

UNIVERSITE KASDI MERBAH OUARGLA
Faculté de Nouvelles Technologie de l'information et de la Communication
Département de l'électronique et de télécommunication



Mémoire
MASTER PROFESSIONNEL
Domaine : Sciences et Technologies
Filière : Génie Électrique
Spécialité : Instrumentation
Présenté par :

Ogal Abdelbari

Hafiane Med Tahar

Thème:

Navigation et évitement d'obstacle d'une voiture autonome

Soutenu publiquement

Le : 24 / 05 / 2017

Devant le jury :

Mr. Benhelal Belkhir

MAA

Président

UKM Ouargla

Dr. Adel Djellal

MCB

Encadreur/rapporteur

UKM Ouargla

Dr. Chakour Chouaib

MAB

Examineur

UKM Ouargla

Année universitaire 2016/2017

Remerciement

À ma famille

Et

À mes amies

*Je remercie ALLAH le tout puissant de m'avoir donné
le courage et la*

*Patience qui m'a permis d'accomplir ce modeste
travail.*

*Je tiens en premier à exprimer ma grande gratitude
envers mon encadreur*

*Mr. Adel Djellal qui m'a apporté son aide et ses
valeureux conseils pour l'accomplissement et le suivi de
ce travail. Je remercie également le membre de jury pour
avoir examiné ce travail*

*Et je présente mes sincères remerciements à tout le
personnel qui m'a aidé*

À la Réalisation de ce travail.

Dédicace

Je dédie ce mémoire:

À ma très chère mère qui a été la lumière de ma vie

Et Mon père

A toute mes familles et mes grandes familles (Ogal et Hafiane)

, pour tous les efforts et les sacrifices qu'ils me présentent.

Et Mon encadreur Mr. Adel Djellal

A tous mes amis

Pour conclure, je le dédie à:

A mon classe 2^{ème} année Master Instrumentation et à tous ce qui

m'ont aimé et m'ont respecté

Ogal Abdelbari

Hafiane Med Taher

Table des matières

Table des matières	4
Table des Figures	6
Introduction générale	7
1. Plateforme utilisée	8
1.1. Introduction	8
1.2. Hardware :	8
1.2.1. Plateforme mobile :	8
1.2.2. Arduino :	8
1.2.3. Moteur à Courant Continu	11
1.2.4. Contrôle des moteurs par L298N	12
1.2.5. Capteur distance :	14
1.3. Software :	15
1.3.1. Plateforme de programmation Arduino :	15
1.3.2. Utilisation de programme :	16
1.4. Conclusion :	16
2. Évitement d'obstacles	17
2.1. Introduction :	17
2.2. Les méthodes d'évitement d'obstacle	17
2.2.1. méthode de champ potentiel	17
2.2.2. Evitement par régulation PID	17
2.2.3 méthode de TOR (Tout Ou Rien)	18
2.3. La Communication	18
2.3.1. Le module Arduino Bluetooth	19
2.3.2. Le module shield Arduino Wifi	19
2.3.3. Le Module XBee	19
2.3.4. Infrarouge	20
2.4. Conclusion	20
3. Application	21
3.1. Introduction	21
3.2. Réalisation de la plateforme	21
3.2.1. La connexion entre la carte 298l et le de moteur et Arduino :	21
3.2.2. La connexion entre l'Arduino et le capteur ultrason :	22

3.2.3. La connexion entre l'Arduino et la carte HC_05 :.....	22
3.2.4. Le schéma global	23
3.3. Communication avec PC	23
3.3.1. La commande par Bluetooth en utilisant Matlab	23
3.3.2. Le programme de commande	24
5.2. La commande par Bluetooth avec Smartphone	30
5.3. Comparaison de résultats :	32
5.4. Conclusion :	32
Conclusion générale.....	33
Bibliographe.....	34

Table des Figures

Figure 1 : Voiture RC Electrique 1/10	8
Figure 2 : Arduino Mega 2560	9
Figure 3 : Différents types des capteurs pour l'Arduino	10
Figure 4 : Différents actionneurs pour l'Arduino.....	11
Figure 5 : les deux parties essentielles de MCC.....	11
Figure 6 : Pont en H	12
Figure 7 : Le L298N	12
Figure 8 ; les différents composants du L298N	13
Figure 9 : Principe du capteur à ultrason	14
Figure 10 : le capteur de distance à Ultrason HC-SR04	14
Figure 11 : Le brochage du module.....	15
Figure 12 : Interface IDE Arduino	16
Figure 13 : évitement obstacle avec méthode champ potentiel.....	17
Figure 14 : Diagramme de commande de PID	18
Figure 15 : boucle de régulation de projet.	18
Figure 16 : Type de modules Bluetooth	19
Figure 17 : Module shield Wi-Fi	19
Figure 18 : Module XBee	19
Figure 19 : Zone de couverture des différents réseaux sans-fil.	20
Figure 20 : La connexion entre la carte 298I et les deux moteurs et Arduino	21
Figure 21 : La connexion entre l'Arduino et le capteur ultrasons	22
Figure 22 : La connexion entre l'Arduino et la carte HC_05	22
Figure 23 : Le schéma globale	23
Figure 24 : Interface Matlab implémenté	24
Figure 25 : La trajectoire de voiture visualisée avec GUI par méthode TOR	29
Figure 26 : La trajectoire de voiture visualisée avec GUI par méthode Champ de potentiel.....	29
Figure 27 : La commande par Bluetooth en utilisant le Smartphone	30
Figure 28 : La voiture réelle (vue du bas)	30
Figure 29 : La voiture réelle (vue de coté)	31
Figure 30 : le voiture complet (vue en haut)	31

Introduction générale

Un des grands avantages de l'automatisation des procédures est la précision des résultats. La robotique est utilisée dans plusieurs industries comme l'automobile, la médecine, l'électroménager et plusieurs autres. Le plus complexe des machines peuvent être assemblées à l'aide de la robotique.

Les robots peuvent être fixes (bras manipulateur) ou mobiles (robot marcheur, robot à roues) selon l'application, les robots fixes sont généralement utilisés dans les usines, les hôpitaux, domaine agricole, etc. Cependant, les robots mobiles sont généralement utilisés dans les environnements dangereux nucléaires, militaires, déminage, etc. Ce dernier Type de robot est notre domaine d'intérêt dans ce mémoire.

Le suivi d'objets en robotique mobile est devenu une des tâches les plus utilisées, notre Projet consiste à réaliser un robot mobile (voiture autonome qui évite les obstacles) en utilisant des capteurs et des actionneurs pour satisfaire ce projet.

Présentation du mémoire

- Le premier chapitre sera consacré à la plateforme et les dispositifs utilisés (software et hardware) pour satisfaire ce projet et leurs fonctions.
- Le deuxième chapitre explique les méthodes d'évitement d'obstacles et leurs différences et les moyens de communications sans fils.
- Le troisième chapitre est une partie expérimentale pour réaliser et satisfaire la conception de la plateforme, Communication avec PC, implémentation des techniques et Comparaison des résultats.

1. Plateforme utilisée

1.1. Introduction

Dans ce chapitre, nous expliquons la plateforme utilisée (hardware et software), ces composants utilisés et leurs tâches (voiture 1/10, moteurs CC, contrôleur L298N, capteur ultrason, etc.) on a présenté aussi une description générale du programme Arduino implémenté sur la carte électronique.

1.2. Hardware :

1.2.1. Plateforme mobile :

Les voitures radiocommandées (RC) sont des versions miniatures de voitures de piste ou tout-terrain en différentes échelles (1/18, 1/16, 1/10, 1/8, 1/5...). Il existe trois motorisations différentes. Il y a les voitures radiocommandées thermiques au nitrométhane, les voitures radiocommandées thermiques au mélange d'essence et d'huile 2 temps et les voitures radiocommandées électriques. Les voitures radiocommandées thermiques sont les modèles qui ressemble le plus aux voitures de taille réelle. Elles sont équipées d'un moteur thermique avec cylindre et piston, carburateur, filtre à air, pont d'échappement, etc.



Figure 1 : Voiture RC Electrique 1/10

1.2.2. Arduino :

➤ Définition du module Arduino en général :

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE

Arduino. [2]

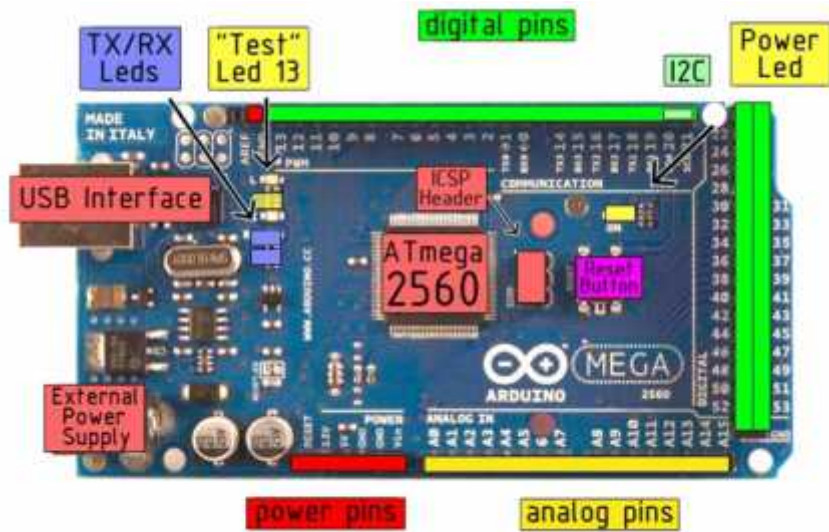


Figure 2 : Arduino Mega 2560

➤ **Synthèse des caractéristiques :**

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 (ATTENTION : 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 utilisée par le bootloader
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

➤ **Alimentation :**

La carte ArduinoMega 2560 peut être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

Les broches d'alimentation sont les suivantes :

- VIN. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- 5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- 3V3. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA
- GND. Broche de masse (ou 0V). The power pins are as follows:

De plus, certaines broches ont des fonctions spécialisées :

- **Communication Serie:** Port Serie Serial : 0 (RX) and 1 (TX); Port Serie Serial 1: 19 (RX) and 18 (TX); Port Serie Serial 2: 17 (RX) and 16 (TX); Port Serie Serial 3: 15 (RX) and 14 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Les broches 0 (RX) and 1 (TX) sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
- **Impulsion PWM (largeur d'impulsion modulée):** Broches 0 à 13. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite().
- **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

Côté entrées, des capteurs qui collectent des informations sur leur environnement comme la variation de température via une sonde thermique, le mouvement via un détecteur de présence ou un accéléromètre, le contact via un bouton-poussoir, etc...



