

REPUBLIQUE DU CAMEROUN

Paix - Travail - Patrie

UNIVERSITE DE YAOUNDE I
ECOLE NORMALE SUPERIEUR
D'ENSEIGNEMENT TECHNIQUE
D'EBOLOWA
DEPARTEMENT DE DE GENIE
INFORMATIQUE



REPUBLIC OF CAMEROUN

Peace - Work - Fatherland

UNIVERSITY OF YAOUNDE I
HIGHER TECHNICAL TEACHER
TRAINING COLLEGE OF
EBOLOWA
DEPARTMENT OF OF
COMPUTER ENGINEERING

**Filière
Informatique Industrielle**

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

Mémoire rédigé et soutenu en vue de l'obtention du Diplôme de
Professeur d'Enseignement Technique deuxième grade (DIPET II)

Par : **MENDOUGA OYONO Nelly Paola**
Ingénieur des travaux informatiques Option génie logiciel

Sous la direction de
Pr. NDJAKOMO ESSIANE Salomé
Maitre de Conférences

Année Académique : 2019 - 2020



DEDICACE

*Ce document est
dédié à mes parents
Mr et Mme OYONO*

REMERCIEMENTS

C'est avec un grand plaisir que je pense à toutes les personnes qui ont cru en moi, qui m'encourage chaque jour à aller toujours en avant au-delà de mes compétences. Mes remerciements vont tout d'abord à l'endroit de la grande famille de l'ENSET d'Ebolowa, particulièrement à :

Pr SALOME NDJAKOMO ESSIANE, Madame Le Directeur de l'ENSET d'Ebolowa, superviseur et encadreur de notre mémoire qui n'a ménagé aucun effort pour nous offrir une formation de qualité ainsi que pour son expertise apportée, ses multiples conseils et attention à notre égard ;

Dr OLLE OLLE DANIEL, Chef de Département Génie Informatique pour son encadrement, sa présence, son soutien, son dynamisme, ses multiples conseils et remarques de nature à nous édifier ;

Mr NYATTE, chef d'équipe de notre encadrement, Enseignant à l'ENSET d'Ebolowa, pour sa présence, sa rigueur dans l'élaboration de ce document, sa rigueur et la transmission de ses connaissances ;

M. MEDZO membre important de notre équipe d'encadrement, Enseignant à l'ENSET d'Ebolowa, pour sa disponibilité, les connaissances et les critiques techniques dont il nous a fait bénéficier durant notre investigation ;

Tout le personnel actif et professoral de l'ENSET d'Ebolowa pour la formation enrichissante et nécessaire ;

A tous mes camarades de l'ENSET d'Ebolowa, en particulier à ceux du département informatique filière Informatique industrielle niveau 5 ;

Ma famille particulièrement mes parents et mes grands-parents pour leur soutien, présence, conseils tout au long de la présente année académique et leurs multiples encouragements pour faire de nous des personnes responsable et autonome vis-à-vis de la société ;

Tous mes frères et sœurs pour leurs esprits de solidarité, pour leur aide financière sans quoi nous ne serons arrivés au bout de ce travail ;

Toutes les personnes qui ont œuvré de près ou de loin dans l'accomplissement de ce mémoire.

SOMMAIRE

DEDICACE.....	i
SOMMAIRE	iii
LISTE DES FIGURES	iv
LISTE DES TABLEAUX	vi
LISTE DES ABREVIATIONS.....	vii
GLOSSAIRE.....	viii
AVANT-PROPOS.....	ix
RESUME.....	x
ABSTRACT	xi
INTRODUCTION GENERALE	1
CHAPITRE I : REVUE DE LA LITTERATURE SUR LA ROBOTIQUE MOBILE.....	3
I.Généralités sur la robotique mobile	4
II.LES METHODES INTELLIGENTES	17
III.ETAT DE L'ART	20
CHAPITRE II : METHODE DE CONCEPTION ET MODELISATION.....	25
I.CAHIER DE CHARGES FONCTIONNEL.....	26
II.METHODOLOGIE DE CONCEPTION	39
III MODÉLISATION DU ROBOT HEXAPODE	41
IV.CONCEPTION DU ROBOT.....	54
V.LA COMMANDE DU ROBOT	64
CHAPITRE III : RESULTATS ET INTERPRETATIONS.....	68
I.PLATEFORME DE PROGRAMMATION ARDUINO	69
II.SCHEMA DU MONTAGE ELECTRONIQUE.....	70
III.SIMULATION DE LA COMMANDE SOUS MATLAB.....	71
IV.PROTOTYPAGE	73
TABLE DES MATIERES.....	81

LISTE DES FIGURES

Figure I. 1:Interaction d'un robot avec son environnement [4]	5
Figure I. 2: La tortue de Grey Walter, source [6]	6
Figure I. 3: A gauche : Robot "Beast" de l'université John Hopkins dans les années 1960. A droite le robot Shakey de Stanford en 1969 [6]	6
Figure I. 4: Le Stanford Cart date de la fin des années 1970 et Le robot Hilare du LAAS [6]	7
Figure I. 5:Architectures de contrôle pour les robots mobiles [7]	8
Figure I.6: Robot de type uni cycle [2]	10
Figure I. 7: Robot de type tricycle [7].....	10
Figure I. 8: Robot mobile de type voiture [7].....	11
Figure I. 9: Robot mobile omnidirectionnel [7].....	11
Figure I.10:Robot mobile à chenilles [8]	12
Figure I.11: Robots marcheurs [5].....	13
Figure I.12:Robot mobile rampant [2]	13
Figure I.13: Exemples de robots commerciaux ou de recherche [3].....	14
Figure I. 14: Représentation de l'apprentissage automatique [8].....	17
Figure 15: Représentation d'un Neurone biologique et d'un Neurone artificiel [8]	18
Figure I. 16: Architecture de contrôle des robots mobiles[4].....	20
Figure I.17: Simulation d'un robot avec évitement d'obstacles[7].....	22
Figure I.18: Simulation d'un robot suiveur de ligne avec évitement d'obstacles [9].....	23
Figure I.19: le diagramme bête à corne exprimant le besoin du robot.....	28
Figure II.20: Diagramme Pieuvre	31
Figure 21: Le diagramme Pieuvre	31
Figure II.22: Le diagramme FAST du robot	36
Figure II.23: Schéma de la chaine fonctionnel du robot	37
Figure II.24 :Organigramme de la chaine fonctionnel [9].....	38
Figure II.25 : Schéma fonctionnelle du mouvement du robot	39
Figure II. 26: Schéma fonctionnel du mouvement du robot.....	40
Figure II.27:choix de la patte du robot hexapode [5].....	41
Figure II.28 : Définition du MGD [8].....	42
Figure II.29 : Notation associé à une chaine arborescente [8].....	43
Figure II.30: Paramètres géométrique pour un corps à plus de deux articulations [8].....	44
Figure II.31: La chaine cinématique d'une patte [8]	46
Figure II.32 : Chaine cinématique d'un banc d'essai de la patte [8].....	47
Figure II.33: MGD d'une patte pour $\theta_1 = 0, \theta_2 = 0, \theta_3 = \pi/4$ [8]	50

