République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université BLIDA-1-

Faculté de Technologie Département d'automatique et d'électrotechnique.

PROJET DE FIN D'ETUDE

Pour l'obtention du diplôme de MASTER en Automatique et Informatique industrielle & Automatique et système.

Thème :

Développement d'un robot mobile assistant au service de la santé.

Préparé par :

HASNI IDRISS

DJOUDI YOUCEF ALAA EDDINE

Dirigé par :

Mme. BOURAINE Sara

Mr. KAZED Boualem

Année Universitaire :

2021/2022

Dédicaces

J’exprime mes sentiments les plus profonds et je dédie ce modeste travail

A mon père, ma mère, pour l’éducation qu’ils ont sue me donner et qui m’a permis avec l’aide de DIEU d’arriver là où je suis.

A ma belle-sœur et mes chers frères et à toute ma famille.

A tous mes amis et tous ceux qui m’ont aidé afin de réaliser ce travail.

YOUCEF ALAA EDDINE

Dédicaces

Je dédie ce mémoire

A mes chers parents, ma mère et mon père pour leur patience, leur amour, leur soutien et leur encouragement.

A ma Chère sœur et Chères frères

A toute ma famille.

A tous mes amis et tous ceux m'aiment et qui m'ont aidé afin de réaliser ce travail.

IDRISS

Remerciements

Tout d’abord, on tient à remercier -ALLAH- Le tout puissant de m’avoir donné le courage et la patience durant toutes mes années d’études.

Nous présentons nos sincères remerciements avec notre profond respect à nos protomotrice : **Madame BOURAINE Sara** pour la confiance qu'elle nous a accordée, ses encouragements, et ses précieux conseils, et pour son effort dans ce travail.

Nous remercions chaleureusement notre encadrant

Monsieur KAZED Boualem pour sa sympathie, ses conseils, son soutien, ses idées et sa disponibilité.

Nous remercions également les membres du jury pour leurs efforts et leur relecture de ce travail.

Nous remercions nos chers parents pour leurs encouragements.

Enfin, nous tenons également à remercier tous ceux qui m'ont aidé de près ou de loin dans la réalisation de ce modeste ouvrage.

Résumé

La tendance actuelle en robotique s'oriente vers le développement de robots assistants en milieu professionnel au service de la santé, l'agriculture, l'éducation ou également dans les foyers pour usages personnels. Toutefois, parmi ces domaines, l'usage en santé gagne plus de terrain tirant parti du contexte sanitaire. Pour ce faire, il est nécessaire de développer des plateformes robotiques qui permettent d'assister le corps médical et de les dégager de certaines tâches quotidiennes pour lui permettre de se consacrer aux soins des malades. Le projet proposé s'inscrit dans cette thématique et a pour objectif de concevoir et de réaliser un robot assistant au service de la santé. La plateforme robotique à développer est de type différentiel, conçue pour transporter des charges telles que des médicaments, des repas, les affaires des malades, etc. La première phase du travail est l'étude et la réalisation de la base mobile. La deuxième phase est liée à l'aspect embarqué qui consiste à développer une architecture matérielle et une architecture logicielle, intégrer le système de perception et le système de navigation. Enfin, la troisième phase est consacrée aux tests et validations.

Mots clés : robot mobile, robot assistant, navigation autonome, Robot Operating System.

ملخص

يتجه الاتجاه الحالي في مجال الروبوتات نحو تطوير الروبوتات المساعدة في مكان العمل في خدمة الصحة أو الزراعة أو التعليم أو أيضًا في المنازل للاستخدام الشخصي.

ومع ذلك، من بين هذه المجالات، يكتسب الاستخدام الصحي المزيد من الأرض، مع الاستفادة من السياق الصحي.

للقيام بذلك، من الضروري تطوير منصات آلية يمكنها مساعدة مهنة الطب وتحريرهم من بعض المهام اليومية للسماح لهم بتكريس أنفسهم لرعاية المرضى.

يندرج المشروع المقترح ضمن هذا الموضوع ويهدف إلى تصميم وإنتاج روبوت مساعد في خدمة الصحة.

المنصة الروبوتيك التي سيتم تطويرها هي من النوع التفاضلي، وهي مصممة لنقل الحمولات مثل الأدوية والوجبات ومتعلقات المرضى، إلخ.

المرحلة الأولى من العمل هي دراسة وبناء القاعدة المتنقلة. تتعلق المرحلة الثانية بالجانب المضمن الذي يتمثل في تطوير بنية الأجهزة وبنية البرامج، ودمج نظام الإدراك ونظام الملاحة. أخيرًا، المرحلة الثالثة مخصصة للاختبار والتحقق من الصحة.

الكلمات المفتاحية: روبوت متحرك، روبوت مساعد، ملاحة مستقلة، نظام تشغيل روبوت.

Abstract

The current trend in robotics is moving towards the development of assistant robots in the workplace in the service of health, agriculture, education or also in homes for personal use. However, among these areas, health use is gaining more ground, taking advantage of the health context. To do this, it is necessary to develop robotic platforms that can assist the medical staff and free them from certain daily tasks to allow them to devote themselves to the care of patients. The proposed project falls within this theme and aims to design and produce an assistant robot in the service of health. The robotic platform to be developed is of a differential robot, designed to transport loads such as medicines, meals, patients' belongings, etc. The first phase of the work is the study and construction of the mobile base. The second phase is related to the embedded aspect, which consists in developing a hardware architecture and a software architecture, integrating the perception and the navigation system. Finally, the third phase is dedicated to tests and validation.

Keywords: mobile robot, assistant robot, autonomous navigation, Robot Operating System.

Table des matières

1 CHAPITRE 1 : INTRODUCTION A LA ROBOTIQUE MOBILE 18

1.1 INTRODUCTION : 18
1.2 HISTORIQUE : 19
1.3 LES DIFFERENTS TYPES DES ROBOTS MOBILES : 22
1.3.1 LES ROBOTS MOBILES A ROUES : 22
1.3.2 LES ROBOTS MOBILES A PATTES : 22
1.4 LES DOMAINES D'APPLICATION DES ROBOTS MOBILES : 24
1.4.1 L'INDUSTRIE : 24
1.4.2 LE DOMAINE MILITAIRE : 24
1.4.3 LA SANTE : 25
1.4.4 UTILISATION CIVILE : 25
1.4.5 L'USAGE DOMESTIQUE : 26
1.5 LES ROBOTS DE TRANSPORT (PORTEUR DE CHARGE) : 26
1.5.1 LES ROBOTS DE TRANSPORT EN MILIEU EXTERIEUR : 26
1.6 CONCLUSION : 28

2 CHAPITRE 2 : ÉTUDE, CONCEPTION ET REALISATION DU ROBOT ADIUTOR 30

2.1 INTRODUCTION : 30
2.2 PARTIE ELECTRIQUE : 30
2.2.1 LA PARTIE COMMANDE : 31
2.2.2 LA PARTIE OPERATIVE : 33
2.2.3 BATTERIE LITHIUM : 38
2.3 PARTIE MECANIQUE : 39
2.3.1 LE CHASSIS : 39
2.3.2 LA COQUE : 42
2.4 CONCLUSION : 43

3 CHAPITRE 3 : NAVIGATION REACTIVE DES ROBOTS MOBILES 45

3.1 INTRODUCTION : 45
3.2 APPROCHES DE NAVIGATION REACTIVE POUR ROBOTS MOBILES : 45
3.2.1 LES ALGORITHMES BUG : 45
3.2.2 LA METHODE BASEE SUR LES CHAMPS DE POTENTIEL PF : 46
3.2.3 LA METHODE BASEE SUR LA FENETRE DYNAMIQUE DW : 46
3.2.4 LA METHODE BASEE SUR LA REPRESENTATION DES OBSTACLES DANS L'ESPACE DES VITESSES VO : 47
3.2.5 METHODE BASEE SUR L'ECHANTILLONNAGE DE L'ESPACE D'ENTREES : 48
3.3 DISCUSSION : 48
3.4 L'APPROCHE DE NAVIGATION REACTIVE PROPOSEE : 50
3.4.1 MODELE CINEMATIQUE DU ROBOT MOBILE : 50
3.4.2 REPRESENTATION DE L'ENVIRONNEMENT : 52
3.4.3 ÉCHANTILLONNAGE DE L'ESPACE LOCAL DU ROBOT : 54

3.4.4 APPROCHE D'ECHANTILLONNAGE DE L'ESPACE D'ENTREE : 55
3.5 CONCLUSION : 57

4 CHAPITRE 4 : IMPLEMENTATION, TEST ET RESULTATS....59

4.1 INTRODUCTION : 59
4.2 SYSTEME D'EXPLOITATION DES ROBOTS « ROS » : 59
4.2.1 LES NOTIONS DE BASE DE ROS : 59
4.2.2 LES OUTILS TRES UTILES DANS ROS : 60
4.3 ARCHITECTURE LOGICIELLE PROPOSEE POUR LE ROBOT MOBILE : 61
4.4 VALIDATION DE LA PLATEFORME ROBOTIQUE : TESTS ET RESULTATS....63
4.4.1 ROBOT MOBILE DEVELOPPE : 63
4.4.2 ASSERVISSEMENT DE LA VITESSE DES MOTEUR (PID) 65
4.4.3 VALIDATION DE LA PARTIE INTELLIGENCE : TESTS ET RESULTATS....67
4.4.4 EVALUATION DU TEMPS DE CALCUL DE L'ALGORITHME ISS 70
4.5 CONCLUSION : 70

Liste des acronymes et abréviations

PWM Pulse Wilde Modulation.

PID Proportional Integral and Derivative.

s Seconde

V volt.

ROS systeme d’exploitation robotique

DC courant continu.

AC courant alternatif

RAM la mémoire vive.

ROM la mémoire de programme.

PF champ de potentielle

DW la fenêtre dynamique.

VO velocity obstacles.

ISS Echantillonnage de l’espace d’entrées.

LIDAR détection et estimation de la distance par la lumière

Liste des figures

FIGURE 1:1 : INTRODUCTION A LA ROBOTIQUE MOBILE [3]. 19
FIGURE 1:2 : LA TORTUE DE GREY WALTER - UNE ILLUSTRATION DE SA TRAJECTOIRE POUR REJOINDRE SA NICHE [3]. 19
FIGURE 1:3 : A GAUCHE : ROBOT "BEAST" DE L'UNIVERSITE JOHN HOPKINS DANS LES ANNEES 1960. A DROITE : LE ROBOT SHAKEY DE STANFORD EN 1970 20
FIGURE 1:4 : LE STANFORD CART DATE DE LA FIN DES ANNEES 1970. 21
FIGURE 1:5 : GENGHIS, DEVELOPPÉ PAR RODNEY BROOKS AU MIT AU DEBUT DES ANNEES 1990 [3]. 21
FIGURE 1:6: LE ROBOT MIR200, POUR L'AUTOMATISATION DE LA LOGISTIQUE ET LE TRANSPORT EN MILIEU INDUSTRIEL. 24
FIGURE 1:7: UN "QUADE" AUTONOME TOUT TERRAIN 24
FIGURE 1:8 : ROBEAR, LE ROBOT INFIRMIER. 25
FIGURE 1:9 : ROBOT DE DESINFECTION DANS LES LIEUX PUBLICS. 25
FIGURE 1:10 : ROBOT ASPIRATEUR. 26
FIGURE 1:11 : DRONE DE TRANSPORT. 26
FIGURE 1:12 : ROBOT DE TRANSPORT DANS UN MILIEU D'EXTERIEUR 27
FIGURE 1:13 : ROBOT DE TRANSPORT L'INTERIEUR DE L'HOPITAL 28
FIGURE 2:1 : LA STRUCTURE GLOBALE DE LA PARTIE ELECTRONIQUE DU ROBOT. 30
FIGURE 2:2 : LE SCHEMA SYNOPTIQUE DE LA PARTIE ELECTRONIQUE DU ROBOT 31
FIGURE 2:3 : EXEMPLES DES DIFFERENTS MICROCONTROLEURS AVR, ARM… .32
FIGURE 2:4 : LA CARTE ARDUINO DUE. 33
FIGURE 2:5 : LA COMPOSITION D'UN MOTEUR BRUSHLESS. 34
FIGURE 2:6 : PRINCIPE DE FONCTIONNEMENT D'UN MOTEUR BRUSHLESS ETAPE PAR ETAPE. 35
FIGURE 2:7 : JYQD-V7.3E3 BRUSHLESS DC MOTEUR DRIVER [8]. 35
FIGURE 2:8 : SLAMTEC RPLIDAR A2 AVEC CARTE DE REDUCTION MICRO USB. 36
FIGURE 2:9 : CONNECTEUR XH2.54-5P ET LA CARTE DE REDUCTION MICRO USB. 37
FIGURE 2:10 : L'ENCODEUR ROTATIF INCREMENTAL OMRON E6B2-CWZ6C. 38
FIGURE 2:11 : UNE BATTERIE LITHIUM 36 V. 39
FIGURE 2:12 : LA CONCEPTION DE CHASSIS SOUS "SOLIDWORKS". 39
FIGURE 2:13 : LE CHASSIS REALISE. 40
FIGURE 2:14 : LES ROUES UTILISEES : A GAUCHE LA ROUE FOLLE, ET A DROITE LE CHASSIS AVEC LA ROUE MOTORISEE. 40
FIGURE 2:15 : LA CONCEPTION DU SUPPORT ENCODEUR SOUS « SOLIDWORKS » 41
FIGURE 2:16 : LE PLACEMENT FINAL DE L'ENCODEUR SUR LE CHASSIS. 41
FIGURE 2:17 : LA CONCEPTION DE SUPPORT ENCODEUR ET ROUE CODEUSE SOUS « SOLIDWORKS » 42
FIGURE 2:18 : L'EMPLACEMENT FINALE DES ENCODEURS LIES AUX ROUES CODEUSES SUR LE CHASSIS 42
FIGURE 2:19 : LA CONCEPTION DE COQUE SOUS "SOLIDWORKS" 43
FIGURE 3:1 : CALCUL DU CHEMIN DU ROBOT PAR LES ALGORITHMES BUG [12]. 45

FIGURE 3:2 : CHAMPS DE POTENTIEL POUR UN ENVIRONNEMENT CONTENANT TROIS OBSTACLES [14]. 46
FIGURE 3:3 : L'APPROCHE DE LA FENETRE DYNAMIQUE. L'ESPACE DES VITESSES EST DIVISE EN REGIONS ADMISSIBLES ET INTERDITES. DW EST REPRESENTEE PAR LE RECTANGLE BLEU CONTENANT LES VITESSES ATTEIGNABLES PAR LE ROBOT DURANT UN INTERVALLE DE TEMPS SPECIFIQUE [12]. 47
FIGURE 3:4 : L'APPROCHE VO : VOA/B POUR A INDUIT PAR B [14]. 47
FIGURE 3:5 : LA TECHNIQUE PAR ECHANTILLONNAGE DE L'ESPACE D'ENTREES. UN ENSEMBLE DE TRAJECTOIRES GENERE PAR L'ECHANTILLONNAGE DE L'ESPACE D'ENTREES, CES TRAJECTOIRES SONT DE COURBURE ET VITESSE LINEAIRE CONSTANTE [12]. 48
FIGURE 3:6 : REPRESENTATION DU CIR D'UN ROBOT MOBILE DIFFERENTIEL...51
FIGURE 3:7 : REPRESENTATION DES MESURES LASER 52
FIGURE 3:8 : UN POINT P DE L'ENVIRONNEMENT AGRANDI PAR LE RAYON DU ROBOT R. 53
FIGURE 3:9 : LE DEPLACEMENT DE LA CHAISE DANS UN PLAN 2D ENCOMBRE D'OBSTACLES STATIQUES. (A) L'ESPACE DE TRAVAIL, (B) L'ESPACE DE CONFIGURATION. 54
FIGURE 3:10 : REPARTITION DES ZONES ECHANTILLONNEES (TROIS ZONES). ..54
FIGURE 3:11 : PRINCIPE DE FONCTIONNEMENT DE L'APPROCHE ISS....55
FIGURE 4:1 : NOTION DE BASE ROS. 59
FIGURE 4:2 : ARCHITECTURE LOGICIELLE PROPOSEE. 62
FIGURE 4:3 : GRAPHE DE L'ENSEMBLE DES NŒUDS ET TOPICS DU SYSTEME....63
FIGURE 4:4 : LE SCHEMA DE CIRCUIT ELECTRIQUE DU ROBOT. 64
FIGURE 4:5 : LE CIRCUIT ELECTRIQUE DU ROBOT (PHOTO REEL). 64
FIGURE 4:6 : LE ROBOT SANS HABILLAGE. 65
FIGURE 4:7: SCHEMA FONCTIONNEL D'UN REGULATEUR PID [24]. 66
FIGURE 4:8 : LES ETAPES NECESSAIRES DE L'IMPLEMENTATION 68
FIGURE 4:9 : ILLUSTRATION DES DIFFERENTS TESTS POUR DIFFERENTS POSITIONS DES OBSTACLES. 69
FIGURE 4:10 : LES RESULTATS DE L'AFFICHAGE SUR ROS POUR LES QUATRE CAS DECRITS DANS LA FIGURE 4.9. 69

Liste des tableaux

TABLEAU 2.1 : LES CARACTERISTIQUES TECHNIQUES DE LA CARTE ARDUINO DUE [6]. 33
TABLEAU 2.2 : LES CARACTERISTIQUES DU DRIVER JYQD-V7.3E3. 35
TABLEAU 2.3 : RPLIDAR A2M8 CARCTRESTIQUE[9]. 37
TABLEAU 3.1 : COMPARAISON DES METHODES REACTIVES. 49
TABLEAU 4.1 : LES VITESSES DU MOTEUR EN MARCHE AVANT ET ARRIERE. 66
TABLEAU 4.2 : TABLEAU REPRESENTANT LE TEMPS DE CALCUL DE L'ALGORITHME PAR RAPPORT AUX NOMBRES DE ZONES EN FIXANT NOMBRE DE CONTROLE = 60. 70

Introduction générale

Introduction générale

La robotique est un domaine en plein essor ces dernières années. Les développements technologiques dépassent sans cesse nos attentes et permettent de créer des solutions technologiques qui s'adaptent à divers problèmes. Ainsi, la robotique est utilisée dans des domaines extrêmement rigoureux et exigeants, comme l'industrie, le domaine militaire, le milieu domestique, mais aussi dans le domaine qui est au centre de notre sujet, à savoir le secteur médical.

