps réel (Un extrait du code de navigation est présenté en ANNEXE 2). Cette approche, combinée aux données du gyroscope pour maintenir le cap, a permis d'obtenir une navigation bien plus fiable pour atteindre les chambres.

3.8. Construction de l'environnement

Pour tester notre robot, nous avons construit un environnement simulant un Couloir hospitalier. Nous avons placé des points de repère au sol et mesuré les distances qui les séparent afin de faciliter la localisation du robot. Ensuite, nous avons programmé différentes missions pour le robot, en choisissant dans notre cas 4 chambres seulement comme destinations. Au départ le robot se trouve à sa position initiale qu'on a appelé pharmacie.

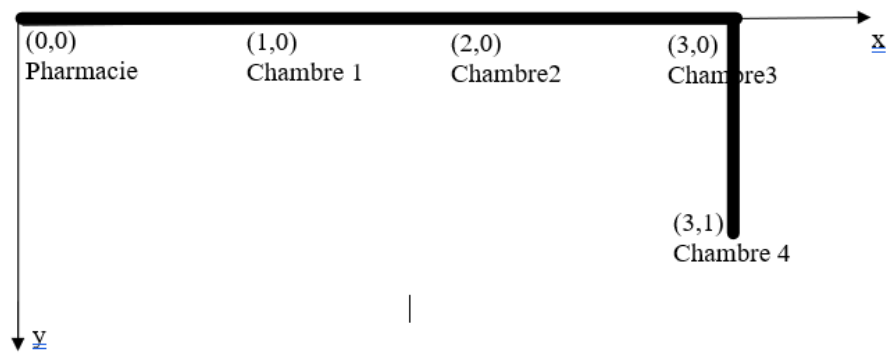


Figure 35: Notre environnement de test

3.9. Les modes de fonctionnement robot

Notre robot peut soit fonctionner en mode autonome (automatique), soit en mode manuelle (téléguidé) ; cela se fait grâce à une simple application mobile conçue pour notre projet (Un extrait du programme de gestion de l'interface est présenté en ANNEXE 3). Le mode automatique correspond au démarrage d'une mission, tandis que le mode Manuelle, le robot peut être dirigé, soit en avant, en arrière, à Gauche, à Droite. La figure 36 représente la capture de notre application mobile.