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

Figure II.34: Schéma du MGD pour l'hexapode pour $\theta_1 = 0, \theta_2 = 0, \theta_3 = \pi/2$ [8].....	51
Figure II. 35: Le châssis de notre robot.....	56
Figure 36 : Vue de la carte Arduino MEGA 2560[8]	59
Figure II. 37: module HC-SR04 [5].....	60
Figure 38: Illustration du signal TRIGGER et ECHO [5].....	61
Figure 39: Brochage du HC-05 [6].....	63
Figure II. 40 SG90 9g Micro Servo	63
Figure II. 41: Synoptique du robot.....	64
Figure 42: Navigation avec évitement d'obstacles	67
Figure I. 43: Vue de l'environnement IDE Arduino	69
Figure 44; Schéma électronique du robot	71
Figure III.45: Schéma d'une patte sur Simulink	72

LISTE DES TABLEAUX

Tableau I. 1: Avantages et inconvénients des différents types de robots mobiles [7]12
Tableau I.2: Récapitulatif des domaines d'application des robots mobiles [7].....15
Tableau II.3:Caractérisation des fonctions32
Tableau II.4: Assignation des repères et définition des paramètres [8].....44
Tableau II.5: Repère de la patte et du banc d'essai.....47
Tableau II.6: Paramètres géométriques relatifs au mécanisme.....48
Tableau 7:Consommation en énergie de tout le montage.....57
Tableau 8; Tableau comparatif Arduino[8]58
Tableau II. 9: Caractéristiques de l'Arduino Mega 2560.....59

LISTE DES ABREVIATIONS

SIGLE	SIGNIFICATION
USB	Universal Serial Bus
PLI	La programmation Inductive logique
RN	réseau neuronal
ISO	International Standard Organization
LAAS	Laboratoire d'Analyse et d'Architecture de systèmes
I.A	Intelligence Artificielle
PLI	programmation Inductive logique
MGI	Les modèles géométriques inverse
MCI	Les modèles cinématique inverse
MDI	Les modèles dynamiques inverse
APTE	Application aux techniques d'entreprise
FAST	Functional Analysist system Technique

GLOSSAIRE

Robot hexapode : est un robot marcheur dont la locomotion est fondée sur trois paires de pattes.

L'autonomie du robot mobile : est une faculté qui lui permet de s'adapter ou de prendre une décision dans le but de réaliser une tâche même dans un environnement peu connu ou totalement inconnu.

Le diagramme Pieuvre : permet de définir les liens c'est-à-dire les fonctions de service entre le système et son environnement, il permet de recenser la plupart des fonctions du système.

AVANT-PROPOS

L'école normale supérieure d'enseignement technique est un établissement de l'université de Yaoundé I créée par le décret présidentiel N°2017 /586 DU 24 Novembre 2017, et dont la mission est la formation des enseignants des collèges et lycées d'enseignement technique industriel et commercial d'une part, et d'autre part le recyclage et le perfectionnement du personnel enseignant dans le cadre de la formation continue. L'entrée dans cet établissement se fait par voie de concours pour les classes de première année (titulaires du Baccalauréat) et en troisième année (titulaires du BTS, DUT) et LICENCE) et, par voie de sélection sur étude des dossiers (ingénieurs et retours sur titre). La formation se fait en deux cycles sanctionnés par un diplôme de professeur d'enseignement technique premier (DIPET I) et deuxième (DIPET II) grade respectivement à la fin du premier et second cycle.

A l'issue de la formation et conformément à l'arrêté ministériel N°03/ B4 du 26/11/1985, l'élève professeur en fin de cycle devra présenter devant un jury constitué d'enseignants, un projet de fin d'études du premier cycle ou du second cycle en vue de l'obtention du diplôme de professeur d'enseignement technique selon le niveau (3 ou 5). C'est dans ce cadre nous avons opté d'orienter le choix de notre étude dans le domaine de l'intelligence artificiel d'où nous avons proposé de présenter comme thème « **CONCEPTION ET REALISATION D'UN ROBOT HEXOPEDE COMMANDE PAR L'ALGORITHME DU Q-LEARNING** »

RESUME

Un robot hexapode est un système mécanique à six pattes qui permet d'étendre les sens et les capacités d'action des humains dans des environnements hostiles ou inaccessibles. Ce projet nous permet de plonger dans le monde du développement des systèmes intelligents autonome dans le cadre de la robotique mobile. L'objectif est de concevoir un robot mobile capable de naviguer de façon autonome étant en mesure de prendre des décisions pour effectuer des mouvements en fonction des informations sur sa position et sur l'environnement qu'il contourne. Pour ce faire, nous allons concevoir un langage de programmation (sous Arduino) capable d'assurer le mouvement du robot dans toutes les directions (avant, arrière, gauche et droite) à l'aide des capteurs, les actionneurs. Cependant, la réalisation implique des systèmes complexes d'où la nécessité de faire recours à l'intelligence artificielle en utilisant l'algorithme du Q-Learning.

Mots clés : robot, hostile, arduino, capteur, programmation, intelligence, artificielle, Q-Learning.

ABSTRACT

A hexapod robot is a six-legged mechanical system that expands human senses and action in hostile or inaccessible environments. This project allows us to delve into the world of the development of autonomous intelligent systems within the framework of mobile robotics. The objective is to design a mobile robot capable of navigating autonomously being able to make decisions to perform movements based on information about its position and the environment around it. To do this, we will design a programming language (under Arduino) capable of ensuring the movement of the robot in all directions (front, rear, left and right) using the sensors, the actuators. However, the realization involves complex systems hence the need to use artificial intelligence using the Q-Learning algorithm.

Keywords: robot, hostile, arduino, sensor, programming, intelligence, artificial, Q-Learning.

INTRODUCTION GENERALE

Depuis la guerre du Vietnam qui a coûté la vie de près de 57000 soldats américains, l'administration BUSH en 2001 a privilégié de nouveaux outils de combats tels que les drones¹ de combats qui permettaient de frapper des groupes ennemis tout en protégeant la vie des militaires. Mais les drones furent critiqués pour le manque de résultat et leurs conséquences psychologiques désastreuses dans les régions visées. La volonté d'économiser les vies des pilotes américains au Vietnam, popularisé par la guerre du Golfe les drones aériens sont l'illustration de la tendance occidentale à envoyer à la guerre le robot plutôt que l'homme [1]. « **L'avenir est aux espions automatiques volants aux robots terrestres, maritimes** » Cécile Jean-Jacques, *'la guerre des robots, les révolutions militaires de demain'* parue le 15.09.2006 aux Editions Monde réel. Aujourd'hui, les Etats-majors des Etats Unies et Européen regardent plus loin et voient dans la robotique l'avenir de la guerre. En 2011, l'Etat-major français envisage ainsi de déployer les premiers robots autonomes pour l'horizon 2035, d'où plusieurs robots terrestres aujourd'hui utilisés dans les armées dans le but de la déshumanisation dans les conflits d'où le coût s'avère largement inférieur au coût d'un soldat lors d'une opération.