Figure 3 : Différents types des capteurs pour l'Arduino

Pour les sorties, des actionneurs qui agissent sur le mode physique telle une petite lampe, un moteur, haut-parleur, etc.



Figure 4 : Différents actionneurs pour l'Arduino

Comme le logiciel Arduino, le circuit électronique de cette plaquette est libre et ses plans sont disponibles sur l'internet. On peut donc les étudier et créer des dérivés. Plusieurs constructeurs proposent ainsi différents modèles de circuits électroniques programmables et utilisables avec le logiciel Arduino.

"Seule le nom « Arduino » n'est pas utilisable librement. De telle sorte à toujours pouvoir identifier le projet de ses dérivés."

1.2.3. Moteur à Courant Continu

Un moteur à courant continu, ou CC, comporte essentiellement deux parties :

Une partie fixe appelée inducteur ou stator constitué par aimant ou un électroaimant qui crée un champ magnétique dirigé vers l'axe du rotor. Il est formé d'aimant pour les petits moteurs et des bobines parcourues par un courant continu.

Une partie mobile appelée induit ou rotor, dans laquelle a lieu la transformation de l'énergie électrique à l'énergie mécanique. Le rotor bobiné relie à un collecteur rotatif inversant la polarité dans chaque enroulement rotorique au moins une fois par tour de façon faire circuler un flux magnétique transversal en quadrature avec le flux statorique.

Les enroulements rotorique sont aussi appelés enroulements d'induits, ou communément induit. Les parties fixes et mobiles sont séparées l'une de l'autre par un entrefer. Dans ces deux parties, on peut distinguer le collecteur, pôles inducteurs et balais et porte 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. Les balais sont en graphite et constituent, en quelques sortes, la pièce d'usure. La graphite en s'usant libère une poussière qui rend le moteur à courant continu sensible à un entretien correct et donc coûteux. [3]

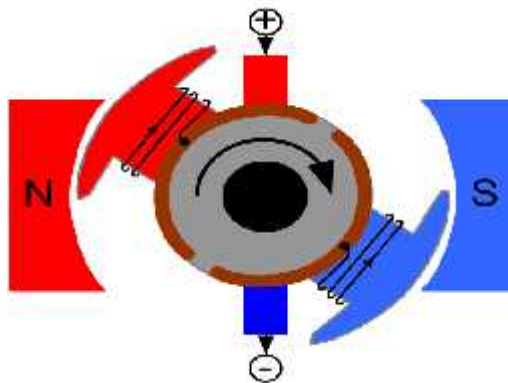


Figure 5 : les deux parties essentielles de MCC

1.2.4. Contrôle des moteurs par L298N

➤ Le pont en H

- Vous avez remarqué que votre moteur peut maintenant tourner plus ou moins vite grâce aux montages précédents. En revanche, il ne tourne que dans un sens.
- Forcément, si vous le connectez à l'envers, il tournera dans l'autre sens. Mais ce n'est quand même pas pratique de refaire les branchements à chaque fois .
- Il existe, en réfléchissant un peu, une méthode qui peut résoudre notre problème : elle consisterait à faire changer l'électricité de sens à volonté.

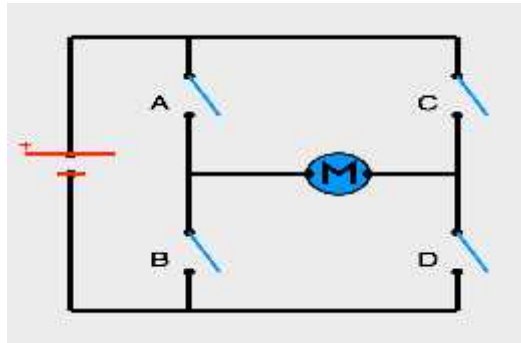


Figure 6 : Pont en H

- Le pont en H
- Voici ce qu'on appelle le pont en H. Il tient son nom de la forme en H du circuit autour du moteur.
- Ce circuit (ici réalisé avec des interrupteurs) permet de résoudre notre problème. Il faut tout de même être attentif à certaines erreurs possibles [4].

➤ Contrôle d'un ou deux moteurs cc avec L298 et Arduino

C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il peut contrôler deux moteurs courant continu ou un moteur pas-à-pas 4 fils 2 phases. Il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleur).

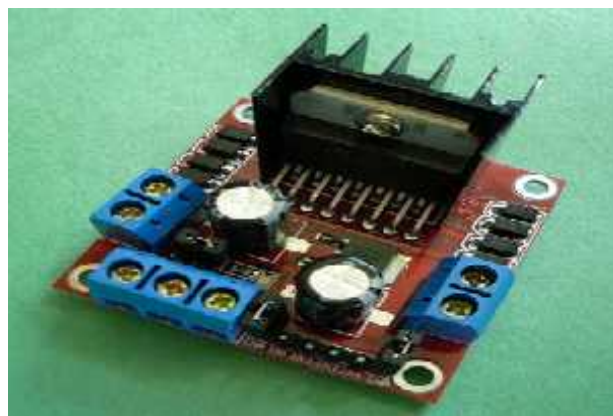


Figure 7 : Le L298N

Le contrôleur de moteur constitué d'un L298N et de tous les composants nécessaires à son bon fonctionnement (radiateur, diodes, résistances, condensateurs, connecteur).

En branchant cette petite carte à un microcontrôleur (qui sera, dans le présent article, une

carte Arduino), il est possible de contrôler simultanément deux moteurs en courant continu, ou un moteur pas à pas.

Description :

- Circuit d'attaque de pont complet double à courant de forte intensité, haute tension
- Deux entrées actives
- Circuit d'attaque de pont complet double ST L298N
- Alimentation électrique du circuit d'attaque : 5 à +46 V

Le Circuit d'attaque de pont en H double ST L298 :

Utilise le circuit d'attaque de pont complet double ST L298N, un circuit intégré monolithique dans un Multi watt 15 au plomb et des paquets PowerSO20. Il s'agit d'un circuit d'attaque de pont complet double conçu pour accepter des niveaux logiques TTL standard et conduire des charges inductives telles que des relais, des solénoïdes, des moteurs pas à pas et CC. Deux entrées actives sont fournies pour activer ou désactiver le dispositif indépendamment des signaux d'entrée. Les émetteurs des transistors inférieurs de chaque pont sont reliés entre eux et le bornier externe correspondant peut être utilisé pour la connexion d'une résistance externe de détection. Une entrée d'alimentation supplémentaire est fournie afin que la logique fonctionne à une tension inférieure.

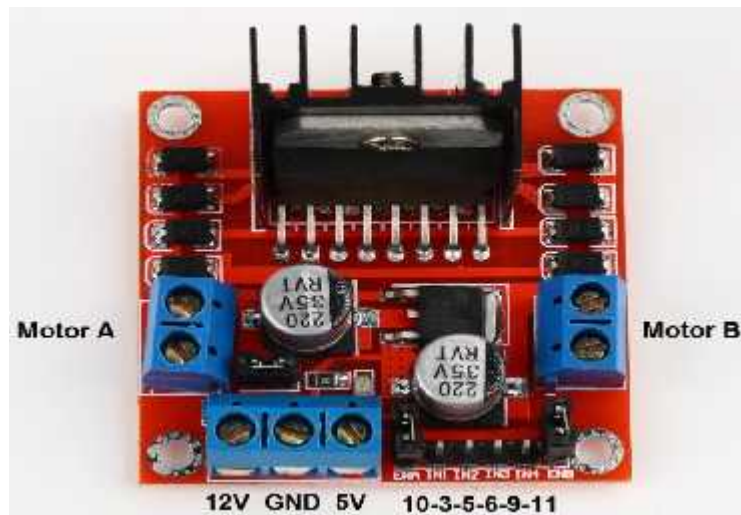


Figure 8 ; les différents composants du L298N

Caractéristiques :

- Léger, petit
- Des capacités hors-pair pour contrôle moteur
- Diodes de protections
- Un dissipateur (pour dissiper la chaleur en cas de forte charge)
- Un sélecteur pour sélectionner la source d'alimentation
- 4 Sélecteurs pour les résistances pull up
- Sortie pour 2 moteurs continu/ 1 moteur pas-à-pas (4 bobines, deux phases)
- Indicateur LED pour sens de rotation moteur
- Indicateur LED pour alimentation 5V
- 4 trous de fixation standard

1.2.5. Capteur distance :

L'ultrason est une onde acoustique dont la fréquence est trop élevée pour être audible par l'être humain.