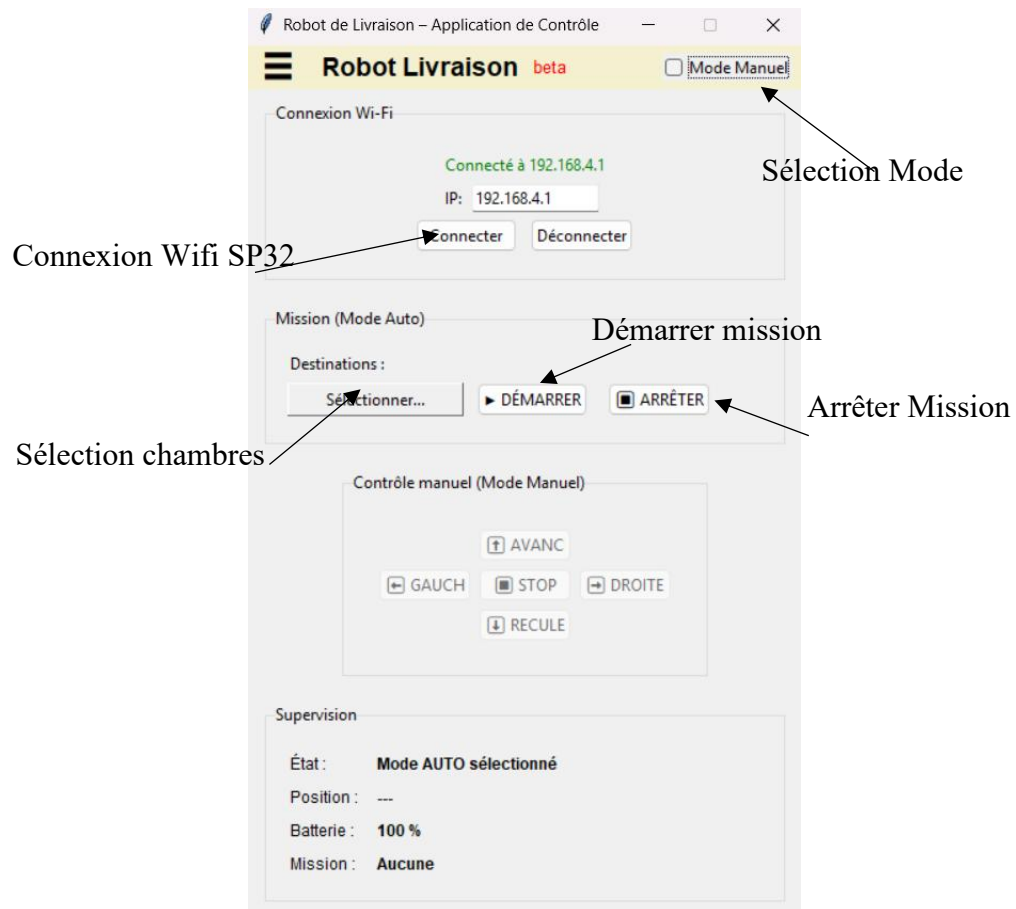


Figure 36: Capture de l'application Mobile

3.10. Le robot physique

Pour la réalisation du robot, nous avons imprimé un châssis en 3D, qui sert de support à notre prototype. Le modèle 3D a été conçu dans SolidWorks. Étant donné que le robot reste un prototype, nous l'avons câblé avec de petits cordons et ajouté de la colle chaude pour que les composants soient bien fixés. Pour certains composants qui l'exigeaient, nous avons également réalisé des soudures avec l'étain. La figure 37 montre une capture de dessus de notre robot.

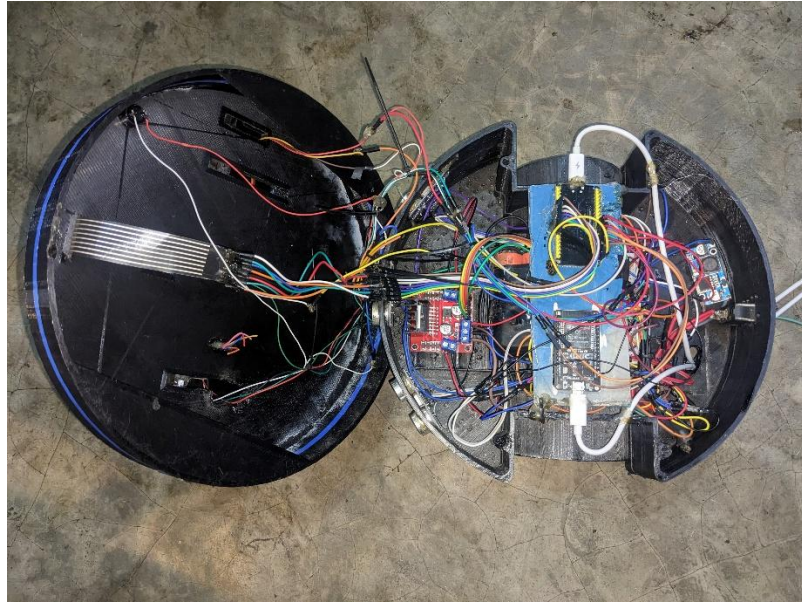


Figure 37: Vue de dessus de notre robot

Aussi nous avons prévue des encodeurs pour pouvoir calculer la distance du robot en temps réel et parvenir à le localiser facilement dans notre environnement. La figure ci-dessous représente une capture de la vue de dessous de notre robot.