En outre, les localités de la partie Ouest de la région de l'extrême-Nord du Cameroun notamment les départements du Mayo-Tsanaga, Mayo-Sava et le Logon et Chari, toutes frontalières au Nigeria subissent depuis février 2015 des attaques à répétition du groupe islamique Boko Haram à partir de ses positions Nigérianes ont causés une centaine de victimes civils et militaires camerounais et de millier de déplacés. Suite aux attentats du 29 juillet 2015 frappé à Maroua et Fotokol les autorités camerounaises ont annoncés une demande de renforcement de sécurité notamment dans le Nord du pays ; d'où notre intérêt à développer un système intelligent autonome dans le cadre de la robotique mobile qui va intervenir dans des tâches variées telles que naviguer sur des terrains accidentés, détecter les obstacles, éviter les obstacles, et enregistrer des images.

L'objectif de notre travail est d'étudier et d'appliquer des techniques de l'intelligence artificielle pour la navigation autonome d'un robot mobile dans un environnement inconnu, afin de permettre au robot de se mouvoir d'une position initiale à une autre finale en évitant les

¹ Aéronefs sans pilote dont le pilotage est automatique ou télécommandé, à usage civil ou au profit des forces armées ou de sécurité.

obstacles. Pour cela, on utilise l'approche comportementale à base de l'algorithme du Q-Learning pour l'optimisation du fonctionnement du robot. Les techniques employées pour aborder ce problème sont basées sur la technique d'apprentissage non supervisée.

La contribution de ce mémoire est principalement l'intégration de l'algorithme du Q-Learning dans la conception d'un robot autonome à pattes.

Afin d'atteindre nos objectifs, le travail a été divisé en trois chapitres à savoir la revue de la littérature, outils et méthodes et résultats et interprétations. Néanmoins nous commençons d'abord par situer le problème par une Introduction générale qui définit la problématique, détermine les hypothèses et présente l'objectif principal et la contribution de cette recherche.

Notre mémoire sera organisé de la manière suivante :

Le premier chapitre est consacré à la revue de la littérature sur la robotique mobile. Un aperçu général sur le domaine de la robotique mobile sera abordé pour examiner en bref les types des robots mobiles, les différentes parties constitutives et les architectures de contrôle existantes pour un robot mobile ; par la suite nous allons présenter les différentes méthodes d'intelligence artificielle et algorithmes de commande des robots. Nous ferons également une étude comparative des différents travaux effectués dans ce domaine.

Dans le deuxième chapitre, nous présentons la méthodologie de conception adoptée. En préambule, nous commençons par définir un cahier de charge fonctionnel du robot, par la suite nous proposons une méthodologie de conception de notre système, l'algorithme de commande du robot. Nous terminons cette partie en présentant la constitution de notre robot notamment le choix des différents outils matériels et logiciels.

Dans le troisième chapitre, nous allons interpréter les résultats de chaque simulation fournis ainsi que le prototype final réalisé.

Nous terminerons notre manuscrit par une conclusion générale récapitulant ce qui a été fait et les perspectives de notre travail.

CHAPITRE I : REVUE DE LA LITTERATURE SUR LA ROBOTIQUE MOBILE

Résumé : dans ce chapitre, il est question de parcourir les généralités sur la robotique mobile et de présenter les différents articles et documents liés à la robotique mobile.

Aperçu

INTRODUCTION

I. GENERALITES SUR LA ROBOTIQUE MOBILE

I.1 Définitions

I.2 Historique des robots mobiles

I.3 Architecture des robots mobiles

I.4 Classification des robots

II. LES METHODES INTELLIGENTES

II.1 Les méthodes d'intelligences artificielles

II.2 Les algorithmes de commande de robots

III. ETAT DE L'ART

CONCLUSION

INTRODUCTION

La robotique est devenue de nos jours un champ de recherche très varié, c'est une science qui s'intéresse aux robots. Il s'agit d'un domaine multidisciplinaire (mécanique, électronique, automatique, informatique). C'est également l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles. L'aspect particulier de la mobilité impose une complexité technologique et un traitement complexe des informations. L'autonomie du robot mobile est une faculté qui lui permet de s'adapter ou de prendre une décision dans le but de réaliser une tâche même dans un environnement peu connu ou totalement inconnu.

L'objectif de ce premier chapitre est de donner un aperçu général de la robotique mobile et de mettre en lumière toutes les notions théoriques essentielles à la conception d'un robot hexapode. Pour faciliter une meilleure compréhension de ce mémoire, nous allons parler des généralités sur la robotique mobile ; par la suite, nous présenterons les différentes méthodes de l'intelligence artificielle ainsi que les différents algorithmes de commande de robot.

A la fin de ce chapitre, nous présenterons les thématiques de recherches actuelles associées au domaine de la robotique mobile ayant abouti au développement total d'un robot mobile totalement autonome au Cameroun à ce jour. A partir des résultats obtenus de ces travaux effectués, nous présenterons notre apport.

I. Généralités sur la robotique mobile

I.1 Définitions

Le petit Larousse définit un robot comme étant un appareil automatique capable de manipuler des objets ou d'exécuter des opérations selon un programme fixe ou modifiable. En fait, l'image que chacun s'en fait est généralement vague, souvent un robot est défini comme un manipulateur automatique à cycle programmables.

Le terme Robot prend son origine du mot «**slave**» qui veut dire en russe travail ou corvée en tchèque. Il désigne aussi une machine à l'aspect humain, capable de se mouvoir et d'agir, d'effectuer certaines opérations, parfois de modifier lui-même son cycle de fonctionnement et d'exercer un certain choix. Par ailleurs, International Standard Organization

(ISO) affirme qu'un robot est « Une machine formée par un mécanisme incluant plusieurs degrés de libertés, ayant souvent l'apparence d'un ou plusieurs bras se terminant par un poignet capable de tenir des outils, des pièces ou un dispositif d'inspection ». La figure suivante est l'illustration d'un robot avec son environnement :

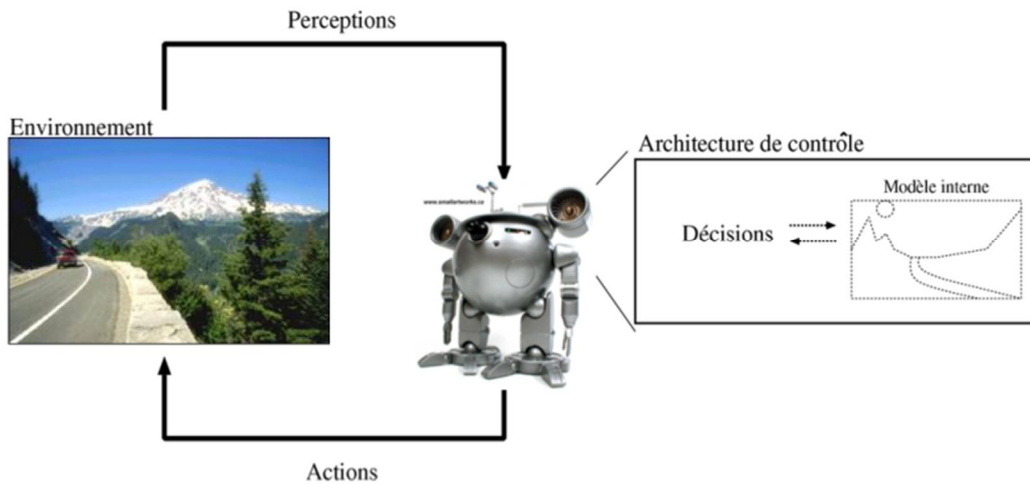


Figure 1.1: Interaction d'un robot avec son environnement [4]

Au regard de toutes ces définitions, nous constatons qu'elles sont énoncées autour de certains concepts qui pour la plupart reviennent. Ainsi pour compléter ces différentes définitions, nous pouvons dire qu'un robot est un système mécanique poly articulé mû par des capteurs et des actionneurs commandés par un ordinateur qui est destiné à effectuer une grande variété de tâches. Dès lors, il convient de parler de l'histoire des robots mobiles.