Le domaine médical devrait être le premier bénéficiaire de cette avancée, car elle présente plusieurs aspects, notamment l'aide aux personnes âgées ou handicapées, et toute personne ayant des difficultés à se déplacer librement dans la société. Elle peut être source d'une meilleure efficacité dans les soins, d'une meilleure prise en charge post-opératoire, ainsi que d'une aide aux soignants dans certaines tâches laborieuses.

Notre étude consiste à développer un robot autonome capable de porter des charges, pour aider et assister le staff médical en les déchargeant des tâches supplémentaires afin qu'il se concentre sur les soins uniquement. Par exemple, assister les personnes ayant besoin de transporter leurs affaires, quel que soit leur état de santé ou le degré de leur handicap, un autre exemple est de livrer la nourriture aux patients, transporter les prises de son au laboratoire, etc.

Pour réussir à développer un tel robot avec un impact aussi important, une étude approfondie a dû être réalisée pour obtenir le meilleur résultat possible. Ce travail comprend deux parties importantes.la première consiste à développer un robot mobile sur le plan matériel (mécanique et électronique) et logiciel. Pour l'architecture logicielle, elle est supervisée par le système ROS (Robot Operating System), un outil très répondu parmi la communauté robotique internationale permettant une intégration facile des algorithmes d'intelligence et la communication entre les différents modules de l'architecture. La deuxième partie revient à doter le robot d'intelligence pour lui permettre de naviguer de manière autonome. Pour ce faire, une approche d'évitement d'obstacle a été implémentée.

Afin d'atteindre ces objectifs, nous avons dû passer par plusieurs étapes, structurées selon les chapitres suivants :

Le premier chapitre expose l'idée de la robotique en général, et mentionne les différents types de robots qui existent et les différents domaines d'application et spécialement le domaine

de santé.

Le deuxième chapitre comprend la conception et la réalisation de l'architecture matérielle de notre robot mobile. Toutes les étapes de la construction sont énumérées en détail, en citant le matériel utilisé.

Dans le troisième chapitre, différentes méthodes de la navigation réactive des robots mobiles ont été étudiées pour l'évitement d'obstacle. Suite à une étude comparative, la méthode la plus appropriée pour notre application a été choisie.

Dans le dernier chapitre, nous présentons la partie software, l'implémentation, les tests et les résultats.

Nous terminons notre mémoire par une conclusion générale incluant un résumé du travail réalisé et quelques perspectives de développements futurs.

Chapitre 1 :

Introduction

à la robotique mobile

1 Chapitre 1 : Introduction à la robotique mobile

1.1 Introduction :

La robotique est un ensemble des disciplines (mécanique, électronique, automatique, informatique), elle se subdivise en deux types : les robots industriels et les robots mobiles. Les robots industriels sont généralement fixes, ils sont utilisés dans des nombreuses applications industrielles : l'assemblage mécanique, la soudure, la peinture, etc. Les robots mobiles sont classifiés selon le type de locomotion (à roues, à chenille, etc.) et le domaine d'application (déminage, surveillance, spatiale, nucléaire, etc.) [1].

Concernant la robotique de service et en particulier d'intervention, son rôle consiste à intervenir là où l'homme ne peut ou ne veut pas intervenir lui-même. Les robots utilisés sont dotés d'outils nécessaires à l'intervention (capteurs, bras de robot, canon à eau, effecteur, fusil, etc.) et doivent pouvoir se déplacer pour atteindre le lieu d'intervention et l'explorer à distance via un dispositif de contrôle fiable. Ce dernier permet de traiter plusieurs fonctions d'où l'accomplissement d'un certain nombre de tâches (parfois non connues à l'avance, et susceptibles d'évoluer dans le temps) sous contrôle d'un opérateur (artificier) humain réduit [2], situé hors de la zone hostile ou derrière des protections.

Dans ce chapitre, nous présentons dans un cadre général la robotique de service et en particulier celle d'interventions dont nous citons quelques exemples de robots utilisés, puis nous étalerons la problématique en question qui est relative à la commande à distance de robot d'intervention, puis nous passons vers l'étude critique de l'existant et nous parachevons par une conclusion.

Introduction aux robots mobiles et manipulateurs :

La robotique est un très bon exemple de domaine pluridisciplinaire qui implique de nombreuses thématiques telles que la mécanique, la mécatronique, l'électronique, l'automatique, l'informatique ou l'intelligence artificielle. En fonction du domaine d'origine des auteurs, il existe donc diverses définitions du terme robot, mais elles tournent en général autour de celle-ci : Un robot est une machine équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a.

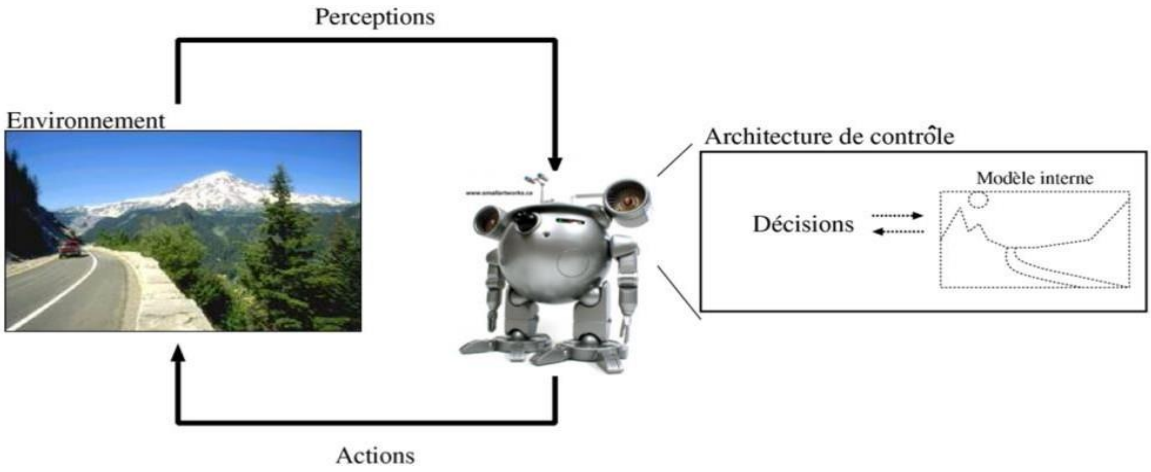

Figure 1:1 : Introduction à la robotique mobile [3].

Cette définition s'illustre par un schéma classique des interactions d'un robot avec son environnement (Figure 1.1). Les différentes notions que nous présenterons dans ce chapitre sont essentiellement issues de cette vision de la robotique, très orientée sur l'Intelligence Artificielle, qui place au centre des préoccupations l'enchaînement de ce cycle Perception/Décision/Action. La manière dont un robot gère ces différents éléments est définie par son architecture de contrôle, qui la plupart du temps va faire appel à un modèle interne de l'environnement qui lui permettra de planifier ses actions à long terme [3].

1.2 Historique :

Le terme de robot apparaît pour la première fois dans une pièce de Karel Capek en 1920: Rossum's Universal Robots. Il vient du tchèque 'robota' (servitude) et présente une vision des robots comme serviteurs dociles et efficaces pour réaliser des taches pénibles mais qui déjà vont se rebeller contre leurs créateurs [3].

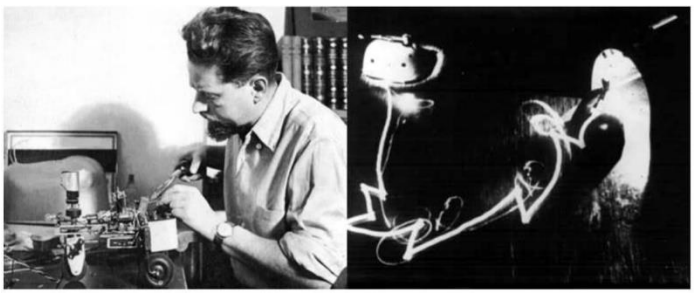

Figure 1:2 : La tortue de Grey Walter - une illustration de sa trajectoire pour rejoindre sa niche [3].

La Tortue construite par Grey Walter dans les années 1950 (Figure 1.2), est l'un des premiers robots mobiles autonomes. Grey Walter n'utilise que quelques composants

analogiques, dont des tubes à vide, mais son robot est capable de se diriger vers une lumière qui marque un but, de s'arrêter face à des obstacles et de recharger ses batteries lorsqu'il arrive dans sa niche. Toutes ces fonctions sont réalisées dans un environnement entièrement préparé, mais restent des fonctions de base qui sont toujours des sujets de recherche et de développement technologiques pour les rendre de plus en plus génériques et robustes.

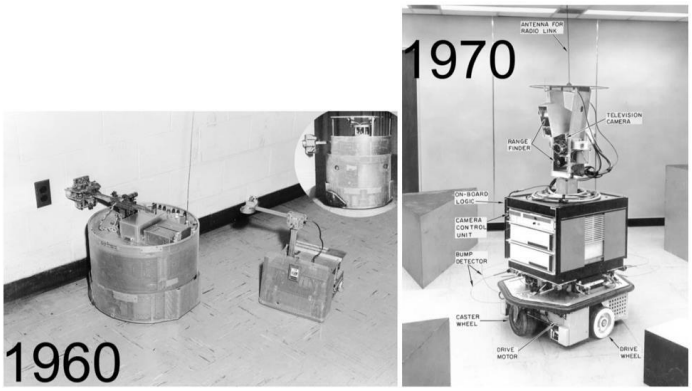

Figure 1:3 : A gauche : Robot "Beast" de l'université John Hopkins dans les années 1960. A droite : Le robot Shakey de Stanford en 1970

Dans les années 60, les recherches en électronique vont conduire, avec l'apparition du transistor, à des robots plus complexes mais qui vont réaliser des tâches similaires. Ainsi le robot "Beast" (Figure 1.3) de l'université John Hopkins est capable de se déplacer au centre des couloirs en utilisant des capteurs ultrason, de chercher des prises électriques (noires sur des murs blancs) en utilisant des photodiodes et de s'y recharger.

Les premiers liens entre la recherche en intelligence artificielle et la robotique apparaissent à Stanford en 1970 avec Shakey (Figure 1.3). Ce robot utilise des télémètres à ultrason et une caméra et sert de plate-forme pour la recherche en intelligence artificielle, qui à l'époque travaille essentiellement sur des approches symboliques de la planification.

La perception de l'environnement, qui à l'époque est considérée comme un problème séparé, voire secondaire, se révèle particulièrement complexe et conduit là aussi à de fortes contraintes sur l'environnement. Ces développements se poursuivent avec le Stanford Cart dans la fin des années 1970, avec notamment les premières utilisations de la stéréovision pour la détection d'obstacles et la modélisation de l'environnement. En France, le robot Hilare est le premier robot construit au LAAS, à Toulouse (Figure 1.4) [3].

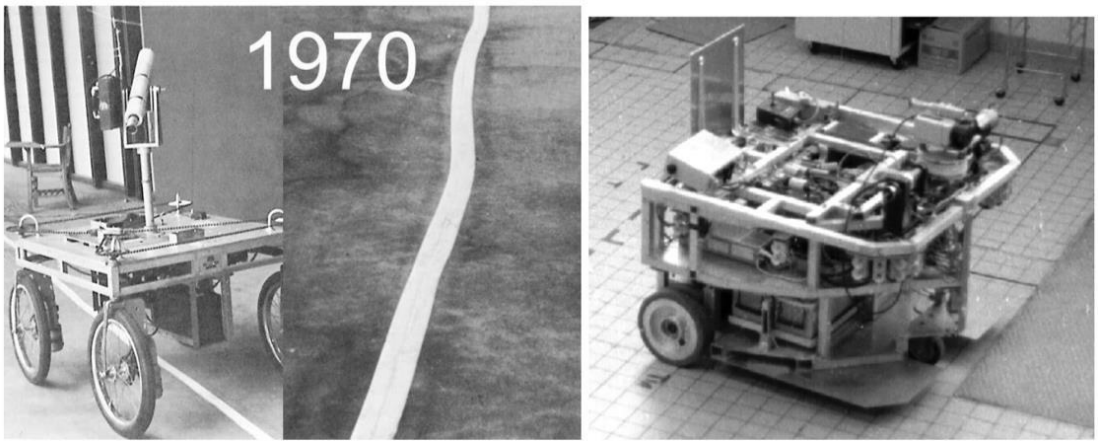

Figure 1:4 : Le Stanford Cart date de la fin des années 1970.

Le robot Hilare du LAAS a été construit en 1977 [3].

Une étape importante est à signaler au début des années 1990 avec la mise en avant de la robotique réactive, représentée notamment par Rodney Brooks. Cette nouvelle approche de la robotique, qui met la perception au centre de la problématique, a permis de passer de gros robots très lents à de petits robots (Figure 1.5), beaucoup plus Réactifs et adaptés à leur environnement. Ces robots n'utilisent pas ou peu de modélisation du monde, problématique qui s'est avérée être extrêmement complexe.

Figure 1:5 : Genghis, développé par Rodney Brooks au MIT au début des années 1990 [3].

Ces développements ont continué, et l'arrivée sur le marché depuis les années 1990 de plates- formes intégrées telles que le pioneer de la société Mobile Robots a permis à de très nombreux laboratoires de travailler sur la robotique mobile et a conduit à une explosion de la diversité des thèmes de recherche. Ainsi, même si les problèmes de déplacement dans l'espace et de modélisation de l'environnement retent difficiles et cruciaux, des laboratoires ont pu par exemple travailler sur des approches multirobots, la problématique de l'apprentissage ou sur les problèmes d'interactions entre les hommes et les robots.

1.3 Les différents types des robots mobiles :

Il est possible de classifier les robots mobiles de nombreuses manières et selon de nombreux critères tels : le domaine d'application, le degré d'autonomie, le degré d'intelligence, etc. Nous avons préféré les classifier, puisque leur mobilité est l'une de leurs caractéristiques essentielles, selon leur mode de locomotion [3].

On peut classer les robots mobiles en cinq catégories selon leur mode de locomotion:

- Les robots mobiles à roues.
- Les robots mobiles à pattes.
- Les robots mobiles à chenilles.
- Les robots mobiles volants (drones).

1.3.1 Les robots mobiles à roues :

Ils sont les plus répandus car les plus simples à concevoir. Il existe deux grands types de robots mobiles à roues selon leur mode de direction :

- Les robots mobiles à roues différentielles (differential drive).
- Les robots mobiles à roues avec essieux classique (steer drive).
- Les robots mobiles à roues avec essieux (steer drive configuration) : Ils sont pour la plupart utilisés dans des applications éducatives car faciles à mettre en œuvre à base de châssis de modèles réduits automobiles.

Le mouvement arrière du robot est assuré par deux roues motrices. La direction quant à elle est calquée sur le modèle automobile.

La commande de ce type de robots est assez simple mais leur capacité en termes de mouvement est réduite. En effet, leur comportement dit « non holonome » les contraint à des manœuvres compliquée (créneau) et limite considérablement leur angle de braquage. C'est pour cette raison qu'ils sont très peu utilisés.

1.3.2 Les robots mobiles à pattes :

On distingue notamment et selon le nombre de pattes :

Les robots à une patte : Un robot doté d'une seule patte a été assemblé au Massachusetts Institute of Technology (MIT). Le robot sautille sur une patte orientable à deux degrés de liberté, dotée d'un actionneur linéaire pour la propulsion verticale. Il n'est pas autonome.

Les robots à deux pattes : Les robots à deux pattes ou bipèdes sont surtout étudiés au Japon ou les progrès dans ce segment sont assez signifiants. En 1997, Honda et son humanoïde P2, font rêver le monde entier de robots domestiques.

Les robots à quatre pattes : Encore une fois, c'est au Japon qu'il faut chercher les robots quadrupèdes les plus évolués : le robot de compagnie AIBO (robot chien) de la firme SONY, notamment dans sa seconde génération est assez « vivant ». Il reproduit en effet une large palette de comportements « canins ».

Les robots à six ou huit pattes : Ce sont les robots à pattes les plus répandus à travers le monde. Avec leurs six pattes ils sont statiques mais aussi facile à programmer et mettre en œuvre, chaque patte n'ayant besoin que de deux degrés de liberté.

Les robots « mille-pattes » : Ces robots sont très utilisés dans l'entretien et la réparation des conduites et autres endroits inaccessibles. La demande professionnelle pour ce type de robots est très importante.

Robots mobiles à chenille : Les chenilles assurent à un mobile une meilleure adhérence au sol. Elles sont employées lorsque le sol est perturbé, essentiellement en environnement extérieur. La commande est de type différentiel (differential drive). On peut citer le robot ANT de la West Virginia University.

Robots volants (drones) : Les drones, petits avions autoguidés ou téléguidés ont de nombreuses applications militaires et éventuellement civiles (surveillance d'autoroutes). Des projets visent à réaliser un avion autonome de moins de 20cm d'envergure, transportant une caméra infrarouge et un transmetteur radio. Il est plus facile de faire voler un avion de 50cm d'envergure ne pesant que quelques grammes (show flyer) ; ces avions indoor sont délicats à construire et fragiles à manipuler mais ils résistent aux obstacles, à cause de leur faible inertie. Le vol stationnaire est encore plus facile à maitriser. Le concours de robots volants organisé chaque année à Atlanta en Georgie montre combien il est difficile de stabiliser en vol un hélicoptère et de commander ses déplacements avec assez de précision.

