République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. MIRA-BEJAIA
Faculté de Technologie
Département de Génie Electrique

Mémoire de Fin d'études

En vue de l'obtention du diplôme de Master en Automatique

Spécialité : Automatique et Informatique Industrielle

Thème

Etude et réalisation d'un robot a base mobile de type unicycle.

Préparé par :

M^{r}. MAZIOUA Toufik

M^{r}.ZEGROUR Abdelhakim

Encadré et dirigé par :

M^{r}.MENDIL Boubekeur

Devant le jury composé de :

M^{r}. HADDAR Hocine

M^{r}. MOKRANI Karim

Soutenu le : 14/09/2020

Année Universitaire : 2019/2020

REMERCIEMENTS

On tient à présenter nos plus sincères remerciements à notre encadreur Mr B.MENDIL pour avoir accepté de nous encadrer et diriger notre travail, pour sa patience, sa disponibilité et ses judicieux conseils, qui ont contribué à la réussite de notre travail.

Nous remercions très respectueusement Mr HADDAR Hocine et Mr MOKRANI Karim, de nous avoir fait l'honneur de juger notre modeste travail.

On remercie également toute l'ensemble pédagogique de l'université de Bejaia et l'ensemble de nos enseignants qui ont contribué à notre apprentissage.

Nous tenons à exprimer toutes nos reconnaissances à tous ceux qui ont contribués de prêt ou de loin à la réalisation de ce modeste travail.

DEDICACES

Nous dédions notre modeste travail :

A nos très chers parents qui n'ont pas cessé de croire en nous deux et de nous encourager et nous soutenir dans les moments difficiles, et qui ont prié Dieu pour nous, et qui ont su Nous entourer de toute leur affection et amour.

A nos camarades et a tout nos amis en particulier : Kouceila, Said, Yanis, Rabah, Wissem, Sara…

SOMMAIRE

CHAPITRE1 : GÉNÉRALITÉS

1.1. Introduction 1
1.2. La robotique 1
1.2.1. Définition 1
1.3. L'histoire de la robotique 1
1.4. Structure d'un robot autonome 3
1.4.1. Les capteurs 4
1.4.1.1. Les types de capteurs 4
1.4.2. Les actionneurs 5
1.4.3. Partie commande 6
1.5. Les types de robots 6
1.5.1. Les humanoïdes 6
1.5.2. Les robots industriels 7
1.5.3. Les robots mobiles 7
1.6. Conclusion 8

CHAPITRE2 : MODELISATION ET NAVIGATION

2.1. Introduction 9
2.2. Bref historique 9
2.3. Définitions 10
2.4. Classification des robots mobiles 11
2.4.1. Les robots terrestres (ou domestiques) 11
2.4.2. Les robots aériens 11
2.4.3. Les robots sous-marins 11
2.4.4. Robots polaires 11
2.5. Les robots à roues (à base mobile) 12
2.5.1. Modes de locomotion 12
2.5.2. Les types de roues 12
2.5.2.1. Roues standard 12

2.5.2.2. Roues omnidirectionnelles 13
2.6. Classes de robots à roues 13
2.6.1. Robot unicycle 13
2.6.2. Robot tricycle 14
2.6.3. Robot voiture 14
2.6.4. Robot omnidirectionnel 15
2.7. Les robots holonomes et non-holonomes 15
2.7.1. Degré de liberté (DLL) 15
2.7.2. Les robots holonomes 15
2.7.3. Les robots non-holonomes 16
2.8. Roulement avec ou sans glissement 17
2.8.1. Relation théorique entre les deux vitesses linéaire et angulaire [$v = f(\omega)$] 17
2.9. Modèle cinématique en posture d'un robot de type unicycle 18
2.9.1. Choix de la commande 18
2.9.2. Modes de locomotion 20
2.10. Navigation autonome 21
2.10.1. La génération de plan 21
2.11. Localisation 22
2.11.1. Localisation relative (Par odométrie) 22
2.11.2. Localisation absolue (Télémétrie) 22
2.12. Architecture de navigation (contrôle) 22
2.12.1. Approche délibérative (Planification-exécution) 23
2.12.2. Approche réactive 23
2.12.3. Approche hybride 23
2.13. Conclusion 23

CHAPITRE3 : DESCRIPTION ET CONTROLE DU ROBOT MOBILE
3.1. Introduction 24
3.2. Structure mécanique générale du robot 24
3.3. Matériel utilisé pour la composition du robot 25
3.3.1. La partie électronique 25
3.3.1.1. Arduino 25
3.3.1.1.1. Le logiciel arduino 26

3.3.1.1.2. Structure d'un programme sous arduino 27
3.3.1.2. ShieldArduino L293D 28
3.3.1.3. Pont en H 30
3.3.1.4. Encodeur rotatif optique 31
3.3.2. La partie mécanique 35
3.3.2.1. Le châssis 36
3.3.2.2. La roue bille 36
3.3.2.3. Moteur à courant continu avec réducteur (MCC/GEARBOX) 36
3.4. Contrôle PID du robot mobile 41
3.4.1. Définition 41
3.4.2. Propriétés d'un système asservi 41
3.4.3. Les boucles : 42
3.4.4. Les régulateurs 44
3.4.4.1. Régulateur proportionnels (P) 44
3.4.4.2. Régulateur proportionnel intégrateur (PI) 44
3.4.4.3. Régulateur proportionnel intégrateur dérivateur (PID) 45
3.4.5. Réglage des coefficients d'un PID 46
3.4.6. Navigation du robot et contrôle de vitesse 46
3.5. Conclusion 51

LISTE DES TABLEAUX

CHAPITRE1 : GÉNÉRALITÉS

CHAPITRE2 : MODELISATION ET NAVIGATION

Tab2. 1.Paramètres du robot type unicycle. 19

CHAPITRE3 : DESCRIPTION ET CONTROLE DU ROBOT MOBILE

Tab3. 1.Brochage et caractéristiques de la carte Arduino. 25
Tab3. 2.Valeurs PID et P_cal premier essai pour la marche avant 48
Tab3. 3.Valeurs PID et P_cal deuxième essai pour la marche avant. 49
Tab3. 4.Valeurs PID et P_cal final pour la marche avant. 50

LISTE DES FIGURES

CHAPITRE1 : GÉNÉRALITÉS

Fig.1. 1. (a) Canard digérateur de Jacques de Vaucanson, (b) Tortue cybernétique de Grey Walter..... 3
Fig.1. 2.Structure d'un robot. 4
Fig.1. 3.Schéma du principe de fonctionnement d'un capteur. 4
Fig.1. 4.Signaux délivrés par les différents types de capteurs. 5
Fig.1. 5. (a) Moteur à courant continu, (b) Vérin hydraulique simple effet 6
Fig.1. 6. (a) Le robot ASIMO de Honda, (b) Le robot NEXI de MIT. 7
Fig.1. 7.Le SCARA d'EPSON robot. 7
Fig.1. 8. (a) L'iRobot 510 Packbot., (b) Le robot de télé présence JAZZ. 8

CHAPITRE2 : MODELISATION ET NAVIGATION

Fig.2. 1. Robot beas de l'université John Hopkins dans les années 1960.....10
Fig.2. 2. (a) Robot a pattes T-HEX.[19]. (b) Robot mobile DJI ROBOMASTER. 11
Fig.2. 3.De haut en bas. Roue : Fixe, Centrée orientable, Décentrée orientable..... 12
Fig.2. 4. (a) Roue suédoise, (b) Roue sphérique. 13
Fig.2. 5.Robot de type unicycle 14
Fig.2. 6.Robot de type tricycle. 14
Fig.2. 7.Robot de type voiture. 14
Fig.2. 8.Robot de type omnidirectionnel. 15
Fig.2. 9.Axes : (a) vue de face. (b) vue de dessus. 16
Fig.2. 10.Modèle de type tricycle à roues différentielles 17
Fig.2. 11.Mouvement circulaire autour d'un axe fixe 17
Fig.2. 12.Modèle cinématique en posture d'un robot de type. 19
Fig.2. 13.Architecture logicielle d'un robot mobile à roues 21

CHAPITRE3 : DESCRIPTION ET CONTROLE DU ROBOT MOBILE

Fig.3. 1. Structure générale du robot mobile 24
Fig.3. 2.Carte ArduinoMega 2560 26
Fig.3. 3.Traduction du programme en langage machine 27
Fig.3. 4.Environnement de développement Arduino 27
Fig.3. 5.Structure de base d'un programme Arduino 28
Fig.3. 6.Le circuit L293D 28
Fig.3. 7.Signification de chaque pin du circuit intégré l293d 29
Fig.3. 8.Shield moteur L293D 29
Fig.3. 9.Forme générale d'un pont en H [36] 30
Fig.3. 10.Sens de rotation du moteur en fonction de l'état des interrupteurs 31

Fig.3. 11.Encodeur rotatif optique. 32
Fig.3. 12.Architecture interne d'un encodeur rotatif optique. 32
Fig.3. 13.Chronogramme du fonctionnement d'un encodeur. 33
Fig.3. 14.Partie déclaration des pins 33
Fig.3. 15.Fonction setup du programme. 34
Fig.3. 16.Fonction du calcul d'impulsions. 35
Fig.3. 17.Châssis de notre robot. 36
Fig.3. 18.Roue bille. 36
Fig.3. 19.Moteur à courant continu 37
Fig.3. 20.Architecture interne d'un moteur à courant continu. 37
Fig.3. 21.(a) Stator, (b) Rotor, (c) Balais. 38
Fig.3. 22. Règle des 3 doigts de LAPLACE 38
Fig.3. 23.Bibliothèque et configuration de pin 39
Fig.3. 24.Initialisation de la vitesse et état du moteur. 39
Fig.3. 25.Exemple rotation du moteur. 40
Fig.3. 26.Posture de notre robot. 40
Fig.3. 27.Plan complexe P. 41
Fig.3. 28.Schéma block en boucle ouverte. 42
Fig.3. 29.Schéma block en boucle fermée. 43
Fig.3. 30.Schéma block régulateur proportionnel 44
Fig.3. 31.Schéma block régulateur proportionnel integrateur 45
Fig.3. 32.Schéma block régulateur PID 45
Fig.3. 33.Schéma bloc général de la boucle de régulation 46
Fig.3. 34.Schéma bloc de la boucle de la marche avant. 47
Fig.3. 35.Réponse indicielle du premier essai de la marche avant. 49
Fig.3. 36.Réponse indicielle du deuxiéme essai de la marche avant. 49
Fig.3. 37.Réponse indicielle finale de la marche avant. 50
Fig.3. 38. Déplacement du robot d'un point A a un point B (De gauche a droite). 50
Fig.3. 39. Le robot atteint son point désiré. 51

LISTE DES SYMBOLES

L : Longueur.

ω : Vitesse angulaire.

v : Vitesse linéaire.

α : Angle de rotation.

$\dot{\varphi}_R$, $\dot{\varphi}_L$: Vitesses angulaires des roues droite et gauche

v_R, v_L : Vitesses linéaires des roues droite et gauche

(x, y) : Position du robot dans le plan

θ : Orientation du robot dans le plan

$(\dot{x}, \dot{y})$: Vitesse du robot dans le plan

r : Rayon des roues droite et gauche

b : Distance entre les 2 points de contact au sol des roues droite et gauche

ρ : Rayon de courbure de la trajectoire circulaire du robot

I : Intensité (A).

l : Longueur(m).

B: Champs magnétique (T).

α : L'angle formé par $\vec{B}$par rapport au conducteur.

k_p : Coefficient proportionnel.

K_i : Coefficient intégrateur.

K_d : Coefficient dérivateur.

$\varepsilon(p)$: Erreur statique.

D_{imp} : Nombre d'impulsions encodeur droit.

$Gimp$: Nombre d'impulsions encodeur gauche.

D : Distance en mètre (m).

R : Rayon de la roue des deux encodeurs.

N : Vitesse intégrée de l'encodeur.

P_{cal} : Facteur de calibrage.

INTRODUCTION GENERALE

Le progrès qu'a connu la technologie et l'expansion de l'automatisation, ont permis à l'humain de l'aider dans ses différents travaux et taches, et ont répondu à son envie d'évolution, en effet l'arrivée des robots qui dans un premier temps a concerné que l'industrie (usines..), a petit à petit progressé pour atteindre plusieurs domaines (militaires, médecines, éducations, domestiques etc..) on rencontre de plus en plus de robots mobiles capables d'effectuer une missions spécifique dans un endroit spécifique.

Un des défis majeur de la robotique mobile est la planification de mouvement, de nombreuse recherches et études ont été réalisées dans ce sens, afin de développer des méthodes pour guider les robots.

Les robots mobiles peuvent prendre plusieurs formes et servir dans de nombreuses applications, dans le cadre de notre projet, nous allons étudier et réaliser un robot mobile à deux roues de type unicycle, en utilisant moteurs à courant continu pour qu'il puisse se déplacer, la commande est faite par la carte Arduino à microcontrôleur ATmega 2560 et le shield L293D, une fois la réalisation terminée, on va procéder à la programmation du robot.

Notre mémoire est organisé en trois chapitres, le premier concerne les généralités, on présente le robot et sa définition et son histoire et domaines d'applications.

Le deuxième chapitre est consacré à la modélisation et navigation de notre robot, divisé en deux parties, la première concerne la classification et types de robots mobile, la deuxième concerne l'aspect de navigation et modèle cinématique.

Le troisième et dernier chapitre traite de la réalisation pratique (réalisation du robot), la carte de commande et son fonctionnement, les différents composants utilisés et le réglage du PID qui permet au robot d'avoir un bon fonctionnement.

Nous terminerons notre manuscrit par une conclusion générale qui résume tout notre travail et connaissances acquise durant le processus de l'élaboration de notre projet, et expose les perspectives de ce travail.

1.1. Introduction

Habituellement, l'image que nous donnons au robot est l'image d'un système mécanique devant articulé des tâches telles que le soudage et la peinture. Mais les robots mobiles à roues sont actuellement peu utilisés dans les applications industrielles. Cependant, ce type de véhicule a de nombreuses applications potentielles : Applications de nettoyages, aide à la mobilité pour les personnes âgées ou handicapés, etc.

Dans ce chapitre, Nous donnerons quelques définitions, un bref historique de la robotique et aussi une petite comparaison entre robot et automate.

1.2. La robotique

La technologie des robots a de nombreux domaines d'application, tels que la technologie des robots industriels ou la technologie des robots de service. Qu'il s'agisse d'un robot civil ou d'un robot militaire, il existe désormais des robots dotés de talents extraordinaires dans de nombreux domaines: les robots compagnons peuvent aider les personnes à domicile ou être responsables de la surveillance et des soins, les robots peuvent assurer la logistique dans les hôpitaux, assister les industriels dans la réalisation de gestes pénibles et répétitifs ou encore permettre le développement de prothèses ou d'orthèses intelligentes.

1.2.1. Définition

La robotique est une Science et technique de la robotisation, de la conception et de la construction des robots.

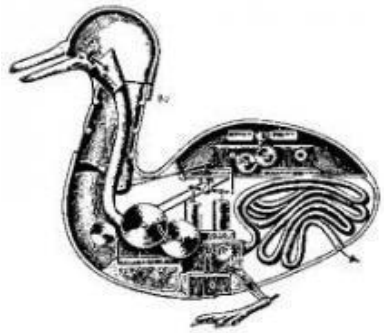
En fait, il s'agit d'un domaine multidisciplinaire : on y trouve des aspects concernant la mécanique, l'informatique, l'électronique. C'est l'ensemble des domaines scientifiques et industriels qui sont en rapport avec la conception et la réalisation de robots. [1]

1.3. L'histoire de la robotique [2]

Ce n'est qu'au tout début du XXe siècle que les robots firent leur apparition, suite aux travaux d'ingénieurs qui voulaient tester des hypothèses émises par des biologistes et des psychologues. [3]

En 1738, **Jacques Vaucanson** crée un canard articulé en cuivre capable de boire, manger, cancaner et digérer comme un véritable animal. C'était en quelque sorte la création la plus brillante du temps.

- 1921 : **Karel Capek** écrit RUR (Rossum's Universal Robots), pièce théâtrale dans laquelle il introduit le mot *robota*(travail en tchèque) signifiant un être artificiel qui travaille dur.
- 1940 : **Isaac Asimov** écrit un ensemble de nouvelles sur les robots, notamment les trois lois de la robotique.
- 1948 : **Grey Walter** invente le premier robot mobile autonome, une tortue se dirigeant vers les sources de lumière qu'elle perçoit. Cependant, ce robot n'est pas programmable.
- 1961 : Premier robot industriel mis en place dans une usine de General Motors : **UNIMATE**.
- 1972 : Nissan ouvre la première chaîne de production complètement robotisée.
- 1978 : **PUMA** (Programmable Universal Machine for Assembly) développé par General Motors (toujours utilisé).
- 1997 : premier robot mobile extra planétaire sur Mars.
- 1999 : Lancement d'**Aibo**.
- 2000 : Lancement d'**Asimo**.
- 2003 : Projet "Mars Exploration Rover" (**Spirit** & **Opportunity**). Diversification des compétitions de robotique. Utilisation de drones en situation réelle (Irak...).
- 2006 : le projet **Aibo** n'est plus assez rentable, fin de la production.
- 2009 : projet "Mars Science Laboratory" succédant au projet Rover, envoi sur Mars de **Curiosity** fin 2011.
- 2011 : Robonaut (R2B) premier robot humanoïde envoyé dans l'espace, conçu et construit par la **NASA** au Johnson Space Center (**JSC**) à Houston (Texas), en collaboration avec General Motors (**GM**) et Oceaneering.



(a)

(b)

(c)

(d)

Fig.1. 1. (a) Canard digérateur de Jacques de Vaucanson, (b) Tortue cybernétique de Grey Walter, (c) Le premier robot industriel Unimate, (d) Le rover Curiosity. [4]

1.4. Structure d'un robot autonome [5]

Un robot autonome est un assemblage complexe de pièces mécaniques et de pièces électroniques, le tout pouvant être piloté par des algorithmes divers de contrôle, de perception et de reconnaissance de formes et de prise de décisions.

Lorsque les robots autonomes sont mobiles, ils possèdent également une source d'énergie embarquée : généralement une batterie d'accumulateurs électriques.