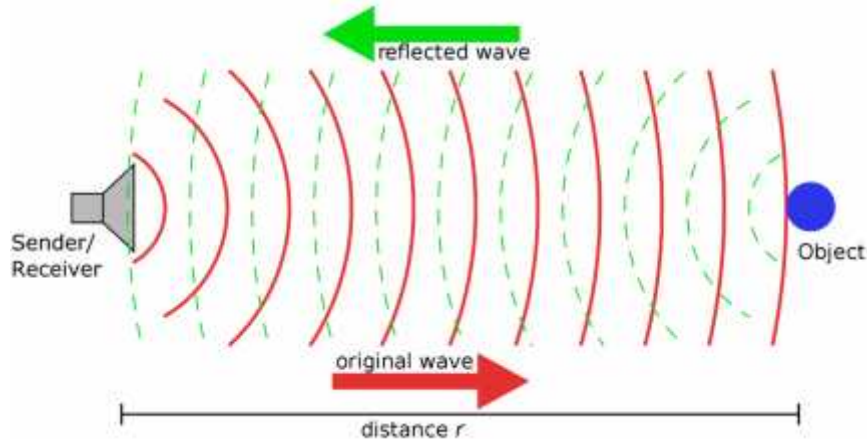


Figure 9 : Principe du capteur à ultrason

L'émetteur et le récepteur sont situés dans le même boîtier. L'émetteur envoie un train d'ondes qui va se réfléchir sur l'objet à détecter et ensuite revenir à la source. Le temps mis pour parcourir un aller-retour permet de déterminer la distance de l'objet par rapport à la source. Plus l'objet sera loin plus il faudra long temps pour que le signal revienne. [6]

➤ Capteur de distance à Ultrasons HC-SR04



Figure 10 : le capteur de distance à Ultrason HC-SR04

Un des deux petits cylindres sur la platine envoie des ultrasons, l'autre cylindre récupère ceux qui reviennent suite à la collision avec un objet. En fonction du temps que l'onde a mis pour revenir, nous pourrions savoir la distance qu'elle a parcourue. [7]

➤ Le fonctionnement :

Il faut envoyer une impulsion niveau haut (à +5V) pendant au moins 10µs sur la broche 'TrigInput' ; cela déclenche la mesure. En retour la sortie 'Output' ou 'Echo', va fournir une impulsion +5V dont la durée est proportionnelle à la distance si le module détecte un objet. Afin de pouvoir calculer la distance on utilisera la formule suivante :

$$Distance = \frac{\text{durée d'impulsion en } \mu s}{58}$$

Les caractéristiques techniques du module sont les suivantes :

- Tension de fonctionnement : 5Vdc
- Courant de fonctionnement : moins de 2mA
- Signal de sortie en tension (0-5Vdc)
- Angle max de capteur : 15°
- Plage de détection : 2cm - 450cm
- Haute précision : 0,3cm
- Fonctionnement à l'impulsion sur la pin Trigger
- Largeur de pulse de trigger : 10µs
- Sortie sur la pin Echo, durée de niveau haut fonction de la distance mesurée



Figure 11 : Le brochage du module

1.3. Software :

1.3.1. Plateforme de programmation Arduino :

➤ Partie programme :

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

➤ L'environnement de la programmation :

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino. [8]

➤ Structure générale du programme (IDE Arduino) :

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

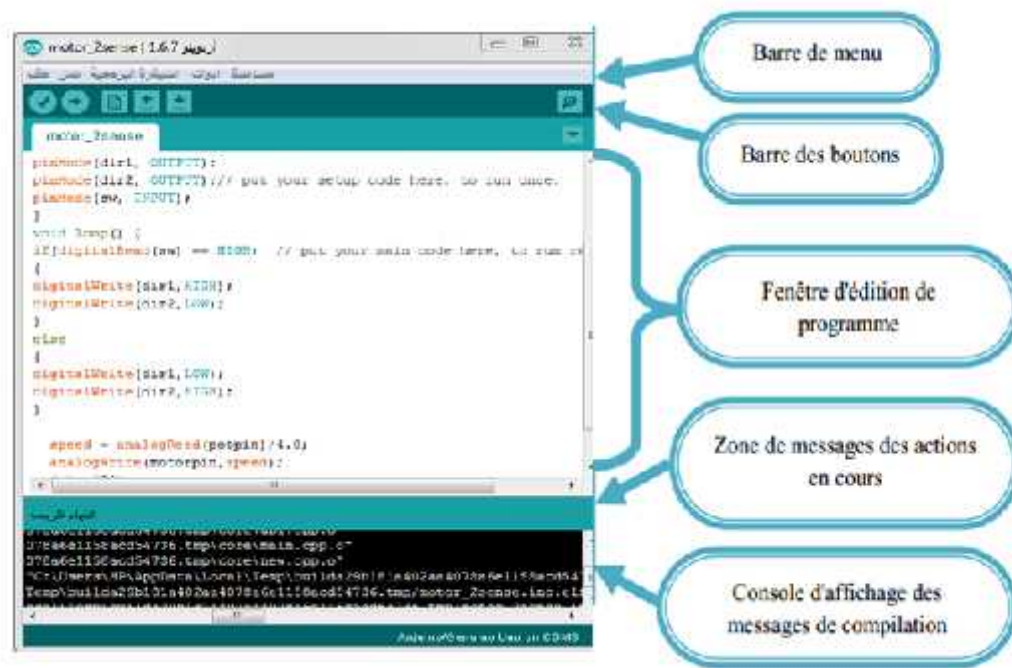


Figure 12 : Interface IDE Arduino

1.3.2. Utilisation de programme :

- Contrôleur hôte Unités.
- Donner "intelligence" pour Androïde.
- Faire des chaînes légères.
- PC a été en mesure de communiquer avec les cartes électroniques, et divers capteurs.
- Commande à distance à partir d'un appareil mobile (modèle).
- Simuler le fonctionnement des portes logiques.
- Permettre l'utilisation de différents capteurs.
- La mise en œuvre et de faciliter la compréhension du réseau informatique.

1.4. Conclusion :

Ce chapitre consiste une présentation générale des outils matériel et logiciel utilisé dans ce travail, à savoir, le système à microcontrôleur (Arduino) et le pré actionneur L298N ainsi que les capteurs ultrason (HC-SR04), etc.

Le chapitre suivant consiste à présenter les différentes techniques connues utilisées à l'évitement d'obstacles.

2. Évitement d'obstacles

2.1. Introduction :

L'évitement d'obstacles est un comportement de base présent dans quasiment tous les robots mobiles. Il est indispensable pour permettre au robot de fonctionner dans un environnement dynamique et pour gérer les écarts entre le modèle interne et le monde réel.

Les méthodes que nous présentons sont efficaces à condition d'avoir une perception correcte de l'environnement. Elles seront par exemple très efficaces avec un télémètre laser, mais donneront des résultats plus bruités avec des sonars.

2.2. Les méthodes d'évitement d'obstacle

2.2.1. méthode de champ potentiel

- Calculer la somme vectorielle des vecteurs répulsifs (distance gauche et droite) et d'attraction (avance vers l'avant)
- Calculer l'intervalle de sécurité *distMax* (un tiers de la distance *vecteurY*)
- Si *vecteurY* est dans l'intervalle $[-distMax, +distMax]$
 - Alors avancer
- Sinon, si *vecteurY* est vers la droite
 - Alors avancer à gauche
- Sinon, si *vecteurY* est vers la gauche
 - Alors avancer à droite
- Si le *vecteurX* est négatif
 - Alors arrêter la voiture.

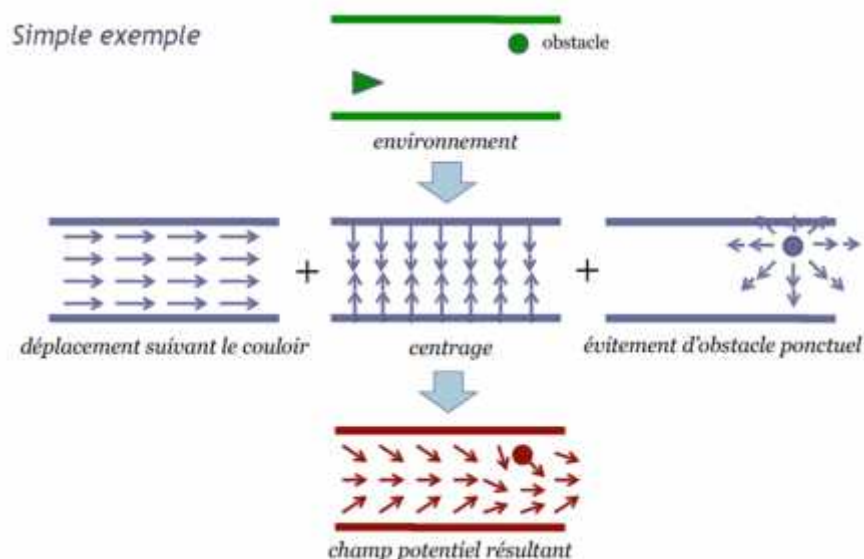


Figure 13 : évitement obstacle avec méthode champ potentiel.