Robots navigants : La construction d'un robot bateau ou sous-marin est facilitée par le fait que le poids n'est pas un facteur limitatif à la mobilité. Les torpilles sont bien connues des militaires, mais il y a peu de recherches civiles qui ont été lancées sur les robots mobiles sous- marins appelés AUV (Autonomous Underwater Vehicle). Ces derniers sont notamment utilisés dans le secteur pétrolier (off-shore) et dans l'exploration sous-marine.

1.4 Les domaines d'application des robots mobiles :

1.4.1 L'industrie :

La robotique industrielle permet l'automatisation de certaines tâches dans la chaîne de production et apporte flexibilité et gain de productivité (Figure 1.6).

Figure 1:6: Le robot MIR200, pour l'automatisation de la logistique et le transport en milieu industriel.

1.4.2 Le domaine militaire :

Actuellement, plusieurs types de robots sont utilisés dans le domaine militaire. Ils peuvent agir dans des missions aussi diverses que le renseignement, les opérations armées, la reconnaissance, le déminage ou encore le transport de matériel. Un exemple de robot de reconnaissance est illustré dans la figure (1.7).

Figure 1:7: Un "Quade" autonome tout terrain

1.4.3 La santé :

Les applications des robots dans le domaine de la santé sont innombrables, chirurgie de pointe, rééducation, prise en charge du handicapé, soutien aux personnes dépendantes, etc. Des gestes simples et une intervention rapide tout en préservant la sécurité des patients et des médecins sont des priorités. Pour ce faire, les roboticiens ont développés plusieurs systèmes, notamment dans le domaine de la robotique mobile, qui sont déjà déployés dans le secteur médical tel que les robots infirmier (Figure 1.8).

Figure 1:8 : Robear, le robot infirmier.

1.4.4 Utilisation civile :

De plus en plus de tâches sont confiées aux robots dans le civil. Ils servent à remplacer les personnes qui sont chargées de tâches civiles (Nettoyer la ville, aider la population, s'occuper des lieux publics, etc.), comme l'exemple présenté dans la figure (1.9).

Figure 1:9 : Robot de désinfection dans les lieux publics.

1.4.5 L'usage domestique :

Les robots domestiques permettent améliorer le quotidien, tels que le robot aspirateur, qui permet le nettoyage autonome de la maison, ou la tondeuse à gazon automatique. Ces robots domestiques ouvrent la voie à de nombreuses autres applications qui permettent de faciliter d'avantage la vie de l'être humain. Nous pouvons voir dans la figure (1.10) un robot domestique de type robot aspirateur.

Figure 1:10 : Robot aspirateur.

1.5 Les robots de transport (porteur de charge) :

C'est un robot conçu pour le transport des palettes et charges lourdes, ou encombrantes.

1.5.1 Les robots de transport en milieu extérieur :

Nous pouvons distinguer principalement plusieur types de robot.

1.5.1.1 Les drones de livraison :

Le drone est un aéronef sans passager ni pilote qui peut voler de façon autonome ou être contrôlé à distance depuis le sol. Un exemple de ces systèmes est présenté dans la figure (1.11). La livraison par drone permet de gagner du temps et de réaliser des opérations à haut risque plus rapidement. En effet, la livraison par l'air est plus rapide que par la route.

Figure 1:11 : Drone de transport.

1.5.1.2 Les robots de livraison par route :

Ce sont des robots mobiles qui peuvent parcourir des longs trajets. Ils font leurs déplacements par un contrôle à distance ou autocontrôle, spécifiquement conçu pour la livraison à domicile,comme illustré dans la figure (1.12)

Figure 1:12 : robot de transport dans un milieu d'extérieur

1.5.1.3 Les robots de transport (robots porteurs de charge) en milieu intérieur :

Ce sont des robots mobiles capables de porter des charges et souvent menés à les transporter vers un endroit bien défini. L'exemple de la figure (1.13) montre un tel robot dans un milieu hospitalier. Dans ce genre d'application, les robots peuvent accomplir leur tâche de manière autonome ou téléguidée. Pour le cas autonome, le déplacement du robot est indépendant, ne nécessitant aucune intervention humaine. Pour ce faire, il suffit de définir la destination à atteindre. Pour le cas téléguidé, le robot a besoin d'une personne pour le guider. Le guidage peut être fait de plusieurs façons telles que la commande par joystick, par télé opération ou un système de reconnaissance pour suivre les personnes.

Figure 1:13 : Robot de transport l'intérieur de l'hôpital

1.6 Conclusion :

Dans ce chapitre, L'objectif de notre étude consistait principalement à l'étude et nous avons donné une idée générale sur la robotique, l'historique des robots et leurs domaines d'applications, et une généralité sur les différents robots mobiles en se focalisant sur les différents types de robots mobiles. Parmi les applications présentées, le domaine visé dans ce travail est la santé, et plus particulièrement, les robots de transport (robots porteurs de charge) dans des milieux hospitaliers.

Ainsi que nous avons défini et énoncé le robot qu'il est le cas d'application de notre étude.

Chapitre 2 :

Étude, conception et réalisation du robot ADIUTOR

2 Chapitre 2 : Étude, conception et réalisation du robot ADIUTOR

2.1 Introduction :

Rappelons l'objectif de notre travail qui vise à développer un robot mobile de transport pour des applications de santé. Pour ce faire, nous présentons dans ce chapitre les différentes phases de conception et de réalisation du robot mobile ADIUTOR développé dans ce projet de fin d'étude. Du point de vue hardware, ce dernier se compose principalement de deux parties : la partie électronique et la partie mécanique, qui vont être présentées dans ce chapitre. Le robot comporte également une architecture logicielle qui sera présenté dans le chapitre 4.

2.2 Partie électrique :

Elle se compose de deux parties distingues mais liées : une partie commande qui concerne le microcontrôleur et une partie opérative qui est constituée d'actionneurs et de capteurs [4].

La circulation des informations de la partie commande vers la partie opérative s'appelle la chaîne d'action. La circulation des informations de la partie opérative vers la partie commande s'appelle la chaîne d'acquisition. Le schéma de la figure (2.1) montre la structure globale de la partie électronique du robot.

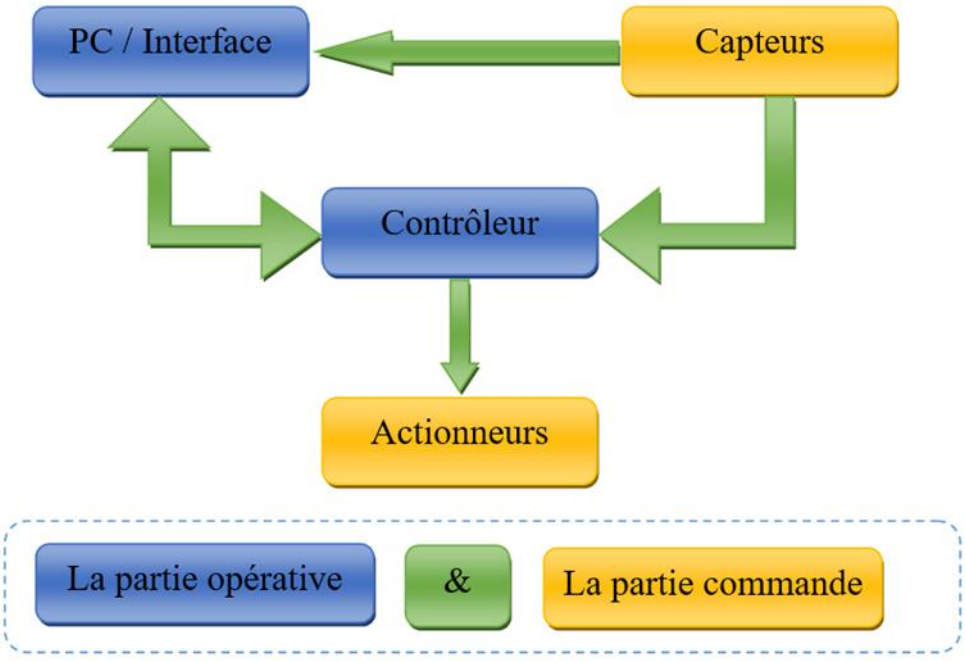

Figure 2:1 : La structure globale de la partie électronique du robot.

Les détails de la partie électronique que nous avons développée sont structurés autour

de l'architecture représentée dans la figure (2.2). Chaque partie de cette architecture possède une fonction précise que nous allons définir ci-dessous.

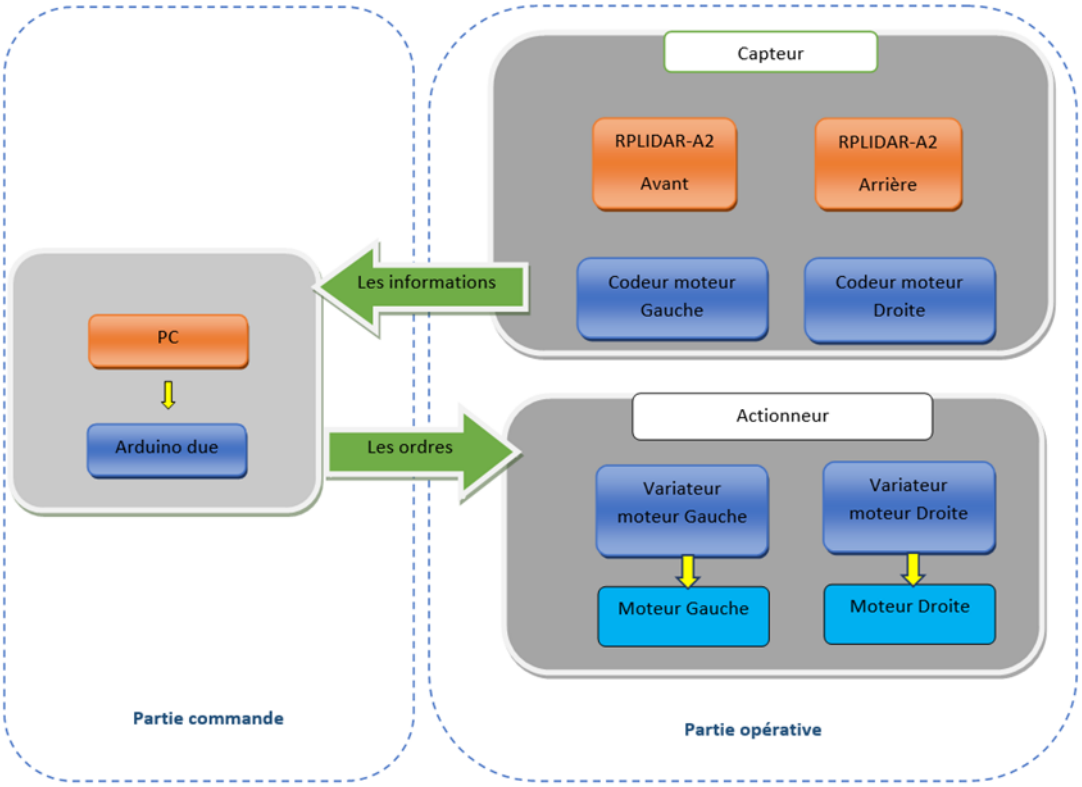

Figure 2:2 : Le schéma synoptique de la partie électronique du robot

2.2.1 La partie commande :

2.2.1.1 Les microcontrôleurs :

L'élément principal de cette partie, notamment celle responsable de la commande des moteurs est le microcontrôleur. Ce dernier est un système informatique contenu dans un seul circuit intégré. Un microcontrôleur se compose de quatre parties. Le microprocesseur assure le traitement des informations et la transmission des instructions. Il est lui-même constitué d'une UAL (unité arithmétique et logique) et d'un bus de données. C'est donc cette partie qui exécute les programmes embarqués dans le microcontrôleur [5].

2.2.1.2 Critères de choix des microcontrôleurs :

Il existe un très grand nombre de modèles de microcontrôleurs, et de nombreux fabricants proposant chacun plusieurs familles de microcontrôleurs, comptant chacune parfois des centaines de modèles.

Figure 2:3 : Exemples des différents microcontrôleurs AVR, ARM…

Généralement, les principaux critères de sélection des microcontrôleurs sont :

• Le nombre de pattes d'entrées-sorties.

• Taille de la mémoire de programme (ROM).

• Taille de la mémoire vive (RAM).

• Consommation électrique (tension de fonctionnement, courant consommé).

• La taille du bus d'adresses dépend du nombre de cellules mémoires.

• La taille du bus de données influence la capacité de traitement du système.

• Le prix.

• L'environnement de développement (matériel et logiciel, Le coût et la disponibilité).

• L'expérience.

2.2.1.3 Solution utilisée : Arduino due :

Dans ce travail, la solution utilisée est la carte Arduino Due (Figure 2.4) qui est une carte très puissant basée sur une architecture ARM 32bit, comparée aux autres gammes d'Arduino qui sont basées sur ATMega, la carte Arduino Due accueille un microcontrôleur Atmel SAM3X8E ARM Cortex-M3. Elle possède 54 entrées/sorties numériques dont 12 peuvent être utilisées comme sorties PWM, 12 entrées analogiques, 4 ports série UART, une horloge à 84 MHz, une connexion compatible USB OTG, 2 DAC (digital to analog), 2 TWI, une prise jack, un header SPI, un header JTAG et enfin un bouton de reset et un bouton d'effacement. La carte Due est utilisée pour gérer le niveau bas du robot, à savoir la commande des moteurs.

Figure 2:4 : La carte Arduino Due.

- **Caractéristiques techniques de la carte Arduino Due**

Les principales caractéristiques de la carte Arduino Due sont présentées dans le tableau (2.1).

Microcontrôleur	AT91SAM3X8E
Tension de fonctionnement	3,3 V
Tension d'alimentation	7- 12 V
Tension d'alimentation (limites)	6 - 20V
Nombre d'entrées/sorties	54 (dont 12 pouvant générer un signal PWM)
Nombre de ports "Analogique/Numérique"	12
Nombre de ports "Analogique"	2 (DAC)
Courant maximal par E/S	130 mA
Courant pour broches 3.3 V	800 mA
Courant pour broches 5 V	800 mA
Mémoire Flash	512 KB
SRAM	96 KB (2 banques 64 KB et 32 KB)
Vitesse d'horloge :	84 MHz

Tableau 2.1 : Les caractéristiques techniques de la carte Arduino Due [6].

2.2.2 La partie opérative :

Cette partie regroupe à son tour deux parties : la partie capteurs et la partie actionneurs, comme illustrée dans le schéma synoptique de la figure (2.2).

2.2.2.1 Les actionneurs :

Les actionneurs transforment l'énergie de puissance fournie au système en énergie physique nécessaire à la réalisation des tâches. Dans ce travail, c'est la motricité du robot qui nécessite les actionneurs, à savoir les moteurs Brushless.

1) Moteur Brushless :

Ce sont des moteurs sans balais synchrones alimentés en courant continu (DC) via un onduleur ou une alimentation à découpage qui produit de l'électricité sous forme de courant alternatif (AC) pour entraîner chaque phase du moteur via un contrôleur en boucle fermée. Le contrôleur fournit des impulsions de courant aux enroulements du moteur qui contrôlent la vitesse et le couple du moteur [7].

1.1) Principe de fonctionnement :

Ces moteurs ont le rotor et le stator inversés par rapport aux moteurs à courant continu, le stator étant constitué de bobines et le côté opposé d'aimants, ce qui permet de retirer les balais. (Figure 2.5), que l'on trouve dans les moteurs à courant continu. Son alimentation est triphasée et un capteur de position fonctionne par effet Hall. Le moteur a besoin de son contrôleur pour le faire tourner, contrairement au courant continu qui a deux pôles. Le moteur a besoin de son contrôleur pour tourner, contrairement au courant continu qui a deux pôles.

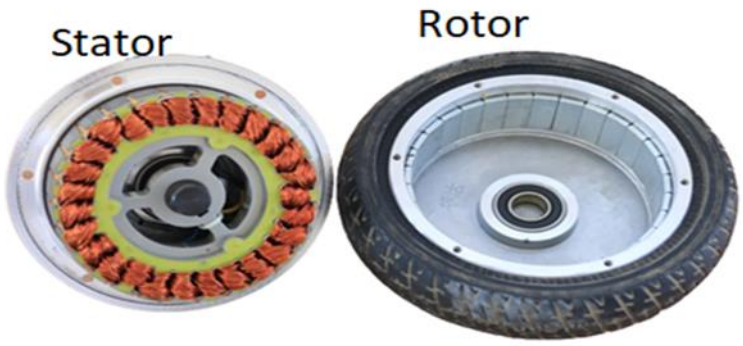

Figure 2:5 : La composition d’un moteur Brushless.

Le fonctionnement du moteur est basé sur l'attraction ou la répulsion entre les pôles magnétiques, dont le principe est illustré dans la figure (2.6). Le processus démarre lorsque le courant traverse l'un des trois enroulements du stator et génère un pôle magnétique qui attire l'aimant permanent le plus proche du pôle opposé. Le rotor se déplacera si le courant passe à un enroulement adjacent. La charge séquentielle de chaque enroulement fait suivre le rotor dans un champ tournant. Le couple dans cet exemple dépend de l'amplitude du courant, le nombre de tours sur les enroulements du stator, la force et la taille des aimants permanents, l'entrefer entre le rotor et les enroulements et la longueur du bras rotatif.