I.2 Historique des robots mobiles

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

La Tortue construite par Grey Walter dans les années 1950, est l'un des premiers robots mobiles autonomes. Grey Walter n'utilise que quelques composants analogiques, dont des tubes à vide, mais son robot est capable de se diriger vers une lumière qui marque un but, de s'arrêter face à des obstacles et de recharger ses batteries lorsqu'il arrive dans sa niche.

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

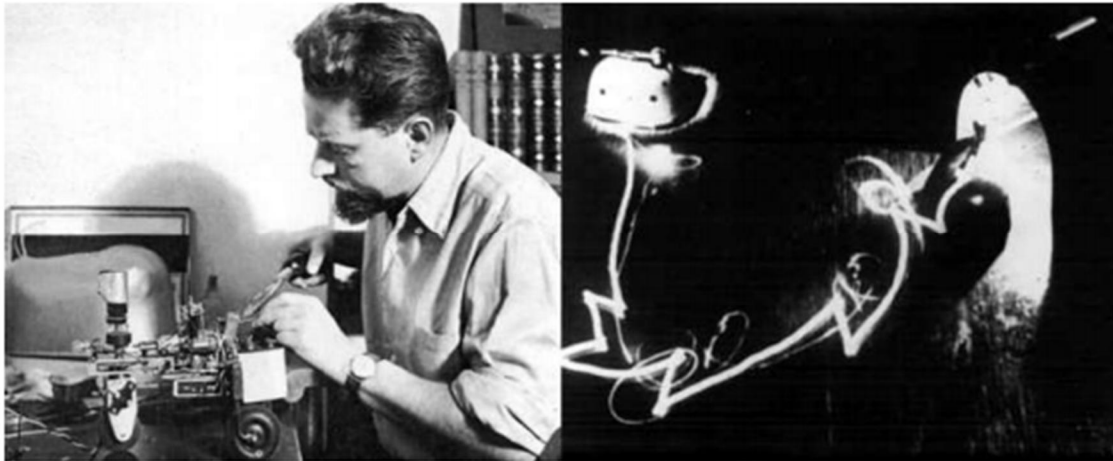


Figure 1. 2: La tortue de Grey Walter, source [6]

Dans les années 60, les recherches en électronique vont conduire, avec l'apparition du transistor, à des robots plus complexes mais qui vont réaliser des tâches similaires. Ainsi le robot "Beast" 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 photo-diodes 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.

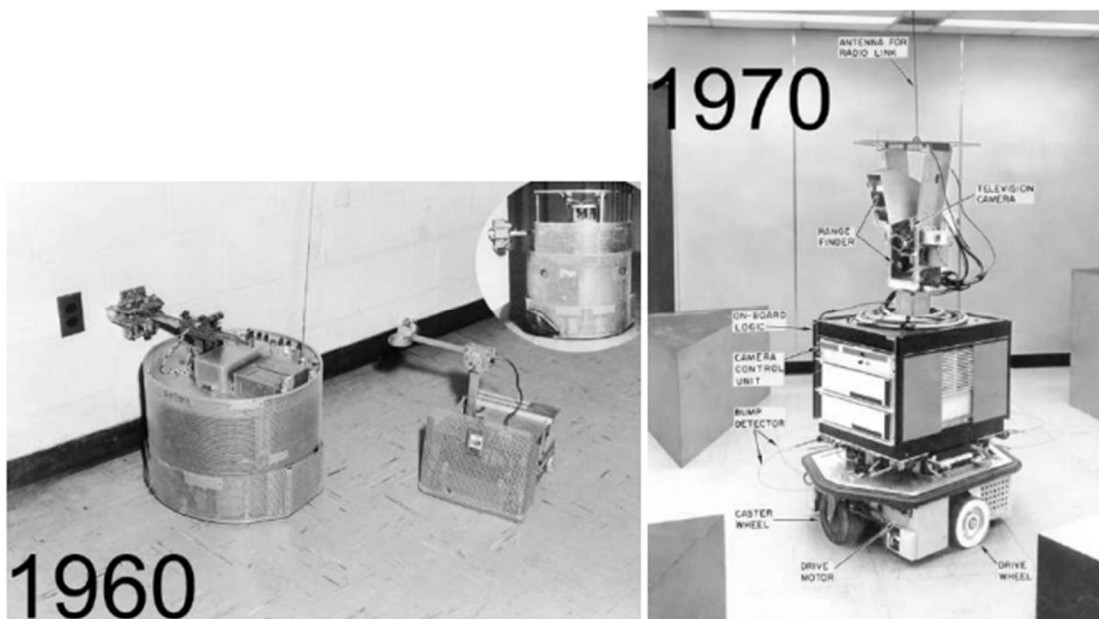


Figure 1. 3: A gauche : Robot "Beast" de l'université John Hopkins dans les années 1960. A droite le robot Shakey de Stanford en 1969 [6]

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

Ces développements se poursuivent avec le Stanford Cart dans la fin des années 1970, avec notamment les premières utilisations de la stéréo-vision pour la détection d'obstacles et la modélisation de l'environnement. En France, le robot Hilare est le premier robot construit au LAAS, à Toulouse de l'environnement. En France, le robot Hilare est le premier robot construit au LAAS, à Toulouse.

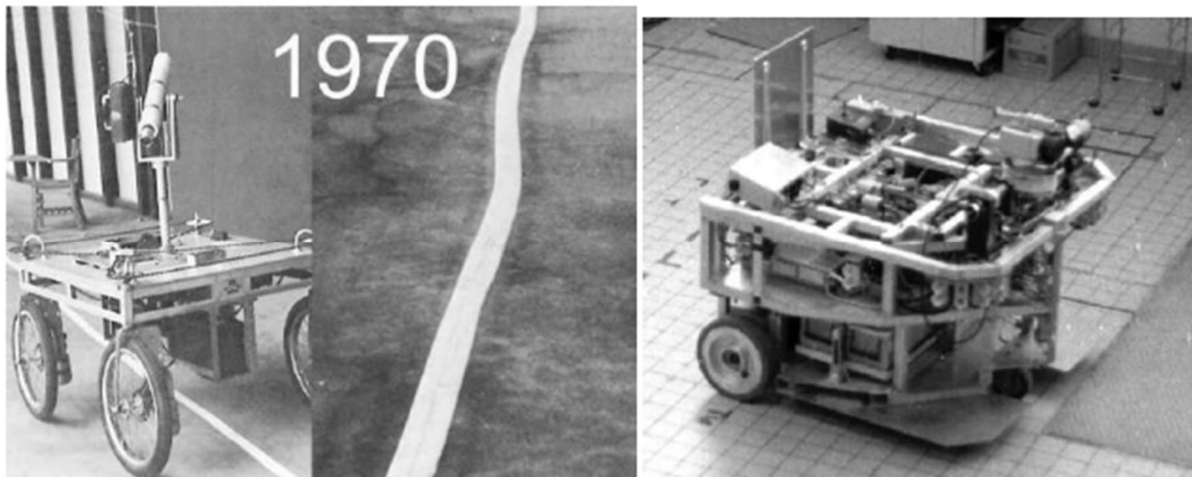


Figure 1. 4: Le Stanford Cart date de la fin des années 1970 et Le robot Hilare du LAAS [6]

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

Ces développements ont continué et l'arrivée sur le marché depuis les années 1990 de plates-formes intégrées telles que le pionnier de la société Mobile Robots a permis à de très nombreux laboratoires de travailler sur la robotique mobile et à conduit à une explosion de la diversité des thèmes de recherche. Ainsi, même si les problèmes de déplacement dans l'espace et de modélisation de l'environnement 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.