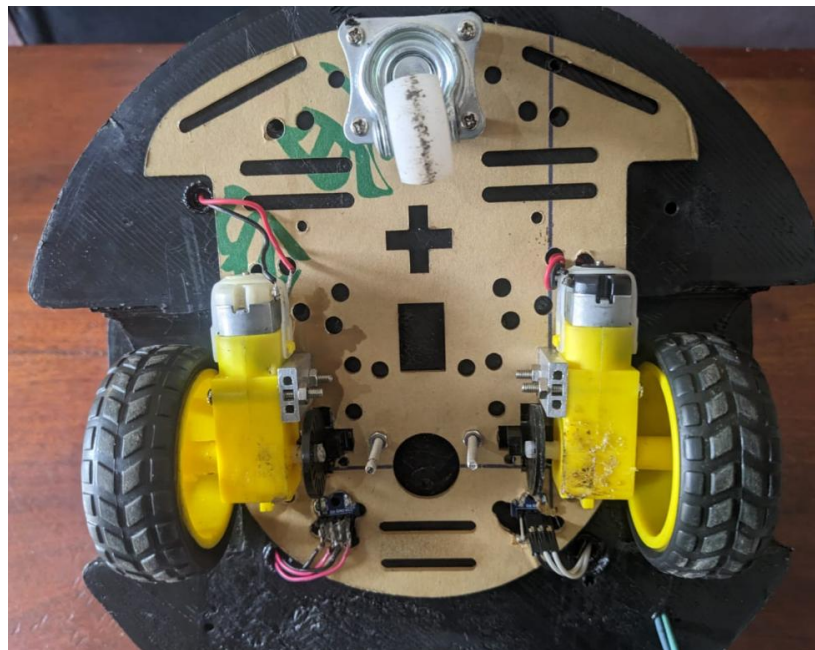


Figure 38: Vue de dessous de notre robot

Sur l'image suivante nous présentons une capture de l'interface utilisateur et le compartiment sécurisé de notre robot (Un extrait du programme utilisé est présenté en ANNEXE 3)



Figure 39: Interface utilisateur de notre robot

3.11 Conclusion Partielle

Dans ce chapitre, nous avons présenté les résultats de la conception de notre robot. Après avoir défini ce que nous voulions réaliser, le premier problème a été de faire avancer le robot en ligne droite. Nous avons réussi à stabiliser son déplacement et à calibrer la mesure des distances parcourues. Ensuite, la sélection des composants et leur intégration sur le châssis ont permis d'obtenir un prototype fonctionnel. Ce prototype a été testé dans un environnement que nous avons spécialement construit pour nos essais. L'ensemble de ces résultats montre que notre robot est viable et capable de réaliser les missions prévues.

Conclusion générale

Au terme de ce travail, nous avons réalisé un prototype de robot de livraison autonome capable de naviguer dans un environnement hospitalier simulé. Face aux obstacles, le robot réagit et prend des décisions grâce à ses capteurs à ultrasons. Une application mobile complète le système et permet de choisir le mode de fonctionnement. Nous avons d'abord présenté le cadre théorique de la robotique mobile, les défis de la navigation autonome et les capteurs utilisés pour la perception. Nous avons ensuite présenté l'approche que nous allions utiliser, basée sur la modélisation cinématique d'un robot différentiel et l'algorithme de planification A* pour la génération de trajectoire. Nous l'avons testée en simulation avec des résultats concluants, montrant une navigation efficace et un évitement d'obstacles (Un extrait du code de simulation est présenté en ANNEXE 1). Nous avons fini par la présentation des difficultés rencontrées, notamment la calibration de l'odométrie et l'intégration matérielle, et nous avons pu réaliser et tester le prototype physique.