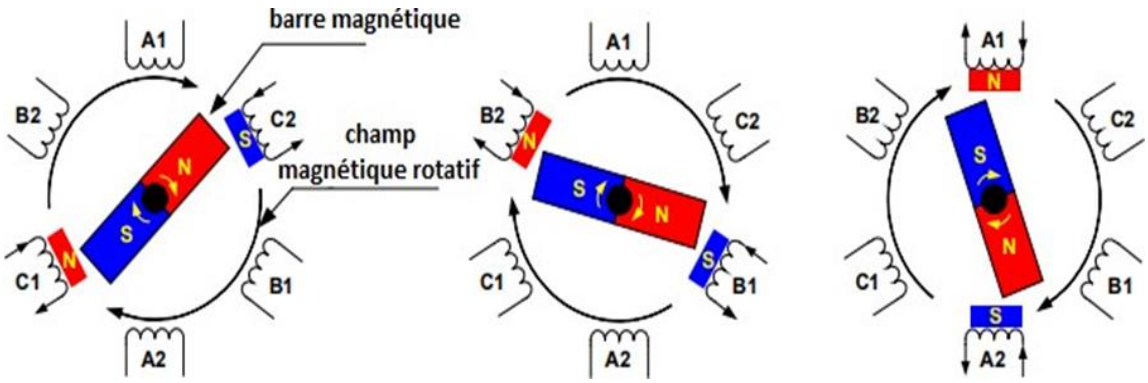

Figure 2:6 : Principe de fonctionnement d'un moteur Brushless étape par étape.

1.2) Driver moteur : JYQD-V7.3E3

Dans ce projet, le driver moteur utilisé est le JYQD-V7.3E3 (voir la figure 2.7). C'est le driver le plus utilisé dans les projets, et très facile à mettre en œuvre. Le module JYQD-V7.3E3 permet de commander un moteur brushless 12 à 36 V. Il fonctionne avec une interface de commande TTL 5V (donc compatible avec Arduino ou avec de nombreux autres microcontrôleurs en basse tension).

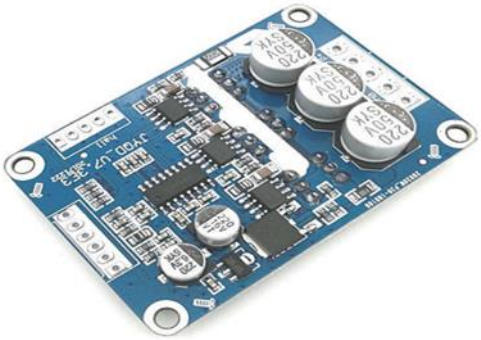

Figure 2:7 : JYQD-V7.3E3 brushless dc moteur driver [8].

Ce tableau nous montre les caractéristiques du driver JYQD-V7.3E3

Modèle	Température de fonctionnement	Voltage de fonctionnement	Courant maximale	Courant continu de fonctionnement	Vitesse de régulation du PWM (1-20KHz)	RÉGULATION DE LA VITESSE EN TENSION ANALOGIQUE
JYQDV7.3E3	-20-------85°C	12----36 V	16 A	15 A	CYCLE DE SERVICE 0----- 100%	0---5V

Tableau 2.2 : les caractéristiques du driver JYQD-V7.3E3.

2.2.2.2 La partie capteurs :

Un capteur est un transducteur capable de transformer une grandeur physique à une grandeur électrique (tension). En effet, nous allons présenter dans cette section les différents capteurs utilisés dans ce projet pour des fonctions variées.

1) Lidar:

L'acronyme LiDAR signifié (Light Detection And Ranging). Il s'agit d'une méthode de calcul qui permet de déterminer la distance entre le capteur et l'obstacle visé. Un Lidar utilise un faisceau laser pour la détection, l'analyse et le suivi. La technologie Lidar est une technologie de télédétection qui mesure la distance entre le capteur et une cible. La lumière est émise par le Lidar et se dirige vers sa cible. Elle est réfléchie sur sa surface et revient à sa source. Comme la vitesse de la lumière est une valeur constante, le Lidar est capable de calculer la distance le séparant de la cible [9].

Figure 2:8 : Slamtec RPLidar A2 avec carte de réduction micro USB.

RPLIDAR A2M8 360° Laser Scanner (Range: 12m)

Le scanner laser RPLIDAR A2 est la version améliorée du lidar 2D. Le système de mesure développé par SLAMTEC est adapté aux applications intérieures et extérieures. Le système peut effectuer jusqu'à 8000 mesures par seconde à grande vitesse.

Le RPLIDAR A2 est la nouvelle génération de scanners laser 2D à 360 degrés (LIDAR), développée par SLAMTEC. Il peut acquérir jusqu'à 8000 échantillons laser par seconde à une vitesse de rotation élevée. Équipé de la technologie brevetée OPTMAG de SLAMTEC, il dépasse les limites de vie des systèmes LIDAR conventionnels pour offrir un fonctionnement stable et de longue durée.

Le scanner laser 360° RPLIDAR A2 fonctionne à 600 tours/minute et permet également une résolution allant jusqu'à 0,9°. Parmis ses principales caracteristiques:

- Soutien total du ROS.
- Particulièrement silencieux grâce au moteur sans balais.
- Particulièrement durable 5 ans / 24h.

Le contrôle ou l'évaluation des données mesurées se fait via RS232 ou USB. La figure (2.9) montre le connecteur XH2.54-5P, qui sort du boîtier RPLIDAR A2. Un module pour la connectivité USB est également compris ce connecteur.

Mesure de la distance	Plage d'angle	Précision	Période d'orbite	Vitesse de circulation	Fréquence de balayage	Consommation électrique	Poids
0,15 - 12m	0-360 degrés	<1	0.25 ms	8000Hz	10Hz	1,5A et 5V	190 g

Tableau 2.3 : RPlidar a2m8 caractéristiques [9].

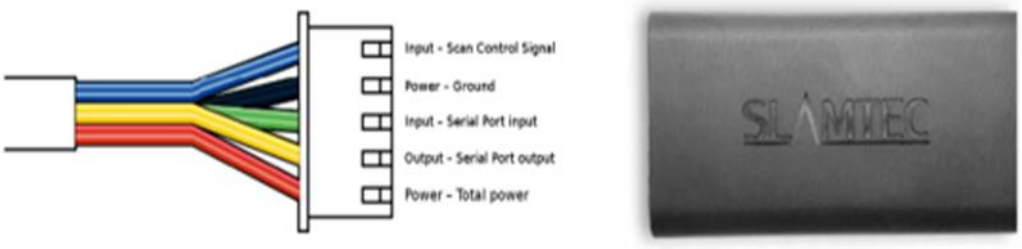

Figure 2:9 : connecteur XH2.54-5P et la carte de réduction micro USB.

Cette partie concerne le lidar, capable de détecter les obstacles présents dans l'environnement d'évolution du le robot. Les données du lidar seront utilisées dans la phase d'évitement d'obstacles.

2) Les codeurs :

Il existe de nombreux types d'encodeurs, mais ils se répartissent en deux grandes techniques de détection: linéaire et rotatif. Au sein de ces catégories, il existe différents types de mesure de codeurs : absolu et incrémental.

Un codeur rotatif collecte des données et fournit un retour d'information basé sur la rotation d'un objet ou, en d'autres termes, d'un dispositif rotatif. Les codeurs rotatifs sont parfois appelés "codeurs d'arbre". Ce type de codeur peut convertir la position angulaire ou le mouvement d'un objet en fonction de la rotation de l'arbre, selon le type de mesure utilisé.Les "codeurs rotatifs absolus" peuvent mesurer des positions "angulaires", tandis que Les "codeurs rotatifs incrémental" peuvent mesurer des éléments tels que la distance, la

vitesse et la position. Dans ce travail, nous avons utilisé un codeur rotatif incrémental : OMRON E6B2-CWZ6C (voir Figure 2.10).

Figure 2:10 : L'encodeur rotatif incrémental OMRON E6B2-CWZ6C.

2.2.3 Batterie lithium :

La source d'énergie utilisée pour alimenter le robot est une batterie lithium (Figure 2.11) apparue en 1991, ce type de batterie a ont une grande capacité de stockage dans un faible volume avec un faible poids. Il existe différentes variantes : le LCO (Lithium cobalt), est apparu ensuite le LMO (lithium manganèse) et le NMC (nickel manganèse cobalt) [10]. Les batteries lithiums ont été très rapidement utilisées pour toutes les applications nomades (téléphone portable, ordinateur portable, etc.) fortement appréciées pour leur grande capacité massique. Leur point faible est leur dangerosité, où elles n'acceptent pas de surcharge sous peine d'exploser. C'est pourquoi une gestion électronique est nécessaire pour éviter ce phénomène. Elles ne supportent pas non plus de chocs trop violents ou un percement qui entraine immédiatement un enflamment de la batterie.

Figure 2:11 : une batterie lithium 36 V.

2.3 Partie mécanique :

2.3.1 Le châssis :

Afin d'atteindre les objectifs visés dans ce travail, il est nécessaire de développer une plateforme mobile qui peut d'une part se mouvoir facilement dans des milieux hospitaliers (présence de couloirs étroits, petites salles, etc.) et d'autre part supporter des grandes charges.

Pour ce faire, la conception du châssis a été effectuée en utilisant de logiciel "SolidWorks" telle que présentée dans la figure (2.12). Il a une forme ovale pour faciliter sa motricité dans les endroits étroits. Les dimensions du châssis sont comme suit :

- La longueur 560 mm.
- La largeur 360 mm.
- La hauteur 450 mm.

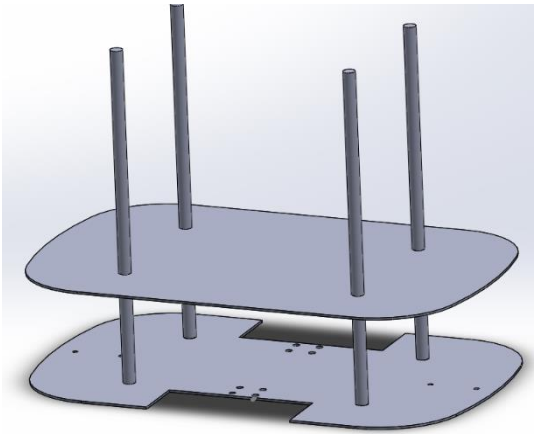

Figure 2:12 : La conception de châssis sous "SolidWorks".

Il a été réalisé à partir d'une plaque en aluminium de 5 mm coupés avec une découpeuse jet d'eau, comme illustré dans la figure (2.13).

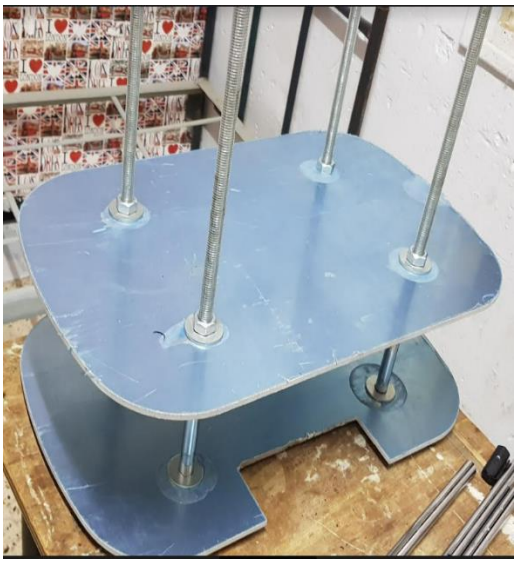

Figure 2:13 : Le châssis réalisé.

Les roues motorisées sont fixées sur le châssis selon le modèle différentiel , en plus de deux roues folles pour assurer la stabilité (voir la Figure 2.14).

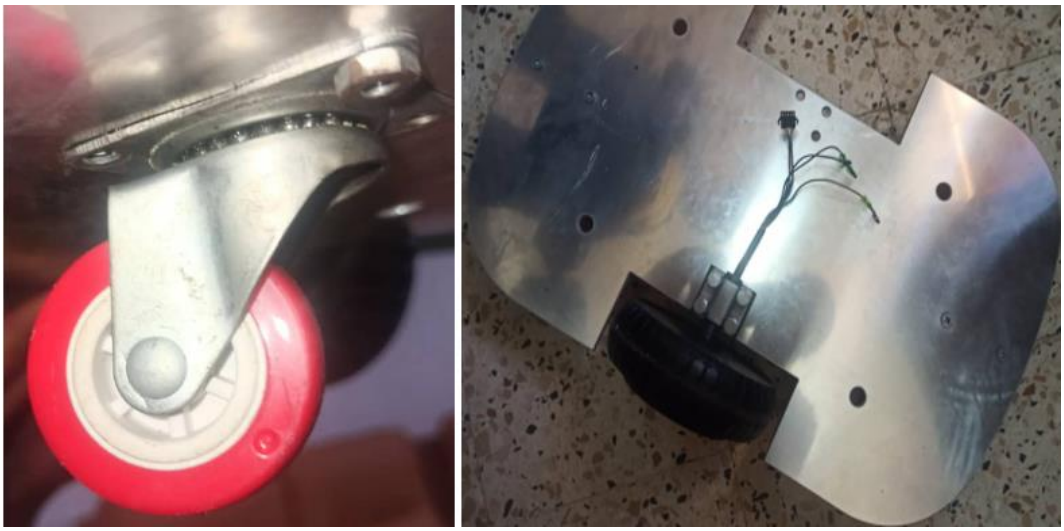

Figure 2:14 : Les roues utilisées : à gauche la roue folle, et à droite le châssis avec la roue motorisée.

Une fois les roues motorisées fixées, On a placé les encodeurs avec deux différentes méthodes :

1- Encodeurs placés en liaison avec les moteurs.

2- Utilisation des roues codeuses.

Pour la 1ère méthode les deux encodeurs sont placés sur un support que nous avons conçu avec « SolidWorks » (voir Figure 2.15).

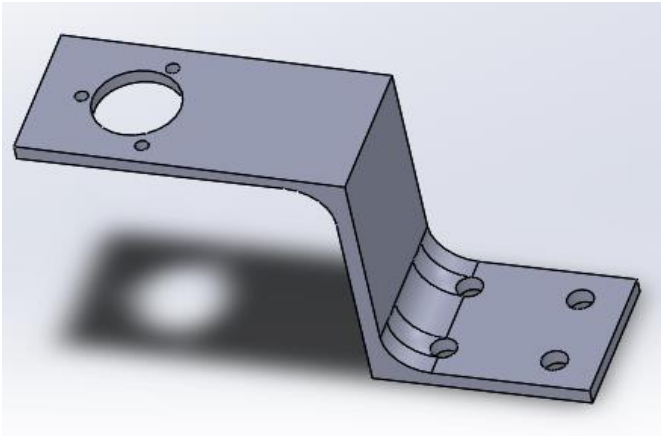

Figure 2:15 : la conception du support encodeur sous « SolidWorks »

Les encodeurs sont reliés à ces dernières via un engrenage conique à denture droite avec arbres concourants perpendiculaires (voir Figure 2.16).

Figure 2:16 : Le placement final de l'encodeur sur le châssis.

Pour la 2ère méthode les deux encodeurs sont placés sur un support lié à des roues codeuses directement en contact avec le sol, que nous avons conçu avec « SolidWorks » (voir Figure 2.17).

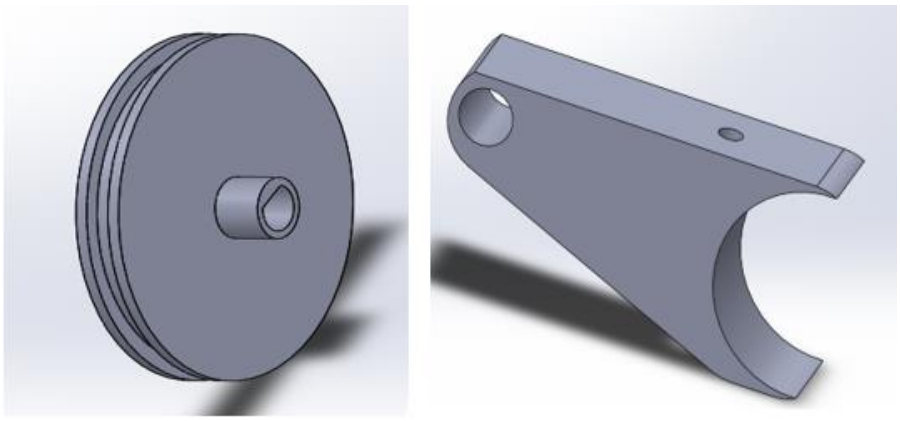

Figure 2:17 : la conception de support encodeur et roue codeuse sous « SolidWorks »

L'emplacement final de l'encodeur sur le châssis est montré dans la figure (2.18).

Figure 2:18 : L'emplacement finale des encodeurs liés aux roues codeuses sur le châssis

2.3.2 La coque :

Pour couvrir la plateforme développée, nous avons confectionné une coque dont la conception sous "SolidWorks" est présentée dans la figure (2.19). Pour la fabrication de cette coque, nous avons utilisé la fibre de verre pour sa solidité et sa facilité à manipuler.

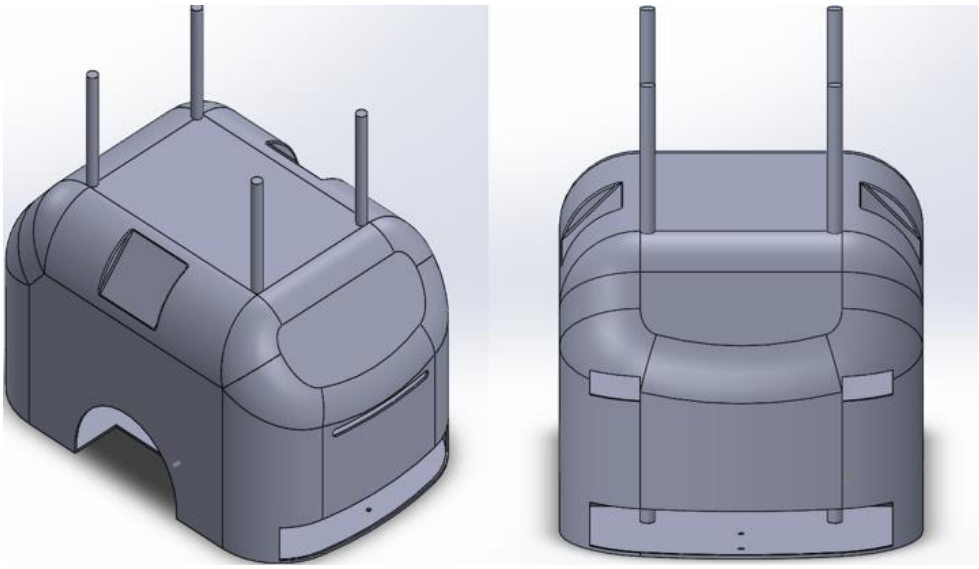

Figure 2:19 : La conception de coque sous "SolidWorks".