A présent la plupart des travaux de recherche portent sur les problèmes de perception. La planification de trajectoires, l'analyse et la modélisation de l'environnement de robot, appliqué sur des robots mobiles commerciaux. Également la recherche actuelle sur la conception

mécanique des robots mobiles pour des applications hautement spécialisées, comme l'exploration sous-marine, les robots volants et le micro robots. Dès lors il convient de présenter l'architecture des robots mobiles

I.3 Architecture des robots mobiles

Un robot est un système complexe qui doit satisfaire des exigences variées en utilisant un ensemble logiciel appelé architecture de contrôle du robot. Cette architecture permet donc d'organiser les relations entre les trois grandes fonctions: la perception, la décision (planification) et l'action. Ces architectures peuvent néanmoins être classées en trois groupes (approches) comme montré sur la figure suivante :

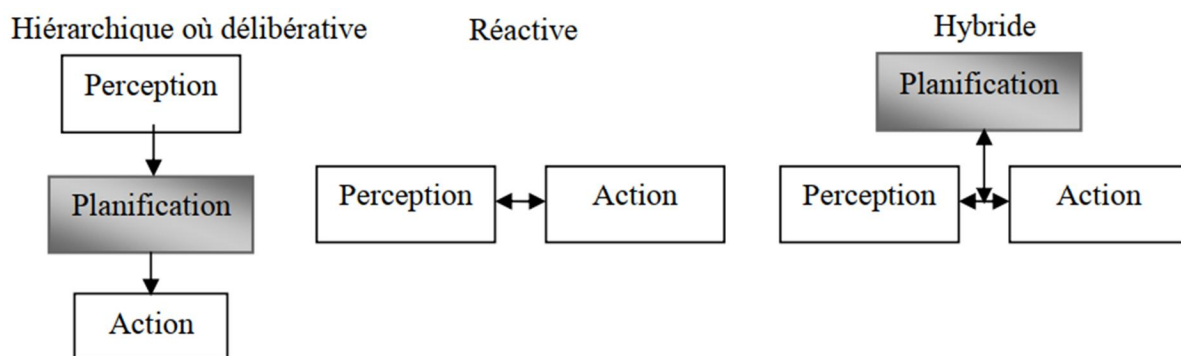


Figure 1. 5: Architectures de contrôle pour les robots mobiles [7]

Il est à noter également qu'un robot est constitué de trois structures :

- **Structure mécanique** : elle assure le mouvement du robot par des roues motrices placées selon le type de mouvement et la précision de la tâche voulue ;
- **Structure instrumentale** : un robot est équipé d'un certain nombre de capteurs de sécurité afin de leur donner une certaine connaissance de l'environnement ;
- **Structure informatique** : une commande numérique est impérative, afin de bien analyser les différentes informations, soit du système de perception ou de localisation. Cette commande peut être à base d'un microprocesseur ou microcontrôleur.

I.4 Classification des robots mobiles

On peut classer les robots mobiles selon leur degré d'autonomie, système de locomotion, leur domaine d'application, leur système de localisation, l'énergie utilisée. Nous allons présenter ici quatre classifications qui semblent être les plus intéressantes.

I.4.1 Classification selon le degré d'autonomie

Un robot mobile autonome est un système automoteur doté de capacités décisionnelles et de moyens lui permettant d'accomplir sous contrôle humain réduit un certain nombre de tâches dans un environnement non complètement connu. On peut citer quelques types :

- Véhicule télécommandé par un opérateur : Ces robots sont commandés par un opérateur qui leur impose chaque tâche élémentaire à réaliser ;
- Véhicule télécommandé au sens de la tâche à réaliser : ce type de véhicule contrôle automatiquement ses actions ;
- Véhicule semi-autonome : Ce type de véhicule réalise des tâches prédéfinies sans l'aide de l'opérateur ;
- Véhicule autonome : Ces derniers réalisent des tâches semi-définies.

I.4.2 Classification selon le type de locomotion

Selon le système de locomotion, on peut distinguer quatre types des robots :

a. Les robots mobiles à roues

La mobilité par roues est la structure mécanique la plus utilisée. Ce type de robot assure un déplacement avec une accélération et une vitesse rapide mais nécessite un sol relativement plat. On distingue plusieurs classes de robots à roues déterminées principalement par la position et le nombre de roues utilisées. Nous citerons ici les quatre classes principales de robots à roues :

- **Robot uni cycle** : Un robot de type uni cycle est actionné par deux roues indépendantes, il possède des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. C'est un robot non holonome. En effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion. Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites.

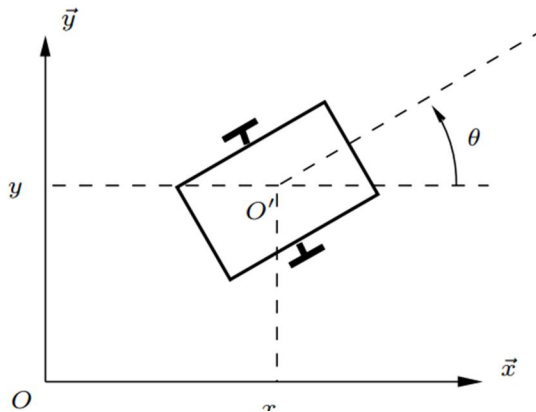


Figure I.6: Robot de type uni cycle [2]

- **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. C'est un robot non-holonyme. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable

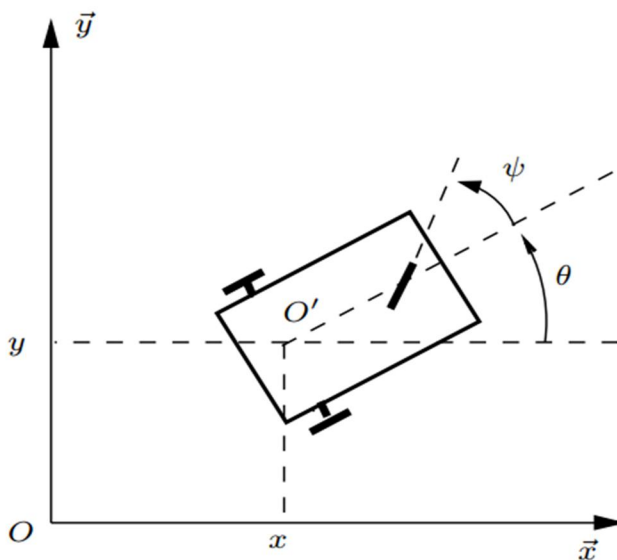
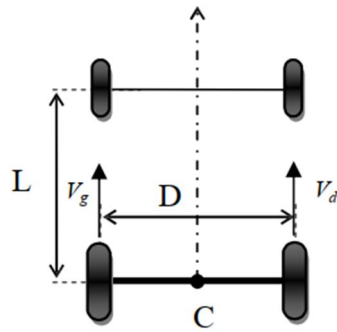


Figure I. 7: Robot de type tricycle [7]

- **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. Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire. Toutes les autres propriétés du robot voiture sont identiques au robot tricycle.