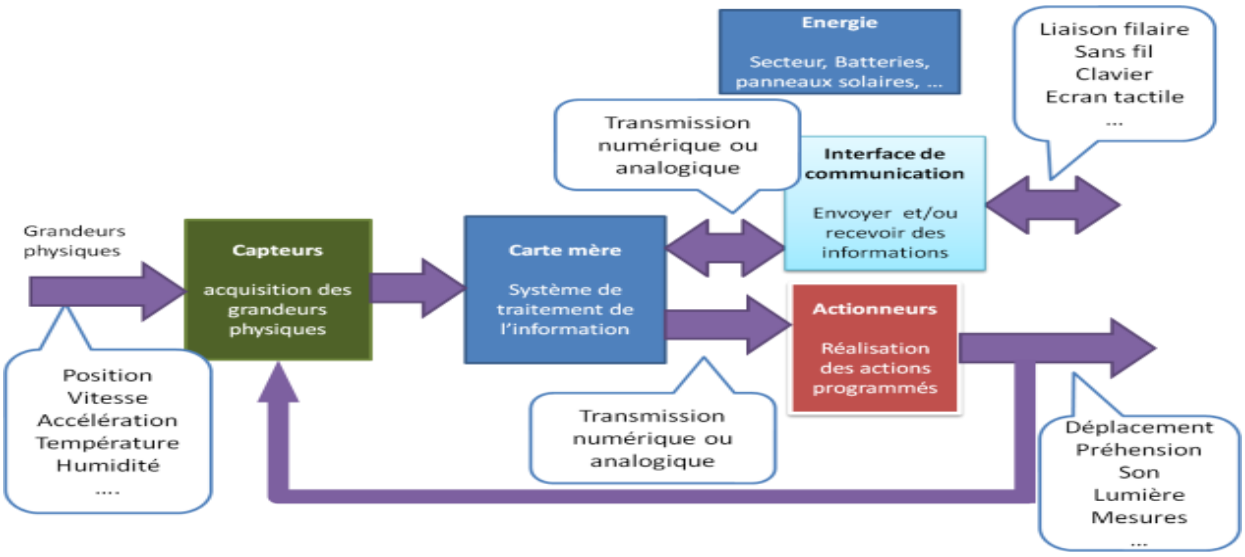


Fig.1. 2.Structure d'un robot.[6]

1.4.1. Les capteurs

Dans un robot, la perception est assurée par l'utilisation de capteurs. Ces derniers donnent des informations à propos de l'environnement ou des composants internes (e.g. position d'un moteur ou d'un vérin, état d'une LED). Cette information est utilisée pour calculer l'ordre approprié à envoyer aux actionneurs. [7]

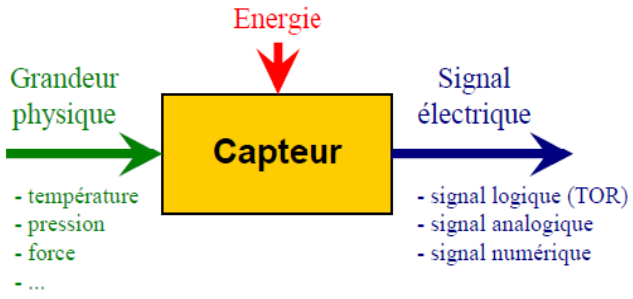


Fig.1. 3.Schéma du principe de fonctionnement d'un capteur.[7]

1.4.1.1. Les types de capteurs [7]

Selon la nature de l'information délivrée, on peut distinguer trois types de capteurs :

- **Capteurs analogiques :** Les capteurs analogiques servent à transformer une grandeur physique en un autre électrique (de variation d'impédance, de capacité, d'inductance ou de tension). Un signal est dit analogique si l'amplitude de la grandeur physique

qu'il représente peut prendre une infinité de valeurs dans un intervalle donné (présence continue).

- **Capteurs numériques :** souvent nommés codeurs ou compteurs, génèrent des signaux numériques, représentés par des codes binaires.
- **Capteurs logique / Tout ou rien (TOR)** : Ils portent le nom de détecteurs. Type de signal de sortie 0V ou 5V.

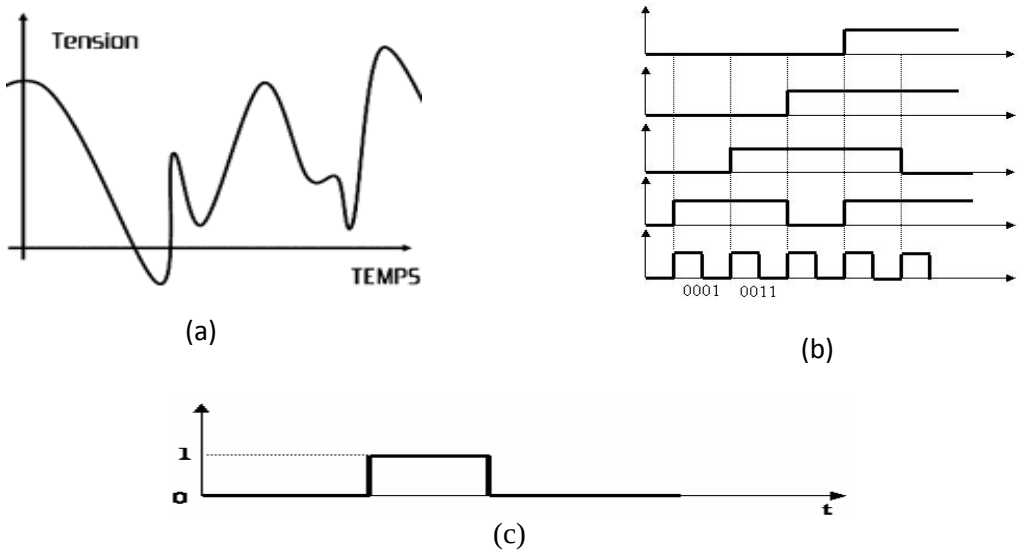


Fig.1. 4.Signaux délivrés par les différents types de capteurs. [7]

1.4.2. Les actionneurs [6]

Les actions des robots sont réalisées à l'aide d'actionneurs. Ce sont des organes qui transforment l'énergie qui leur est fournie en un phénomène physique utilisable comme des mouvements.

Les actionneurs les plus usuels sont :

- Des moteurs électriques rotatifs qui sont fréquemment associés à des réducteurs mécaniques à engrenages.
- Des vérins hydrauliques reliés par une tuyauterieà des pompes fournissant des pressions élevées.

Généralement, un actionneur peut être considéré comme un constituant d'un système mécanique (exemple : bras, patte, roue motrice...) et correspond à un degré de liberté.

(a) (b)

Fig.1. 5. (a) Moteur à courant continu, (b) Vérin hydraulique simple effet [8]

1.4.3. Partie commande [9]

La partie commande (cerveau) du robot permet d'analyser les données provenant des capteurs et d'envoyer les ordres relatifs aux actionneurs. La partie commande est matérialisée physiquement par un calculateur, tel que un microcontrôleur qui est un cerveau électronique spécialement conçu pour l'interagir avec des capteurs et des actionneurs.

1.5. Les types de robots [9]

1.5.1. Les humanoïdes

Catégorie la plus connue, en grande partie grâce à leur promotion faite par la science-fiction, elle regroupe tous les robots anthropomorphes, ceux dont la forme rappelle la morphologie humaine.

Ces robots ont généralement un torse, une tête, deux bras et deux jambes. On peut citer le robot Asimo de Honda. Parfois, certains de ces robots ne représentent qu'une partie du corps, comme le robot Nexi développé par le MIT.

Lorsqu'un robot anthropomorphe imite non seulement l'apparence physique, mais aussi les comportements humains, on l'appelle un androïde. Un parfait exemple d'androïde est l'Actroid-DER de la société Kokoro.

(a) (b)

Fig.1. 6. (a) Le robot ASIMO de Honda[10], (b) Le robot NEXI de MIT. [11]

1.5.2. Les robots industriels

La majorité de ces robots sont à base fixe. Quand la base n'est pas fixe, elle est généralement montée à un rail. On retrouve dans cette catégorie les robots de manipulation, type « Pick And Place », des robots soudeurs ou encore des robots de peintures. Cela représente la majorité des robots actuellement en état de service. On peut citer le robot SCARA, le robot anthropomorphe (vulgairement appelé bras 6-axes) ou encore le robot delta.

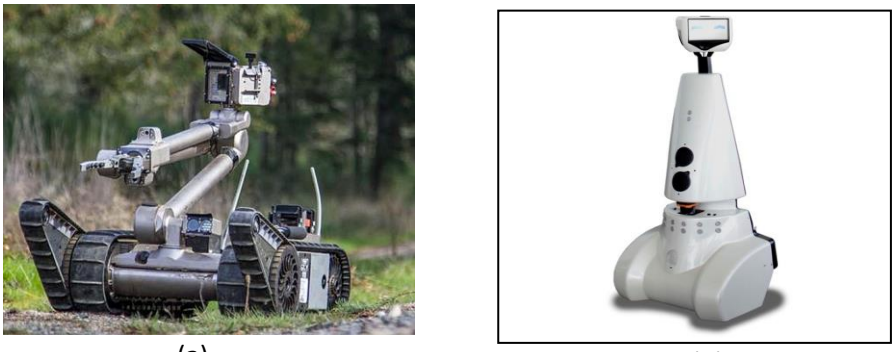
Fig.1. 7.Le SCARA d'EPSON robot. [12]

1.5.3. Les robots mobiles

Cette catégorie englobe tous les robots à base mobile, mais elle désigne de façon générale la sous-catégorie des robots mobiles à roues ; les autres robots étant généralement appelés par leur nom de catégorie correspondant à leur fonctionnalité.

Les robots mobiles à roues sont appelés en anglais UGV (Unmmaned Ground Vehicles). Cette catégorie regroupe les robots à base actionnée par des roues ou par des chenilles. Ces robots sont généralement exploités pour faire de l'exploration, on appelle ces robots des rovers (vagabonds en anglais). Les plus connus sont le Curiosity qui a été envoyé sur Mars par la Nasa, pour explorer et identifier le terrain martien, et le iRobot 510 Packbot qui est utilisé par l'armée américaine pour vérifier le terrain avant d'envoyer les soldats ou à déminer

des bombes et des mines anti personnelles. Ils sont également utilisés à des fins ludiques comme la plateforme de développement POB Robotics Suite ou pour l'assistance à la personne comme le robot Jazz de Gostai.



(a) (b)

Fig.1. 8. (a) L'iRobot 510 Packbot. [13], (b) Le robot de télé présence JAZZ. [14]

1.6. Conclusion

Dans ce chapitre, on a donné une idée sur la robotique en générale et mobile sans introduire la théorie et la partie technique et pratique, chose qu'on verra dans les deux prochains chapitres.

2.1. Introduction [15]

Les robots mobiles sont un axe majeur de la recherche actuelle. Ils sont présents, notamment dans les laboratoires des grandes universités et également dans les environnements industriels, militaires et de sécurité.

Les robots domestiques sont des produits de consommation, y compris les robots de divertissement et ceux qui effectuent certaines tâches ménagères telles que l'aspiration ou le jardinage. On peut estimer que les robots mobiles a à roues constituent le gros des robots mobiles. Historiquement, leur étude est venue assez tôt, suivant celle des robots manipulateurs, au milieu des années 70. Leurs faibles complexités en a fait de bons premiers sujets d'étude pour les roboticiens intéressés par les systèmes autonomes.

Cependant, malgré leurs simplicités apparentes (mécanismes plans, à actionneurs linéaires), ces systèmes ont soulevé un grand nombre de problèmes difficiles, un nombre de ceux-ci ne sont d'ailleurs toujours pas résolus, en particulier, au niveau de la reconnaissance et de prise de décisions.

2.2. Bref historique [16]

Dans les années 60, les recherches en électronique et l'apparition des transistors ont conduit au développement de robots plus complexes. Ainsi le *robotBeast* **(Fig.2.1.)** 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 blanc) 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 1969 avec *Shakey*. 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, considérée à l'époque 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 stéréo-visionpour la détection d'obstacles et la modélisation de l'environnement. En France, le robot Hilare est le premier robot construit au LAAS, à Toulouse.

Ces développements ont continué et l'arrivée sur le marché depuis les années 1990 de plateformes intégrées telles que le Pioneer de la société Mobile Robots a permis à de très nombreux laboratoires de travailler sur la robotique mobile et à conduire 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 restent difficiles et cruciaux, des laboratoires ont pu par exemple travailler sur des approches multi-robot, la problématique de l'apprentissage ou sur les problèmes d'interactions entre les hommes et les robots.

Fig2. 1.Robot Beast de l'université John Hopkins dans les années 1960. [17]

2.3. Définitions [16]

2.3.1. Définition

Un robot mobile est un système mécanique, électronique et informatique agissant physiquement sur son environnement en vue d'atteindre un objectif qui lui a été assignée.

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 a. Cette machine est polyvalente et capable de s'adapter à certaines variations de ses conditions de fonctionnement. Elle est dotée de fonction de perception, de décision et d'actions.

2.4. Classification des robots mobiles [18]

Les robots mobiles peuvent être classés par L'environnement dans lequel ils évoluent.

2.4.1. Les robots terrestres (ou domestiques)

Sont généralement appelés véhicules terrestres sans pilote. Ils sont le plus souvent à roues ou à chenilles, mais comprennent également des robots à pattes avec deux jambes ou plus (humanoïdes ou ressemblant à des animaux ou des insectes).

2.4.2. Les robots aériens

Sont généralement appelés véhicules aériens sans pilote.

2.4.3. Les robots sous-marins

Sont généralement appelés véhicules sous-marins autonomes.

2.4.4. Robots polaires

Conçus pour naviguer dans des environnements glacés, remplis de crevasses.

Aussi, Une classification est proposée dans la littérature qui définit le degré d'autonomie du robot mobile.

- Véhicule télécommandé par un opérateur.
- Véhicule télécommandé au sens de la tâche à réaliser.
- Véhicule semi-autonome réalisant sans l'aide de l'opérateur des tâches prédéfinies.
- Véhicule autonome qui réalise des tâches semi-définies.

(a)

(b)

Fig.2. 2.(a) Robot a pattes T-HEX.[19]. (b) Robot mobile DJI ROBOMASTER. [20]

2.5. Les robots à roues (à base mobile) [21]

2.5.1. Modes de locomotion

On peut distinguer deux fonctions pour les systèmes de locomotion :

- Appui vis-à-vis du milieu.
- Propulsion (Moteurs et mécanisme de déplacement).

Le mode de locomotion propre d'un robot diffère selon le type de roues et leur disposition.

2.5.2. Les types de roues [22]

2.5.2.1. Roues standard

❖ **Roue fixe**

1 degré de liberté.

- Rotation autour de l'axe de la roue.

❖ **Roue centrée orientable**

2 degrés de liberté.

- Rotation autour de l'axe de la roue.
- Rotation autour du point de contact.

❖ **Roue décentrée orientable**

3 degrés de liberté.

- Rotation autour de l'axe de la roue.
- Rotation autour du point de contact.
- Rotation autour du pivot.

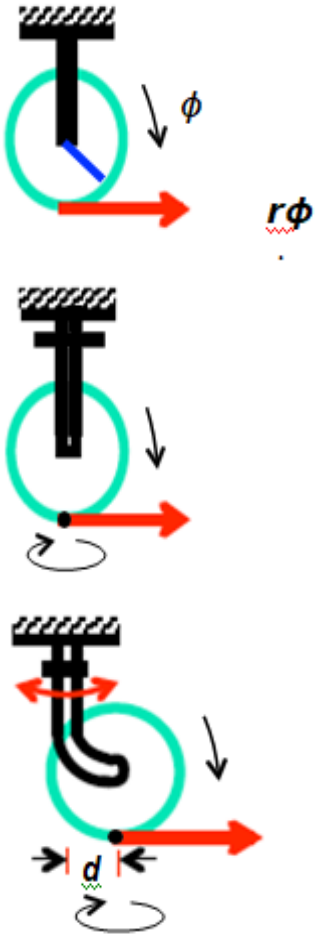


Fig.2. 3.De haut en bas. Roue : Fixe, Centrée orientable, Décentrée orientable. [22]

$R\varphi$: Vitesse tangentielle.

2.5.2.2. Roues omnidirectionnelles

- **Roues suédoises**

3 degrés de liberté.

- Rotation autour de l'axe motorisé de la roue.
- Rotation autour des roulettes transverses.
- Rotation autour du point de contact.

Roue sphérique

- Principe inverse de la souris PC.
- Réalisation ardue (difficile).

(a) (b)

Fig.2. 4. (a) Roue suédoise, (b) Roue sphérique.

2.6. Classes de robots à roues [23]

Il existe plusieurs classes de robots à roues déterminées, principalement par la position et le nombre de roues utilisées.

2.6.1. Robot unicycle

Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices.

Caractéristiques : Stable, Complexité mécanique faible, non-holonome.

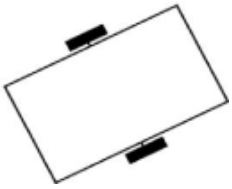


Fig.2. 5.Robot de type unicycle

2.6.2. Robot tricycle

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable.

Caractéristiques : Peu stable, Complexité mécanique faible, non-holonome.

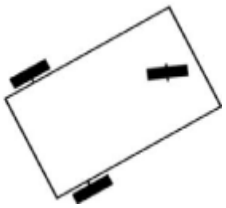


Fig.2. 6.Robot de type tricycle.

2.6.3. Robot voiture

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe.

Caractéristiques : Stable, Complexité mécanique modéré, non-holonome.

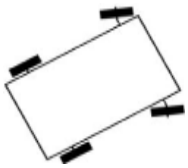


Fig.2. 7.Robot de type voiture.

2.6.4. Robot omnidirectionnel

Un robot omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral.

Caractéristiques : Stable, Complexité mécanique importante, holonome.

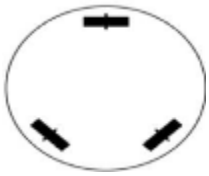


Fig.2. 8.Robot de type omnidirectionnel.

2.7. Les robots holonomes et non-holonomes [24]

Avant de voir le type d'un robot, il nous faut définir ce que sont les degrés de liberté d'une plate-forme mobile.

2.7.1. Degré de liberté (DLL)

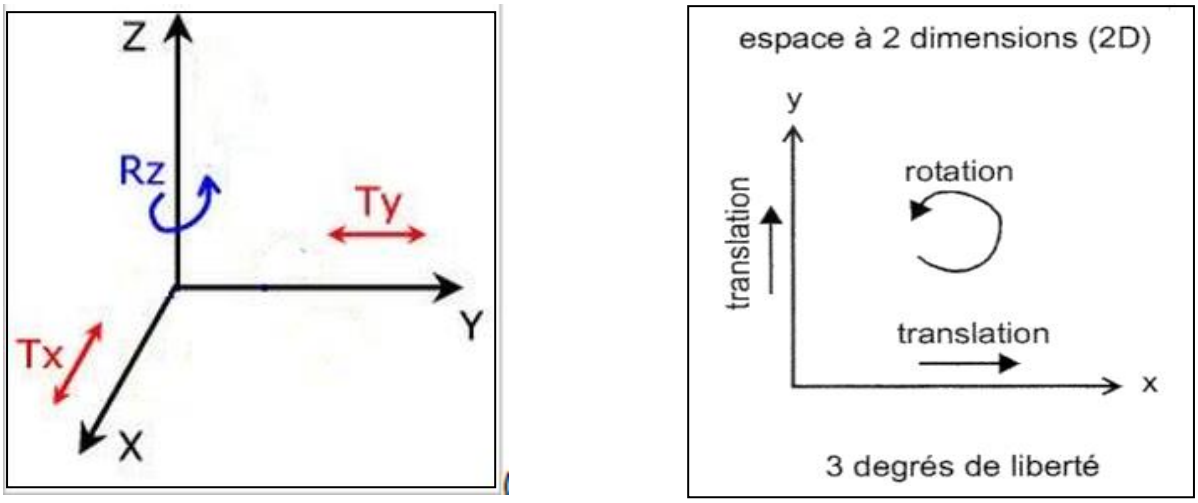
Le nombre de degrés de liberté d'un robot mobile est défini comme le nombre de mouvements indépendants que ce robot peut faire par rapport à un système de coordonnées déterminé.