2.4 Conclusion :

Ce chapitre a été consacré à la conception et réalisation du robot mobile. Il été Ce chapitre a été consacré à la conception et réalisation du robot mobile. Il été principalement structuré en deux parties la première partie concernait l'architecture électronique du robot. La seconde partie quant à elle concernait sa structure mécanique. Une fois le robot réalisé, il devient possible de le doter d'intelligence, ce qui fait l'objet du chapitre suivan

Chapitre 3 :

Navigation réactive des robots mobiles

3 Chapitre 3 : Navigation réactive des robots mobiles

3.1 Introduction :

Dans le chapitre précédent, nous avons présenté la plateforme robotique développée. Cette plateforme doit être dotée d'intelligence pour lui permettre de naviguer de manière autonome, d'où l'objectif exposé dans ce chapitre.

Ce chapitre se structure selon, trois sections importantes : un état de l'art sur les méthodes de navigation réactive, Discussion, L'approche de navigation réactive propose.

3.2 Approches de navigation réactive pour robots mobiles :

Les approches réactives, connues généralement comme approches d'évitement d'obstacles, utilisent les informations des capteurs embarqués afin de calculer à chaque pas de temps le contrôle à appliquer aux actionneurs. Ce qui fait d'elles des approches très adaptées pour une exécution en temps réel. Une présentation non exhaustive de ces méthodes est donnée dans les paragraphes suivants.

3.2.1 Les algorithmes Bug :

Les algorithmes Bug1 et Bug2 font partie des méthodes les plus anciennes et les plus simples [11,12]. Ces algorithmes supposent que le robot est un point qui opère dans un plan 2D et qui utilise un capteur tactile pour détecter les obstacles. Ces algorithmes raisonnent selon deux comportements : "déplacement vers le but" et "suivi des limites des obstacles ".

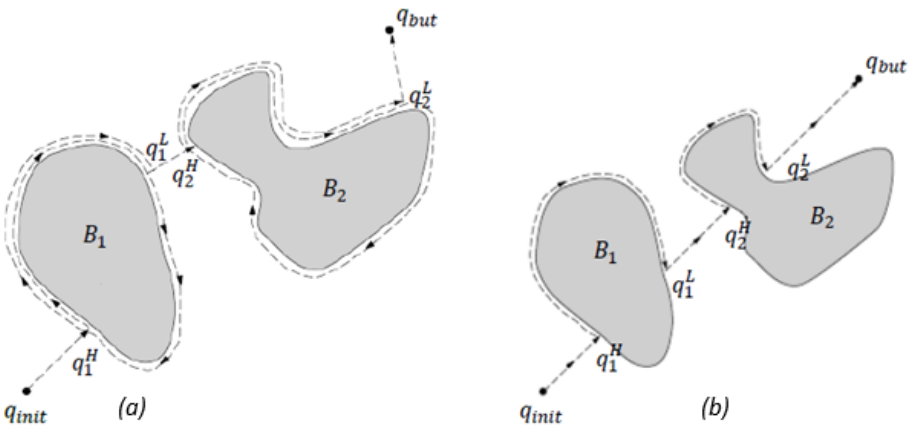

Figure 3:1 : Calcul du chemin du robot par les algorithmes Bug [12].

Ces méthodes naïves souffrent de beaucoup de problèmes dont le problème de sûreté de mouvement, le fait qu'ils ne prennent pas en considération les contraintes dynamiques et cinématiques du robot, et considèrent l'environnement comme statique.

3.2.2 La méthode basée sur les champs de potentiel PF :

La technique des champs potentiels artificiels (Potential Fields) [13] est une technique de navigation réactive très populaire utilisant la notion de force. Le principe est simple, la destination lui est appliquée une force attractive qui attire le robot, alors que les obstacles ont une force répulsive. Intuitivement il peut paraître qu'avec ce concept le robot ne peut jamais entrer en collision, et c'est le cas, mais l'algorithme ne garantit pas l'arrivée à la destination, et il arrive très facilement que le robot se bloque dans un minimum local et n'arrive pas à s'en sortir.

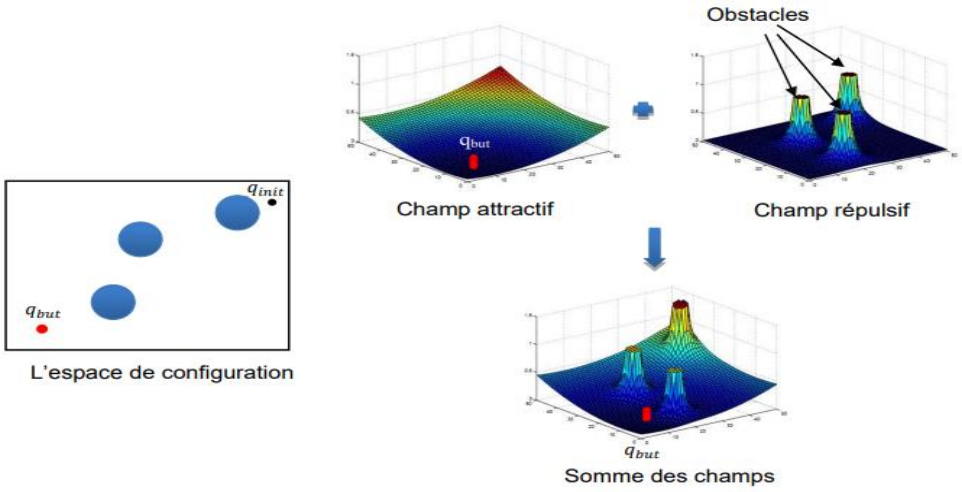

Figure 3:2 : Champs de potentiel pour un environnement contenant trois obstacles [14].

3.2.3 La méthode basée sur la fenêtre dynamique DW :

L'approche de la fenêtre dynamique a été proposée par Burgardand et Thrun [15]. Elle opère dans l'espace des vitesses du robot, permettant de prendre en considération les contraintes cinématiques et dynamiques en limitant l'espace de recherche uniquement à un espace de vitesses atteignables. A partir d'une perception locale de l'environnement, DW vise à sélectionner un couple (ν,ω) représentant la vitesse linéaire et angulaire du robot qui permet d'éviter les obstacles perçus. A partir des différents couples possibles, DW sélectionne le couple de vitesses le plus pertinent. Ces vitesses sont appelées vitesses admissibles. La zone de recherche est limitée par une fenêtre dynamique qui désigne les limitations dynamiques du robot et les vitesses admissibles qui peuvent être atteintes durant un intervalle de temps donné (voir la figure 3.3). Cette méthode a été initialement développée pour des environnements statiques, mais par la suite a été étendue pour des environnements dynamiques [47].

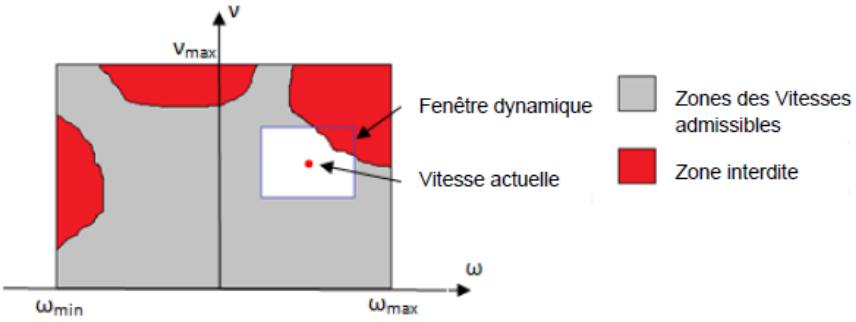

Figure 3:3 : L'approche de la fenêtre dynamique. L'espace des vitesses est divisé en régions admissibles et interdites. DW est représentée par le rectangle bleu contenant les vitesses atteignables par le robot durant un intervalle de temps spécifique [12].

3.2.4 La méthode basée sur la représentation des obstacles dans l'espace des vitesses VO :

L'approche "velocity obstacles" (VO) a été proposée par Fiorini et Shiller [16] afin de prendre en compte la dynamique de l'environnement et celle du robot. VO représente l'ensemble des vitesses qui mènent le robot vers une collision avec les obstacles voisins. Les obstacles statiques et dynamiques sont modélisés dans l'espace des vitesses du robot, i.e. la VO d'un obstacle circulaire planaire B qui se déplace avec une vitesse constante est un cône dans l'espace des vitesses de A, où A est réduit à un point et est élargi du rayon de B (comme illustré dans la figure 3.4. Chaque point de VO représente un vecteur de vitesse dont l'origine est le point correspondant à A. Toute vitesse A qui pénètre dans VO est une vitesse qui va conduire le robot vers une collision avec B. Avec une telle représentation, une manœuvre d'évitement peut être facilement calculée en sélectionnant une vitesse à l'extérieur de VO.

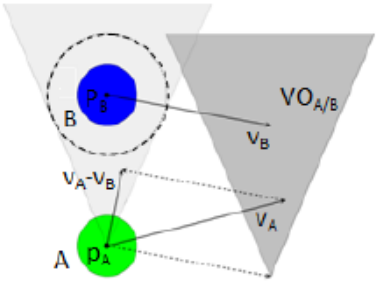

Figure 3:4 : L'approche VO : $VO_{A/B}$ pour A induit par B [14].

Toutefois, cette méthode a certaines limitations ; elle est moins performante en environnements fermés et considère uniquement des formes circulaires pour le robot et les obstacles.

3.2.5 Méthode basée sur l'échantillonnage de l'espace d'entrées :

Le principe de l'approche basée sur l'échantillonnage de l'espace d'entrées (en anglais: Input Sampling Space) (ISS) est simple; à partir d'un état initial, le modèle prédictif du système est utilisé pour générer un ensemble de contrôles (leur forme dépend du modèle du système et de son état initial. Ces trajectoires peuvent être triées par rapport à une fonction de coût donnée. L'exemple d'un ensemble de trajectoires généré en utilisant la technique par échantillonnage de l'espace d'entrées est illustré dans la figure 3.5.

Cette technique reste un sujet de recherche actif ; plusieurs améliorations concernant l'optimisation de l'algorithme (en réduisant le nombre de trajectoires testées pour non collision) [17] et le modèle prédictif utilisé [18] ont été apportées. Cependant, tous les travaux développés autour de cette technique n'ont traité que les obstacles statiques (ex. des roches dans le cas de l'exploration spatiale). Toutefois, en vue de sa conception de base, cette méthode peut être adaptée pour des environnements dynamiques.

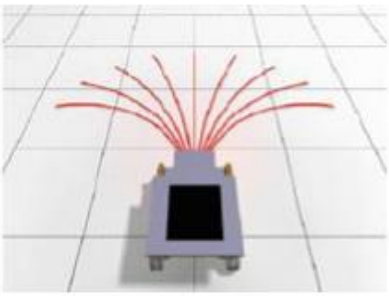

Figure 3:5 : La technique par échantillonnage de l'espace d'entrées. Un ensemble de trajectoires généré par l'échantillonnage de l'espace d'entrées, ces trajectoires sont de courbure et vitesse linéaire constante [12].

3.3 Discussion :

L'objectif de ce travail est de développer un système d'évitement d'obstacle de notre robot mobile qui lui permettra d'évoluer dans un environnement dynamique inconnu en se basant uniquement sur l'information de son système sensoriel. Pour cela, nous avons effectué une recherche bibliographique sur les travaux élaborés dans le domaine de la navigation. A partir de là nous avons mis en exergue les avantages et les inconvénients des différentes méthodes de navigations :

Les méthodes « algorithme bug » et « champ de potentiel PF » sont faciles à mettre en œuvre et peu gourmandes en termes de mémoires et temps de calcul mais elles ne sont appliquées dans des environnements dynamiques, de plus elles ne prennent en aucun cas

compte de la cinématique et de la dynamique du robot. Toutefois, des nouvelles variantes ont été proposées pour remédier à ce problème [19,20].

Par ailleurs, les méthodes « fenêtres dynamiques DW » et «velocity obstacles VO» présentent l'avantage de s'adapter à des environnements à la fois statiques et dynamiques tout en prenant en considération le modèle cinématique du robot, notamment leurs variantes TVDW [21] et GVO [22].

Quant à la méthode d'échantillonnage de l'espace d'entrées (ISS), elle a l'avantage de prendre en considération la cinématique et la dynamique du système robotique mais a l'inconvénient d'évoluer principalement dans des environnements statiques. Malgré cette limitation, la méthode ISS se révèle très intéressante relativement à sa simplicité, son efficacité (elle échantillonne directement l'espace de contrôles) mais surtout elle peut être facilement adaptée pour intégrer d'autre concepts et extensions. Toutefois, récemment cette méthode a été étendue pour des environnements dynamiques [23].

Par conséquent, pour notre application nous avons opté pour cette méthode d'échantillonnage de l'espace d'entrées. Le tableau 3.1 présente un résumé de l'étude comparative effectuée.

Méthodes réactives	Contraintes cinématiques	Contraintes dynamiques	Utilisation d'information sensorielle	Environnement inconnu	Environnement Dynamique	Simplicité
Algs. Bug	Non	Non	Oui	Oui	Oui ([19])	Très bonne
PF	Non	Non	Oui	Oui	Oui ([20])	Bonne
DW	Oui	Oui	Oui	Oui	Oui ([21])	Bonne
VO	Oui ([22])	Non	Oui	Oui	Oui	Moyenne
ISS	Oui	Oui	Oui	Oui	Oui ([23])	Très bonne

Tableau 3.1 : Comparaison des méthodes réactives.

((x) : la caractéristique peut être vérifiée sous certaines conditions).

3.4 L'approche de navigation réactive proposée :

L'approche développée est basée sur la méthode d'échantillonnage de l'espace d'entrée (ISS).

Cette approche est basée sur la génération d'un ensemble de contrôles en utilisant le modèle prédictif du système, à savoir son modèle cinématique pour notre cas. Ces contrôles sont par la suite vérifiés si elles sont sans collision et cela en utilisant les données sensorielles du système. Pour ce faire, avant de présenter le principe de l'approche, il est nécessaire de définir tout d'abord le modèle cinématique du robot et la représentation de l'environnement.

3.4.1 Modèle cinématique du robot mobile :

Le robot mobile développé dans ce projet possède une configuration différentielle il comporte deux roues fixes non orientables commandées indépendamment et deux roues folles placées à l'avant et à l'arrière pour assurer sa stabilité. Les roues motrices ayant le même axe de rotation, nous pouvons définir le centre instantané de rotation CIR du robot comme étant un point de cet axe (voir figure 3.6).

Soit *R* le rayon de courbure de la trajectoire du robot, c'est-à-dire la distance du *CIR* au point *O*' (figure3.6). Soit 2L la distance qui sépare les deux roues, et ω la vitesse angulaire du robot par rapport au *CIR*. Alors les vitesses des roues droite et gauche, respectivement notées vd et vg vérifient :

$$vd = (R + L)\omega \qquad (1)$$

$$vg = (R - L)\omega \qquad (2)$$

Ce qui permet de déterminer *R* et ω à partir des vitesses des roues :

$$\omega = \frac{Vd - Vg}{2L} \qquad (3)$$

$$R = L\frac{vd + vg}{vd - vg} \qquad (4)$$

La vitesse linéaire *v* du robot au point *O* ' est :

$$v = \frac{vd + vg}{2} \qquad (5)$$

Ces équations expliquent deux propriétés particulières du mouvement des robots différentiels : si vg = vd, la vitesse angulaire ω sera nulle et le rayon de courbure R est infinie donc le robot se déplace en ligne droite ; si *vg* = -*vd*, ω≠0 et R est nulle alors le robot effectue une rotation sur lui-même. Cependant, dans le cas où vg ≠*vd,* le déplacement du robot est un virage à gauche ou à droite selon que *vd* soit supérieur ou inférieur à *vg* (dans une direction qui correspond à la vitesse inférieur).

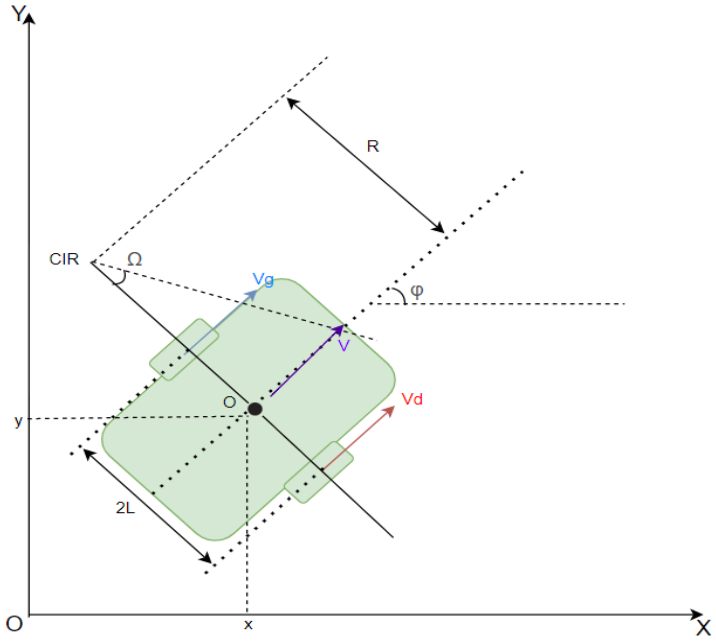

Figure 3:6 : Représentation du CIR d'un robot mobile différentiel.