- **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. L'énorme avantage du robot omnidirectionnel est qu'il est holonome puis qu'il peut se déplacer dans toutes les directions.

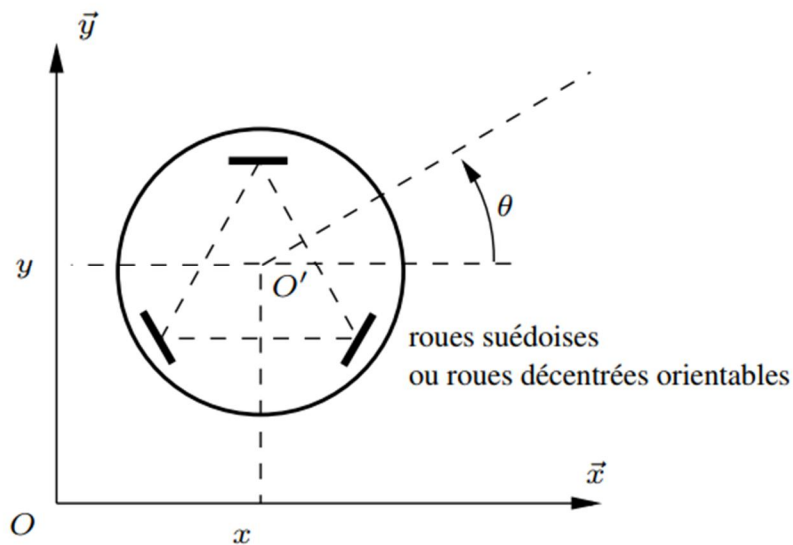


Figure I. 9: Robot mobile omnidirectionnel [7]

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

Le tableau récapitulatif ci-après les avantages et les inconvénients des robots mobiles roues :

Tableau I. 1: Avantages et inconvénients des différents types de robots mobiles [7]

Type de robots mobiles	Avantages	Inconvénients
Uni cycle	<ul style="list-style-type: none"> - Stable - Rotation sur soi-même - Complexité mécanique faible 	Non-holonome
Tricycle	Complexité mécanique modérée	<ul style="list-style-type: none"> - Non-holonome - Peu stable - Pas de rotation sur soi-même
Voiture	<ul style="list-style-type: none"> - Stable - Complexité mécanique modérée 	<ul style="list-style-type: none"> - Non holonome - Pas de rotation sur soi-même
Omnidirectionnel	<ul style="list-style-type: none"> - Holonome - Stable - Rotation sur soi-même 	Complexité mécanique importante

b. Robots mobiles à chenilles

L'utilisation des chenilles présente l'avantage d'une bonne adhérence au sol et d'une faculté de franchissement d'obstacles. L'utilisation est orientée vers l'emploi sur des sols accidentés ou de mauvaise qualité au niveau de l'adhérence (présence de boue, herbe...).



Figure I. 10: Robot mobile à chenilles [8]

c. Les robots mobiles marcheurs

Les robots mobiles marcheurs sont destinés à réaliser des tâches variées dont l'accès au site est accidenté et dangereux à l'homme. On distingue les robots marcheurs à deux jambes (humanoïdes), à quatre pattes (type cheval), et à six pattes (type araignée).

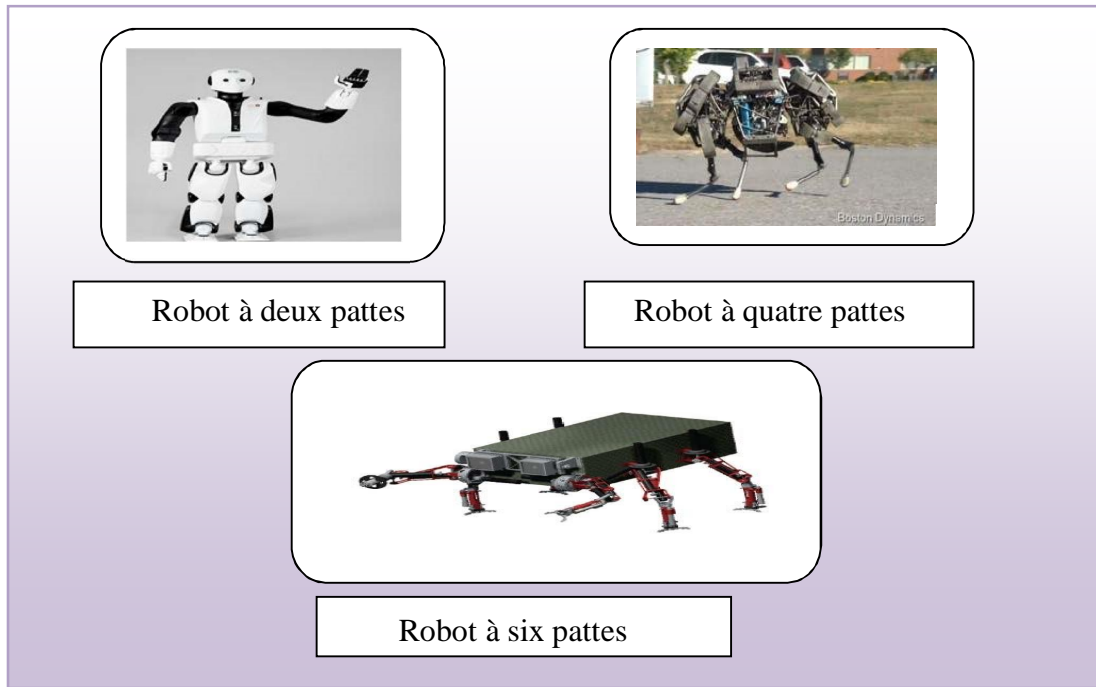


Figure I.11: Robots marcheurs [5]

d. Les robots mobiles rampants

Le système est composé d'un ensemble de module ayant chacun plusieurs mobilités. Ici aussi les techniques utilisées découlent des méthodes de locomotion des animaux et des insectes.



Figure I.12: Robot mobile rampant [2]

I.4.3 Classification selon le domaine d'application

Bien que le champ d'application des robots mobiles reste illimité, nous présentons ici quelques domaines d'application.

- Les robots industriels et de service (hôpital, bureau, maison);
- Les robots militaires ;
- Les robots de laboratoires ;
- Les robots de loisirs (jouet, 'robot compagnon').



Figure I.13: Exemples de robots commerciaux ou de recherche [3]

En outre, il existe d'autres moyens de locomotion différente des précédentes. On peut citer les robots sous-marins, les robots d'exploration spatiale et les robots volant. Le champ d'application des robots reste très varié.