2.7.2. Les robots holonomes

Un robot holonome ou omnidirectionnel est un robot qui possédant trois degrés de liberté :

- Une translation selon X, Avance/Recule.
- Une translation selon Y, Droite/Gauche.
- Une rotation selon Z, Droite/Gauche.

Le robot holonome est donc capable de se déplacer dans n'importe quelle direction quelque soit son orientation.



(a) (b)

Fig.2. 9.Axes : (a) vue de face. (b) vue de dessus.

Remarque :

Mécaniquement, on obtient ces trois degrés de liberté en utilisant trois roues dites "*holonomes*" (Roues sphérique, roues suédoise…) qui sont en faites des roues à galets. Ce sont des roues qui peuvent tourner librement sur un axe perpendiculaire à l'axe de rotation de la roue.

2.7.3. Les robots non-holonomes

Plate-forme mobile qui ne dispose que de 2 degrés de liberté sur un plan puisque les translations latérales sont impossibles à réaliser :

- Une translation Avance/Recule.
- Une rotation Droite/Gauche.

Un exemple typique de robot non-holonome est celui d'une voiture : en interdisant les glissements des roues, le vecteur de vitesse d'un point de l'essieu arrière d'une voiture est toujours tangent à l'axe de la voiture.

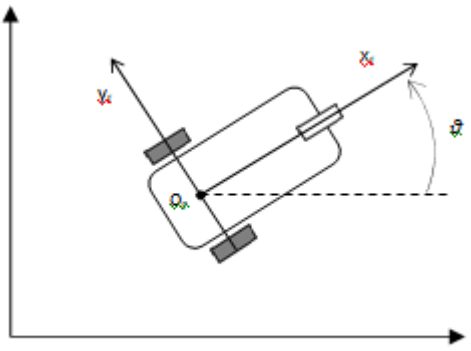


Fig.2. 10.Modèle de type tricycle à roues différentielles. [25]

2.8. Roulement avec ou sans glissement

La locomotion se fait grâce au frottement entre la roue du véhicule et le sol, et l'efficacité du mouvement dépend notamment du type de sol. [26]

Théoriquement, pour vérifier la condition de r.s.g. il faut réunir les hypothèses suivantes:

- Le contact entre la roue et le sol est ponctuel.
- La roue est indéformable, de rayon fixe r.
- Le sol est parfaitement plat.

En réalité, le contact sur le sol se fait sur une surface avec le pneu de la roue. Pour certaines méthodes de positionnement, le glissement au sol est la principale source d'erreur.

Contact continu avec le sol, essentiel pour l'**odométrie** d'un robot qui sert à déterminer la position relative d'un robot par rapport à son point de départ, à partir de la mesure dunombre de tours parcourus par chaque roue. [22]

2.8.1. Relation théorique entre les deux vitesses linéaire et angulaire [$v = f(\omega)$]

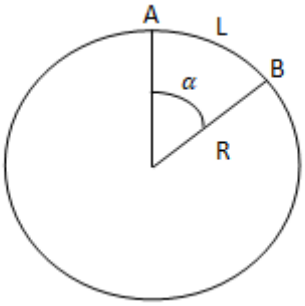


Fig.2. 11.Mouvement circulaire autour d'un axe fixe

$v = \omega . R$

Avec :

L : Longueur.

ω : Vitesse angulaire.

v : Vitesse linéaire.

α : Angle de rotation.

2.9. Modèle cinématique en posture d'un robot de type unicycle

On désigne par unicycle un robot actionné par deux roues indépendantes et possédant éventuellement un certain nombre de roues folles assurant sa stabilité. Les roues folles n'interviennent pas dans la cinématique, dans la mesure où elles ont été judicieusement placées. [26]

2.9.1. Choix de la commande

En ce qui concerne la commande ; si l'on se contente de traiter le cas cinématique, on peut considérer que celle-ci est donnée par les vitesses de rotation des roues. Ceci étant, on préfère généralement exprimer cette commande par la vitesse longitudinale du robot et sa vite de rotation.

$$\begin{cases} \dot{x} = v \cos\theta \\ \dot{y} = v \sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{2.5}$$

$\boldsymbol{v}$:Vitesse linéaire ; $\boldsymbol{\omega}$: Vitesse angulaire.

$$\begin{cases} v_R = r\dot{\varphi}_R = (\rho + \frac{b}{2})\omega \\ v_L = r\dot{\varphi}_L = (\rho - \frac{b}{2})\omega \end{cases} \tag{2.6}$$

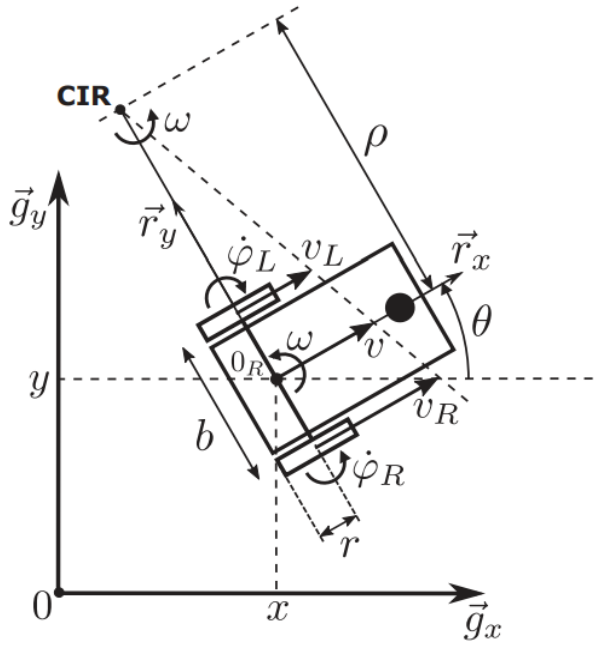


Fig.2. 12.Modèle cinématique en posture d'un robot de type. [27]

Tab2. 1.Paramètres du robot type unicycle. [27]

Symboles	Unités	Paramètres
$\dot{\varphi_R}$, $\dot{\varphi_L}$	Rad/s	Vitesses angulaires des roues droite et gauche
v_R, v_L	m/s	Vitesses linéaires des roues droite et gauche
(x, y)	m	Position du robot dans le plan
θ	rad	Orientation du robot dans le plan
$(\dot{x}, \dot{y})$	m	Vitesse du robot dans le plan
v	m/s	Vitesse linéaire du robot
ω	Rad/s	Vitesse angulaire du robot
r	m	Rayon des roues droite et gauche
b	m	Distance entre les 2 points de contact au sol des roues droite et gauche
ρ	m	Rayon de courbure de la trajectoire circulaire du robot

De (2.5) – (2.6), on obtient :

$$\begin{cases} (1) + (2) \Leftrightarrow r(\dot{\varphi_R} + \dot{\varphi_L}) = 2\,\rho\omega \\ (1) - (2) \Leftrightarrow r(\dot{\varphi_R} - \dot{\varphi_L}) = b\omega \end{cases} \quad (2.7)$$

Comme $v = \rho\omega$, la première équation du système précédent peut se réécrire :

$$r(\dot{\varphi_R} + \dot{\varphi_L}) = 2v \quad (2.8)$$

Il y'a en effet équivalence entre les deux représentations. D'une part on a :

$$\begin{cases} v = \frac{v_R + v_L}{2} = \frac{r(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \\ w = \frac{r(\dot{\varphi}_L - \dot{\varphi}_R)}{b} \end{cases} \quad (2.9)$$

Ces deux dernières équations représentent ce qu'on appelle le modèle cinématique inverse du robot.

En inversant ces relations, on obtient alors les équations du modèle cinématique direct. Les équations permettent de passer des vitesses linéaires et angulaires du robot aux vitesses linéaires des roues sont donc donnés par :

$$\begin{cases} v_R = v + \frac{b\omega}{2} \\ v_L = v - \frac{b\omega}{2} \end{cases} \quad (2.10)$$

Pour arriver à la vitesse de rotation des roues (rad/s), on divise simplement par r :

$$\begin{cases} \dot{\varphi}_R = \frac{v}{r} + \frac{b\omega}{2r} \\ \dot{\varphi}_L = \frac{v}{r} - \frac{b\omega}{2r} \end{cases} \quad (2.11)$$

2.9.2. Modes de locomotion

Si :

$\dot{\boldsymbol{\varphi}}_{\boldsymbol{R}} = -\dot{\boldsymbol{\varphi}}_{\boldsymbol{L}}$ **:** Le robot tourne sur lui-même.

$\dot{\boldsymbol{\varphi}}_{\boldsymbol{R}} = \dot{\boldsymbol{\varphi}}_{\boldsymbol{L}}$ **:** Le robot se déplace en ligne droite.

Remarque :

L'utilisation de ces deux seuls modes de locomotion, bien que limitée, permet de découpler les mouvements et de fournir une solution simple pour amener le robot d'une posture a une autre. C'est sans doute là une des raisons du succès de ce type de robots. Pour élaborer une stratégie plus fine de déplacement, il est cependant intéressant de savoir comment la posture du robot est reliée à la commande de ses roues.

2.10.Navigation autonome [18]

La navigation est une étape très importante en robotique mobile. Bien entreprise, elle permet une large autonomie à un robot mobile.

On s'intéresse plutôt à la localisation. On a choisi de réaliser un système de localisation permettant au robot de se mouvoir de manière autonome (sans intervention humaine) dans un milieu plan partiellement connu. Cette étude d'un cas simple de robotique fera ressortir les difficultés posées par la mise au point de mécanismes d'automatisation nettement plus sophistiqués, à savoir : le choix des capteurs, leur positionnement et la localisation du mobile.

2.10.1. La génération de plan

Consiste à établir la manière dont le robot se déplace par rapport à des connaissances à priori (statiques) ou obtenues en cours d'évolution (dynamiques). La génération de plus repose sur trois concepts :

1. **La stratégie de navigation.**
2. **La modélisation de l'espace :** La connaissance du milieu dans lequel évolue le robot mobile n'est établie en général qu'après avoir effectué une campagne de mesure de l'ensemble des éléments constituant l'environnement.
3. **La planification de trajectoire :** Connaissant la carte et la position actuelle du robot, prévoir les mouvements à effectuer afin de rejoindre un but fixe.

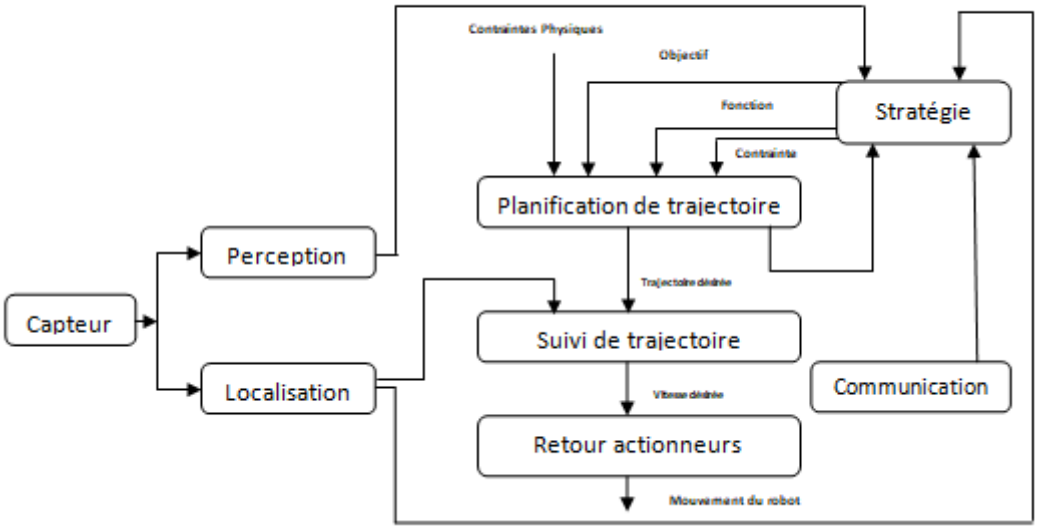


Fig.2. 13.Architecture logicielle d'un robot mobile à roues

2.11.Localisation [18]

La localisation instantanée est un des points les plus importants et les plus délicats des robots mobiles. Elle permet de définir le positionnement :

- Du robot dans l'environnement.
- D'éléments particuliers de l'environnement par rapport au mobile.

La localisation consiste à définir la position en termes de coordonnées d'un point du mobile par rapport à un référentiel de base.

2.11.1. Localisation relative (Par odométrie)

Estimation de la position courante du robot par rapport à sa position initiale.

- **Odométrie**

La technique de l'odométrie est la plus utilisée pour la localisation des robots mobiles à roues. Elle permet de déterminer la position et le cap d'un véhicule par intégration de ses déplacements élémentaires, et ce, par rapport à un repère lié à sa configuration initiale.

$$x(t) = \int_0^t \dot{x}(t).\,d\tau, \qquad y(t) = \int_0^t \dot{y}(t).\,d\tau\ ,\theta(t) = \int_0^t \dot{\vartheta}(t).\,d\tau \qquad (2.12)$$

2.11.2. Localisation absolue (Télémétrie)

Localisation en se référant à des points de référence de l'environnement.

- **Télémétrie**

La technique de la télémétrie permet de positionner les objets présents dans la scène par rapport au robot.

Le capteur renvoie généralement l'angle de gisement (direction de la mesure) et la distance au corps ayant réfléchi l'onde émise.

2.12. Architecture de navigation (contrôle) [28]

La navigation d'un robot mobile consiste à trouver un mouvement qui amène le robot d'une configuration initiale à une configuration finale. On peut distinguer trois approches de navigation.

2.12.1. Approche délibérative (Planification-exécution)

L'approche délibérative, dite aussi *hiérarchique*, consiste à relier d'une manière séquentielle les trois actions : percevoir, planifier et agir [Chatila et Laumond, 1985].

2.12.2. Approche réactive

L'approche réactive a été développée pour faire face aux limites de l'approche délibérative dans des environnements dynamiques et inconnus [Seraji et Howard, 2002]. C'est une approche généralement utilisée en temps réel.

Elle ne nécessite pas de modéliser l'environnement ni de planifier une trajectoire à l'avance. Il s'agit d'agir directement après la perception [Ye et Wang, 2000]. Cette architecture génère des commandes de contrôle basées sur l'environnement actuellement perçu.

2.12.3. Approche hybride

Cette approche combine l'aspect de temps réel et la rapidité des réponses retrouvées dans l'approche réactive dans des environnements dynamiques ou inconnus, et l'efficacité de la planification de l'approche délibérative.

L'architecture de contrôle hybride a été développée pour présenter de nouvelles approches pour obtenir des systèmes de contrôle de supervision qui utilisent des architectures de contrôle réactives et délibératives.

2.13.Conclusion

Dans ce chapitre, on a présenté la robotique mobile, sa définition, ses nombreuses classifications, ainsi que son modèle cinématique et les différentes approche de navigation et de localisation. Dans les chapitres suivant on verra la partie pratique et la réalisation de notre robot mobile.

3.1. Introduction

Généralement, dans la robotique, pour décrire un problème on se réfère souvent au terme de commande. Il s'agit bien évidemment d'un problème de commande dans le cas où l'on cherche à élaborer les lois qui permettent au robot de suivre une trajectoire et ou d'atteindre un point (but) précis.

Dans ce 3éme chapitre, on présentera l'architecture mécanique du robot mobile, ainsi que les différentes approches de contrôle, en se basant sur quelques notions théoriques essentielles dans le domaine de la robotique mobile.

3.2. Structure mécanique générale du robot

On peut résumer la structure de notre robot avec le schéma de Fig.3.1

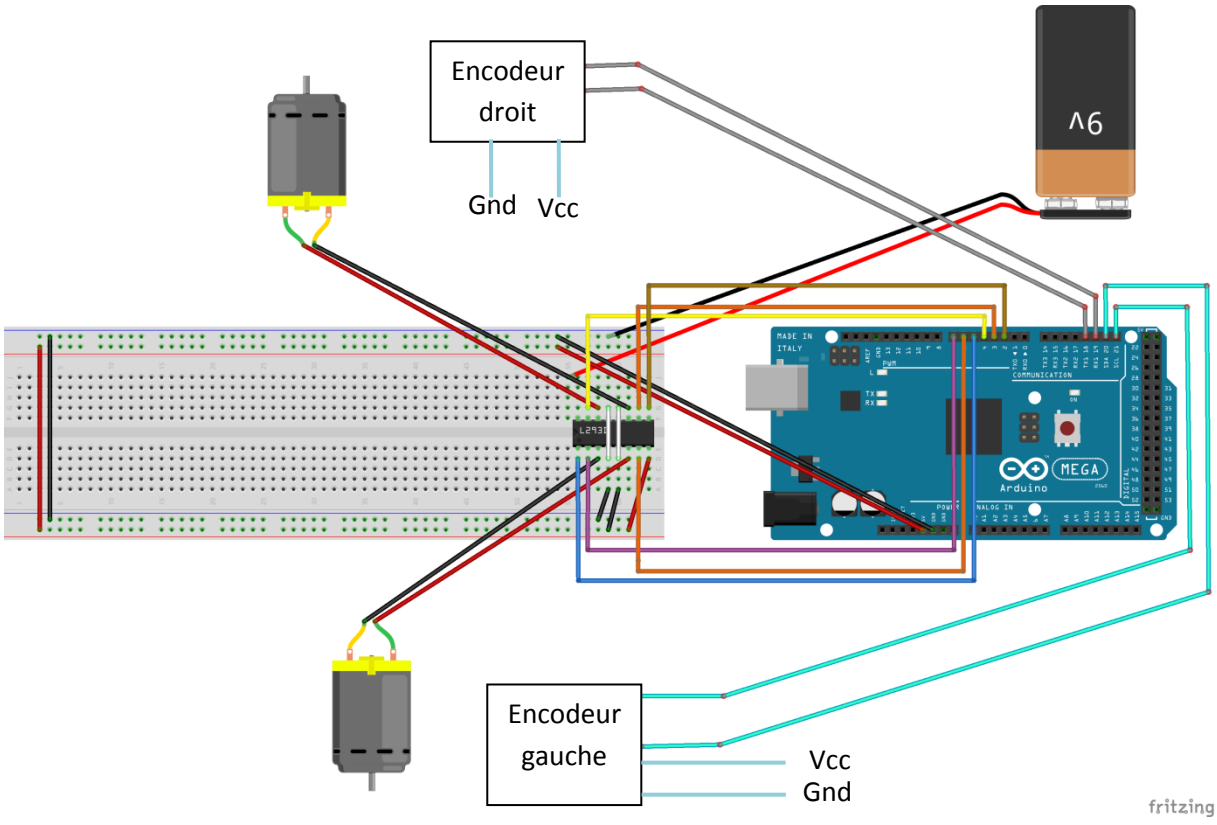


Fig.3. 1. Structure générale du robot mobile

3.3. Matériel utilisé pour la composition du robot

On peut distinguer 2 parties fonctionnelles essentielles : la partie électronique et la partie mécanique.