Ainsi à partir de la figure 3.6 nous pouvons déduire les équations du modèle cinématique du robot différentiel.

$$\dot{x} = v\cos\varphi \qquad (6)$$

$$\dot{y} = v\sin\varphi \qquad (7)$$

$$\dot{\varphi} = \omega \qquad (8)$$

Ces équations relient la dérivée de la position (*x*, *y*, φ) du robot à la commande u = (v, ω)^T, avec φ rotation instantanée du robot par rapport au repère (O, X, Y). De ce fait la position du robot est donnée par :

$$x(t) = \int_0^t v(\sigma)\cos\big(\varphi(\sigma)\big)\,d\sigma \qquad (9)$$

$$y(t) = \int_0^t v(\sigma)\sin\big(\varphi(\sigma)\big)\,d\sigma \qquad (10)$$

$$\varphi(t) = \int_0^t \omega(\sigma)d\sigma \qquad (11)$$

3.4.2 Représentation de l'environnement :

3.4.2.1 Représentation des mesures laser

Etant donné que le laser est un capteur directif dans notre cas réparti en 567 directions (un champ de vision de 180° avec une résolution de 0,3°). La mesure est interprétée comme étant la distance au plus proche obstacle pour chacune des directions. Donc les mesures seront modélisées comme celles d'un capteur directif et cela pour chaque direction du capteur laser.

Nous exprimons les données laser par rapport au repère capteur R_c puis par rapport au repère local R_L (voir figure 3.7).

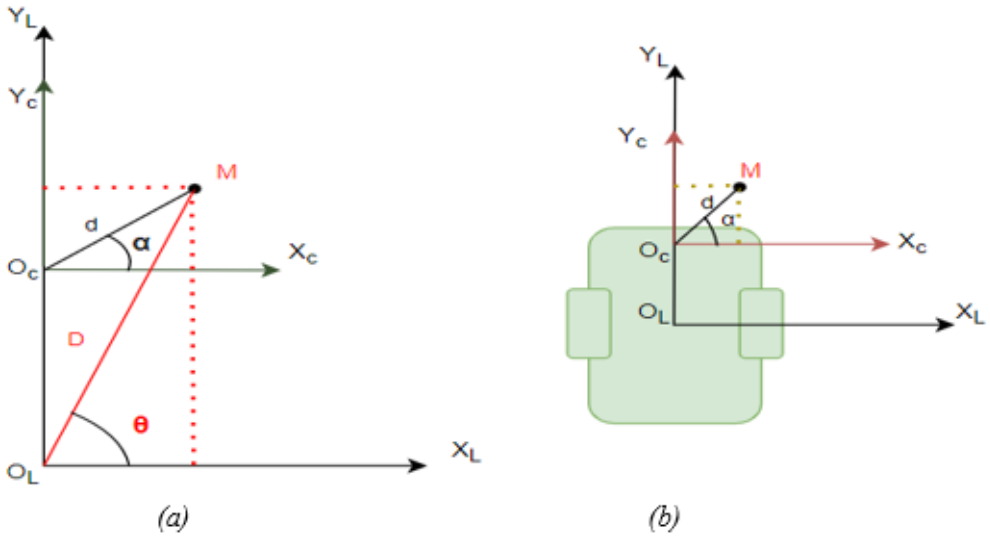

Figure 3:7 : Représentation des mesures laser

(a) Représentation de la mesure laser dans le repère capteur et le repère local

(b) l'emplacement du laser sur le robot mobile.

Représentation du point M dans le repère R_c

$$\begin{cases} x_C = \mathrm{d}\ \cos\alpha \\ y_C = \mathrm{d}\ \sin\alpha \end{cases} \qquad (12)$$

Avec : (x_C, y_C) la position de M dans Rc et α son orientation.

Représentation du point M dans le repère R_L :

En remplaçants la valeur de la position du repère R_c ((a,b,β)=(a,b,0)) dans R_L dans l'équation (13), Avec (a,b) la position du repère R_c dans le repère R_L et β son orientation. la position de M dans R_L est obtenue:

$$\begin{cases} x_L = \mathrm{d} * \cos(\alpha) + \mathrm{a} \\ y_L = \mathrm{d} * \sin(\alpha) + b \end{cases} \qquad (13)$$

Représentation du point M dans le repère R_L en coordonnées polaires (D, θ)

$$\begin{cases} D = \sqrt{{x_L}^2 + {y_L}^2} \\ \theta = \arctan\frac{y_L}{x_L} \end{cases} \quad (14)$$

3.4.2.2 Construction des zones de détection :

Représentation en espace de configuration :

Afin de pouvoir se déplacer sans contrainte dans l'environnement il est primordial de prendre en considération la géométrie du robot mobile, car aucun point du robot ne doit toucher les obstacles de l'environnement.

Pour ce faire, une représentation en espace de configuration s'impose où le robot est représenté par un point et les obstacles sont agrandis [14] Cette représentation permet de simplifier le problème de navigation pour un objet dimensionné en un problème de navigation pour un point. Par conséquent, chaque point de l'environnement est dilaté de façon à avoir un cercle de rayon R correspondent au rayon du robot mobile comme illustré dans la figure 3.8.

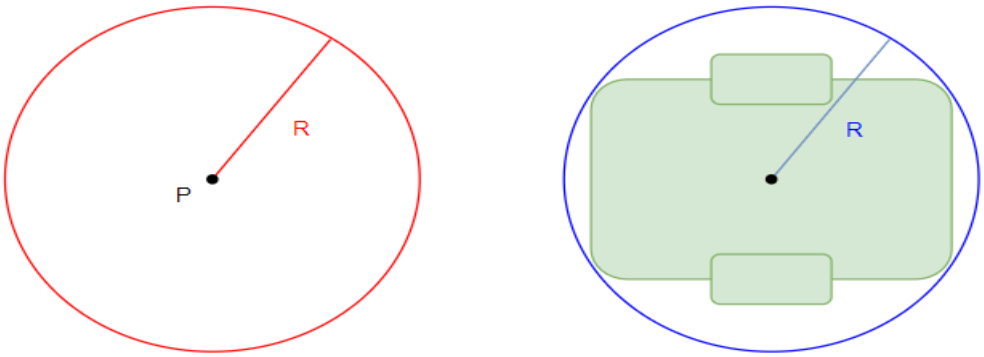

Figure 3:8 : Un point P de l'environnement agrandi par le rayon du robot R.

La Figure 3.9.a montré l'exemple de notre robot mobile qui peut se mouvoir librement dans l'espace de travail selon un chemin libre de collision à partir de la position initiale jusqu'à la position but du robot. L'espace de configuration correspondant est illustré dans la figure *3.9.b* où la région interdite est représentée en gris et la région libre est représentée en blanc (l'empreinte des obstacles est représentée uniquement à titre indicatif). Le chemin (libre de collision) du robot est également indiqué.

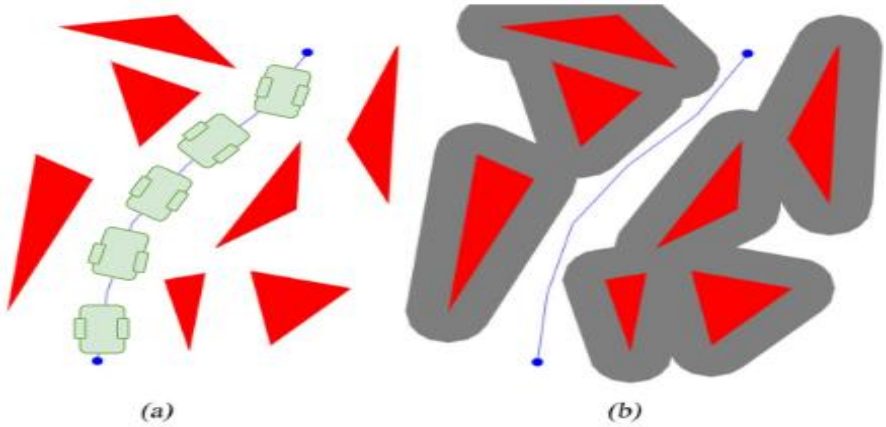

Figure 3:9 : Le déplacement de la chaise dans un plan 2D encombré d'obstacles statiques.
(a) L'espace de travail, (b) l'espace de configuration.

3.4.3 Échantillonnage de l'espace local du robot :

Afin de mettre en œuvre notre approche de navigation, nous avons échantillonné l'espace de configuration en plusieurs zones de détection (voir figure 3.10), où chaque zone a un angle d'ouverture α_{zone}.Ces zones sont construites dans le repère local R_L relié à le robot mobile. Chaque point de l'environnement appartient à une zone bien précise. En conséquent, chaque zone lui sera affectée une valeur désignant si elle est occupée ou libre (selon la présence ou non d'un obstacle). Ainsi, la position d'un point détecté par le système de perception est tous d'abord défini dans le repère local R_L, puis selon une distance minimale de détection, chaque zone est testée pour savoir si elle est occupée ou non (i.e. espace interdit ou admissible).

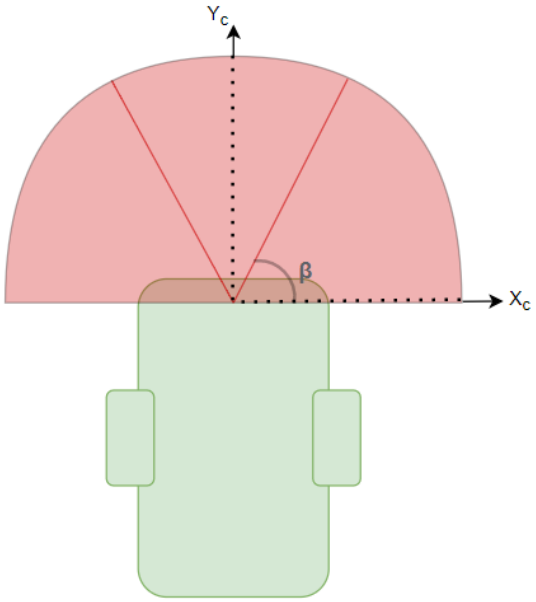

Figure 3:10 : Répartition des zones échantillonnées (trois zones).

3.4.4 Approche d'échantillonnage de l'espace d'entrée :

3.4.4.1 Principe de l'approche ISS :

ISS est une méthode de navigation réactive qui agit pendant une période de temps δt. A chaque pas de temps δt, son but est de calculer un contrôle u qui sera appliqué au système robotique A durant le pas de temps suivant. Son principe de base consiste à échantillonner l'espace de contrôles U en sélectionnant un sous-ensemble de contrôles $U_{ctrl} \subset U$et par la suite de choisir un contrôle $u = (v, \omega)$ qui doit être admissible (voir figure 3.11). Ce contrôle est sélectionné à partir d'un ensemble de contrôles admissibles ${U_{ctrl}}^* \subset U_{ctrl}$.

Un contrôle u est admissible s'il est faisable et sans collisions. u est faisable dans un sens où il respecte les contraintes cinématique (grâce à l'équation (12, 13 et 14) de A. Ce contrôle est sans collision s'il n'est en intersection avec aucun obstacle de son environnement, i.e. intersecte la région admissible de l'espace (région libre).

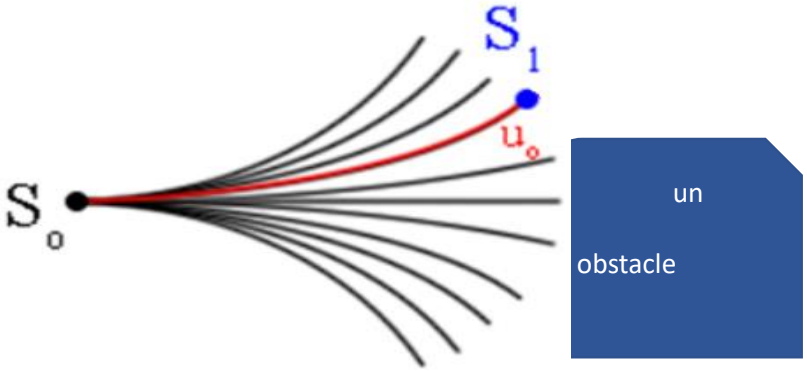

Figure 3:11 : Principe de fonctionnement de l'approche ISS.

3.4.4.2 Algorithme de l'approche ISS

Les étapes requises pour calculer un contrôle (v, ω) sans collision sont décrites dans l'algorithme 1.

Cet algorithme a pour entrées "Points" qui représentent tous les points de l'environnement perçus par les différents capteurs.

L'étape 1 de l'algorithme vise à rapporter les données du capteur dans le nouveau repère local tout en agrandissant chaque point en cercle comme expliqué précédemment (grâce aux fonctions ChangeReference et GrowPoint, lignes 7 et 8 de l'algorithme 1).

L'étape 2 consiste à échantillonner l'espace local de A en un certain nombre de zones

(NbrZone) puis affecter une valeur 0 pour zone libre ou 1 pour zone occupée (à l'aide de la fonction DetermineAdmissibleZone, ligne 11).

L'étape 3, A partir d'un ensemble de contrôle faisable U, nous sélectionnons un sous-ensemble Uctrl comportant NbrCtrl nombre de contrôle répartie uniformément sur nos zones (ligne 13).

Etape 4 vis à construire l'ensemble des contrôles admissibles U*ctrl et ce en récupérant uniquement les contrôles faisable d'ont les zones sont libres (fonction CheckCollisionFree, ligne 17 à 21).

Etape 5, Sur la base de l'ensemble U*ctrl obtenu précédemment nous sélectionnons (via la fonction SelecteControl (ligne 23) un contrôle u sans collision selon l'application.

Algorithme 1 : ISS ;

1 **Input:** Points: points perçus par l'environnement,
2 So= (x, y, vo, 0o) //état p-sûr actuel de A,
3 δt, NbrCtrl, NbrZone
4 **Output:** u : contrôle (v, ω) //respectivement vitesse linéaire et vitesse angulaire
5 i ← 0 ;
6 **foreach** point ∈ Points **do**
7 Change Reference (point) ;
8 PCercle[i] ← GrowPoint (point); //PCercle[i] est le point agrandi
9 i = i + 1;
10 **end**
11 Z← Determine Admissible Zone (PCercle, so, NbrZone);
12 // U est l'ensemble de tous les contrôles faisables
13 Sample U UCtrl = {u1, U2, ..., UNbrCtrl}://Sélectionner un sous ensemble de contrôle
14 de taille NbrCtrl
15 Uctrl=0;//initialiser l'ensemble des contrôles admissibles
16 //Sélectionner les contrôles admissibles
17 **foreach** u_i ∈ Uctrl **do**
18 //vérifier que u, est sans collision
19 **if** CheckCollisionFree (u_i, so, Z) == true then
20 $U^*_{ctrl} = U^*_{ctrl} \cup u_i$

21 **end**

22 **end**

23 u←SelectControl (U^*_{ctrl}, null,null);//Sélectionner un contrôle admissible u=(v,ω)

24 return u ;

3.5 Conclusion :

Ce chapitre a été consacré à une recheche sur les méthodes de navigation réactive, et à la méthode proposée après comparaison des avantages et inconvénients de chaque méthode. L'approche est nommée ISS, permettant l'évitement des obstacles. L'algorithme développé a été implémenté sur le robot, où les détails de l'implémentation seront présentés dans le chapitre 4.

Chapitre 4 :

Implémentation, Test et Résultats

4 Chapitre 4 : Implémentation, Test et Résultats

4.1 Introduction :

Dans ce chapitre, nous présentons l'environnement software (partie programmation) qui comprend essentiellement ROS (système d'exploitation pour robots), Nous illustrons par la suite l'architecture logicielle développée et les procédures d'implémentation. Nous présentons aussi les différents tests et nous finissons par présenter les résultats obtenus.

4.2 Système d'exploitation des robots « ROS » :

Comme son nom l'indique, ROS (Robot Operating System) est un système d'exploitation pour robots. De même que les systèmes d'exploitation pour PC, serveurs ou appareils autonomes, ROS est un système d'exploitation complet pour la robotique de service. C'est un environnement de développement open source composé d'un ensemble de bibliothèques logicielles, d'outils et d'algorithmes pour le développement et la programmation des systèmes robotiques et d'automatisation avancée. Pour utiliser ROS, nous avons utilisé un système d'exploitation (OS) open source basée sur la distribution Debian GNU / Linux. Ubuntu intègre toutes les fonctionnalités d'un système d'exploitation Unix avec une interface graphique personnalisable supplémentaire.

4.2.1 Les notions de base de ROS :

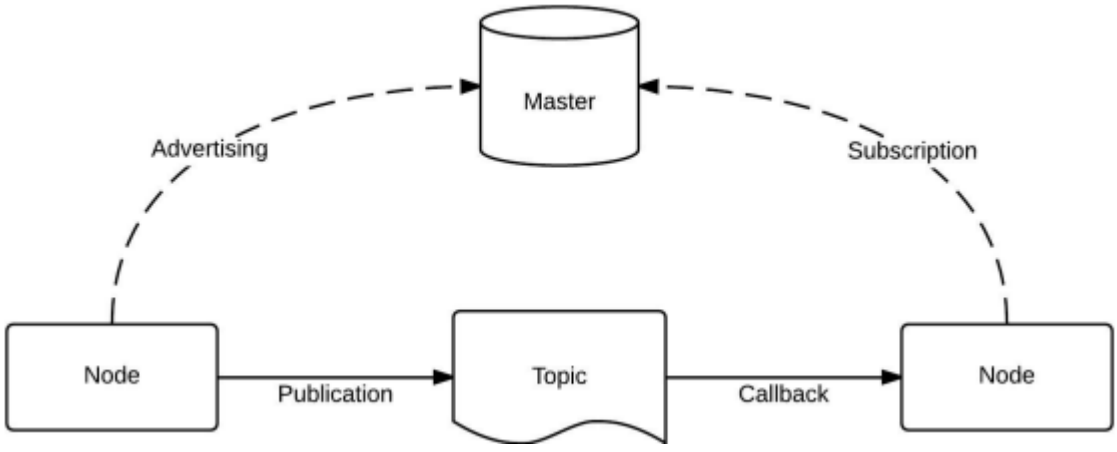

Figure 4:1 : Notion de base ROS.