2.2.2. Evitement par régulation PID

- Principe de la régulation PID :

- Calculer la différence entre la distance à droite et à gauche (*dif*)
- Calculer la commande *out* pour assurer $dif = 0$
- Si *out* est positive
 - Alors avancer vers la droite
- Si *out* est négative
 - Alors avancer vers la gauche

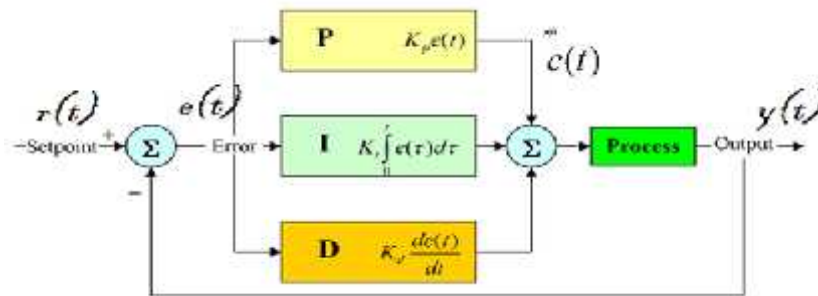


Figure 14 : Diagramme de commande de PID

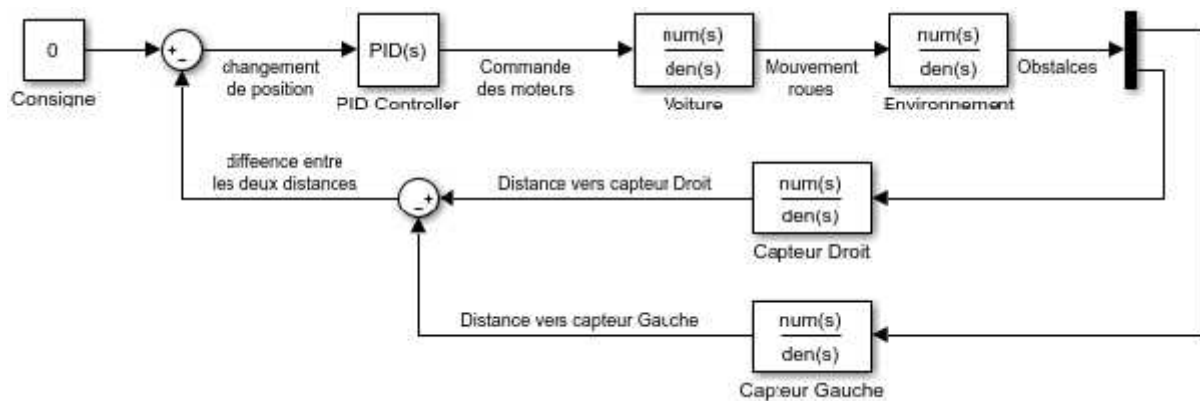


Figure 15 : boucle de régulation de projet.

2.2.3 méthode de TOR (Tout Ou Rien)

- Calculer la différence entre la distance à gauche et à droite (*dif*)
- Calculer l'intervalle de sécurité *distMax* (un tiers de la distance *dif*)
- Si *dif* est dans l'intervalle de sécurité $[-distMax, +distMax]$
 - Alors avancer
- Si *dif* est hors l'intervalle et vers la droite
 - Alors avancer à gauche
- Si *dif* est hors l'intervalle et vers la gauche
 - Alors avancer à droite

2.3. La Communication

Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de communications, on peut éclaircir actuellement quelques types. [11]

2.3.1. Le module Arduino Bluetooth

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion série Bluetooth à la place d'une connexion USB, très faible consommation d'énergie, très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant.



Figure 16 : Type de modules Bluetooth

2.3.2. Le module shield Arduino Wifi

Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.



Figure 17 : Module shield Wi-Fi

2.3.3. Le Module XBee

Ce module permet de faire de la transmission sans fil, faible distance /consommation /débit/ prix. [6]



Figure 18 : Module XBee

2.3.4. Infrarouge

Le rayonnement infrarouge est un rayonnement électromagnétique qui permet une connexion à l'aide d'onde de faible longueur. De ce fait, une interaction entre deux appareils est possible, à la seule est unique condition que les appareils en question soient proche et l'un en face de l'autre. La liaison reste cependant

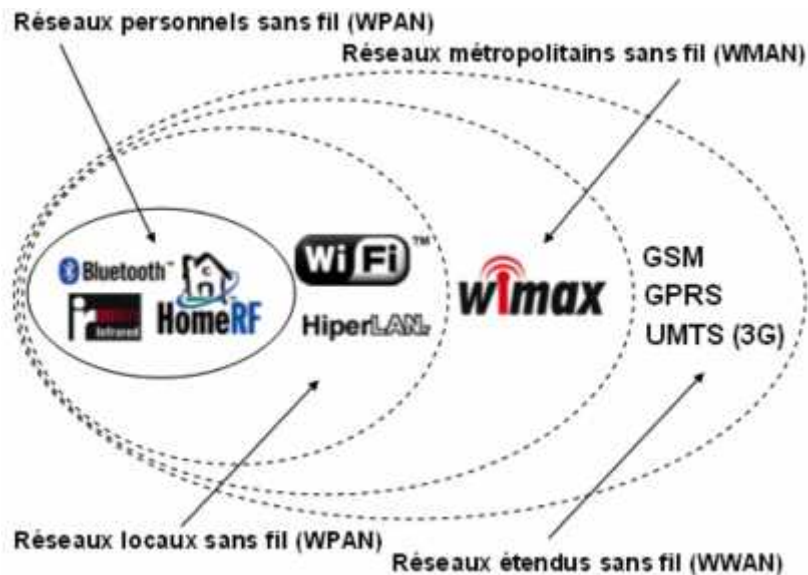


Figure 19 : Zone de couverture des différents réseaux sans-fil.

2.4. Conclusion

Ce chapitre est consacré à la présentation des techniques utilisées pour l'évitement d'obstacles ainsi que les différents modules utilisés pour la communication entre équipements Arduino et PC.

Le chapitre suivant consiste à la présentation des différentes étapes de la réalisation ainsi que les techniques de navigation implémentées.

3. Application

3.1. Introduction

Dans ce chapitre, nous expliquons la plateforme réalisée par étape, chaque fois nous ajoutons un dispositif et donnons le schéma électrique en utilisant la plateforme gratuite *Fritzing* [12] et plus expliquer le programme de branchement et en fin proposer la plateforme finale (moteurs, carte L298N, capteurs ultrason, Arduino, etc.).

Et expliquons la communication sans fil (par Bluetooth) avec deux méthodes, avec PC par programme Matlab et avec Smartphone réalisé avec le programme *Bluetooth Terminal* [13].

3.2. Réalisation de la plateforme

3.2.1. La connexion entre la carte 298l et le de moteur et Arduino :

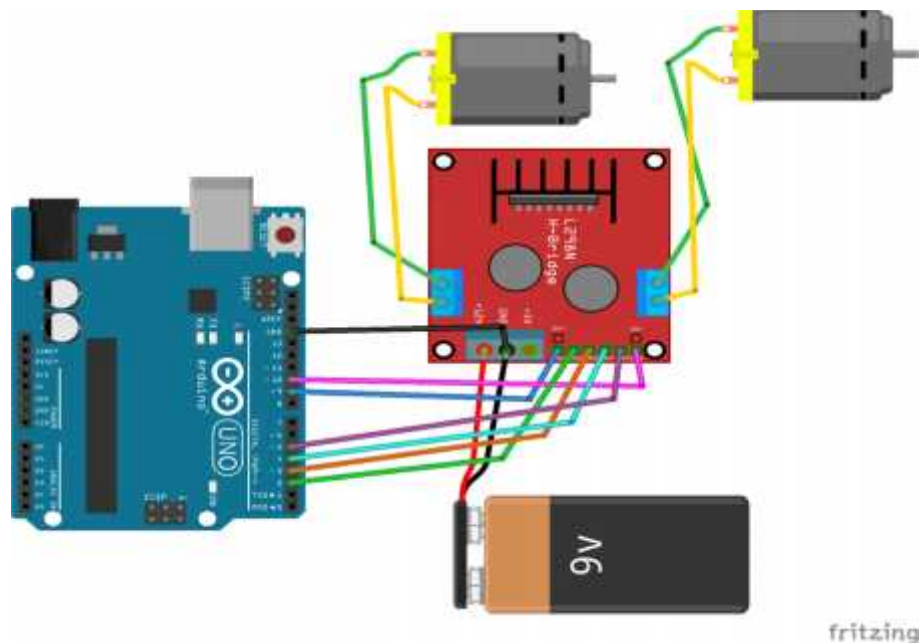


Figure 20 : La connexion entre la carte 298l et les deux moteurs et Arduino

Le branchement :

- enA : à relier au la sortie 10 de Arduino
- enB : à relier au la sortie 5 de Arduino
- in1 : à relier au la sortie 9 de Arduino
- in2 : à relier au la sortie 8 de Arduino
- in3 : à relier au la sortie 7 de Arduino
- in4 : à relier au la sortie 6 de Arduino
- 12v : à relier au le 9v de la batterie
- GND : à relier au GND de Arduino et GND de la batterie