3.3.1. La partie électronique

C'est la partie la plus importante de notre travail. Pour que les moteurs s'actionnent, le robot se déplace et pour commander le tout, on doit s'appuyer sur circuit programmable. Pour cela, on a utilisé une carte Arduino ATMEGA 2560.

3.3.1.1. Arduino

Le leader des cartes de développement et des logiciels libres, Arduino Corporation, est connu pour sa carte de haute qualité, appelée Arduino Uno. La carte ArduinoMega 2560, bien qu'elle soit un peu moins connue, est la grande sœur, plus rapide, de la carte Uno. La carte ArduinoMega 2560, dont les capacités équivalent à quatre cartes Uno combinées, comporte un microcontrôleur et est construite autour du puissant microprocesseur ATmega2560. Cette carte robuste résiste à presque tout [29]. Ses caractéristiques sont données par Tab.3.1.

Tab3. 1.Brochage et caractéristiques de la carte Arduino. [30]

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont **8 KB** sont utilisés par le bootloader
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

- **Le microcontrôleur :** C'est le cerveau de notre carte. Il a une mémoire, un processeur, des interfaces avec le monde extérieur. Les microcontrôleurs ont des

performances réduites, mais sont de faible taille et consomment peu d'énergie, les rendant indispensables dans toute solution d'électronique embarquée. [31]

- **L'alimentation :** Pour fonctionner, la carte Arduino a besoin d'alimentation. Le microcontrôleur fonctionne à 5V, La carte peut être fournie avec une tension de 5 V via un port USB ou une alimentation externe entre 7V et 12V. Ensuite, le régulateur se chargera de réduire la tension à 5V pour que la carte fonctionne correctement. [33]
- **La connectique :** A part une LED sur la broche 13, la carte Arduino ne possède pas de composants (résistances, diodes, moteurs...) qui peuvent être utilisés pour un programme. Il est nécessaire de les rajouter. Mais pour cela, il faut les connecter à la carte. C'est là qu'interviennent les connecteurs, aussi appelés broches. [31]

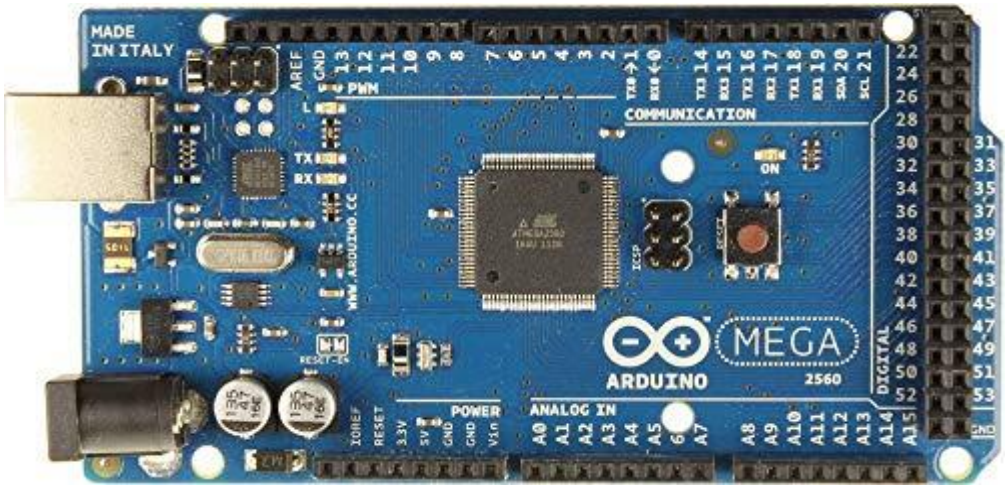


Fig.3. 2.Carte ArduinoMega 2560 [29]

Programmation en Arduino

Un programme est une liste d'instructions qui est exécutée par un système.

3.3.1.1.1. Le logiciel arduino

- **Un logiciel :** C'est Un programme informatique exécuté sur un ordinateur.
- **Un compilateur :** En informatique, ce terme désigne un logiciel qui est capable de traduire un langage informatique (programme) utilisant un langage prédéfinie.

Fig.3. 3.Traduction du programme en langage machine

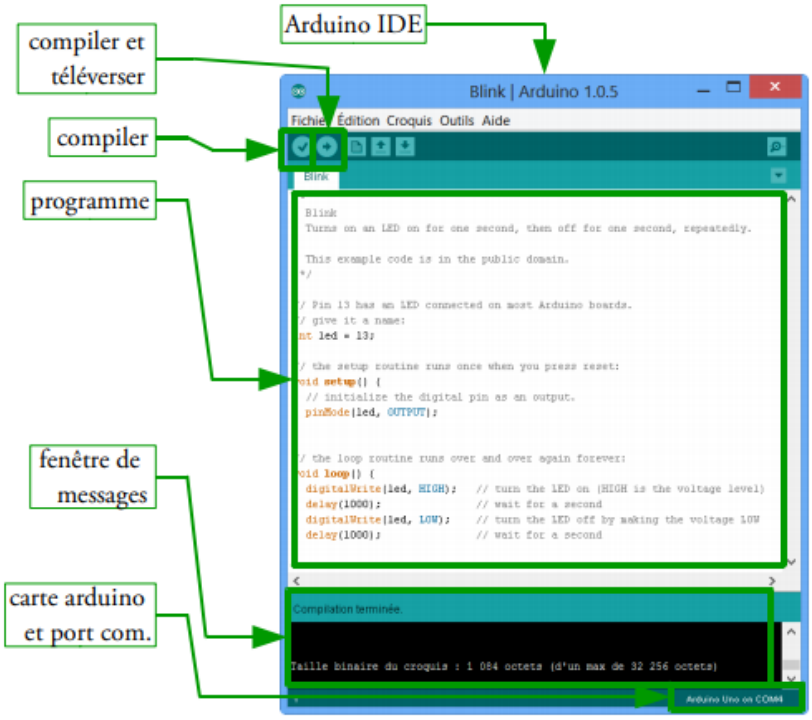


Fig.3. 4.Environnement de développement Arduino

3.3.1.1.2. Structure d'un programme sous arduino

Les programmes Arduino peuvent être divisés en 3 parties : la Structure, les Valeurs (variables et constantes) et les Fonctions. La Structure d'un programme Arduino est obligatoirement composée de ces deux fonctions principales :

- La fonction **setup() :** C'est la fonction d'initialisation d'Arduino. C'est la première qui sera activée au démarrage de l'Arduino. Ici s'effectue la configuration de divers objets, protocoles de communication, configuration d'entrée / sortie, etc.
- La fonction **Loop() :**La fonction loop() est la 2ème fonction à être lancée sur Arduino, juste après la fonction setup().

Cette fonction a la particularité, comme son nom l'indique, d'être une boucle (*loop* en anglais). En effet, cette fonction reboucle sur elle-même. Lorsque la dernière ligne de la fonction loop() sera atteinte par le programme, ce dernier remontera à la première ligne.

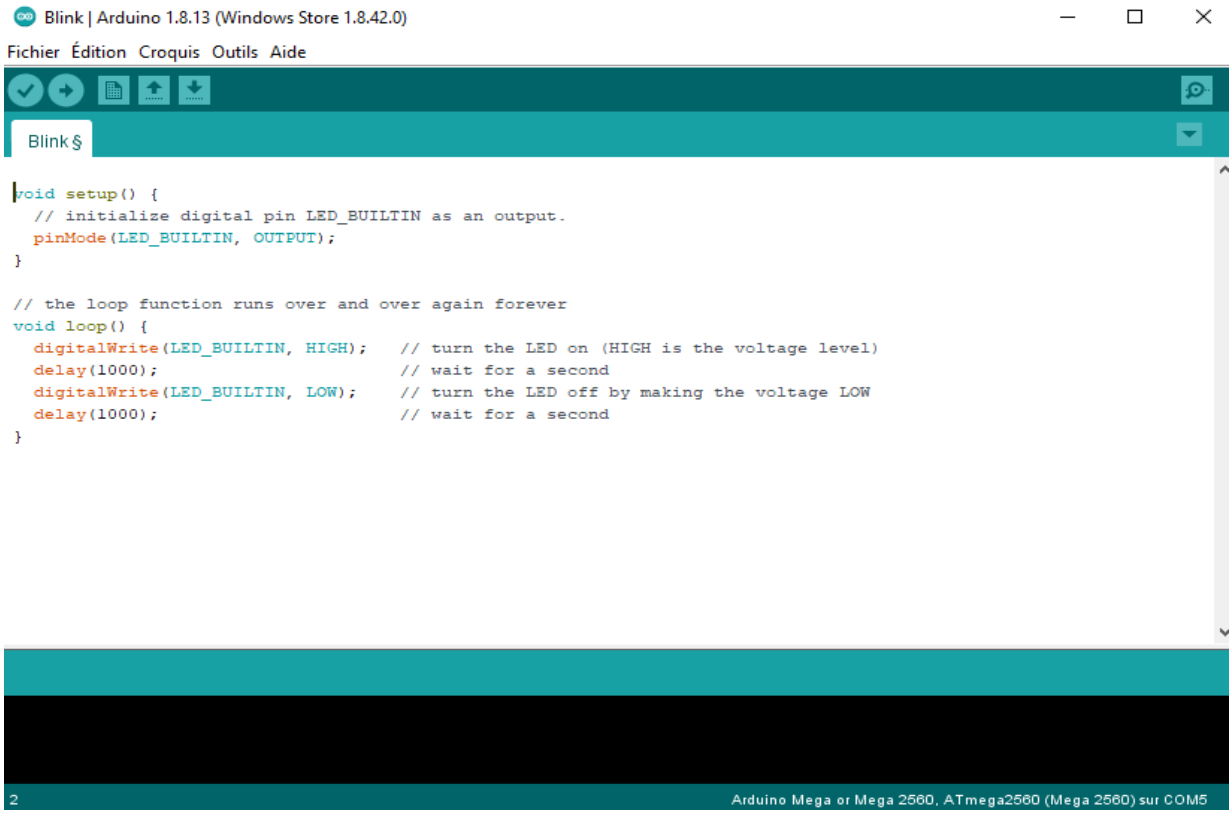


Fig.3. 5.Structure de base d'un programme Arduino

3.3.1.2. ShieldArduino L293D [32]

Basé sur un pilote de moteur le circuit intégré L293D qui est conçue pour fournir des courants de commande bidirectionnels allant jusqu'à 1,2 A chaque pont, avec une protection contre les coupures thermiques à des tensions de 4,5 V à 36 V.

Ce moteur shield contient deux L293D, ce qui permet de piloter 4 moteurs à partir des deux circuits intégrés contenus sur cette plateforme. **Le L293D** est un circuit intégré monolithique, à haut voltage, grand courant, 4 canaux pilotes. Le L293D est aussi connu pour être une sorte de Pont-H.

Fig.3. 6.Le circuit L293D. [33]

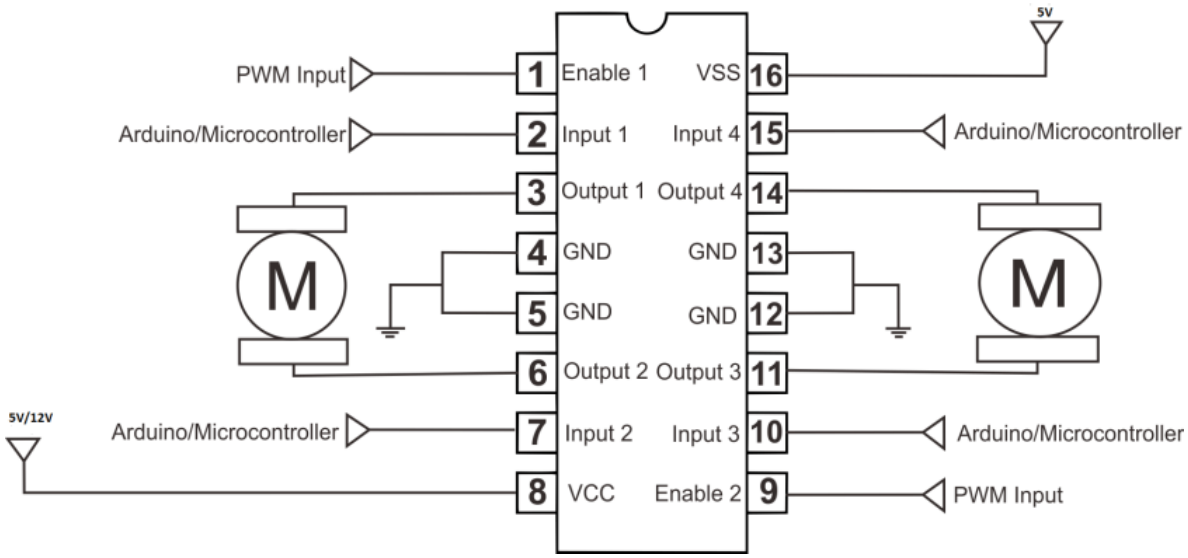


Fig.3. 7.Signification de chaque pin du circuit intégré l293d [34]

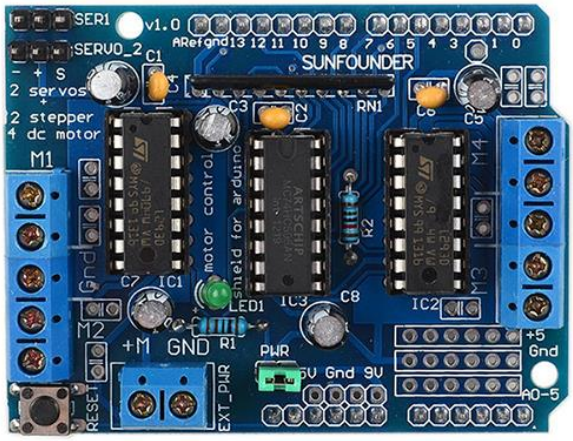


Fig.3. 8.Shield moteur L293D [35]

- **Caractéristiques techniques [35]**

- 2 interfaces pour **servomoteurs** 5V, connectés au timer haute résolution de l'Arduino.
- Peut piloter **4 moteurs à courant continu** DC, ou **2 moteurs pas à pas**, ou **2 servo** à la fois.
- Jusqu'à 4 moteurs DC bidirectionnels avec sélection de la vitesse individuelle (sur 8 bit).
- Jusqu'à 2 moteurs pas à pas (unipolaire ou bipolaire) avec une seule bobine, double bobine, ou demi-pas.

- 4 ponts en H (H-Bridges).
- Fournit **0,6 A** par pont (1.2A en courant de crête) avec protection thermique.
- Pilotage des moteurs à courant continu de **4.5V à 36V**.
- Des résistances pull down désactivent les moteurs au cours de la mise sous tension.
- Bouton de réinitialisation (Reset).
- 2 interfaces d'alimentation pour séparer la partie logique de la partie puissance (moteurs).

- Compatible avec les cartes ArduinoMega, Diecimila&Duemilanove.

3.3.1.3. Pont en H [36]

Le pont en H est une structure électronique servant à contrôler la polarité aux bornes d'un dipôle. Il est composé de quatre éléments de commutation généralement disposés schématiquement en une forme de H d'où le nom. Les commutateurs peuvent être des relais, des transistors, ou autres éléments de commutation en fonction de l'application visée.

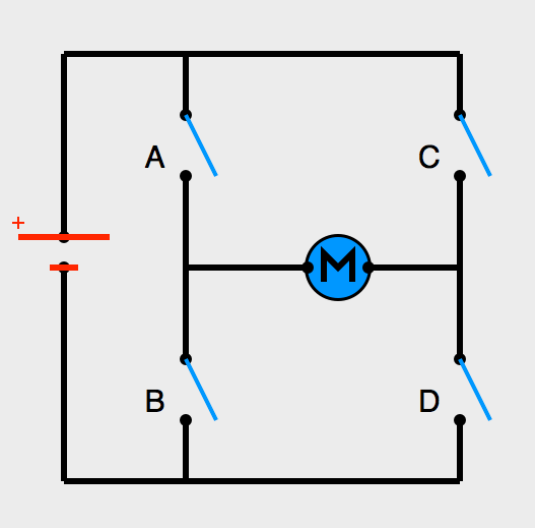


Fig.3. 9.Forme générale d'un pont en H [36]

- **Principe de fonctionnement:**

Les commutateurs fonctionnent par paires. A est associé à D et B est associé à C. Dans **Fig.3.9**, puisque tous les interrupteurs sont ouverts (ils ne laissent pas passer le courant), rien ne se passe. Le moteur est à l'arrêt.

En générale : (Explication illustré sur **Fig.3.10.**)

- A et D fermés, B et C ouverts $\Rightarrow \boldsymbol{U_M = +U}$
- A et D ouverts, B et C fermés $\Rightarrow \boldsymbol{U_M = -U}$

Avec :

$\boldsymbol{U_M}$ **:** Tension aux bornes du moteur.

NB : Les signes (+) et (-) indiquent le sens de rotation du moteur.