Par rapport aux hypothèses prises au début de ce travail, nous avons pu montrer que les méthodes de navigation référencées capteurs sont bien adaptées aux couloirs d'hôpitaux, en utilisant avec succès des ultrasons pour la détection. Nous avons démontré par le biais de simulations et du prototype que le robot était capable de générer une trajectoire vers des points de livraison prédéfinis (pharmacie, chambres) et de la suivre. Enfin, nous avons validé qu'une interface simple (application mobile, clavier, écran) permettait une interaction efficace avec le personnel médical et que les mécanismes de sécurité de base (évitement, compartiment sécurisés) étaient fonctionnels.

Comme perspective à ce travail pour les futurs chercheurs, nous pensons qu'avec des moyens matériels supplémentaires, il serait possible d'implémenter un système de localisation plus robuste (comme le SLAM avec un LiDAR low-cost) proposé dans le cadre théorique. Aussi, pour ceux qui veulent aller plus loin, ils pourraient perfectionner la gestion des obstacles dynamiques en intégrant des algorithmes de prédiction de trajectoire pour les piétons.

Bibliographie

- [1] G. F. Elena De Momi, *Medical Robotics*, Londres: Academic Press(Elsevier), 2022.
- [2] W. H. Organization, «Chronic staff shortfalls stifle Africa's health systems,» OMS Bureau régional de l'Afrique, 15 avril 2022. [En ligne]. Available: <https://www.afro.who.int/news/chronic-staff-shortfalls-stifle-africas-health-systems-who-study>. [Accès le 12 juillet 2025].
- [3] M. S. F. A. Campaign, «Left Behind by the HIV Response,» Médecins Sans Frontières, 12 juin 2017. [En ligne]. Available: https://msfaccess.org/sites/default/files/MSF_assets/HIV_AIDS/Docs/MSF%20Kinshasa%20Report%20ENG%20WEB.pdf. [Accès le 12 juillet 2025].
- [4] A. P. (AP), «Cameroon or Canada? Poorly paid doctors and nurses are choosing to leave...,» The Associated Press, 11 juin 2024. [En ligne]. Available: <https://apnews.com/article/2becf962f5b207e20b2ca9d8a84fd3b3>. [Accès le 12 juillet 2025].
- [5] L. G. Juan Martinez, *Mobile Robots in Healthcare: Improving Medicine Delivey With Autonomous Systems*, Barcelone: IEE/Springer, 2020.
- [6] Université de Lille, *Cours de Robotique Mobile*, Lille: Presses Universitaires de Lille, 2020.
- [7] Faculté des sciences – Université de Montréal, «Laboratoire de robotique – Travaux sur la navigation autonome,» Université de Montréal, Service des Publications Scientifiques, 10 Septembre 2019. [En ligne]. Available: <https://robotique.umontreal.ca>. [Accès le 16 août 2025].
- [8] U. d. T. d. C. (UTC), «Robotique et systèmes embarqués appliqués à la santé,» Presses Universitaires de Compiègne, Compiègne, 2021.
- [9] INRIA, «Systèmes robotiques autonomes et perception,» 5 juin 2024. [En ligne]. Available: <https://inria.fr>. [Accès le 16 août 2025].