3.2.2. La connexion entre l'Arduino et le capteur ultrason :

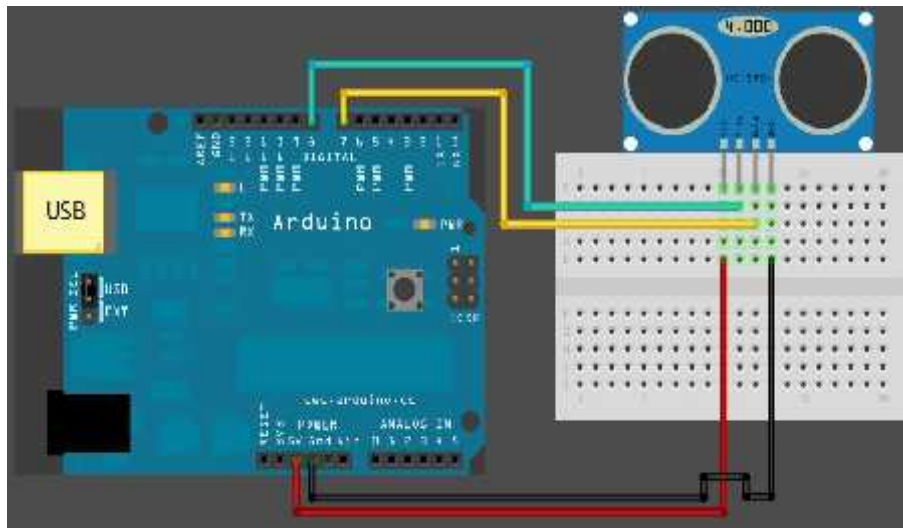


Figure 21 : La connexion entre l'Arduino et le capteur ultrasons

Le branchement :

En utilisé réellement deux capteur ultrason gauche (Left) et droit (Right)

- VCC : à relier au 5v de Arduino
- GND : à relier au GND de Arduino
- TrigR : à relier au l'entrée 22 de Arduino
- TrigL : à relier au l'entrée 24 de Arduino
- EchoR : à relier au sortie 23 de Arduino
- EchoL : à relier au sortie 25 de Arduino

3.2.3. La connexion entre l'Arduino et la carte HC_05 :

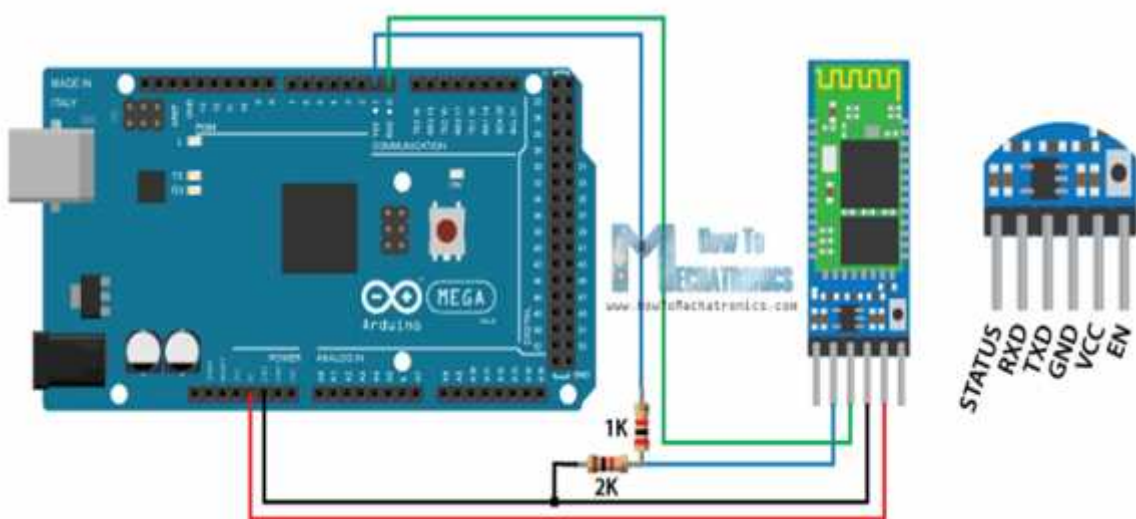


Figure 22 : La connexion entre l'Arduino et la carte HC_05

➤ **Le branchement :**

Il y a 6 pattes dont 4 uniquement sont utilisées couramment. Dans l'ordre :

- KEY : non utilisé (sauf configuration du HC05).
- VCC : alimentation en 5V du Arduino.
- GND : à relier au GND du Arduino.
- TXD : à relier au RX du Arduino. Le signal émis vers Arduino est de 0 ou 3.3V mais cela suffit.
- RXD : à relier au TX du Arduino, mais par le biais d'un diviseur de tension. En effet, la tension d'entrée acceptée est de 0 ou de 3.3V, alors que le TX de l'Arduino émet du 0 ou du 5V.
- STATE : non utilisé

3.2.4. Le schéma global

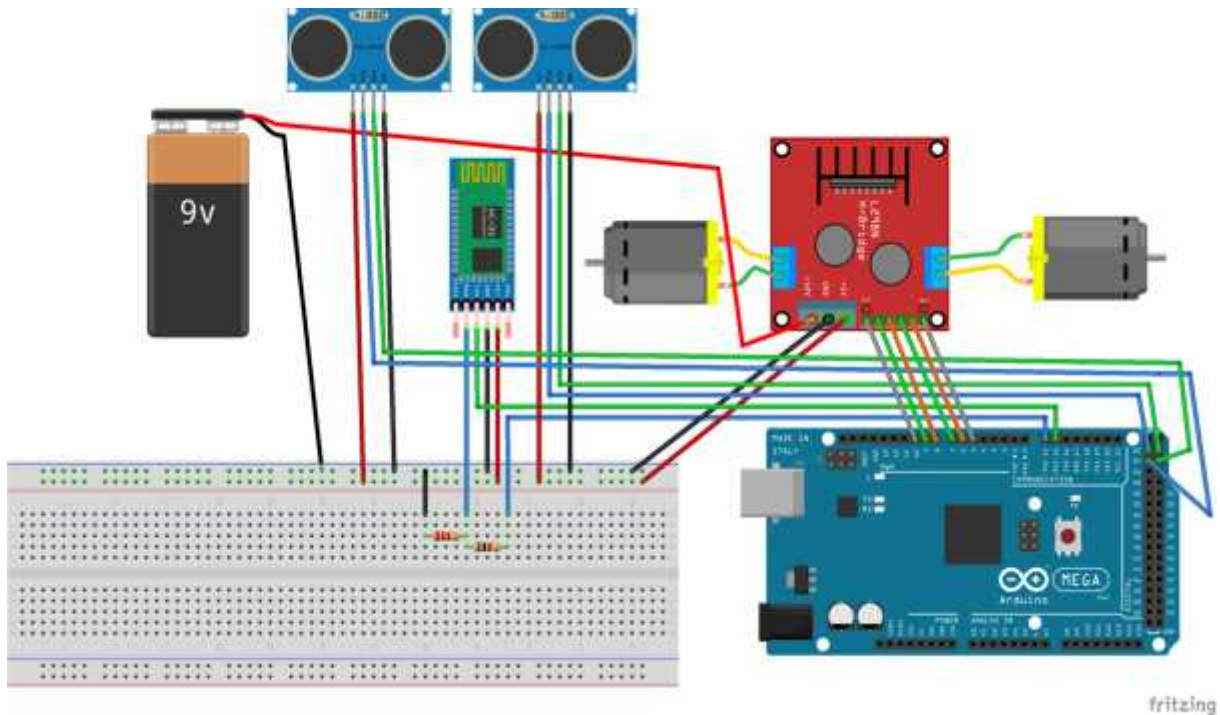


Figure 23 : Le schéma globale

3.3. Communication avec PC

3.3.1. La commande par Bluetooth en utilisant Matlab

En utilisant la fenêtre de GUIDE. Après avoir enregistré la fenêtre, nous pouvons obtenir le programme automatique pour réaliser l'objectif visé Il doit y avoir une modification dans le programme.



Figure 24 : Interface Matlab implémenté

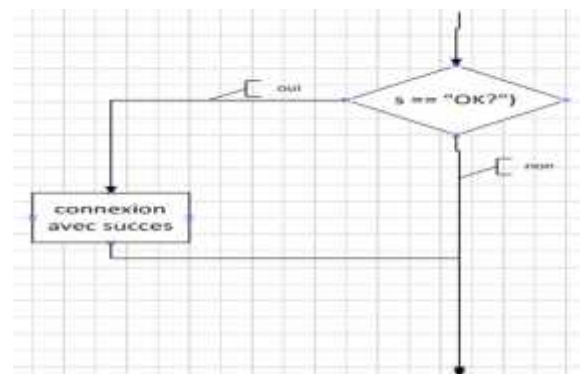
- 1) Graphe des distances
- 2) Communication Bluetooth
- 3) Technique d'évitement d'obstacles
- 4) Paramètres du régulateur PID
- 5) Commande manuelle
- 6) Application de la technique d'évitement d'obstacles
- 7) Zone d'affichage des messages d'erreur, notification, etc.