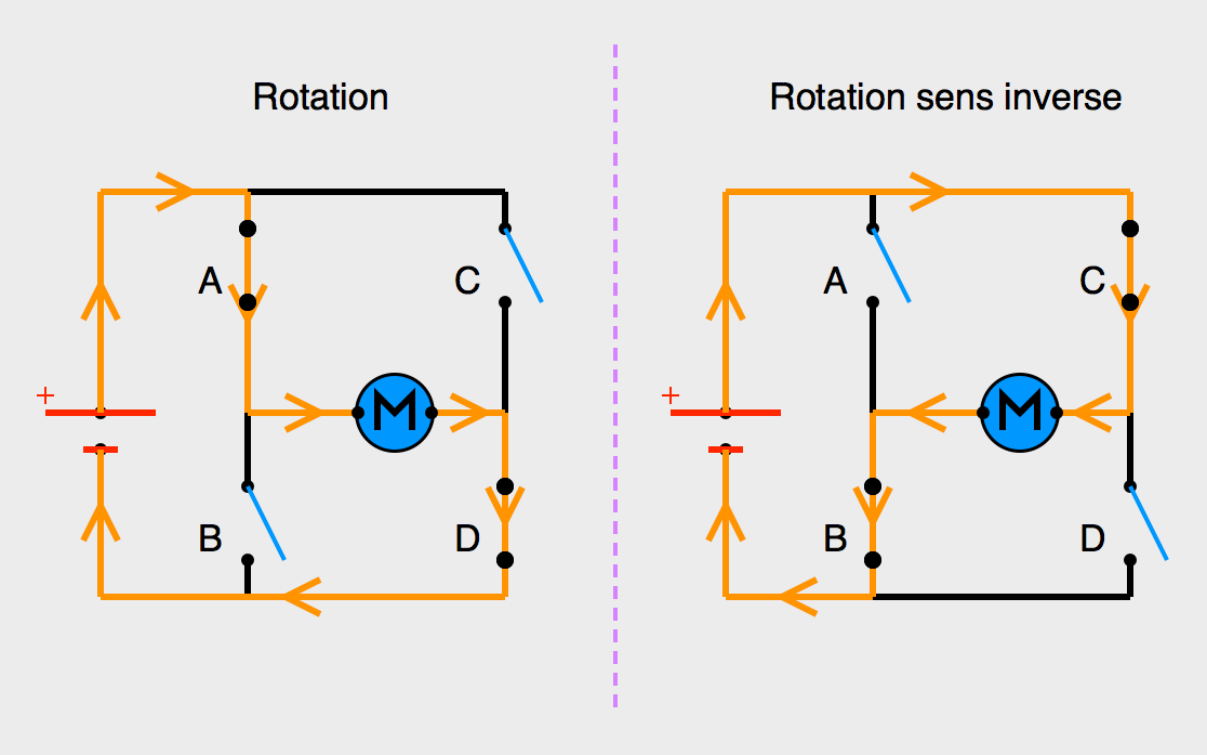


Fig.3. 10.Sens de rotation du moteur en fonction de l'état des interrupteurs [36]

3.3.1.4. Encodeur rotatif optique [37]

Un encodeur optique (parfois appelé roue d'encodeur optique) est un composant électromécanique qui peut prendre de nombreuses formes différentes. Il ressemble beaucoup à un potentiomètre. Mais contrairement à ce dernier, le mouvement de l'arbre du codeur ne se limite pas à la gauche ou à la droite: il peut tourner indéfiniment dans les deux sens.

Fig.3. 11.Encodeur rotatif optique. [38]

Un encodeur permet d'obtenir une information en quasi-temps réel sur la position et vitesse du moteur et donc d'appliquer des corrections (on parle de **boucle de contrôle**) si le comportement du moteur n'est pas conforme à l'ordre qu'on lui a transmis. Ce qui arrive fréquemment en fonction de la pente, de la capacité de la roue à agripper sur le sol (la roue tourne plus vite si elle patine) et de la charge utile.

- **Principe de fonctionnement [37] :**

Le codeur optique dispose d'un disque avec de nombreuses fenêtres (le nom technique est tic).Ces fenêtres peuvent être éclairées. La LED du codeur émet un faisceau de lumière qui tente de traverser le disque. Lorsque la fenêtre apparaît, le faisceau passe et est capté par le phototransistor. Les segments opaques bloquent la lumière alors que les segments transparents la laissent passer. Ceci génère des impulsions d'onde carrée qui peuvent ensuite être interprétées comme position ou mouvement.

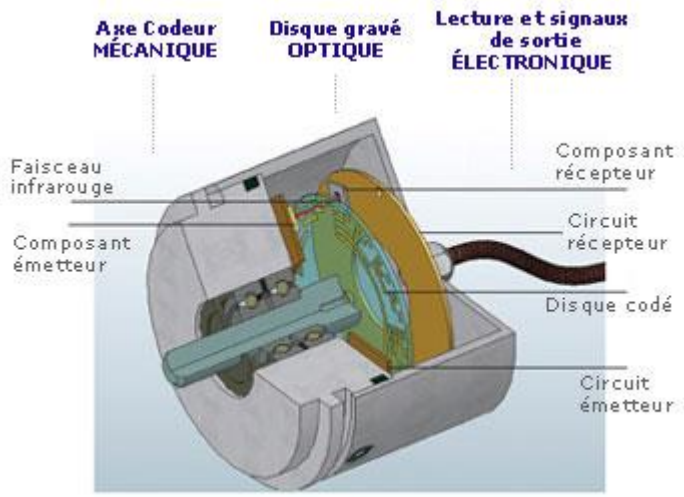


Fig.3. 12.Architecture interne d'un encodeur rotatif optique. [39]

Pour déterminer le sens de rotation de l'axe, on voit d'abord la phase A puis la phase B ou l'inverse. Le chronogramme de Fig.3.13 montre comment on peut utiliser les informations générées par les deux phases.

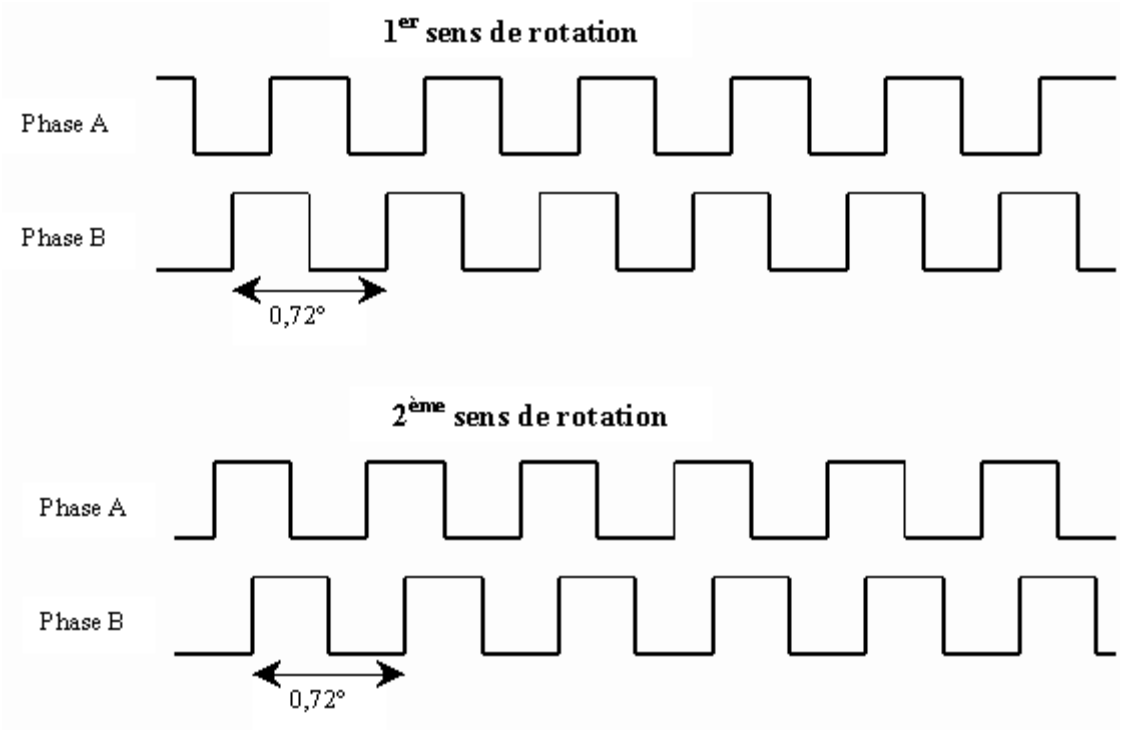


Fig.3. 13.Chronogramme du fonctionnement d'un encodeur. [40]

NB : Pour 500 impulsions, chaque alternance "noir transparent" représente : 360° / 500 = 0,72°.

- **Programmation d'un encodeur sous arduino**

Au niveau de la partie déclarative, on commence par déclarer les constantes utiles et on précise les broches utilisées pour chaque chaine de l'encodeur.

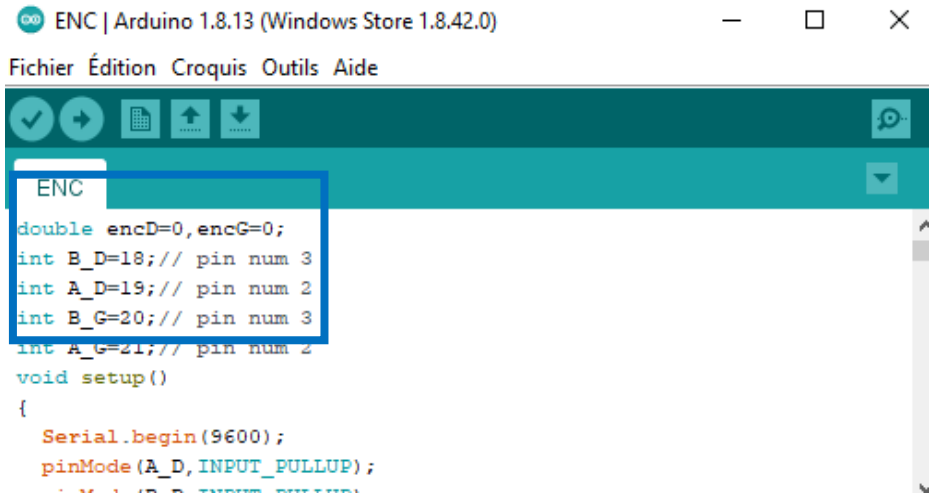


Fig.3. 14.Partie déclaration des pins

Au niveau de la fonction setup, on configure les broches ou sont connecté les chaines de l'encodeur en entrés. La configuration s'effectue en utilisant la fonction pinMode**(broche, Mode)**. On peut utiliser aussi la résistance interne de la carte arduino comme sur la figure (**Fig3.15.**).

Dans la même partie, on a utilisé ***attachInterrupt(pin, fct, mode)***,qui spécifie la fonction à appeler lorsqu'une interruption externe se produit. Elle Remplace toutes les autres fonctions qui accompagnent cette interruption. Le mode CHANGE déclenche l'interruption chaque fois que la broche change de valeur.

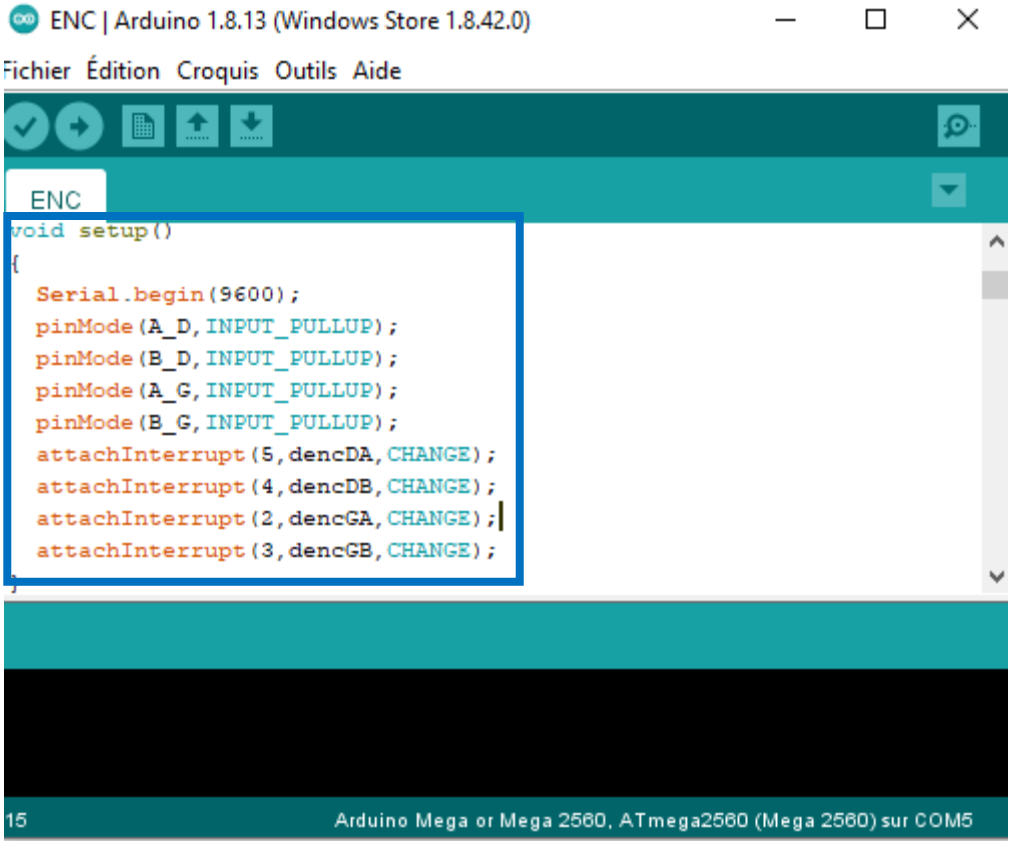


Fig.3. 15.Fonction setup du programme.

Le nombre d'impulsions générées par l'encodeur s'obtient en injectant une fonction précise pour chaque chaîne du capteur optique.

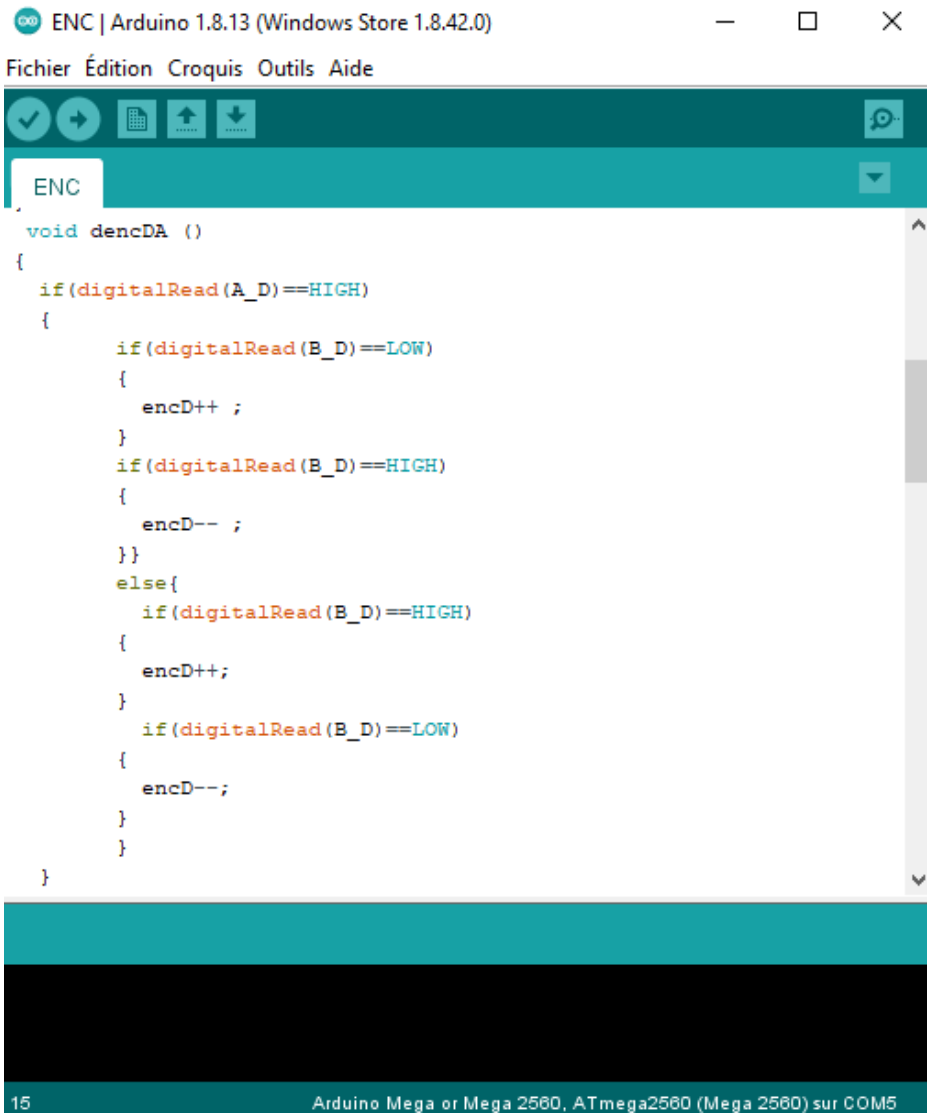


Fig.3. 16.Fonction du calcul d'impulsions.

3.3.2. La partie mécanique

Le Choix de structure est souvent effectué parmi un panel de solutions connues et pour lesquelles les problèmes de modélisation, planification et commande sont résolus. Choix classique des actionneurs et de l'alimentation : moteurs électriques à courant continu avec ou sans collecteur, alimentés par des convertisseurs de puissance fonctionnant sur batterie.

On peut dire que l'aspect mécanique est l'une des problématiques qu'on retrouve souvent dans le monde la robotique.

3.3.2.1. Le châssis

En robotique, Le châssis est une plaque qui représente l'ossature du véhicule sur laquelle tous les composants du robot sont placés et équilibrés.

Fig.3. 17.Châssis de notre robot.

3.3.2.2. La roue bille

La propulsion du robot se fera pas les deux moteurs grâce aux roues placées sur leurs tiges, la roue bille sera donc placée à l'avant.

La roue bille permettra de maintenir le robot en équilibre par rapport aux roues arrières et permettra le déplacement facile du véhicule dans n'importe quelle direction.

Fig.3. 18.Roue bille.

3.3.2.3. Moteur à courant continu avec réducteur (MCC/GEARBOX) [41]

C'est est un convertisseur d'énergie, totalement réversible, qui peut fonctionner soit en moteur, convertissant de l'énergie électrique en énergie mécanique, soit en génératrice, convertissant de l'énergie mécanique en énergie électrique. Dans les deux cas, un champ magnétique est nécessaire aux différentes conversions. Donc c'est un convertisseur électromécanique.

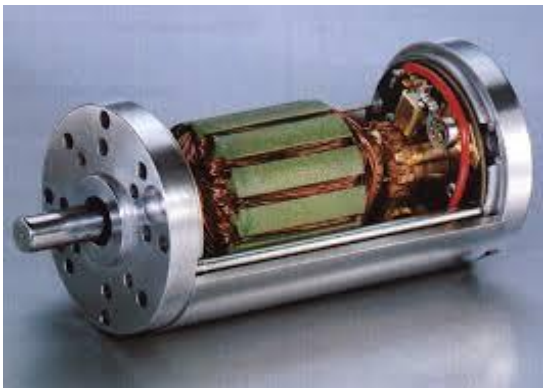


Fig.3. 19.Moteur à courant continu [42]

- **Principe de fonctionnement [43]**

Le moteur à courant continu se compose de:

- Inducteur ou du stator.
- Induit ou du rotor.
- Collecteur et des balais.

Un moteur à courant continu est constitué de deux parties électriques : le stator et le rotor. Lorsqu'on alimente le moteur, il se crée une interaction magnétique qui met le moteur en mouvement. Lorsqu'on inverse le sens de la tension qui alimente le moteur, il tourne en sens inverse.