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

Tableau I.2: Récapitulatif des domaines d'application des robots mobiles [7]

Domaines	Applications
Industrie nucléaire	Surveillance de sites ; Manipulation de matériaux radioactifs ; Démantèlement de centrales.
Sécurité civile	Neutralisation d'activité terroriste ; Démunage ; Pose d'explosif ; Surveillance de munitions.
Chimique	Surveillance de site ; Manipulation de matériaux toxiques.
Mine	Assistance d'urgence.
Agricole	Cueillette de fruits ; Traite, moisson, traitement des vignes.
Nettoyage	Coque de navire ; Nettoyage industriel
Espace	Exploration
industrie	Convoyage; Surveillance.
Sous-marine	Pose de câbles ; Recherche de navires immergés ; Inspection des fonds marins.
Militaire	Surveillance ; Pose d'explosif ; Manipulation de munitions

I.4.4 Classification selon la motricité et l'énergie

Le déplacement des robots est réalisé par des moteurs de types :

- Électrique ;
- Thermique ;
- Hydraulique.

I.5 Caractéristiques d'un robot

Un robot doit être bien choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre en compte :

- La charge maximale transportable ;
- Le volume de travail ;
- La répétabilité ;
- La vitesse de déplacement.

I.6 Avantages et inconvénients des robots

Les avantages des robots sont :

- Les robots peuvent travailler dans un environnement dangereux ;
- Les robots n'ont pas besoin de l'éclairage, de la climatisation ;
- Les robots travaillent continuellement, sans ressentir une fatigue ou l'ennui, et ne nécessitent pas une assurance médicale ou de vacances ;
- Les robots peuvent être beaucoup plus précis que les humains. Précision linéaire d'un robot typiquement est de 20 à 10 microns. L'inconvénient des robots est qu'ils manquent de capacité de réagir en cas d'urgence, à moins que les situations comprises et les réponses sont inclut dans le système. Inconvénients des robots comprennent :
 - Le manque de pouvoirs prendre une décision ;
 - Consommation de l'énergie ;
 - Ils peuvent causer des dommages à des autres appareils, et la blessure de l'homme...

Après avoir présenté la généralité sur la robotique mobile, nous allons parler de quelques méthodes d'intelligentes artificielles existantes utilisés pour la conception de ces robots.

II LES METHODES INTELLIGENTES

Cette partie consiste à présenter quelques méthodes d'intelligence artificielle et des algorithmes de commandes de robot. De prime à bord, il convient de définir certaines notions au préalable.

L'intelligence artificielle est la reproduction, par des moyens artificiels de toutes les formes de l'intelligence humaine pour un objectif final, qui s'intéresse à la conception des systèmes intelligents. L'IA est la capacité à raisonner, à apprendre, et à s'adapter face à de nouveaux changements qu'ils sont des éléments principaux que nous recherchons.

D'après Ferber, **un agent** est une entité autonome réelle ou abstraite qui est capable d'agir sur elle-même et sur son environnement, et peut communiquer avec d'autres agents, dont le comportement est la conséquence de ses observations, ses connaissances et de ces interactions avec son environnement. Il doit être (autonome, interactif, adaptatif, rationnel, coopératif et intelligent...). La notion d'agent se diffère selon l'utilisation.

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.

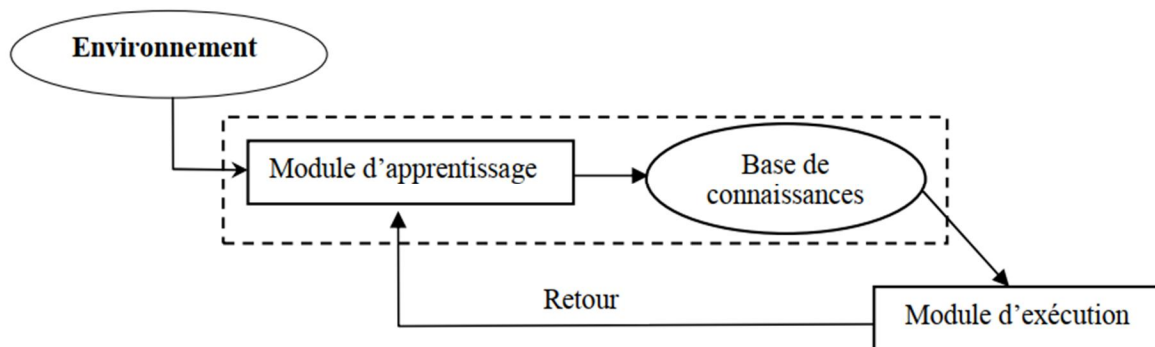


Figure I. 14: Représentation de l'apprentissage automatique [8]

II.1 LES METHODES D'INTELLIGENCES ARTIFICIELLES

Il existe plusieurs méthodes permettant d'implémenter l'intelligence artificielle. On peut citer entre autre :

- **Le réseau de Neurones Artificielles :**

Généralement appelé « réseau neuronal » (RN), est un algorithme d'apprentissage qui est inspiré par la structure et les aspects fonctionnels des réseaux de neurones biologiques. En outre, ce sont des systèmes de traitement de l'information dont la structure s'inspire de celle du système nerveux. Ils sont destinés à effectuer des tâches auxquelles les approximateurs traditionnels semblent moins adaptés. Dans le domaine de traitement du signal et de l'automatique, les dernières années ont été marquées par des avancées significatives tant du point de vue fondamental que du point de vue des applications industrielles. Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau. Le neurone est une cellule composée d'un corps cellulaire et d'un noyau, tel que représenté dans la figure suivante :

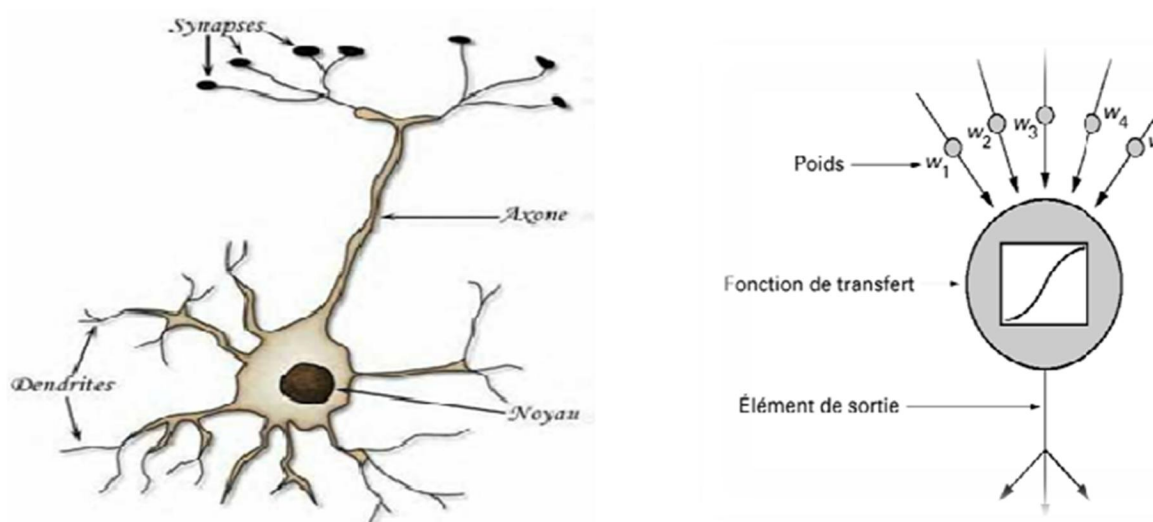


Figure 15: Représentation d'un Neurone biologique et d'un Neurone artificiel [8]

- La logique floue (LF)

La logique floue est une méthode de raisonnement qui ressemble au raisonnement humain. L'approche de LF imite la manière de prendre des décisions chez l'homme qui implique toutes les possibilités intermédiaires entre les valeurs numériques OUI et NON. C'est une technologie émergente importante et est sans aucun doute l'un des domaines des technologies pour la conception et le Systèmes de contrôle complexes. La logique floue a trouvé des applications dans des domaines divers tels que la reconnaissance, le traitement de l'image et du signal, l'IA

et l'aide à la décision, et maintenant, l'un des domaines où la logique floue trouve son essor est l'industrie de la robotique.