3.3.2. Le programme de commande

➤ La fonction de vérification de la connexion

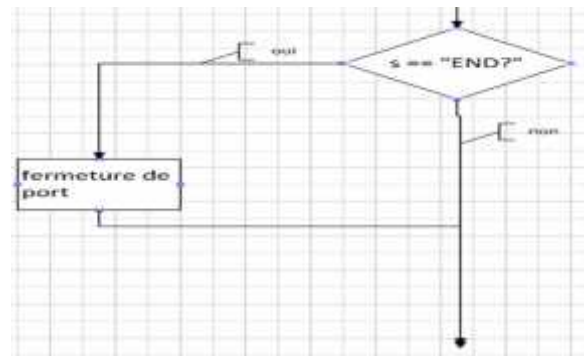
Si(s) égal à ("OK?")

Alors la connexion avec succès



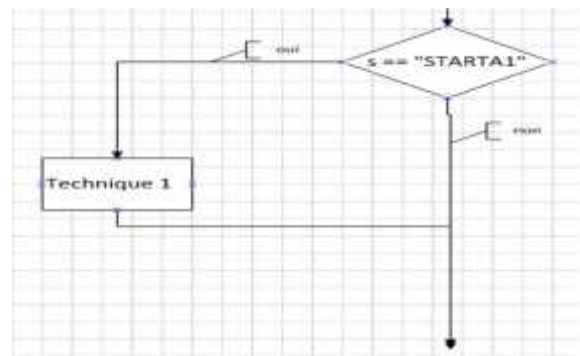
➤ **La fonction de fermeture de port**

Si (s) égal à ("END")
Alors fermeture de port



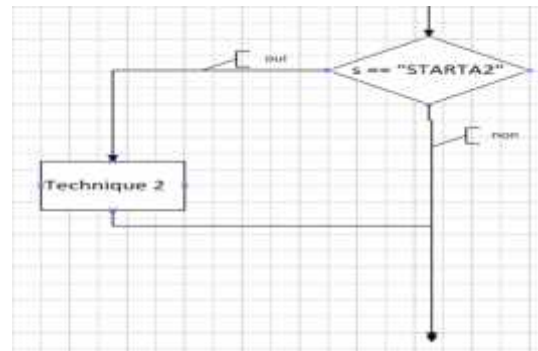
➤ **La fonction de démarrage de la navigation par la technique 1 (Tout ou Rien)**

Si (s) égal à ("STARTA1")
Alors on utilise la technique 1



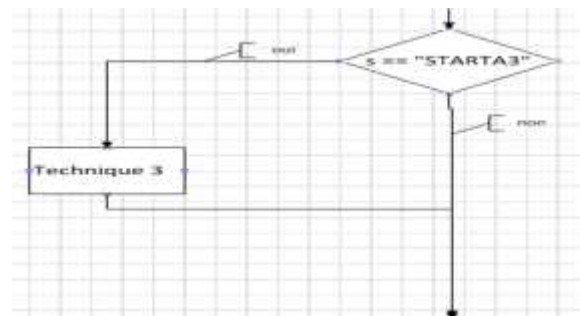
➤ **La fonction de démarrage de la navigation par la technique 2 (champ de potentiel)**

si (s) égala ("STARTA2")
Alors en utilise la technique 2



➤ **La fonction de démarrage de la navigation par la technique 3 (régulateur PID)**

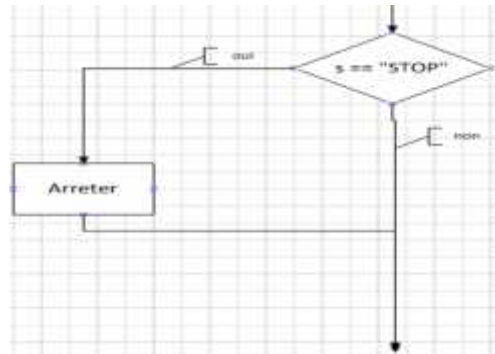
si (s) égala ("STARTA3")
Alors en utilise la technique 3



➤ **La fonction d'arrêt de voiture**

Si (s) égala ("STOPA")

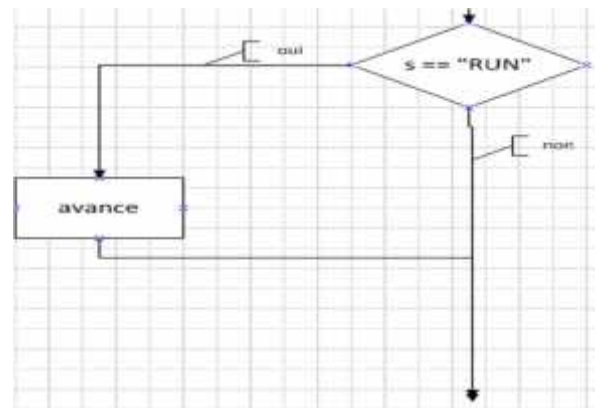
Alors arrête le voiture



➤ **La fonction de marche avant de voiture**

Si (s) égala ("RUN")

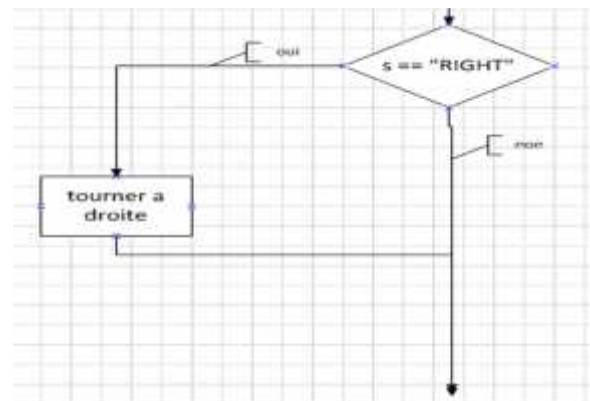
Alors avancé le voiture



➤ **La fonction de marche à droit**

Si (s) égal à ("RIGHT")

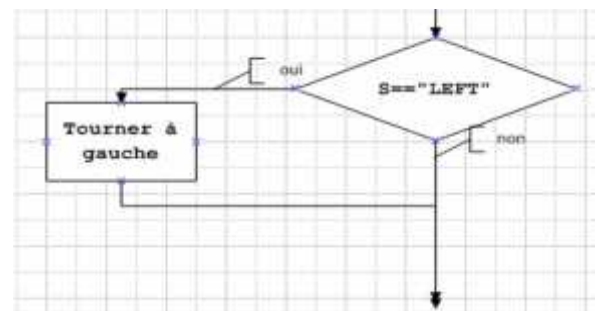
Alors marche à droit le voiture



➤ **La fonction de marche à gauche**

Si (s) égal a ("LEFT")

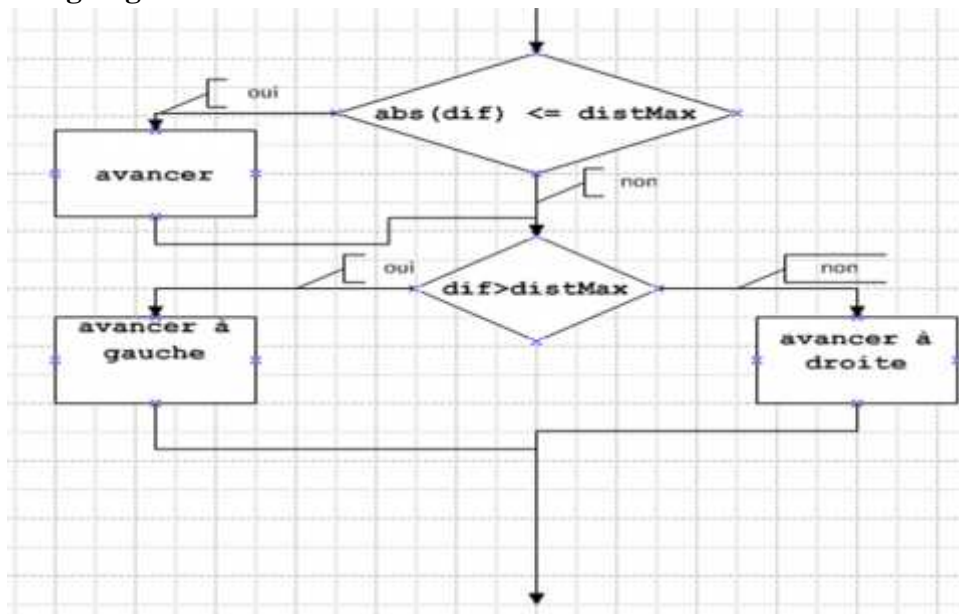
Alors marche à gauche le voiture



4. La fonction de détection de la distance par méthode TOR

- o Calculer la différence entre la distance à gauche et à droite (*dif*)
- o Calculer l'intervalle de sécurité *distMax* (un tiers de la distance *dif*)
- o Si *dif* est dans l'intervalle de sécurité $[-distMax, +distMax]$
 - Alors avancer
- o Si *dif* est hors l'intervalle et vers la droite
 - Alors avancer à gauche
- o Si *dif* est hors l'intervalle et vers la gauche
 - Alors avancer à droite