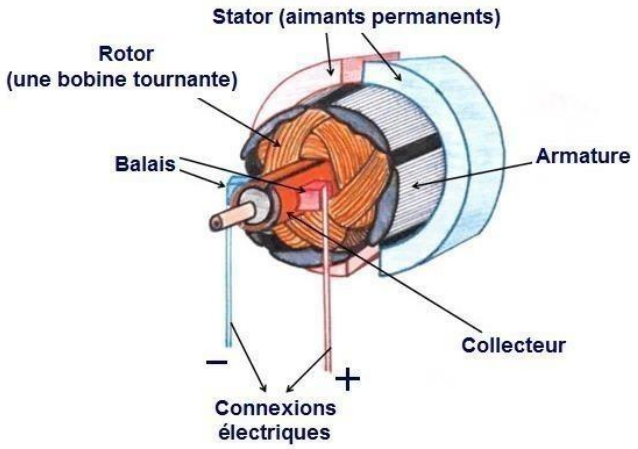


Fig.3. 20.Architecture interne d'un moteur à courant continu.

Définitions :

- **Inducteur (stator) :** La partie fixe du moteur (*statique* = qui ne bouge pas).
- **Induit (rotor) :** la partie en *rotation* du moteur. C'est lui qui tourne. Il est constitué du bobinage induit.
- **Balais :** Les balais assurent le passage du courant électrique entre l'alimentation et les bobinages de l'induit sous forme d'un contact par frottement.

Fig.3. 21.(a) Stator[44], (b) Rotor[45], (c) Balais.

Force de Laplace :

Une partie du conducteur traversée par un courant et placée dans le champ magnétique est soumise à une force électromagnétique appelée force de Laplace.

- **Principe de fonctionnement**

La direction de la force de Laplace F est donnée par la convention appelée règle des trois doigts, qui détermine la direction de F.En théorie :

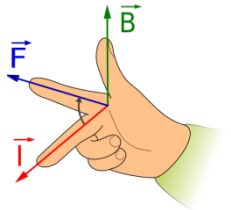


Fig.3. 22. Règle des 3 doigts de LAPLACE

$$\boldsymbol{F = I.B.l \sin \alpha} \qquad (3.1)$$

Avec :

$\boldsymbol{I}$ **:** Intensité (A).

$\boldsymbol{l}$ **:** Longueur(m).

$\boldsymbol{B}$**:** Champs magnétique (T).

$\boldsymbol{\alpha}$: L'angle formé par $\vec{\boldsymbol{B}}$par rapport au conducteur.

Programmation d'un moteur sous Arduino :

Au niveau de la partie déclarative, on commence par ajouter la bibliothèque approprié aux moteurs à courant continu **<AFMotor.h>** et on déclare les constantes utiles et on précise les broches utilisées.

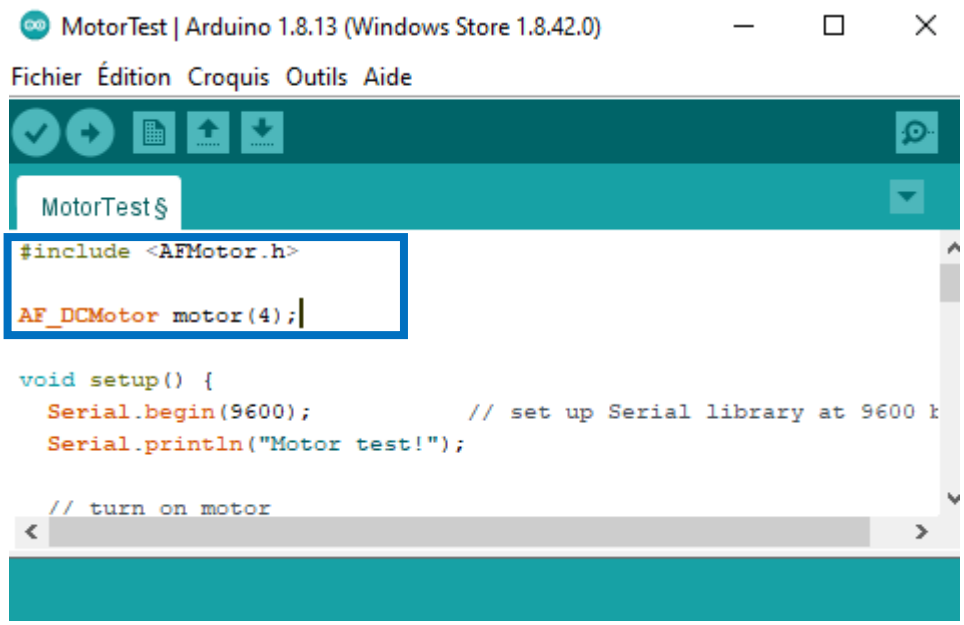


Fig.3. 23.Bibliothèque et configuration de pin

Au niveau de la fonction setup, on va initier la vitesse du robot et le mettre en arrêt (mise à zéro) en utilisant deux fonctions issues de la bibliothèque <AFMotor.h>.

Fig.3. 24.Initialisation de la vitesse et état du moteur.

Au niveau de la fonction *loop()*, on va réaliser un test simple de rotation du moteur dans un sens et dans l'autre, avec incrémentation de la vitesse jusqu'à remise à zéro (répétition en boucle infini).

Fig.3. 25.Exemple rotation du moteur.

Pour conclure la partie mécanique, l'ensemble des éléments a été assemblé a l'aide de plaques d'aluminium, ce qui nous a permis de concevoir la posture illustré dans la figure **(Fig3.26.).**

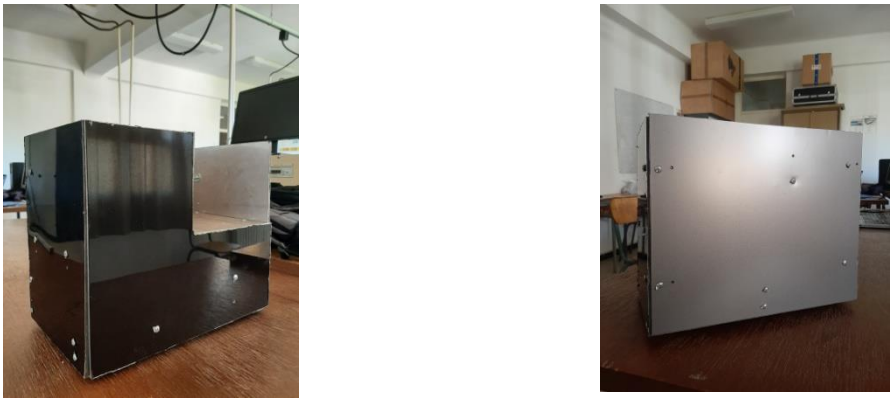


Fig.3. 26.Posture de notre robot.

3.4. Contrôle PID du robot mobile

Les commandes de type PID sont implantées dans tous les contrôleurs de robots industriels actuels. Le système est considéré comme un système linéaire et chacune de ses articulations est asservie par une commande décentralisée de type PID à gains constants. [46]

3.4.1. Définition [47]

C'est un système d'auto régulation (boucle fermée), qui cherche à réduire l'erreur entre la consigne et la mesure. La régulation PID sert à atteindre la valeur souhaitée pour une des variables du système (vitesse, position…).

3.4.2. Propriétés d'un système asservi [48]

- **Stabilité :** Le système asservi doit fonctionner automatiquement. Il est indispensable qu'il soit stable. Autrement, le système évolue en s'éloignant de son point (ou trajectoire) d'équilibre, ce qui peut engendrer des saturations voire une dégradation du système.
- **Précision :** En régime permanent, la sortie du système asservi doit suivre la référence sans erreur et rejeter rapidement les perturbations.

Un système linéaire continu à temps invariant est asymptotiquement stable si et seulement si les pôles de sa fonction de transfert sont à parties réelles strictement négatives

- Un système est stable si à une entrée bornée, la sortie du système est bornée.
- La stabilité d'un système est déterminée par la position des racines de l'équation caractéristique dans le plan complexe.

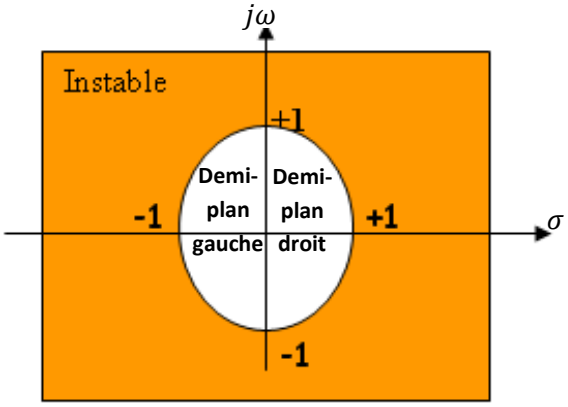


Fig.3. 27.Plan complexe P. [49]

3.4.3. Les boucles :

- **Boucle ouverte**

La commande en boucle ouverte consiste à ajuster la commande directement à partir de la consigne, sans tenir compte de la valeur du signal de sortie. [50]

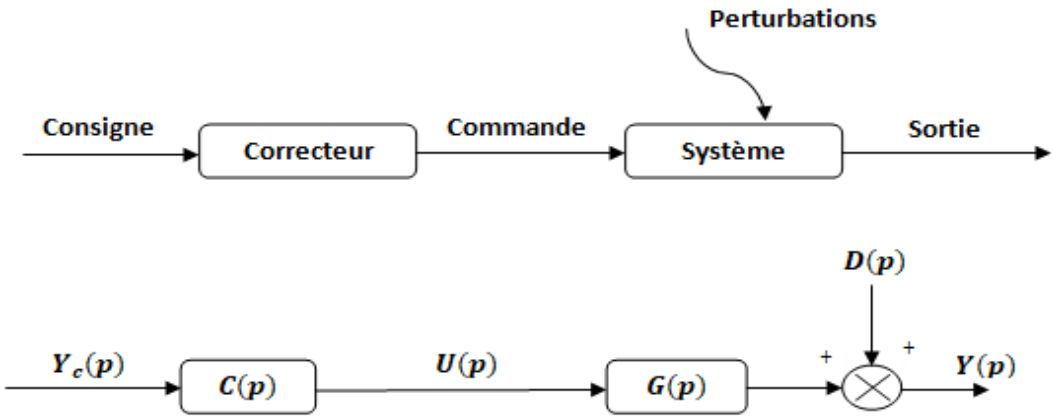


Fig.3. 28.Schéma block en boucle ouverte.

En théorie, la FTBO s'écrit de la forme :

$$\boldsymbol{Y(s) = C(p).G(p).Y_c(p) + D(p)}$$

Le correcteur parfait doit être choisi comme l'inverse du modèle du système afin de compenser la dynamique de celui-ci, c'est-à-dire :Si $\boldsymbol{D(s) = 0}$

$$\text{Afin que } \boldsymbol{Y(p) = Y_c(p) :\ C(p).G(p) = 1 \Rightarrow C(p) = 1/G(p)}$$

Remarque :

- Présence de perturbations ($\boldsymbol{D(p) \neq 0}$) impossible à compenser.
- Contrôle est peu robuste.
- Contrôle peu performant en pratique.

- **Boucle fermée**

En automatique continue, quand on est en présence d'un système qui comporte une boucle de rétroaction, on parle système en boucle fermée. [51]

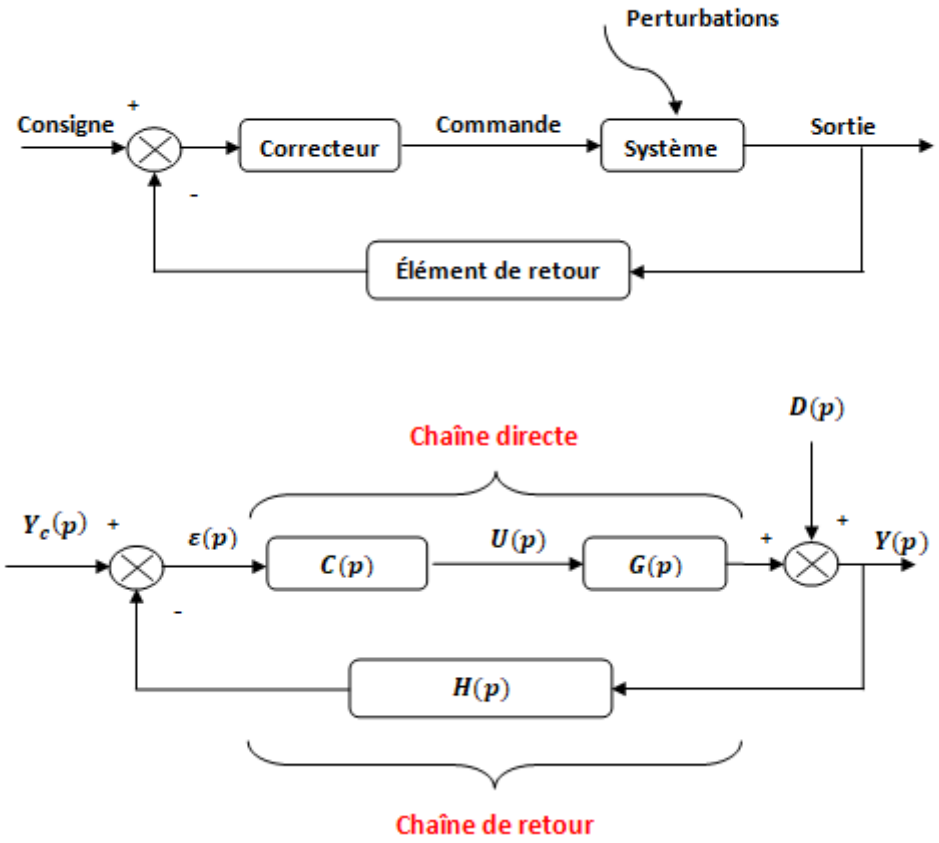


Fig.3. 29.Schéma block en boucle fermée.

En théorie, la FTBF s'écrit de la forme :

$$\text{Si } \boldsymbol{D(p)=0}\,;\boldsymbol{Y(p)}=\frac{\boldsymbol{C(p).G(p)}}{\boldsymbol{1+C(p).G(p).H(p)}} \qquad (3.4)$$

On distingue l'étude de la stabilité lors :

- Des variations de consigne : **problème de consigne ou de suivi.**
- De la présence de perturbations sur le processus : **problème de régulation.**

3.4.4. Les régulateurs [51]

3.4.4.1. Régulateur proportionnels (P)

L'asservissement de type P est le plus simple qui soit. Il s'agit d'appliquer une correction proportionnelle à l'erreur corrigeant de manière instantanée tout écart de la grandeur à régler :

$$U(t) = k_p.\varepsilon(t) \tag{3.5}$$

Avec $\varepsilon(t)$: $L'erreur\ statique$.

En utilisant la transformée de Laplace :

$$U(p) = k_p.\varepsilon(p) \tag{3.6}$$

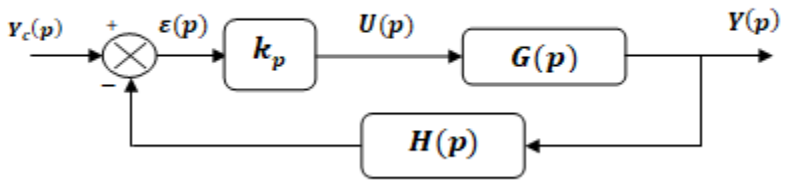


Fig.3. 30.Schéma block régulateur proportionnel

3.4.4.2.Régulateur proportionnel intégrateur (PI)

Le terme intégral complète l'action proportionnelle. Puisqu'il permet de compenser l'erreur statique et d'augmenter la précision en régime permanent. L'idée est d'intégrer l'erreur depuis le début et d'ajouter cette erreur à la consigne : lorsque l'on se rapproche de la valeur demandée, l'erreur devient de plus en plus faible.

L'intégrale agissant comme un filtre sur le signal intégré. Elle permet de diminuer l'impact des perturbations (bruit, parasites), et il en résulte alors un système plus stable. Malheureusement, un terme intégral trop important peut lui aussi entraîner un dépassement de la consigne, une stabilisation plus lente, voire même des oscillations divergentes.

L'asservissement de type PI est un asservissement de type P auquel on a ajouté un terme intégral :

$$U(t) = K_p.\varepsilon(t) + K_i.\int_0^t \varepsilon(\tau)\,d\tau \tag{3.7}$$

En utilisant la transformée de Laplace :

$$U(p) = K_p.\varepsilon(p) + K_i.\frac{\varepsilon(p)}{p} = \varepsilon(p).\left[K_p + \frac{K_i}{p}\right] \quad (3.8)$$

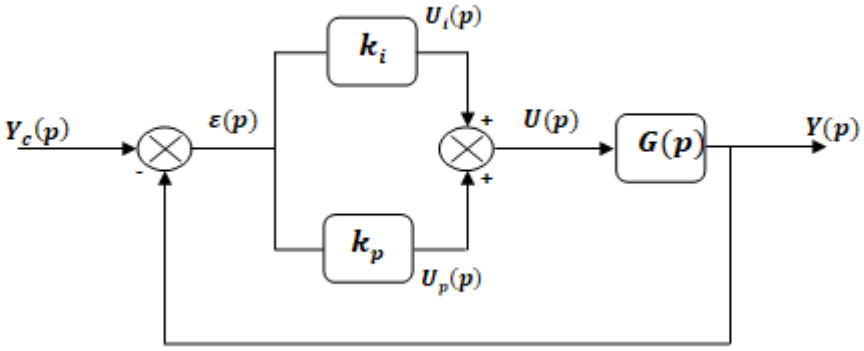


Fig.3. 31.Schéma block régulateur proportionnel intégrateur

3.4.4.3.Régulateur proportionnel intégrateur dérivateur (PID)

Les termes proportionnel et intégral peuvent amener un dépassement de la consigne et des oscillations. Pour limiter ce phénomène indésirable, on introduit un troisième élément : le terme dérivé.

Son action va dépendre du signe et de la vitesse de variation de l'erreur, et sera opposée à l'action proportionnelle : elle freine alors le système, limitant le dépassement et diminuant le temps de stabilisation :

$$U(t) = K_p.\varepsilon(t) + K_i.\int_0^t \varepsilon(\tau)\,d\tau + K_d.\frac{d}{dt}\varepsilon(t) \quad (3.9)$$

En utilisant la transformée de Laplace :

$$U(p) = K_p.\varepsilon(p) + K_i.\frac{\varepsilon(p)}{p} + K_d.p.\varepsilon(p) = \varepsilon(p).\left[K_p + \frac{K_i}{p} + K_d.p\right] \quad (3.10)$$

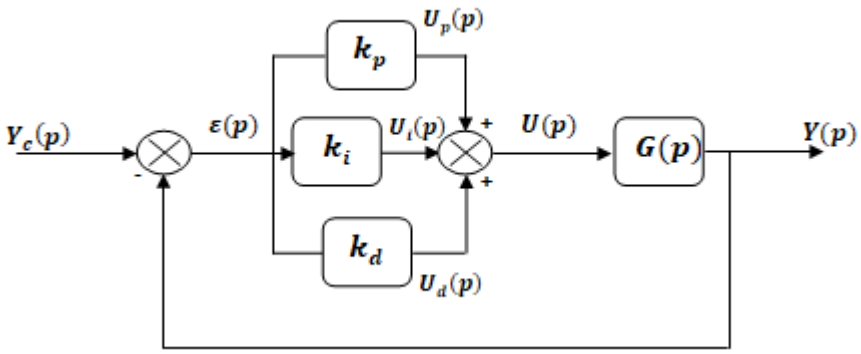


Fig.3. 32.Schéma block régulateur PID

3.4.5. Réglage des coefficients d'un PID [52]

Le réglage des coefficients $\boldsymbol{k_p}, \boldsymbol{k_i}, \boldsymbol{k_d}$d'un PID peut se faire expérimentalement par essais/erreurs.