- [10] U. d. Sherbrooke, «Robotique mobile et localisation,» Université de Sherbrooke, Service des Publications Académiques, 12 avril 2020. [En ligne]. Available: <https://usherbrooke.ca>. [Accès le 16 août 2025].
- [11] ISO, «ISO 8373: Robots and robotic devices — Vocabulary,» Organisation internationale de normalisation (ISO Central Secretariat), 15 Novembre 2012. [En ligne]. Available: <https://www.iso.org/standard/55890.html>. [Accès le 16 août 2025].
- [12] Craig, J. J., Introduction to Robotics: Mechanics and Control (4e éd.), Boston: Pearson Education, 15 janvier 2018.
- [13] B. & K. O. (. Siciliano, Springer Handbook of Robotics, Berlin: Springer-Verlag, 10 juin 2016.
- [14] T. & O. H. Doi, Robots in Healthcare: Applications and Challenges, Amsterdam: Elsevier, 20 septembre 2020.
- [15] M. M. J. J. & C. M. C. Tavakoli, Robotics in Healthcare: Design and Applications, Hoboken: Wiley-IEEE Press, 8 mars 2021.
- [16] B. S. L. V. L. O. G. Siciliano, Modélisation, Planification et Contrôle, Springer, Éd., Londres, Royaume-Uni: Springer, 2010.
- [17] S. M. LaValle, Algorithmes de Planification, Cambridge, Royaume-Uni: Cambridge University Press, 2006.
- [18] R. M. Murray, Introduction Mathématique à la Manipulation Robotique, Boca Raton, FL, États-Unis: CRC Press, 1994.
- [19] ResearchGate, « Difference between the motion of the Holonomic and Non-Holonomic robots,» 2020. [En ligne]. Available: https://www.researchgate.net/figure/Difference-between-the-motion-of-the-Holonomic-and-Non-Holonomic-robots_fig1_360850849. [Accès le 17 août 2025].
- [20] E. Guizzo, «How Google's self-driving car works,» *IEEE Spectrum*, 2008.
- [21] P. Wurman, R. D'Andrea et M. Mountz, «Coordinating hundreds of cooperative, autonomous vehicles in warehouses,» *AI Magazine*, vol. 29, n° 11, pp. 9-20, 2008.

- [22] V. (. d. Automobilindustrie), «Automated Guided Vehicles in the Industry,» Verband der Automobilindustrie, 2020. [En ligne]. Available: <https://www.vda.de/en/services/Publications/Automated-Guided-Vehicles.html>. [Accès le 17 août 2025].
- [23] B. Van Laarhoven et e. al., «Autonomous Mobile Robots for Hospitals: A Survey,» *Robotics and Autonomous Systems*, vol. 115, pp. 102-118, 2019.
- [24] J. Worth, «Finding the Right Fit for your Industrial Automation Need - AGVs or AMRs,» 2024. [En ligne]. Available: <https://ez.analog.com/ez-blogs/b/engineerzone-spotlight/posts/finding-the-right-fit-for-your-industrial-automation-need---agvs-or-amrs>. [Accès le 17 août 2025].
- [25] Aethon, «TUG-ROBOT autonome de transport hospitalier,» Aethon, 2023. [En ligne]. Available: <https://aethon.com/>. [Accès le 12 Juillet 2025].
- [26] S. Healthcare, «Relay-Solution de robot mobile autonome,» Swisslog Healthcare, 2023. [En ligne]. Available: <https://www.swisslog-healthcare.com/>. [Accès le 12 juillet 2025].
- [27] M. I. Robots, «Robots mobiles MiR pour la logistique hospitalière,» Mobile Industrial Robots, 2023. [En ligne]. Available: <https://mobile-industrial-robots.com/fr/les-industries/hospitals>. [Accès le 12 Juillet 2025].
- [28] M. Simon, «This Incredible Hospital Robot Is Saving Lives. Also, I Hate It,» WIRED, 11 février 2015. [En ligne]. Available: <https://www.wired.com/2015/02/incredible-hospital-robot-saving-lives-also-hate/>. [Accès le 17 août 2025].
- [29] R. Robotics, «Relay Robotics,» "Relay Robotics proposes autonomous navigation for indoor delivery robots", 31 octobre 2024. [En ligne]. Available: <https://relayrobotics.com/relay-robotics-proposes-autonomous-navigation-for-indoor-delivery-robots/>. [Accès le 17 août 2025].
- [30] C. Walton, «Logistics Manager,» "RARUK unveils MiR250 autonomous mobile robot", 20 avril 2020. [En ligne]. Available: <https://www.logisticsmanager.com/raruk-unveils-mir250-autonomous-mobile-robot/>. [Accès le 17 août 2025].
- [31] J.-C. Latombe, *Robot Motion Planning*, Boston, MA: Springer Science & Business Media, 1991.