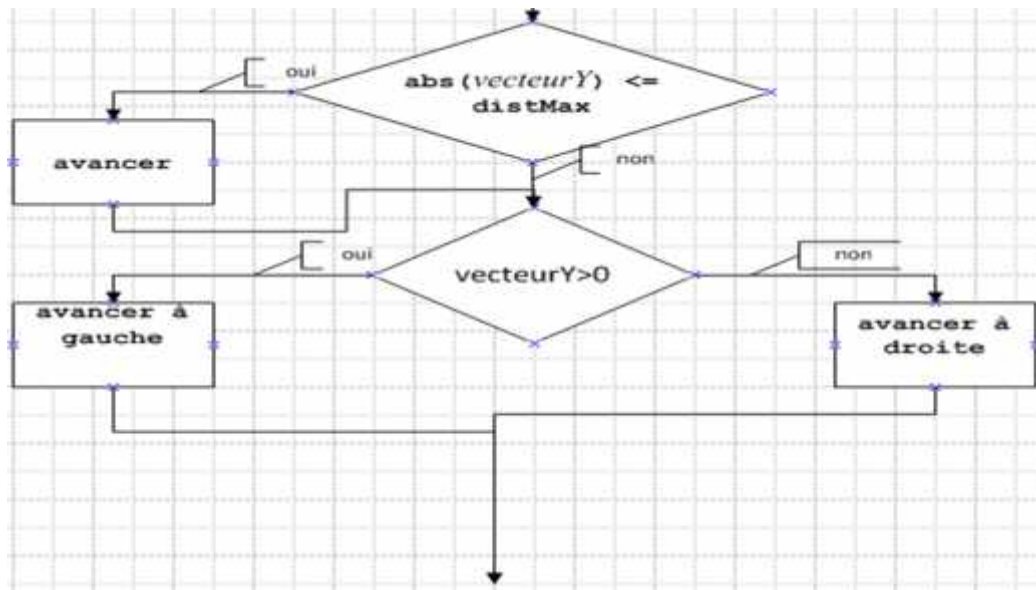
L'organigramme :



5. La fonction détectée la distance avec méthode potentiel

- o Calculer la somme vectorielle des vecteurs répulsifs (distance gauche et droite) et d'attraction (avance vers l'avant)
- o Calculer l'intervalle de sécurité *distMax* (un tiers de la distance *vecteurY*)
- o Si *vecteurY* est dans l'intervalle $[-distMax, +distMax]$
 - Alors avancer
- o Sinon, si *vecteurY* est vers la droite
 - Alors avancer à gauche
- o Sinon, si *vecteurY* est vers la gauche
 - Alors avancer à droite
- o Si le *vecteurX* est négatif
 - Alors arrêter la voiture

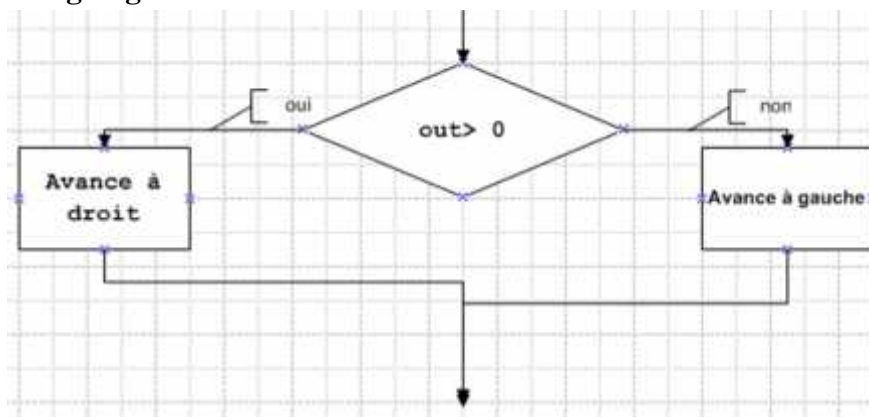
Organigramme :



➤ **La fonction détectée la distance avec méthode PID**

- o Calculer la différence entre la distance à droite et à gauche (*dif*)
- o Calculer la commande *out* pour assurer *dif* = 0
- o Si *out* est positive
 - Alors avancer vers la droite
- o Si *out* est négative
 - Alors avancer vers la gauche

L'organigramme :



➤ **La trajectoire de voiture visualisée avec GUI par méthode TOR**

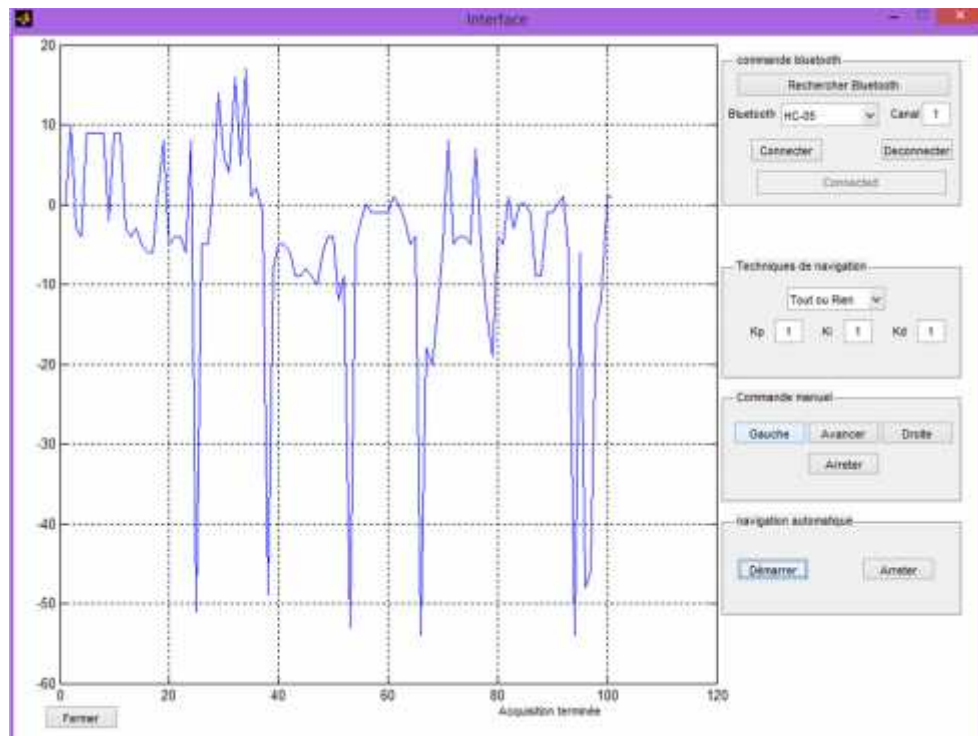


Figure 25 : La trajectoire de voiture visualisée avec GUI par méthode TOR

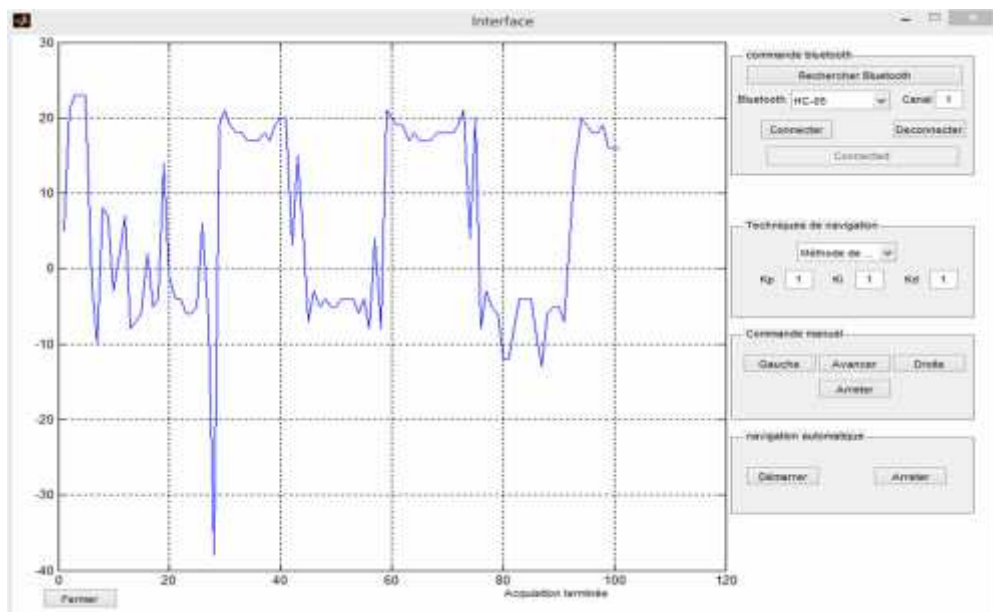


Figure 26 : La trajectoire de voiture visualisée avec GUI par méthode Champ de potentiel

➤ **Les étapes :**

- 1) Cliquez sur le bouton (Recherche Bluetooth) pour capté le Bluetooth de HC_05.
- 2) Cliquez sur le bouton (connecter).
- 3) Choisissez la méthode de fonctionnement manuel ou automatique (TOR ou potentiel ou PID).
- 4) Démarrer la voiture et affiché la trajectoire.