Vouloir régler les trois coefficients en même temps ne sert à rien. Car, Il y a trop combinaisons possibles et trouver un triplé performant relèverait de l'exploit. Il vaut mieux y aller par étape :

- Tout d'abord, il faut mettre en place un simple régulateur proportionnel, il faut régler $\boldsymbol{k_p}$afin d'améliorer le temps de réponse du système et qu'il se rapproche très vite de la consigne en gardant sa stabilité ($\boldsymbol{k_p \neq 0}$, $\boldsymbol{k_i = k_d = 0}$).
- Ensuite le coefficient $\boldsymbol{k_d}$ qui permet de rendre le système plus stable, il permet de diminuer les oscillations.
- Une fois $\boldsymbol{k_p}$ et $\boldsymbol{k_d}$ réglés, on passe au coefficient $\boldsymbol{k_i}$qui permettra d'annuler l'erreur finale du système afin que ce dernier respecte exactement la consigne.

Remarque :

En général, pour régler ces coefficients, on donne au système une consigne fixe (exp : pour un moteur : tourne à 3 tours par seconde) et on observe la réponse du système (exp : l'évolution du nombre de tours par seconde du moteur au cours du temps).

3.4.6. Navigation du robot et contrôle de vitesse

L'obtention des valeurs parfaite du PID se fait après plusieurs essais. Le rôle de l'encodeur optique devient crucial. Le contrôle se fait par l'observation des courbes générées par la rotation des roues de l'encodeur.

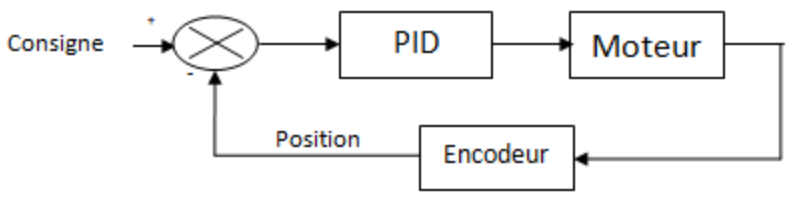


Fig.3. 33.Schéma bloc général de la boucle de régulation

Marche avant du robot

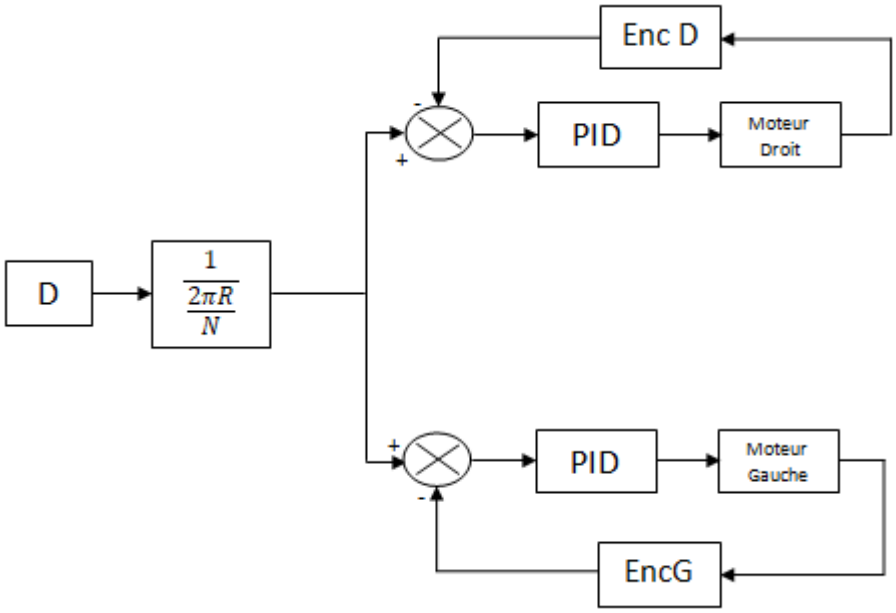


Fig.3. 34.Schéma bloc de la boucle de la marche avant.

Le système asservi en boucle fermé (retour unitaire) de la figure **(Fig.3.34)** illustre l'effet des deux encodeurs gauche et droite relié aux deux moteurs gauche et droit, respectivement.

Remarque : L'encodeur nous renvoie le nombre d'impulsions correspondant à la position angulaire du moteur.

A partir du schéma bloc on peu retirer les deux équations suivantes :

$$\begin{cases} D_{imp} = \dfrac{D}{\dfrac{2\pi R}{N}} & (3.11) \\ G_{imp} = \dfrac{D}{\dfrac{2\pi R}{N}} & (3.12) \end{cases}$$

Avec :

D_{imp} : Nombre d'impulsions encodeur droit.

G_{imp}: Nombre d'impulsions encodeur gauche.

D : Distance en mètre (m).

R : Rayon de la roue des deux encodeurs.

N : Vitesse intégrée de l'encodeur.

La réponse la plus correcte s'obtient après plusieurs essaies et modifications des valeurs contenues dans le programme, le changement se fait sur les coefficients du régulateur PID afin de contrôler en position le déplacement du robot grâce au calcul du nombre d'impulsions.

La méthode utilisée est assez suffisante pour pouvoir obtenir la sortie désirée, néanmoins un résultat plus fiable pourrait être obtenu en ajoutant des éléments ou en modifiants l'équation qui nous génère le nombre d'impulsions des encodeurs, d'où l'équation ci-dessous :

$$D_{imp} = \frac{D}{\frac{2\pi R}{N_1}} \quad (3.13)$$

$$G_{imp} = \frac{D}{\frac{2\pi R}{N_1}} \quad (3.14)$$

Equations prises en considération.

Où : $$N_1 = P_{cal}.N \quad (3.15)$$

Avec :

P_{cal} : Facteur de calibrage.

Pour illustrer le changement des coefficients du régulateur PID, on s'est appuyer sur les ces courbes obtenue après simulation grâce au traceur série d'Arduino pour des valeurs proportionnel intégrateur et dérivateur différente.

Tab3. 2.Valeurs PID et P_cal premier essai pour la marche avant

	Proportionnel	Intégrateur	Dérivateur	P_{cal}
Moteur droit	0.31	0.033	0.0012	4
Moteur gauche	0.33	0.05	0.0012	4

A partir des valeurs du tableau (**Tab.3.2.)** on obtient les courbes de la figure **(Fig3.34.)**:

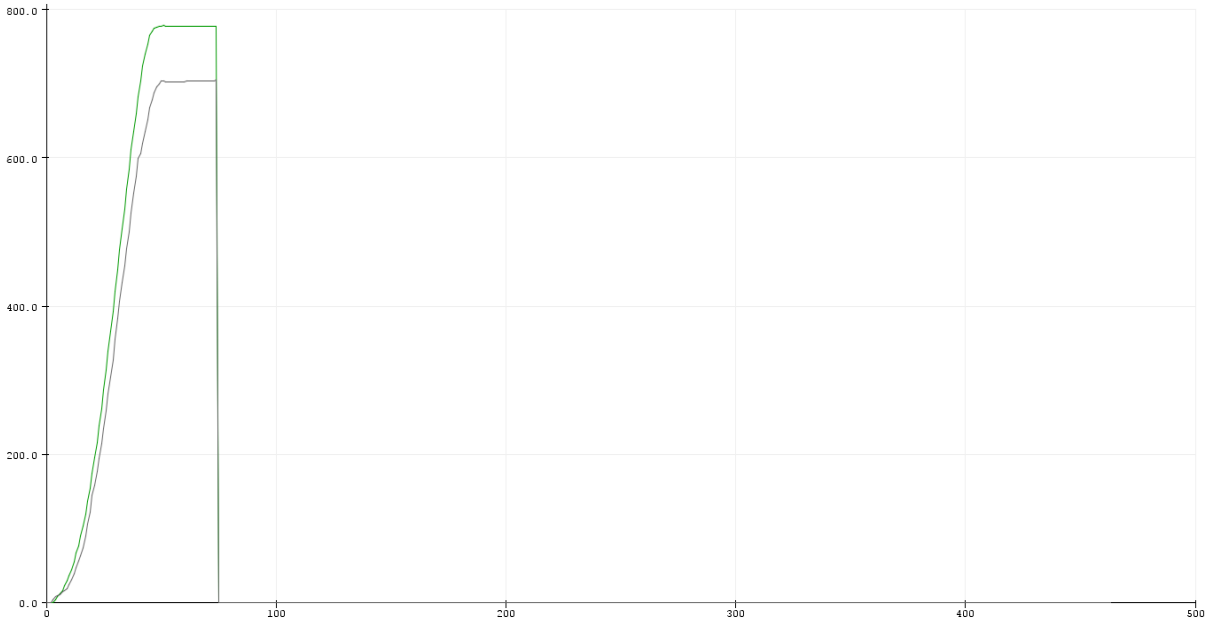


Fig.3. 35.Réponse indicielle du premier essai de la marche avant.

Tab3. 3.Valeurs PID et P_cal deuxième essai pour la marche avant.

	Proportionnel	Intégrateur	Dérivateur	P_{cal}
Moteur droit	0.29	0.04	0.0011	4
Moteur gauche	0.31	0.033	0.0012	4

A partir des valeurs du tableau (**Tab3.3.)** on obtient les courbes de la figure **(Fig3.35.)** :

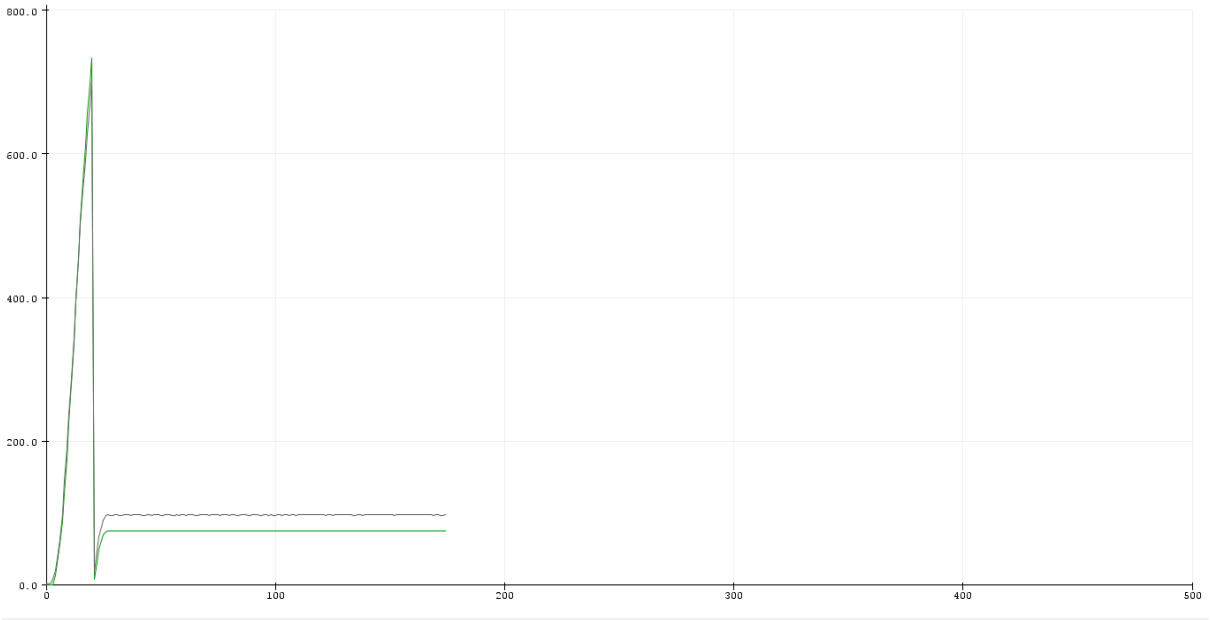


Fig.3. 36.Réponse indicielle du deuxiéme essai de la marche avant.

Tab3. 4.Valeurs PID et P_{cal} final pour la marche avant.

	Proportionnel	Intégrateur	Dérivateur	P_{cal}
Moteur droit	0.28	0.04	0.0011	4
Moteur gauche	0.3	0.04	0.0011	4

A partir des valeurs du tableau (**Tab3.4.)** on obtient les courbes de la figure **(Fig3.36.)** :

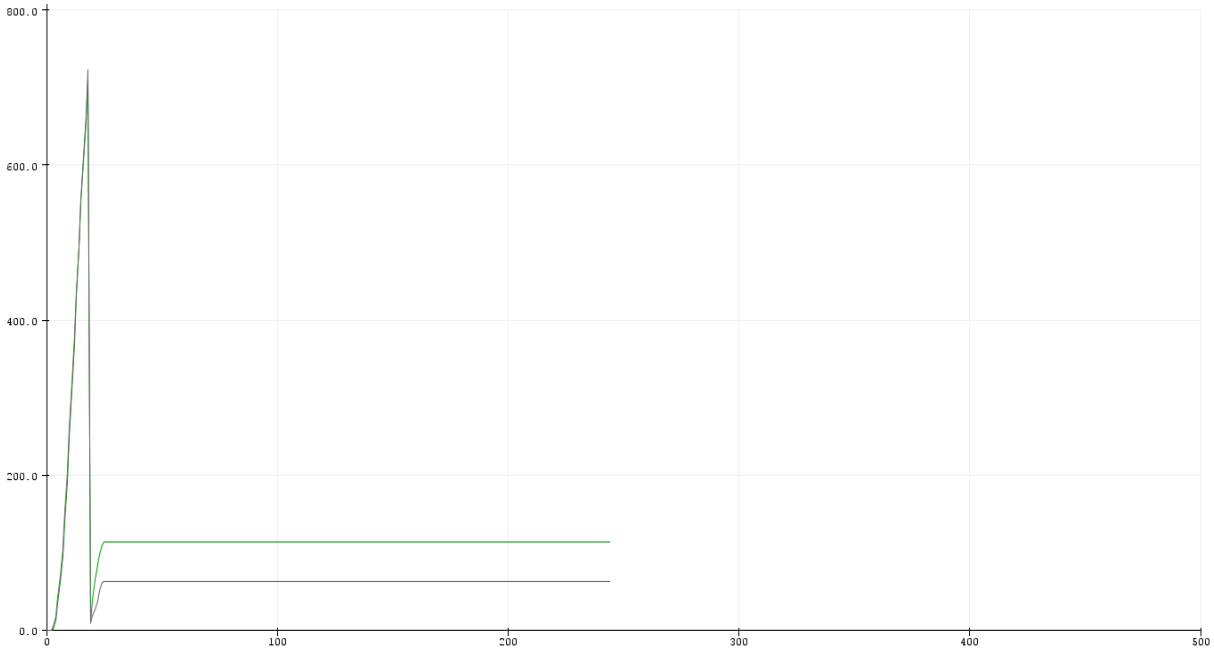


Fig.3. 37.Réponse indicielle finale de la marche avant.

Le déplacement réel du robot est illustré dans la figure **(Fig.3. 38.)**

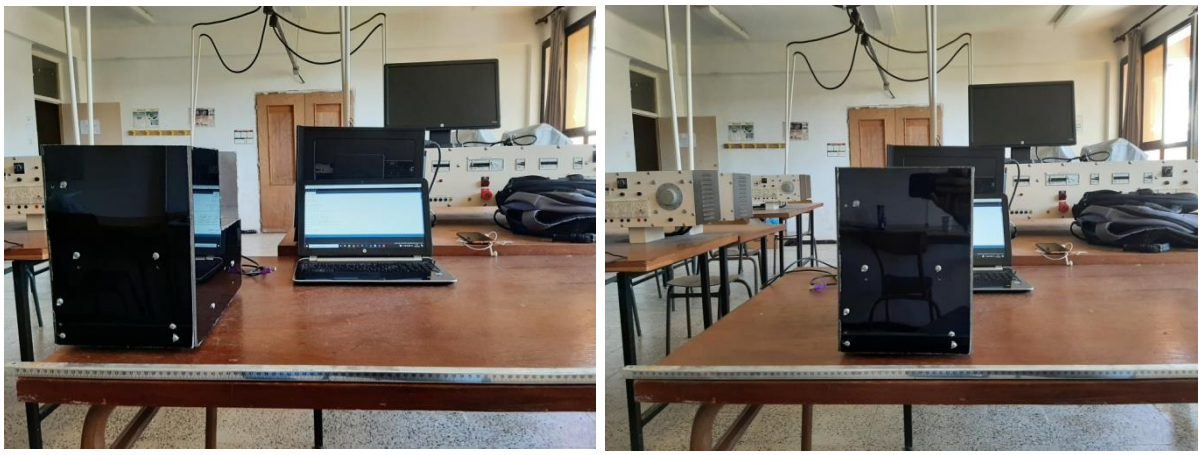


Fig.3. 38. Déplacement du robot d'un point A a un point B (De gauche a droite).

Lors du déplacement en marche avant, le robot suit une trajectoire prédéfinie dans le programme. Dans notre cas on a donné comme instruction le déplacement du robot sur une ligne droite de 300mm, une distance qui sera converti en impulsions selon les équations (3.13) et (3.14).

Après les modifications de commande qu'on a fait subir aux moteurs de notre robot, ce dernier atteint son but avec une légère erreur négligeable (Fig.3. 39.).

Fig.3.39. Le robot atteint son point désiré.

3.5. Conclusion

Dans ce chapitre, on a exposé l'essentiel de notre travail. Après la description des différents constituants de notre réalisation et le schéma global du robot, on a procédé à sa commande. Ainsi, quelques rappels théoriques ont été exposés et les différents réglages effectués ont été donnés avec les résultats obtenus.

Ce mémoire est une étude de projet pratique qui concerne la réalisation d'un robot de type unicycle, après avoir présenté la robotique de manière générale ainsi que les différents types de robots, on s'est intéressés à la modélisation et navigation de notre robot, et à l'élaboration du modèle cinématique.

Concernant la réalisation pratique, pour le châssis on a utilisé plaque d'aluminium qui a été découpé par nous-même. On a aussi utilisé deux roues indépendantes relié a deux moteurs, une roue bille assurant la stabilité et l'équilibre, deux encodeurs pour chaque roue conçu spécialement pour la calcule et contrôle de distance, tous ces composants sont connectés entre eux pour former un robot mobile par la Carte Arduino ATmega2560 ainsi que le shield moteur L293D. Après avoir terminé l'aspect mécanique on est passé a la partie programmation ou on a dû contrôler le robot en réglant les paramètres du PID, la programmation est assurée par le langage C++.