- [32] S. M. LaValle, *Planning Algorithms*, Cambridge, UK: Cambridge University Press, 2006.
- [33] R. N. I. R. & S. D. Siegwart, *Introduction to Autonomous Mobile Robots* (2nd ed.), Cambridge, MA: MIT Press, 2011.
- [34] W. B. D. F. Sebastian Thrun, *Probabilistic Robotics*, vol. 4, Cambridge, Massachusetts: MIT Press, 2005.
- [35] R. R. Murphy, *Introduction to AI Robotics* (2^e édition), Cambridge, Massachusetts: MIT Press, 2019.
- [36] G. B. Greg Welch, «An Introduction to the Kalman Filter,» University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC, 1995.
- [37] J. H. W. B. Daniel Meyer-Delius, «Probabilistic Occupancy Grids for Dynamic Environments,» chez *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-12)*, Toronto, Canada, 2012.
- [38] M. S. C. S. W. B. Henrik Kretzschmar, «Socially Compliant Mobile Robot Navigation via Inverse Reinforcement Learning,» *The International Journal of Robotics Research (IJRR)*, vol. 35, n° 11, pp. 1289-1307, 2016.
- [39] G. Tagne Fokam, Thèse de doctorat, -Université de Technologie de Compiègne (UTC), 2015.
- [40] P. Giguère, *Planification de trajectoire*, N. d. c. / s. p. PDF, Éd., Université Laval, Département d'informatique et de génie logiciel, 2017.
- [41] O. Khatib, «Real-Time Obstacle Avoidance for Manipulators and Mobile Robots,» *The International Journal of Robotics Research*, vol. 5, n° 1, pp. 90-98, 1986.
- [42] D. B. W. & T. S. Fox, «The Dynamic Window Approach to Collision Avoidance,» chez *IEEE International Conference on Robotics and Automation (ICRA)*, Albuquerque, USA, 1997.
- [43] M. N. Zennir, *Nouvelles approches en robotique pour la navigation autonome dans les environnements partiellement connus*, Université de Jijel, 2017.

- [44] W. B. D. F. Sebastian Thrun, Probabilistic Robotics, Cambridge, Massachusetts, USA: MIT Press, 2005.
- [45] Y. M. e. A. MILOUDI, *Développement d'algorithmes d'évitement d'obstacles statiques et dynamiques*, uargla: Université Kasdi Merbah-Ouargla, 2012.
- [46] J. W. C. e. J. D. Creswell, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, Los Angeles, 2018, p. 275.
- [47] Z. Y. KACHOUR, CONTRÔLE DU MOUVEMENT D'UN ROBOT MOBILE, Juin 2019.
- [48] ROS-Mobile-Robots, «ROS Mobile Robots,» REMO robot front side view, [En ligne]. Available: https://ros-mobile-robots.com/resources/remo/remo_front_side.jpg. [Accès le 15 Septembre 2025].
- [49] B. Electronics and Robotics Club (ERC), «Drive Mechanism,» [En ligne]. Available: https://erc-bpgc.github.io/handbook/mechanical/drive_mechanism/?utm_source=chatgpt.com.
- [50] [En ligne]. Available: https://www.google.com/search?sca_esv=f04038ad81816743&udm=2&fbs=AIJpHxU7SXXniUZfeShr2fp4giZgjZjkPoOvWxp274gTXGx4B9C470XSt2qfqhRlvIaZCIYajdCQTdl-75V9J1b-sAlJ1h_Gb0aAHKERO-0xZm57BshyJHqojEukQ6enjXsrM-xcTLPzxLw8s1YW9K-eePQReQqhSRrCNPH6TF3wvy4Y0EemJA&q=une+.
- [51] B. A. Séraphin, CONCEPTION ET REALISATION D'UN ROBOT NETTOYEUR, Goma: Faculté de Science et technologies ULPGL, 2022-2023.
- [52] H. OUBADJI, Commande non linéaire et hybride d'un robot mobile unicycle, Juin 2010.
- [53] R. Kumar, «DataCamp,» 2024 Novembre 2024. [En ligne]. Available: <https://www.datacamp.com/fr/tutorial/a-star-algorithm>. [Accès le 17 Octobre 2025].
- [54] R. Kumar, «hme A* : un guide complet.»,» ataCamp, 14 Février 2025. [En ligne]. Available: <https://www.datacamp.com/fr/tutorial/a-star-algorithm>. [Accès le 20 Janvier 2026].