5.2. La commande par Bluetooth avec Smartphone

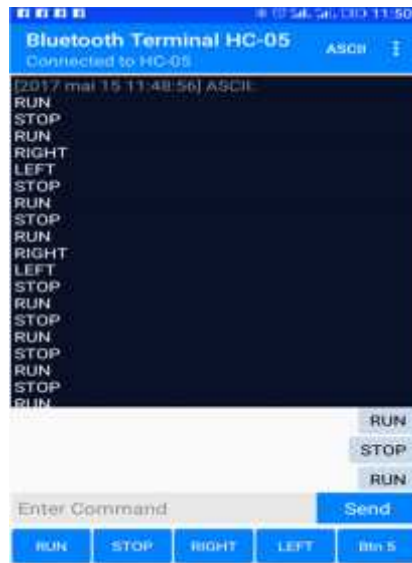


Figure 27 : La commande par Bluetooth en utilisant le Smartphone

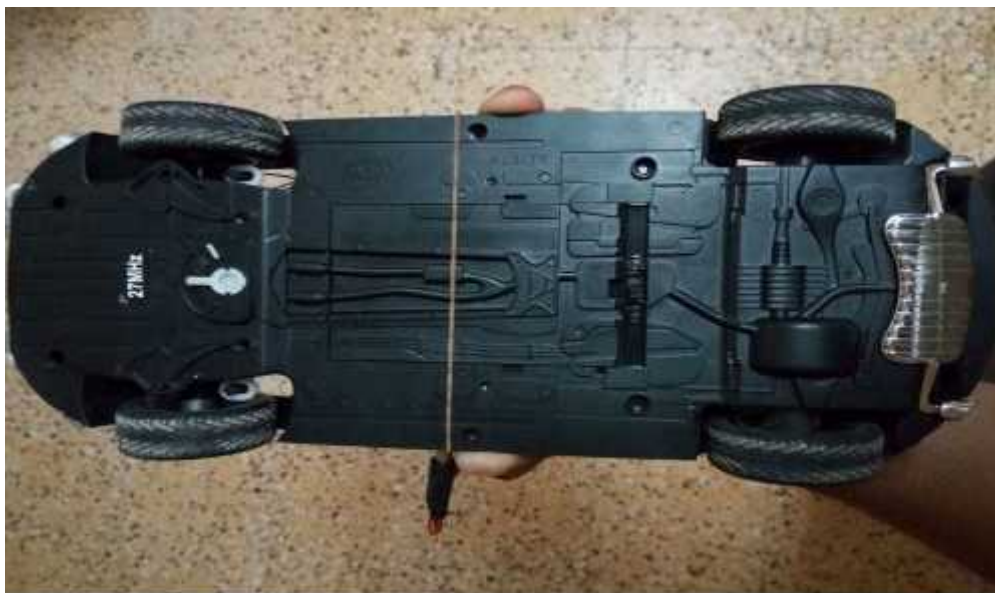


Figure 28 : La voiture réelle (vue du bas)



Figure 29 : La voiture réelle (vue de coté)

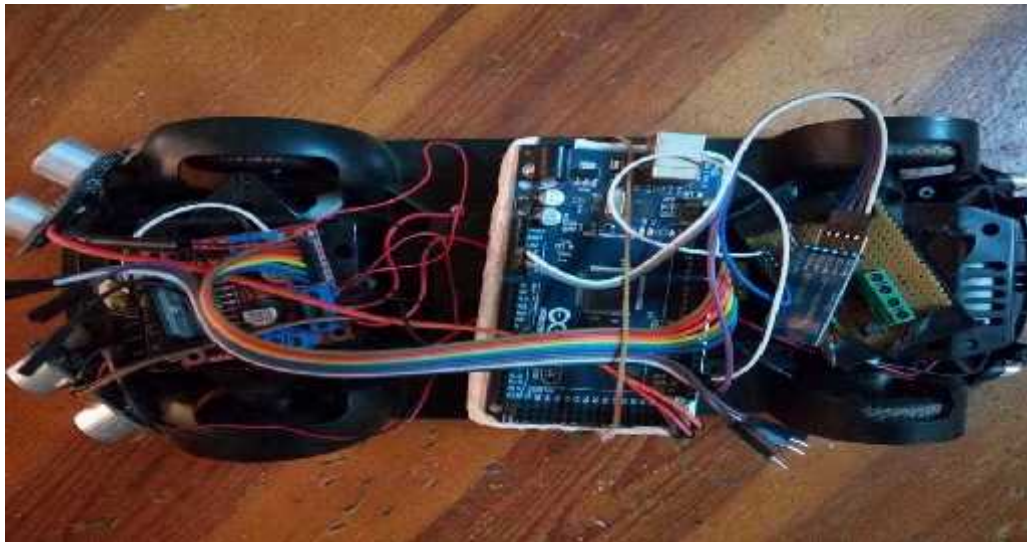


Figure 30 : le voiture complet (vue en haut)

Description :

1) Les dimensions :

- Longueur de voiture 18cm
- Largeur de voiture 12.3cm
- La distance entre deux capteurs 8 cm
- L'angle de rotation de roue
- La vitesse de voiture 125 cm/s
- Largeur de roue 1.5 cm
- Diamètre de roue 4.5 cm

2) L'alimentation :

- L'alimentation de deux moteurs par Cinq batteries de puissance 1.5v / $1.5 \times 5 = 7.5v$
- L'alimentation de L'Arduino : en utilisé l'alimentation externe batterie 9v pour donner l'énergie nécessaire.

- En utilisant le diviseur de tension dans la connexion entre l'Arduino et la carte Bluetooth Hc_05 pour adapter la valeur de signal traversant à partir de l'Arduino vers la carte HC_05
- Le fonctionnement de moteur avant est de changer de direction.
- Le fonctionnement de moteur arrière est le marché avant et marche arrière.

5.3. Comparaison de résultats :

Après exécution des techniques TOR, potentiellement remarquée les résultats suivants :

- La technique la plus précise est la technique de champ potentielle.
- La technique TOR c'est une technique bien mais n'est pas précise.

5.4. Conclusion :

Après réalisation de projet de système robotique, nous avons identifié certains points :

- Le branchement entre équipements utilisés.
- Comment traitons-nous une fenêtre de MATLAB (Guide) du côté des applications utilisées et comment modifier des programmes.
- Comment traitons-nous la fenêtre de programme Terminale Bluetooth et Comment changer les paramètres.
- Créer le programme Arduino dans chaque partie (communication, détection de distance, etc.) et en déduire le programme final.
- Régler l'énergie nécessaire de chaque composant pour Obtenir de bons résultats en termes de vitesse et de capacité.
- La comparaison entre les techniques réalisées et choisir la technique la plus intéressante.

Conclusion générale

Ce projet couvre un grand nombre de domaines :

- La partie hardware : on définit les composants généraux utilisé dans ce robot à partir des moteurs à courant continue, l'Arduino Mega, etc.
- La partie software : on définit le côté programmation de la carte Arduino et on définit le programme IDE pour comprendre comment commander la carte Arduino (et la voiture) par l'interface élaboré par Matlab.

L'objectif nécessaire dans ce robot est d'éviter les obstacles donc nous expliquons les méthodes d'évitement d'obstacles et créons le programme Arduino correspondant.

Ce projet nécessite la commande sans fil donc nous avons utilisé la communication sans fil par Bluetooth avec deux méthodes PC et smartphone et on définit la communication sans fil en général.

Ce travail couvre d'une façon bien détaillée la partie réalisation de la voiture et explique d'une façon générale les techniques d'évitement d'obstacles proposées. Par conséquent, un travail plus détaillé sur la différence entre les techniques de navigation, ainsi qu'une implémentation robuste de la technique de commande par régulateur PID est nécessaire dans le futur.

L'utilisation d'une voiture plus grande avec des actionneurs plus puissants est essentiel pour éliminer les problèmes d'alimentation et de commande de direction observée dans le modèle utilisé.

Bibliographie

- [1] modelisme-rc .voiture RC Thermique et Brushless . <https://modelisme-rc.net/3-voiture-radiocommandee-modelisme-rc>
- [2] Génération Robot. Arduino le technologie LDY. <http://www.generationrobots.com/fr/152-arduino> . Consulter le: mars 2015.
- [3] M .Keslenko et L .Piotrovski Machines électrique « machines à courant continu et transformateur» édition:1979.
- [4] Le pont en H et le circuit auteur de moteur. Moteur à courant continu(partie2). <https://openclassrooms.com/courses/programmez-vos-premiers-montages-avec-arduino/le-moteur-a-courant-continu-partie-2-le-pont-en-h-et-les-circuits-integres>
- [5] Arduino modules. L298N Dual H Bridge Motor Controller. <http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/>
- [6] Sensopart . capteur de mesure de distance .<http://www.sensopart.com/fr/produits/capteurs-de-mesure-de-distance>
- [7] Itechnofrance . utilisation du module ultrason HC-SR04 avec l'arduino. <https://itechnofrance.wordpress.com/2013/03/12/utilisation-du-module-ultrason-hc-sr04-avec-larduino/>
- [8] Jean- Noël, « livret Arduino en français » , centre de ressources art sensitif .
- [9] MIS. Navigation ; ch2; part 4.1 . http://home.mis.u-picardie.fr/~fabio/Eng/documenti/Teaching/LNR14-15/LNR_Ch2_partie4p1.pdf
- [10] Mémoire fin d'étude ingénieur en Automatismes « La commande de vitesse d'un Moteur à courant continu par logique floue » Université de l'USTO
- [11] Le nouvelliste. Quels sont les types de communication sans fil? . <http://lenouvelliste.com/lenouvelliste/article/157434/Quels-sont-les-types-de-communication-sans-fil> .
- [12] Fritzting, Paul-Linke-Ufer 39/40, 10999 Berlin, Germany. <http://fritzing.org/home/>
- [13] Bluetooth Terminal HC-05, mightyIT, 13, Harishnagar Society, Below Sardar Bridge, Adajan Patiya, Surat-395009. India, <http://www.memighty.com/>