Le manuscrit a été partagé en trois chapitres. Le premier traite des généralités, le second parle de navigation et modélisation du robot, le troisième chapitre expose de l'aspect pratique et la régulation PID.

Ce travail nous a permis de nous familiariser avec la programmation de la carte Arduino et de découvrir de façon encore plus précise une nouvelle technologie qui est la robotique. En réalisant ce projet nous avons pu acquérir de nouvelles connaissances et approfondir celles qu'on a acquis durant notre formation à l'université, grâce à ce travail nous avons compris comment travailler en équipe et comment concevoir et mener à bien un projet.

Nous avons pu atteindre notre objectif qui est de concevoir et contrôler un robot mobile unicycle, néanmoins notre travail pourra évidemment être amélioré, en ajoutant d'autres fonctionnalités comme l'évitement d'obstacle ainsi diversifier ses taches et actions, en implémentant la rotation, chose qu'on n'a pas pu effectuer et réaliser a temps.

[1] Dictionnaire Larousse, Edition en ligne, 2008.

[2] Laëtitia Matignon « *Introduction à la robotique* », Support de cours, Université de Caen, France, 2011/2012.

[3] Agnès Guillot, « *Histoire de la robotique : automate aux premiers robots* », Dossiers FuturaTech, site web : www.futura-sciences.com, Visité le 02/07/2020.

[4] EYRAUD, Charles-Henri. « *Horloges astronomiques au tournant du XVIIIe siècle : de l'à-peu-près à la précision* ». Thèse de doctorat. Université de Lyon 2, 2004.

[5] Lionel Genève,« *Système de déploiement d'un robot mobile autonome basé sur des balises* », Thèse de doctorat, Université de Strasbourg, 2017.

[6] Marc Silanus, Cours de robotique, 2014, Site web :www.silanus.fr consulté le 02/07/2020.

[7] Document «Schneider Electric », site web : www.se.com, visité le : 02/07/2020.

[8] Fabricant Jacksking, site web :www.amazon.fr, visité le : 03/07/2020.

[9] Site web :www.Roboticbeast.com, visité le : 03/07/2020.

[10] Dossiers FuturaTech, site web : www.futura-sciences.com, visité le 03/07/2020.

[11] Site web : Robotic.media.mit.edu, visité le : 03/07/2020.

[12] Site web :www.epson.fr, visité le 03/07/2020.

[13] Site web :www.army-technology.com visité le 03/07/2020.

[14] Site web :www.directindustry.fr, visité le 03/07/2020.

[15] « Histoire de la robotique, L'informatique ingénierie Technologie »,Site web :www.hisour.com,visité le 05/07/2020.

[16] David Filliat, Support de cours «*Navigation pour les systèmes autonomes* », Ecole Nationale supérieure de Techniques avancées, ParisTech, 2019.

[17] Beast Autonomous Robot Mod II (with Vision), site web : www.cyberneticzoo.com consulté le 09/07/2020.

[18] Slimane Noureddine,« *SYSTEME DE LOCALISATION POUR ROBOTS MOBILES* », Thèse de doctorat, Université de Batna, 2005.

[19] Site web: www.zentasrobots.com, visité le 10/07/2020.

[20] Site web : www.frandroid.com, visité le 10/07/2020.

[21] Hocine TAKHI et ATTACHI Redouane Cherif. « *Conception et réalisation d'un robot a base mobile*», Mémoire de fin d'études, Université de Leghouat, 2014.

[22] Fabio Morbidi « *Perception avancée et robotique mobile* », Cours M2 EEAI, Université de Picardie 2014/2015.

[23] Stéphane Lens « *locomotion d'un robot mobile* » mémoire de fin d'études, Université de liège, institut Montefiore, MAI 2008.

[24] Site web www.intelligence-artificielle-robotique.weebly.com consulté le 12/07/2020

[25] HAMANI MILOUD, « *Navigation des robots mobiles non-holonomes sous contrôle flou* », Thèse de Doctorat en sciences, Université Ferhat Abbas Sétif-1, le 11/12/2016.

[26] Nicolas Morette. « *approche par modèle direct et commande prédictive* ». Cours de contribution à la navigation de robots mobiles, Université d'Orléans, 2009.

[27] Lionel Geneve. « *Système de déploiement d'un robot mobile autonome basé sur des balises* », Université de Strasbourg, 2017.

[28] Hela BenSaid *« Navigation autonome et commande référencée capteurs de robots d'assistance à la personne* ». Thèse de Doctorat. l'Université de Limoges Science et Ingénierie pour l'Information, Mathématiques, Le 23 mars 2018.

[29] Site web :www.arrow.com, consulté le 10/08/2020.

[30] DATASHEET « Arduino UNO », Components 101, 2018.

[31] Document « *PROJETS EXPÉRIMENTAUX DE PHYSIQUE*», Université Paris Sud.

[32] Luky Larry, « *Control a DC motor with Arduino and L293D chip*».

[33] Fréderic Giamarchi, Cours, IUT de Nîmes, 2010.

[34] Site web : www.elektrifikasyon.com, consulté le 11/08/2020.

[35] Site web : www.euro-makers.com consulté le 11/08/2020.

[36] Article de Alexandre SCHMITT, Site web : www.rco.fr.nf, Consulté le 20/08/2020.

[37] DZUINO online electronic store, Site web : www.dzduino.com consulté le 11/08/2020.

[38] Site web : www.codechamp.fr, consulté le 15/08/2020.

[39] Site web : www.astrosurf.com, consulté le 15/08/2020.

[40] Dr. ELKHEIR Merabet, Machines électrique, université de Batna, 2013/2014.

[41] Site web : www.technologieelkhalil.blogspot.com, consulté le 20/08/2020.

[42] « Groupe Ward-Leonard », Site web : www.energieplus-lesite.be/, Consulté le 15/08/2020.

[43] Site web : www.jr-international.fr, consulté le 25/08/2020.

[44] Summum, Site web : www.fr.dreamstime.com,consulté le 25/08/2020.

[45] Oscar Andrès Vivas. « *Contribution à l'identification et à la commande des robots parallèles* ». Université Montpellier II - Sciences et Techniques du Languedoc, 2004.

[46] Etienne Deguine, Daniel Ross, « *régulation PID* », 26/02/2010.

[47] « Cours Systèmes linéaires asservis : analyse de la stabilité », Institut National des sciences appliquées Rouen Normandie.

[48] « *Stabilité et précision des systèmes discrets* », site web : www.uvt.rnu.tn, consulté le 25/08/2020.

[49] Hugues GARNIER. « *Automatique continue, Systèmes bouclés : commande, stabilité & performances* ». université de Lorraine, version 24 septembre 2019.

[50] J.M. Dutertre , Support de Cours « *Automatique linéaire1* », 2016.

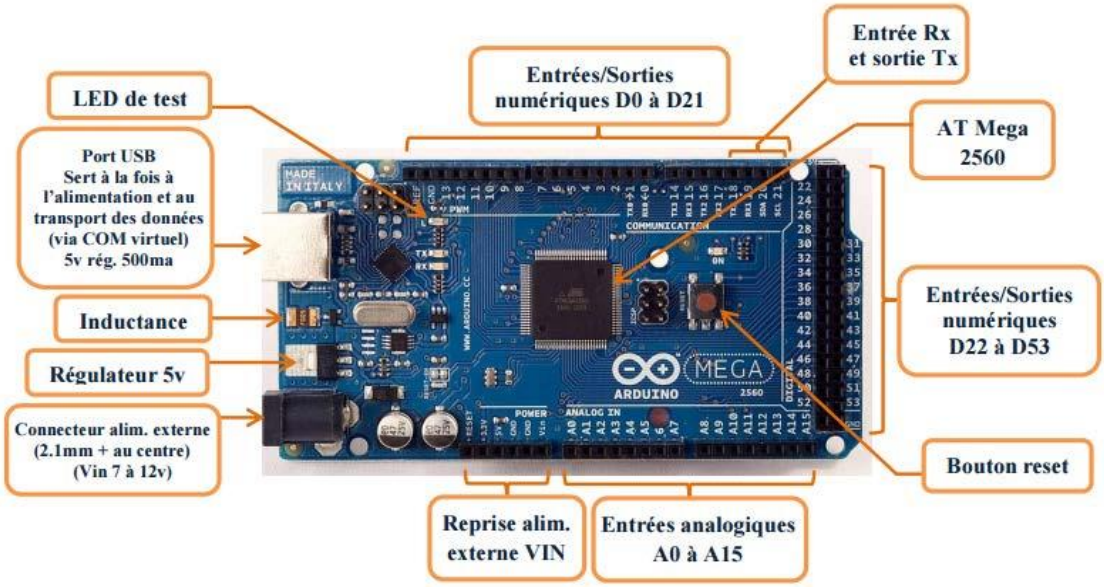
[51] Christophe Le Lann, Support de cours « *Le PID utilisé en régulation de position et/ou de vitesse de moteurs électriques* », 2006/2007.

[52] Support de Cours Classe de terminale SI « *Implémenter un PID sans calcul* ».

ANNEXES

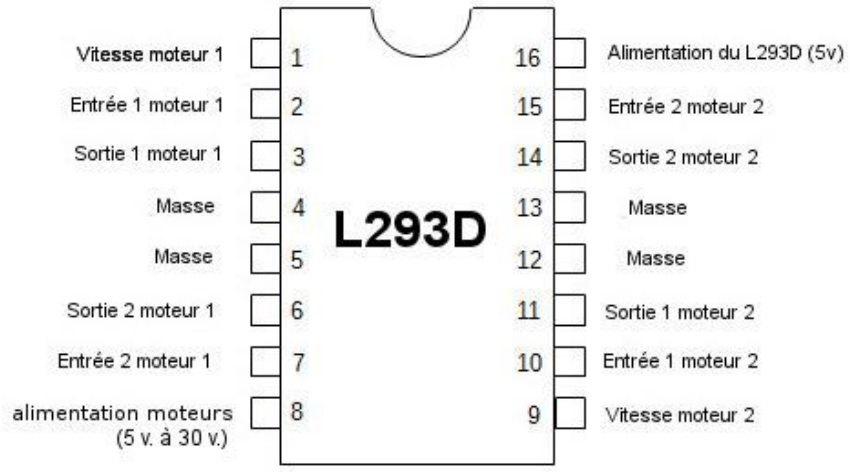
ANNEXE A

Brochage de la carte Arduino Atmega 2560.



ANNEXE B

Brochage du shield moteur L293D



ANNEXE C

La fonction de commande de la marche avant

```
void AVANT(double dist,int A)
{
if(steep==A)
{

D_Setpoint=dist/(2*pi*R/(N));
G_Setpoint=dist/(2*pi*R/(N));
 D_PID.Compute();
if(D_Output>=0)
{
  D_PID.SetTunings(0.29, 0.04, 0.0011);//good
  G_PID.SetTunings(0.31 , 0.033, 0.0012); //good

  MoteurD.run(FORWARD);
  MoteurD.setSpeed(D_Output);
}
else
{
  MoteurD.run(BACKWARD);
  MoteurD.setSpeed(-D_Output);
  }
G_PID.Compute();
if(G_Output>=0)
{
  D_PID.SetTunings(0.3, 0.04, 0.0011);//good
  G_PID.SetTunings(0.28 , 0.04, 0.0011); //good
MoteurG.run(FORWARD);
MoteurG.setSpeed(G_Output);
}
else
{
MoteurG.run(BACKWARD);
MoteurG.setSpeed(-G_Output);
}
int eD=D_Setpoint-encD;
int eG=G_Setpoint-encG;
int ermax=250;
int delayenc=500;
float prevMillis;

if((eD<ermax && eD>-ermax) &&(eG<ermax && eG>-ermax))
{
{
  if(millis()-prevMillis>delayenc)
steep++;
Serial.println("avantfini");
encD=0;encG=0;
prevMillis=millis();
}
}
}
}
```

ANNEXE D

Programme complet de notre réalisation

```
#include <AFMotor.h>
#include <Servo.h>
#include <PID_v1.h>

AF_DCMotor MoteurD(4);
AF_DCMotor MoteurG(3);
//-----------------
double encD=0,encG=0;
double encDLast=0,encGLast=0;
int B_D=19;// pin num 3
int A_D=18;// pin num 2
int B_G=20;// pin num
int A_G=21;// pin num
float Nl=300,R=20,L=270,pi=3.14159,Pcal=4,N=Pcal*Nl;
int steep;
//--------------------
double Kpd=0.15, Kid=0.01, Kdd=0.0001;
double Kpg=0.15, Kig=0.01, Kdg=0.0001;
//double Kpd=0.15, Kid=0.02, Kdd=0.0001;
//double Kpg=0.15, Kig=0.02, Kdg=0.0001;
double G_Setpoint, G_Output=0, Gspeed;
PID G_PID(&encG,&G_Output,&G_Setpoint,Kpg,Kig,Kdg,DIRECT);
double D_Setpoint, D_Output =0, Dspeed;
PID D_PID(&encD,&D_Output,&D_Setpoint,Kpd,Kid,Kdd,DIRECT);
//----------------------------------
void setup()
{
//MoteurD.run(RELEASE);
//MoteurG.run(RELEASE);
//-----------------------
Serial.begin(9600);
//-------------
pinMode(A_D,INPUT_PULLUP);
pinMode(B_D,INPUT_PULLUP);
pinMode(A_G,INPUT_PULLUP);
pinMode(B_G,INPUT_PULLUP);
attachInterrupt(5,dencDA,CHANGE); // pin
attachInterrupt(4,dencDB,CHANGE); // pin
attachInterrupt(2,dencGA,CHANGE); // pin
attachInterrupt(3,dencGB,CHANGE); // pin
//---------------------------
D_PID.SetMode(AUTOMATIC);
D_PID.SetOutputLimits(-250,200);
D_PID.SetSampleTime(1);
G_PID.SetMode(AUTOMATIC);
G_PID.SetOutputLimits(-250,200);
G_PID.SetSampleTime(1);
}
```

```
//------------------------------
void loop()
{
  Serial.print("encD=");
encShowR();

AVANT(100,0);
AVANT(100,1);
AVANT(100,2);
AVANT(100,3);
AVANT(100,4);
}
void AVANT(double dist,int A)
{
if(steep==A)
{

D_Setpoint=dist/(2*pi*R/(N));
G_Setpoint=dist/(2*pi*R/(N));
 D_PID.Compute();
if(D_Output>=0)
{
  D_PID.SetTunings(0.29, 0.04, 0.0011);//good
  G_PID.SetTunings(0.31 , 0.033, 0.0012); //good

  MoteurD.run(FORWARD);
  MoteurD.setSpeed(D_Output);
}
else
{
  MoteurD.run(BACKWARD);
  MoteurD.setSpeed(-D_Output);
  }
G_PID.Compute();
if(G_Output>=0)
{
  D_PID.SetTunings(0.3, 0.04, 0.0011);//good
  G_PID.SetTunings(0.28 , 0.04, 0.0011); //good
MoteurG.run(FORWARD);
MoteurG.setSpeed(G_Output);
}
else
{
MoteurG.run(BACKWARD);
MoteurG.setSpeed(-G_Output);
}
int eD=D_Setpoint-encD;
int eG=G_Setpoint-encG;
int ermax=250;
int delayenc=500;
float prevMillis;

if((eD<ermax && eD>-ermax) &&(eG<ermax && eG>-ermax))
{
{
  if(millis()-prevMillis>delayenc)
```

```
steep++;
Serial.println("avantfini");
encD=0;encG=0;
prevMillis=millis();
}
}
}
}
void dencDA ()
{
if(digitalRead(A_D)==HIGH)
{
      if(digitalRead(B_D)==LOW)
      {
        encD++ ;
      }
      if(digitalRead(B_D)==HIGH)
      {
        encD-- ;
      }}
      else{
        if(digitalRead(B_D)==HIGH)
      {
        encD++;
      }
        if(digitalRead(B_D)==LOW)
      {
        encD--;
      }
      }
}

void dencDB()
{
  if(digitalRead(B_G)==HIGH)
{
      if(digitalRead(A_G)==HIGH)
      {
        encG++ ;
      }
      if(digitalRead(A_G)==LOW)
      {
        encG-- ;
      }
      }
      else
      { if(digitalRead(A_G)==LOW)
      {
        encG++;
      }
        if(digitalRead(A_G)==HIGH)
```

```
          if(digitalRead(A_G)==HIGH)
        {
          encG--;
        }
        }
 }
 void dencGA ()
 {
 if(digitalRead(A_G)==HIGH)
 {
       if(digitalRead(B_G)==LOW)
        {
          encG++ ;
        }
        if(digitalRead(B_G)==HIGH)
        {
          encG-- ;
        }}
        else{
          if(digitalRead(B_G)==HIGH)
        {
          encG++;
        }
          if(digitalRead(B_G)==LOW)
        {
          encG--;
        }
        }
 }
 void dencGB()
 {
{
  if(digitalRead(B_G)==HIGH)
{
      if(digitalRead(A_G)==HIGH)
      {
        encG++ ;
      }
      if(digitalRead(A_G)==LOW)
      {
        encD-- ;
      }
      }
      else
      {
        if(digitalRead(A_G)==LOW)
      {
        encD++;
      }
        if(digitalRead(A_G)==HIGH)
      {
        encG--;
      }
      }
}
```

```
      }
}
void encShowR()
{
  Serial.print("R : ");
  Serial.print(encD);
  Serial.print(" *** L: ");
  Serial.println(encG);
}
```

RESUMÉ

Ce projet de Master concerne l'étude et la réalisation d'un robot mobile de type unicycle à base d'une carte Arduino ATmega2560, et de lui donner la capacité de se déplacer dans son environnement. Il est équipé de deux moteurs à courant continu avec réducteur reliés à la carté Arduino grâce au circuit de puissance L293d (Shield) qui nous a permis de commander les deux moteurs, afin de suivre et réguler le déplacement du robot, des encodeurs rotatifs optiques sont connectés à chaque roue. La programmation est faite à base du langage C++.

ABSTRACT

This Master's project concerns the study and construction of a unicycle-type mobile robot based on an Arduino ATmega2560 board, and to give it the ability to move around in its environment. It is equipped with two direct current motors with reducer connected to the Arduino card thanks to the L293d (Shield) power circuit which allowed us to control the two motors, in order to follow and regulate the movement of the robot, optical rotary encoders are connected to each wheel. The Different control modes have been implemented using C ++ code.

ملخص

يتعلق مشروع الماجستير هذا بدراسة وبناء روبوت متحرك من نوع الدراجة الأحادية على أساس لوحة اردوينو ميغا، و منحه القدرة على التحرك في بيئته. وهو مجهزة بمحركين للتيار المباشر مع مخفض متصل بالبطاقة بفضل دائرة الطاقة التي سمحت لنا بالتحكم في المحركين ، المشفرات الضوئية الدوارة هي متصلة بكل عجلة من أجل متابعة وتنظيم حركة الروبوت. تتم البرمجة على أساس لغة ++C.