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

1. Ros Master : est un service de déclaration et d'enregistrement des nœuds qui permet ainsi à des nœuds de se connaître et d'échanger de l'information.

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

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

4. Ros Messages : un message est une structure de données définissant le type d'un sujet, elle est composée d'une structure imbriquée de « entiers, booléens, flottants, chaîne de caractères, »..

5. Les bags : Les « bags » sont des formats pour stocker et rejouer les messages échangés. Ce système permet de collecter par exemple les données mesurées par les capteurs et les rejouer ensuite autant de fois qu'on le souhaite afin de faire de la simulation sur des données réelles. Ce système est également très utile pour déboguer un système a posteriori

4.2.2 Les outils très utiles dans ROS :

ROS est une collection d'outils. Certains sont très utilisés lors de la programmation, de la simulation ou de l'exécution des comportements des robots. Citons quelques outils ou algorithmes que le programmeur de ROS retrouvera souvent :

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

Catkin : C'est un système de gestion de paquets, de génération de code automatique et de compilation. Chacun de ces outils a également ses propres limites, et de nombreuses alternatives existent dans les paquets fournis sur ROS.

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

Dans ce travail, nous utilisons principalement Rviz et catkin.

4.3 Architecture logicielle proposée pour le robot mobile :

Dans l'architecteur logicielle proposé on a 3 modules (voir figure 4.2):

Module 1 : ce module est implémenté sur le PC et représente le superviseur qui gère les différents modules de l'architecture. Le superviseur communique avec les autres modules afin de recevoir et envoyer les données nécessaires pour un bon fonctionnement. Suivant les différentes informations récoltées, il est ainsi responsable du contrôle (v, ω) (avec v la vitesse linéaire du robot et ω sa vitesse angulaire), garantissant un déplacement sûr quelles que soient les conditions grâce à ISS.

Module 2 : ce module est géré par un microcontrôleur (Arduino Due 1). Il possède comme entrées les données reçues par les codeurs, ainsi que le contrôle (v, ω) émis par le module 1. En sortie le module envoie une valeur PWM vers les contrôleurs brushless. Sa fonction principale est l'asservissement et la gestion des deux moteurs brushless.

Module 3 : est responsable du traitement des données du capteur lidar sous ROS. Ces dernières sont émises vers le module1 (superviseur).

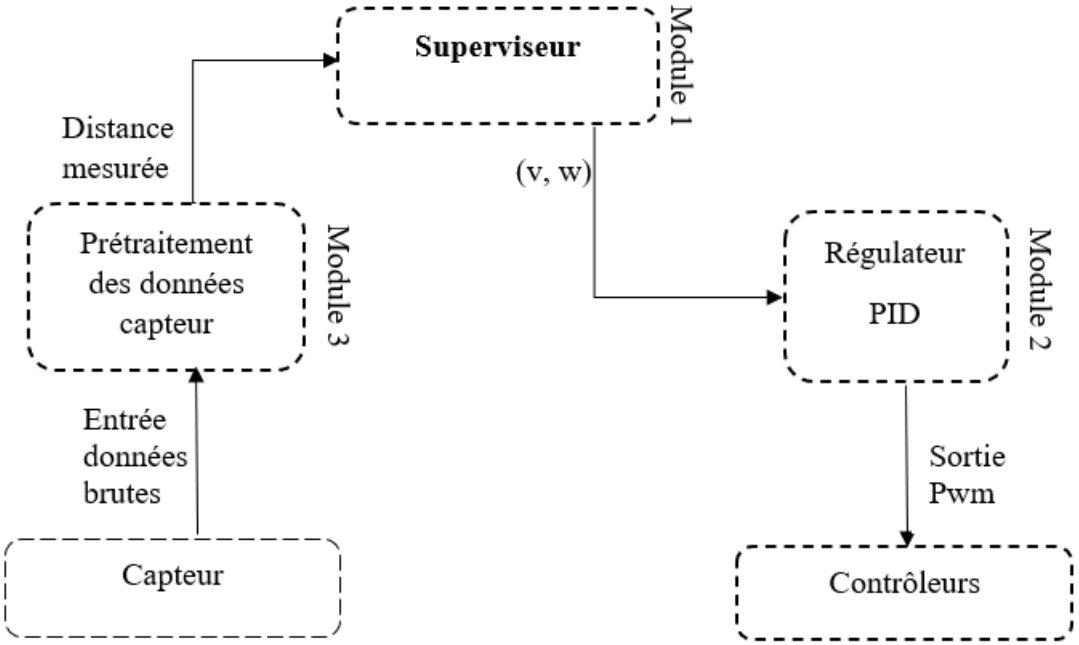

Figure 4:2 : Architecture logicielle proposée.

Notre espace de travail ROS contient trois packages. Nous avons utilisé deux packages ROS_SERIAL » et « RPLIDAR_ROS » et nous avons créé le troisième package que nous avons nommé « PROJET_ADIUTOR ».

Le premier package « ROS_SERIAL » a pour but de gérer la commnication entre le superviseur et l'Arduino Due. Il contient un nœud et un topic sur lequel ce nœud va publier ou souscrire des données. L'Arduino Due Contient le programme de gestion des moteurs et de l'asservissement (correcteur proportionnel intégral dérivé PID).

Le deuxième package « RPLIDAR_ROS » est spécifique au capteur laser RPlidar A2 et s'occupe de la collecte des données provenant de ce dernier.

Le troisième package « PROJET_ADIUTOR » a pour rôle de gérer la partie navigation du robot. Il contient un seul nœud principal « SUPERVISOR ». Le graphe de l'ensemble des nœuds et des topics du système est illustré sur la figure 4.3.

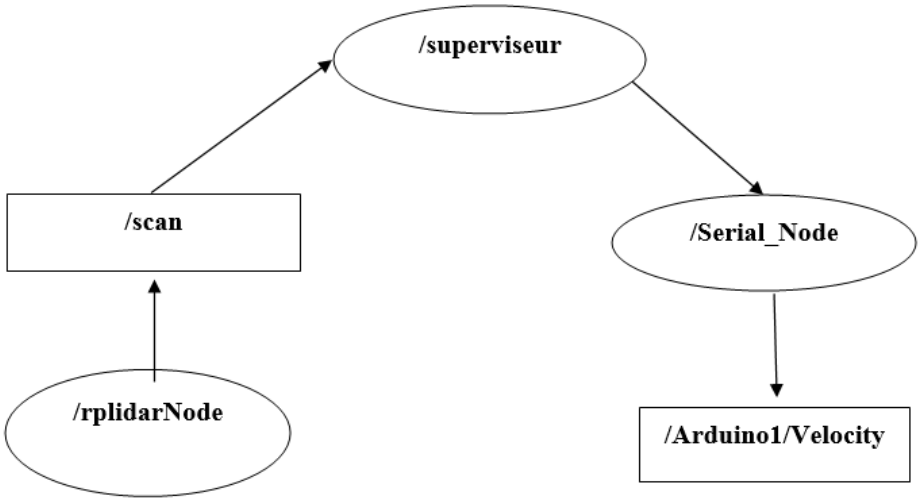

Figure 4:3 : Graphe de l'ensemble des nœuds et topics du système.

Avec :

Dans le package Ros_serial :

/serial_node : Représente le nœud de communication avec Arduino.

Dans le package projet_adiutor :

/Arduino1/Velocity C'est le topic dans lequel est publiée la commande des moteurs envoyé par le nœud SUPERVISOR.

Dans le package rplidar_ros :

/rplidarNode c'est le nœud qui contient le programme du télémètre laser.

/scan C'est le topic dans lequel sont publiées les données du télémètre laser

4.4 Validation de la plateforme robotique : tests et résultats

4.4.1 Robot mobile développé :

Après avoir étudier la partie électrique et la partie mécanique dans le chapitre 2 ainsi que l'architecture logicielle dans ce chapitre, le montage global a pu être réalisé.

Nous avons réaliser une carte liaison comportant la carte Arduino, la partie alimentation (basée sur les DC /DC). Elle représente la carte de commande. Cette carte permet de relier la partie opérative avec la partie commande Nous allons commencer par l'encodeurs qui est connecté avec les pins d'entrées interruptions. Le raccordement des contrôleurs moteur se fait via deux pins pour chaque contrôleur : la première pour la vitesse et la deuxième pour le sens.

Le contrôleur est rattacher directement au moteur (Figure 4.4). Tous les composants mentionnés précédemment nécessite une alimentation de 5V.

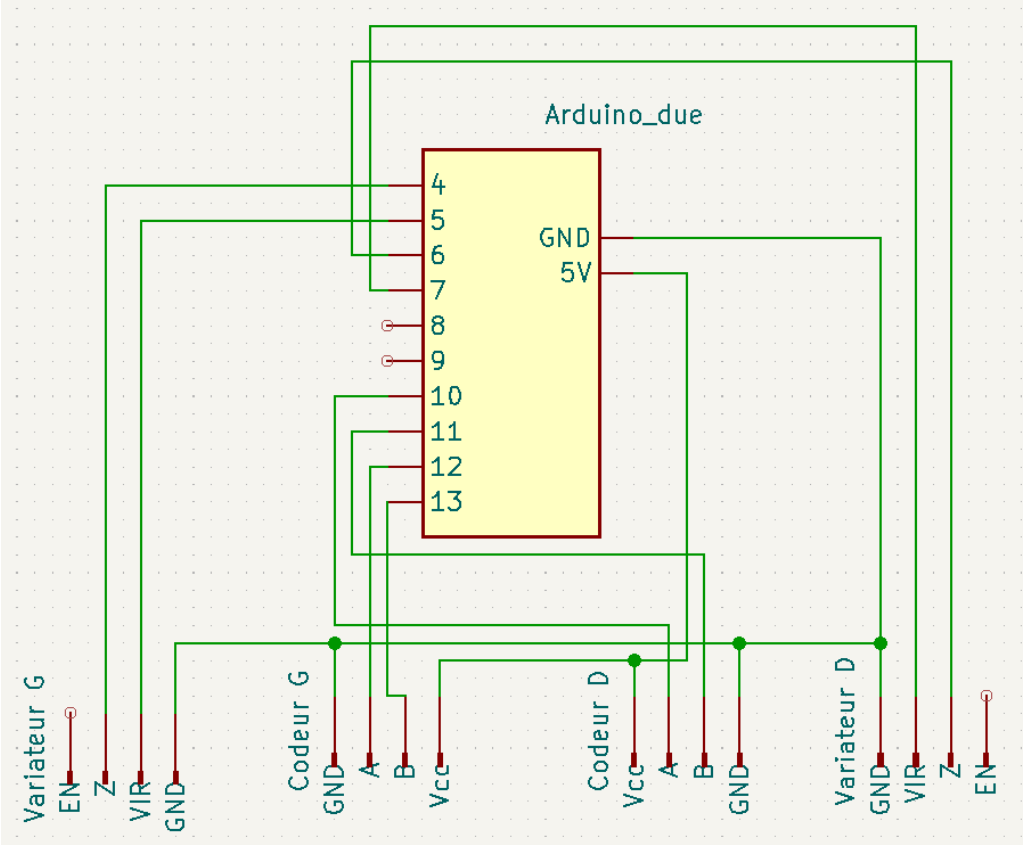

Figure 4:4 : le schéma de circuit électrique du robot.

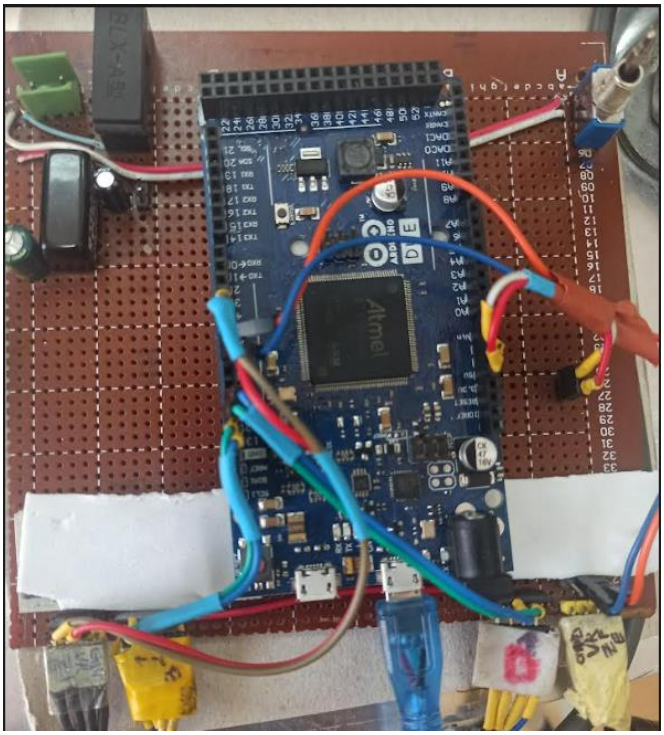

Figure 4:5 : le circuit électrique du robot (photo réel).

À la fin, le montage final du châssis rassemble la carte de commande Arduino Due, le capteur laser, la batterie lithium (tension de 36V et sa Capacité de 144WH) et les contrôleurs des moteurs Brushless (figure 4.6).

Figure 4:6 : le robot sans habillage.

4.4.2 Asservissement de la vitesse des moteur (PID)

4.4.2.1 Calcul de la vitesse des moteurs :

Avant de procéder à l'asservissement des moteurs, nous avons mesuré les vitesses des moteur droite et gauche en marche avant et arrière et nous avons relevé la tension correspondante. Le tableau 4.1 représente les résultats obtenus.

		MOTEUR G		MOTEUR D	
PWM	Tension(V)	Vitesse avant (RPM)	Vitesse arrière	Vitesse avant	Vitesse arrière
90	1	233	474.95	455.12	266.09
100	1,2	234.15	464.89	515	232.16
110	1,35	248,18	511.7	590.05	255.07
120	1,45	251.79	535.66	619.43	261.48
125	1,57	258.74	562.56	648.57	274.32
130	1,77	245.25	579.13	673.13	286.55
135	1,8	248.7	593.67	691.22	298.92
255	3,2	464.26	748,22	722.83	492.22

Tableau 4.1 : les vitesses du moteur en marche avant et arrière.

4.4.2.2 Régulateur PID :

Le régulateur PID (Proportionnel, Intégral, Dérivé) est un système d'auto régulation (boucle fermée), qui cherche à réduire l'erreur entre la consigne et la mesure [23] (voir figure 4.7).

E = Consigne – Mesure

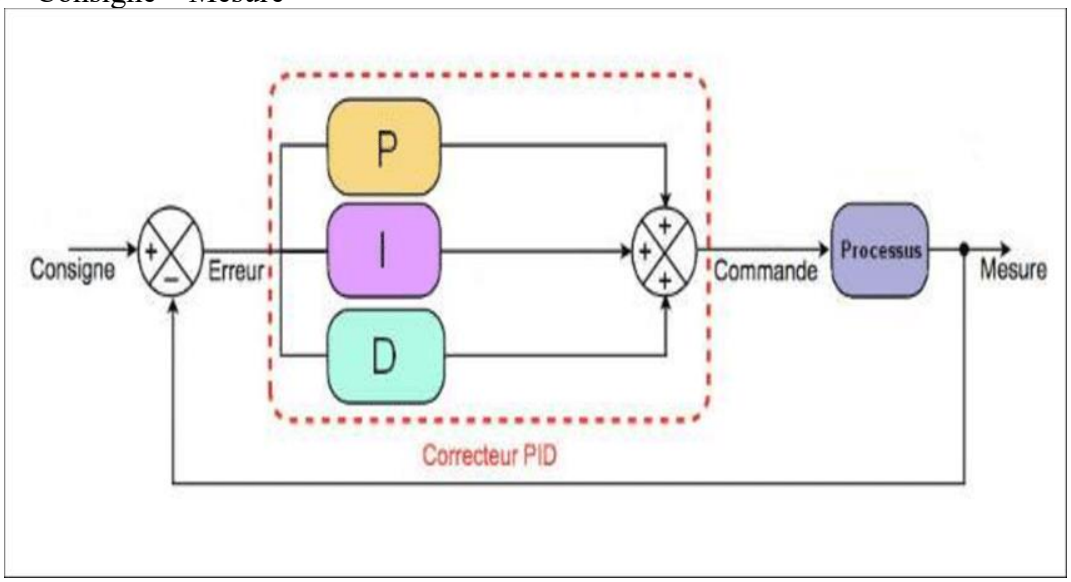

Figure 4:7: Schéma fonctionnel d'un régulateur PID [24].

Le régulateur PID sert à atteindre la valeur souhaitée pour une des variables du système (vitesse, position…).

L'action Proportionnel :
L'erreur est multipliée par une constante Kp

$$u(t) = Kp \times e(t) \quad (15)$$

$$Gain\ Kp = u(p) = Kp\ e(p) \quad (16)$$

Plus Kp est grand, plus la réponse est rapide mais on obtient une Erreur statique.

L'action Intégral

L'erreur est intégrée sur un intervalle de temps, puis multipliée par une constante Ki

$$u(t) = Ki \times \int e(t)\, dt \quad (17)$$

$$Gain\ Ki = u(p) = Ki \times e(p)/p \quad (18)$$

Le corrige l'erreur statique, Plus Ki est élevé, plus l'erreur statique est corrigée.