- [55] M. Shakour, Conception et implémentation d'un algorithme de planification de chemin dans un jeu vidéo comportant un environnement triangulisé (Mémoire de maîtrise), Québec, Canada: Université du Québec à Montréal, 2012.
- [56] Y. e. al., Development of Obstacle Avoidance Control for Robotic Products Using Potential Method, vol. 15, Mitsubishi Heavy Industries Technical Review, mars 2014.
- [57] E.-D. U. G. Espressif Systems. [En ligne]. Available: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/user_guide.html?utm_source=chatgpt.com. [Accès le 22 01 2026].
- [58] S. E. Components, LM393 Speed Sensor Module - Technical Specifications and Application, Shenzhen, Chine: SEC Publishing, 2021.
- [59] H. C. P. O. Robert Martinez, Optical Encoding Disks: Principles and Applications in Motion Control Systems, New York, USA: Springer Robotics Series, 2021.
- [60] C. m. A. Le Disrupteur Dimensionnel. [En ligne]. Available: <https://ledisrupteurdimensionnel.com/arduino/claviers-matriciels/>. [Accès le 23 01 2026].
- [61] 1. I. O. D. AZ-Delivery. [En ligne]. Available: <https://www.az-delivery.de/fr/products/1-3zoll-i2c-oled-display>. [Accès le 23 01 2026].
- [62] led. [En ligne]. Available: <https://www.techitall.co/produit/led-verte/>. [Accès le 23 01 2026].
- [63] Buzzer. [En ligne]. Available: https://www.google.com/search?sca_esv=fd565832262b37c0&sxsrf=ANbL-n4wWysXVbsZkYN1IJJ1t2hDvbWuJA:1769127672029&udm=2&fbs=ADc_1-aN0CWEZBOHjofHoaMMDiKp26dgc35auQCScVYlfJtMBpoLTs0sZLQzbb7x_3_gmMxoXObiJDsFHAb1yXtxdvIakbtXor4xU0_sGOXUojDy9zaiKFpbXiu0UFoM_2FsJy. [Accès le 23 01 2026].
- [64] D. Electronics, «3S 10A Li-ion Battery Protection Board Datasheet,» 2024. [En ligne].
- [65] M. J. G. Dudek, Computational Principles of Mobile Robotics, Cambridge University Press, 2010.

- [66] N. Correll, INTRODUCTION TO AUTONOMOUS ROBOTS, University of Colorado at Boulder, 13/01/2026.
- [67] P. Tournassoud, *Planification de trajectoire et contrôle en robotique : application aux robots mobiles et manipulateurs. Mémoire de recherche*, Université de Tlemcen, Algérie., 1992.
- [68] K. AG, «AGV vs AMR: Quelle est la différence ?,» KNAPP, juillet 2023. [En ligne]. Available: <https://www.knapp.com/fr/savoir/blog/difference-agv-vs-amr/>. [Accès le 12 juillet 2025].
- [69] D. Filliat, «Introduction à la robotique mobile,» Developpez.com, 2008. [En ligne]. Available: <https://david-filliat.developpez.com/tutoriel/robotique-mobile/>. [Accès le 12 Juillet 2025].
- [70] [En ligne]. Available: http://lsc.univ-evry.fr/public/presentation/Rapport_activite/Rapport2001/RVVision.html.
- [71]
- [72] K. M. L. a. F. C. Park, MODERN ROBOTICS MECHANICS, PLANNING, AND CONTROL, May 3, 2017.