L'action Dérivé :

L'erreur est dérivée par rapport au temps, puis multipliée par une constante Kd

$$u(t) = Kd \times \partial e(t)/\partial t \quad (19)$$

$$Gain\ Kd = u(p) = Kd \times e(p) \times p \quad (20)$$

Réduit le dépassement et le temps de stabilisation mais aussi notre système sera Sensible au bruit.

On résume alors :

$$u(t) = Kp \times e(t) + Ki \times \int e(t)dt + Kd \times \partial e(t)/\partial t \quad (21)$$

$$u(p) = e(p) \times (Kp + Ki/p + Kd \times p) \quad (22)$$

$$gain\ Totale = gain\ kp + gain\ ki + gain\ Kd \quad (23)$$

4.4.3 Validation de la partie intelligence : tests et résultats

Nous avons implémenté l'algorithme d'évitement d'obstacle ISS sur un ordinateur doté d'un processeur Intel Core i5 (11ème génération). Ce dernier communique avec la carte Arduino et le capteur lidar afin de recevoir et envoyer des données nécessaires pour le fonctionnement de l'ensemble et pour permettre d'accomplir les différentes tâches requises. Toute l'architecture logicielle de notre robot est gérée par le méta-système d'exploitation ROS.

Les étapes nécessaires pour l'implémentation sont les suivantes :

1- Implémentation du code de gestion des moteurs en boucle fermée sur à la carte Arduino.

2- Implémentation de l'algorithme ISS sur le PC via ROS. Après en compile tous les packages dans ros par catkin_make, ces derniers sont lancés comme suit (figure 4.8) :

– lancer roscore.

– lancer le nœud « rplidar_Node »

– lancer le nœud « serial_node ».

– lancer le nœud « SUPERVISOR ».

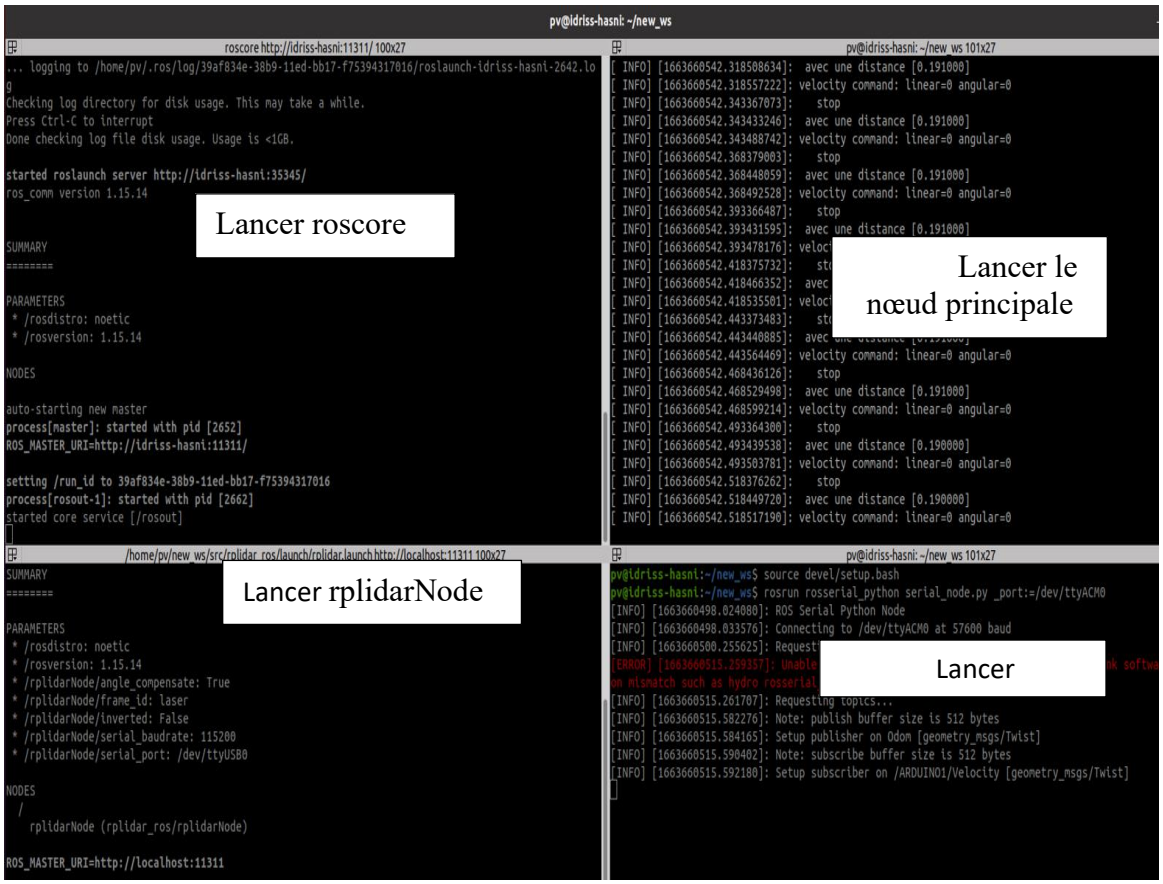

Figure 4:8 : les étapes nécessaires de l'implémentation

Les tests :

Pour valider l'approche, nous l'avons testé en premier dans un environnement comportant trois obstacles. On a commencé notre test par fixer le nombre de zones à 3 zones. Le test est illustré dans la figure 4.9 et 4.10.

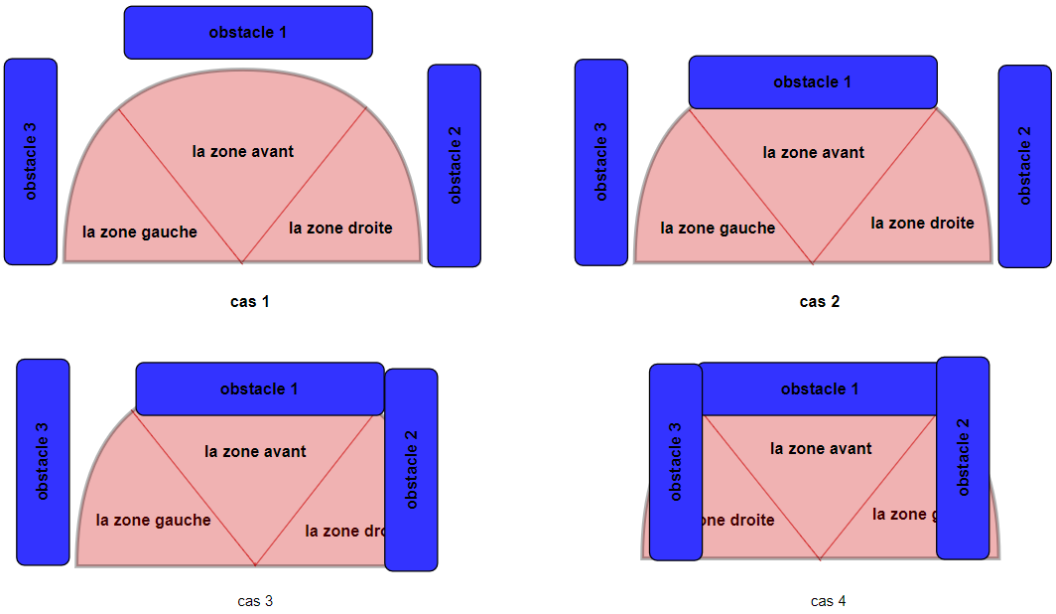

Figure 4:9 : Illustration des différents tests pour différents positions des obstacles.

Cas 1 : toute les zones sont libres, donc le robot peut aller vers l'avant.

Cas 2 : les zones gauche et droite sont libres alors que la zone avant est occupée, donc le robot peut tourner vers la droite).

Cas 3 : les zones avant et droite sont occupées et la zone gauche est libre, donc le robot peut tourner vers la gauche.

Cas 4 : toute les zones sont occupées, donc le robot s'arrête.

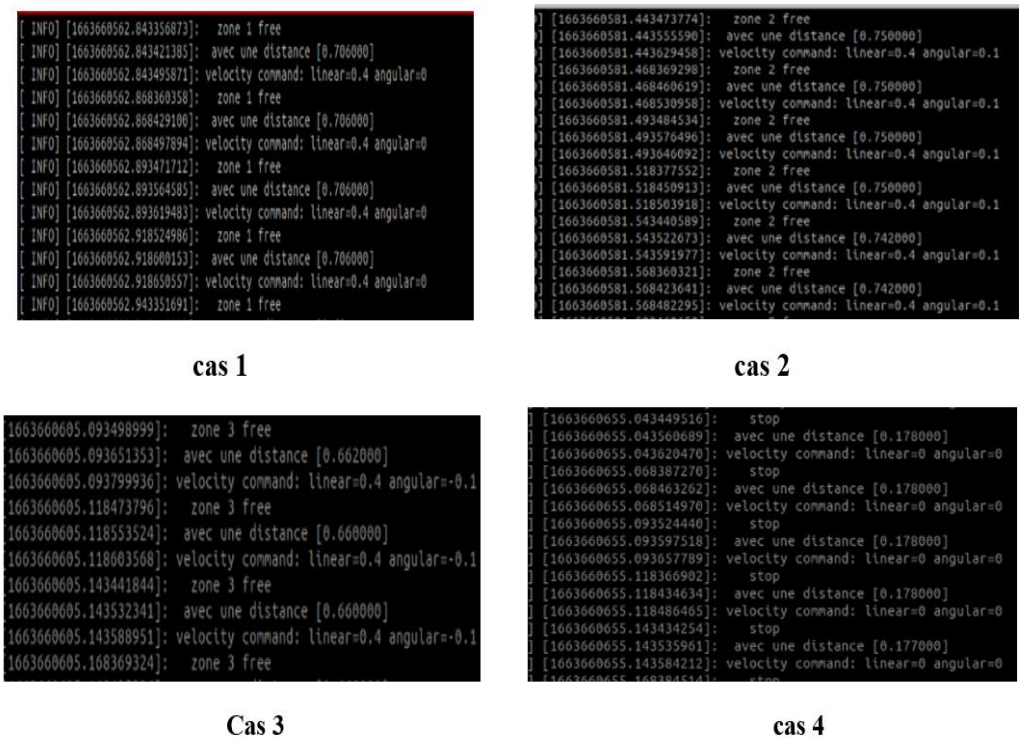

Figure 4:10 : les résultats de l'affichage sur ROS pour les quatre cas décrits dans la figure 4.9.

4.4.4 Evaluation du temps de calcul de l'algorithme ISS

Le temp de calcule augmente avec le nombre de zones, ce qui est une résultat logique (voir tableau 4.1).

toute fois nous constatent même si le nombre de zones est elevé le temp de calcule rest trés réduit ce la prouve que l'algorithme ISS au plus de sont simplicité il est trés adapté pour des application au temp réel.

Nombre de zones	Temp de calcule (ms)
3	0.32 ~ 0.39
9	0.56 ~ 0.63
15	0.79 ~ 0.82
30	1 ~ 1.3
45	1.9 ~ 2.1
60	2.5 ~ 2.2

Tableau 4.2 : Tableau représentant le temps de calcul de l'algorithme par rapport aux nombres de zones en fixant nombre de contrôle = 60.

4.5 Conclusion :

Au cours de ce chapitre nous avons exposé l'architecture logicielle développée. Nous avons opté pour ROS afin de gérer l'ensemble de l'architecture. Cela nous à simplifier le travail notamment côté communication et drivers. Nous avons également illustré les tests effectués pour valider la plateforme réalisée et l'algorithme d'évitement d'obstacles implémenté.

Conclusion générale

Conclusion générale

Dans ce présent travail s'inscrivant dans le cadre de projet de fin d'études, nous avons réalisé un robot de service pour une application de santé et nous l'avons doté d'intelligence. Le robot mobile développé est nommé ADIUTOR.

Plusieurs travaux ont été réalisés dans ce domaine pour résoudre le problème de transport dans un milieu hospitalier (transport des médecines, affaires des patients, repas, etc..).

Tout d'abord, nous avons effectué des recherches dans le domaine de la robotique mobile, pour pouvoir ainsi comprendre les différents types de solutions qui peuvent s'adapter à notre concept. A l'issue de cette étape, nous avons étudié les architectures matérielle et logicielle du robot, y compris la conception mécanique. Nous avons muni notre robot du système ROS pour qu'il soit conforme aux plateformes robotiques développées au niveau international, permettant d'intégrer facilement les fonctionnalités mise à la disposition de la communauté robotique.

Une fois la réalisation du robot achevé, nous l'avons doté d'intelligence. Pour ce faire, une recherche bibliographique sur les différentes méthodes de navigation réactive a été réalisée en premier. Après comparaison des avantages et inconvénients de chaque méthode, notre choix s'est porter sur la méthode d'échantillonnage de l'espace d'entrée (ISS) qui est une méthode simple et précise.

La réalisation de ce projet nous a permis de découvrir un nouveau domaine très important et passionnant qui est la robotique mobile en général et la navigation autonome en particulier.

Il nous a permis aussi d'apprendre de nouvelles notions dans ce domaine.

Les utilisations du robot Adiutor dans le domaine médical sont divers, par exemple, il peut transporter les repas aux lits des malades, suivre le médecin et déplacer ses affaires, livrer des échantillons au laboratiore, etc.

Les résultats des tests effectués ont permis la validation de la platefome développée et ont montré l'efficacité de la méthode d'évitement d'obstacle ISS.

Comme perspective, il est intéressant de réaliser ce projet dans un environnement réel car il est très bénéficiaire à l'être humain et permet de gagner du temps, de l'argent et le plus

important, la santé. Notre projet peut être également utilisé dans d'autres domaines, comme l'industrie, l'agriculture, le secteur militaire, etc…

Références Bibliographiques:

[1] C. Boukazzoula, “Etude, conception et réalisation d’une partie de commande à distance d’une plateforme mobile à roues,” mémoire de master, UMBB, Boumerdes, 2015.

[2] M. Ziat, “Etude et réalisation d’un robot mobile,” projet de fin d’etude, ENP, Alger, 2004.

[3] D. Filliat, “Introduction,” in Robotique Mobile, D. Filliat, Fondation. Paris Tech, ENSTA Eds. UNIT : 2004, pp. 9–14.

[4] O.BELAIDI,L.BELHACENE ,“Automatique industriel “28/06/2018

[5] Pierre-Yves Rochat, “ un microcontrôleur ? “, EPFL, École polytechnique fédérale de Lausanne , Nov. 2013.

[6] https://store.arduino.cc/products/arduino-due.

[7] J.Zhao, “Brushless DC Motor Fundamentals Application Note“,juillet, 2011.

[8] https://www.bldcmotor-driver.com/sale-11476358-jyqd-v7-3e3-3-phase-bldc-motor-driver-15a-current-pwm-speed-control.html

[9] https://www.generationrobots.com/blog/fr/qu-est-ce-que-la-technologie-lidar.

[10] parlonssciences.ca (02/2020)

[11] Milnor J., ―Morse Theory‖, Princeton University Press, Princeton, NJ, (1963).

[12] Choset H., Lynch K. M., Hutchinson S., Kantor G. A., Burgard W., Kavraki L. E. and Thrun S., ―Principles of Robot Motion: Theory, Algorithms, and Implementation‖, Livre, MIT Press, (2005).

[13] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots.In Autonomous robot vehicles, pages 396–404. Springer, 1986.

[14] Sara Bouraine. Contribution à la planification de mouvements en environnements dynamiques pour des environnements dynamiques pour des robots mobiles de type voiture : cas du robucar. Thèse de doctorat, 2016.

[15] Fox D., Burgardand W. and Thrun S., ―The dynamic window approach to collision avoidance‖, IEEE Robotics and Automation Magazine V. 4, Issue 1, (1997), pages 23-33

[16] Fiorini P. and Shiller Z., ―Motion planning in dynamic environments using velocity obstacles‖, Int. Journal Robotics Research V. 17, Issue 7, (1998), pages 760-772

[17] Knepper R., Srinivasa S. and Mason M., ―An equivalent relation for local path sets‖, In: Proceedings of the ninth international workshop on the algorithmic foundations of robotics, V. 68, (2010), pages 19-35

[18] Rogers-Marcovitz F. and Kelly A., ―On-line mobile robot model identification using integrated perturbative dynamics‖, In Proceedings of the 12th international

symposium on experimental robotics, (2010), pages 417-431.

[19] Haro, F. and Torres, M., ―A Comparison of Path Planning Algorithms for OmniDirectional Robots in Dynamic Environments‖, IEEE 3rd Latin American Robotics Symposium, LARS '06, (2006), pages 18-25.

[20] GE S.S. and CUI Y.J., ―Dynamic Motion Planning for Mobile Robots Using Potential Field Method‖, Autonomous Robots, V. 13, Issue 3, (2002), pages 207–222.

[21] Seder M., Petrovic I., ―Dynamic window based approach to mobile robot motion control in the presence of moving obstacles‖, In: IEEE Int. Conf. Robotics and Automation, Roma (IT), (2007), pages 1986-1991

[22] Wilkie D., van den Berg J. and Manocha D., ―Generalized velocity obstacles‖, In Proceedings IEEE International Conference on Intelligent Robots and Systems, (2009), pages 5573–5578

[23] S. Bouraine, Y. F. Loumachi and M. Boukhecheba, “a Safe Reactive Motion Planner for Service Robots”, The 2nd edition of the International Workshop on Human-Robot Interaction: New Trends in Service and Industry HRI-SI2022.

[24] https://www.researchgate.net/profile/Mohamed-Lamine-Berrandjia

[25] 31st IEEE International Conference on Robot & Human Interactive Communication IEE RO-MAN 2022, Italie, 2022.