ANNEXES

ANNEXE 1

Extrait code python pour la simulation

```
from controller import Robot
import math
import socket
import threading
import queue

# INITIALISATION

robot = Robot()
TIME_STEP = int(robot.getBasicTimeStep())

# COMMUNICATION SOCKET

command_queue = queue.Queue()
mission_active = False
chambers_to_deliver = []
current_chamber_index = 0
state = "EN ATTENTE"
def socket_server():
    """Serveur socket dans un thread séparé"""
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(('127.0.0.1', 9999))
    server.listen(1)
    print(" Serveur socket: 127.0.0.1:9999")
    while True:
        try:
            client, addr = server.accept()
            print(f" Client connecté: {addr}")
            while True:
                data = client.recv(1024).decode().strip()
                if not data:
```

```

        break
    print(f' Commande: {data}')
    ...

```

ANNEXE 2

Extrait code Arduino

```

#include <WiFi.h>
#include <Wire.h>
#include <MPU9250_asukiaaa.h>
#include <Ticker.h>
#include <math.h>
#include <vector>

//===== 1. CONFIGURATION RÉSEAU=====
const char* ssid = "ROBOT_LIVRAISON";
const char* password = "12345678";
WiFiServer server(5000);
WiFiClient client;

//===== 2. CONFIGURATION MATÉRIELLE=====
namespace Pins {
    const int I2C_SDA = 21;
    const int I2C_SCL = 22;
    const int LEFT_ENC = 13;
    const int RIGHT_ENC = 15;
    const int MTR_ENA = 25;
    const int MTR_IN1 = 26;
    const int MTR_IN2 = 27;
    const int MTR_ENB = 14;
    const int MTR_IN3 = 32;
    const int MTR_IN4 = 23;
    const int TRIG_AVANT = 5;
    const int ECHO_AVANT = 34;
    const int TRIG_GAUCHE = 18;
    const int ECHO_GAUCHE = 35;
    const int TRIG_DROIT = 19;
    const int ECHO_DROIT = 4;
}

//=====3. PARAMÈTRES DE NAVIGATION=====
namespace Config {
    const float WHEEL_RADIUS = 0.030;

```

```

const float WHEEL_BASE    = 0.147;

const int  TICKS_PER_REV  = 40;
const float METERS_PER_TICK = (2.0f * PI * WHEEL_RADIUS) /
(float)TICKS_PER_REV;

const float L_FAC = 1.00f;
const float R_FAC = 1.18f;

const int VITESSE_MISSION = 220;
const int VITESSE_RETOUR  = 220;
const int VITESSE_EVITEMENT = 180;
const int VITESSE_MANUEL  = 220;

const int GRID_SIZE = 100;
const float CELL_SIZE = 0.10f;
const float DISTANCE_TOLERANCE = 0.08f;
const int DIST_OBSTACLE = 28;
const int SIDE_CLEARANCE_MARGIN = 10;

const int NUM_SAMPLES = 10;
const float MAX_VALID_CM = 250.0f;
...

```

ANNEXE 3

Extrait du code Arduino pour l'interface utilisateur

```

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH110X.h>
#include <Keypad.h>
#include <ESP32Servo.h>

// ===== OLED SH1106 =====
#define i2c_Address 0x3c
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SH1106G display = Adafruit_SH1106G(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, OLED_RESET);
const int LED_PIN = 5;
const int BUZZER_PIN = 19;
const int SERVO_PIN = 18;
Servo porteServo;

```

```
// ===== CLAVIER 4x4 =====  
const byte ROWS = 4;  
const byte COLS = 4;  
...
```