- **Réseaux Bayésiens**

Un réseau bayésien encore appelé modèle graphique acyclique orienté est un modèle graphique probabiliste qui représente un ensemble de variables aléatoires et leurs indépendances conditionnelles par un graphe orienté acyclique. Par exemple, un réseau bayésien pourrait représenter les relations probabilistes entre les maladies et les symptômes. Compte tenu des symptômes, le réseau peut être utilisé pour calculer les probabilités de la présence de diverses maladies. Des algorithmes efficaces existent qui effectuent l'inférence et l'apprentissage.

- **Programmation logique inductive**

La programmation Inductive logique (PLI) est une approche à la règle d'apprentissage en utilisant la programmation logique comme une représentation uniforme pour des exemples d'entrée, les connaissances de base et des hypothèses. La Programmation inductive est un domaine connexe qui considère tout type de langages de programmation pour représenter des hypothèses (et non seulement la programmation logique), tels que des programmes fonctionnels.

- **Arbre de décision**

L'apprentissage par arbre de décision utilise un arbre de décision comme un modèle prédictif, qui mappe observations sur un élément à des conclusions au sujet de la valeur cible de l'élément.

Voilà ainsi présenter quelques méthodes d'intelligence artificielle, il convient dès lors d'énumérer quelques algorithmes de commande de robots.

II.2 Les algorithmes de commande de robot

Un robot est un système complexe qui doit satisfaire des exigences variées en utilisant un ensemble logiciel appelé architecture de contrôle du robot. Cette architecture permet donc d'organiser les relations entre les trois grandes fonctions: la perception, la décision (planification) et l'action représentées comme suit :

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

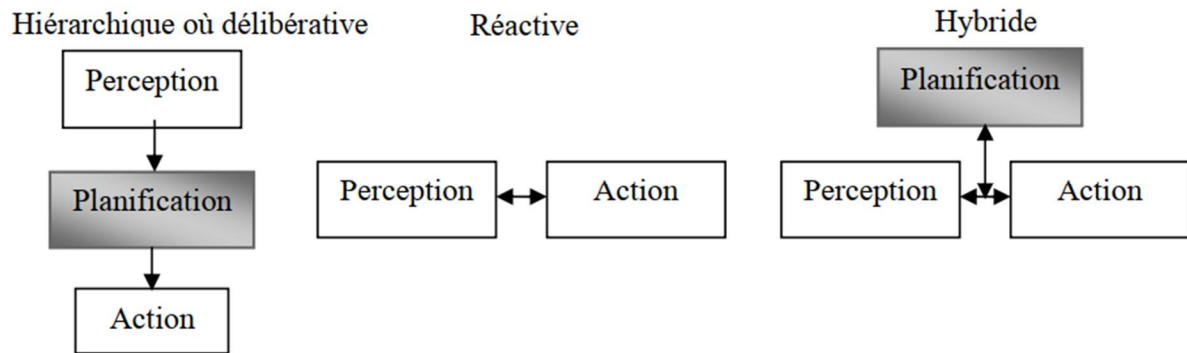


Figure I. 16: Architecture de contrôle des robots mobiles[4]

Les différents algorithmes développés jusqu'à ce jour pour la commande des robots sont nombreuses, dans le cadre de notre étude, nous allons citer quelques algorithmes :

- **L'algorithme TD (0)** : c'est l'algorithme élémentaire d'apprentissage par renforcement, dit algorithme de « différence temporelle ». Cet algorithme repose sur une comparaison entre la récompense que l'on reçoit effectivement et la récompense que l'on s'attend à recevoir en fonction des estimations construites précédemment. Par ailleurs, il repose également sur deux processus de convergence couplés, le premier estimant de plus en plus précisément la récompense immédiate reçue dans chacun des états et le second approchant de mieux en mieux la fonction de valeur résultant de ces estimations en les propageant de proche en proche.
- **L'algorithme Sarsa** : L'algorithme SARSA est similaire à l'algorithme TD (0) à ceci près qu'il travaille sur les valeurs des couples $(s;a)$ plutôt que sur la valeur des états. Son équation de mise à jour est identique à celle de TD (0) en remplaçant la fonction de valeur par la fonction de valeur d'action.
- **L'algorithme du Q-Learning** : se présente comme une simplification de l'algorithme SARSA par le fait qu'il n'est plus nécessaire pour l'appliquer de déterminer un pas de temps à l'avance quelle sera l'action réalisée au pas de temps suivant. L'algorithme Q-learning effectue les mises à jour en fonction des actions optimales mêmes si ce ne sont pas ces actions optimales que l'agent réalise, ce qui est plus simple.

III. ETAT DE L'ART

Les robots marcheurs à six pattes (robots hexapodes) font l'objet de nombreux travaux de recherche car ils présentent un bon compromis entre complexité de conception, stabilité et mobilité [5]. Ils peuvent être utiles pour accéder à des zones hostiles pour l'homme comme les

sites nucléaires ou l'espace. Ces robots possèdent un nombre élevé de degrés de liberté (≥ 2 par patte) et leur commande nécessite de résoudre simultanément des problèmes de coordination de mouvements, de contrôle de l'équilibre et de contrôle de la posture. Les travaux de recherche portent sur la structure mécanique, l'élaboration de modèles géométriques, cinématiques ou dynamiques ainsi que sur les méthodes de commande du robot. Dans un environnement inconnu ou incertain, la planification des mouvements n'est pas efficace. Il est alors intéressant de recourir à un apprentissage tel que l'apprentissage par renforcement qui se base sur un processus essai / erreur pour acquérir le comportement souhaité. La résolution de tels problèmes d'apprentissage est menée à partir de modèles simplifiés des robots hexapodes décrivant la séquence des mouvements et la marche obtenue.

C. Touzet et A. Johannet (Touzet, 1992), (Johannet, 94) ont proposé une architecture de commande distribuée à base de six réseaux de neurones stochastiques pilotant les mouvements de poussée ou de transfert de chaque patte. L'apprentissage est mené selon l'algorithme de 'pénalité / récompense' de G. Barto (Barto, 1985). Le robot dont un prototype physique a été réalisé (poids 3 kg, dimension (en mm) : 300x150x150), apprend à marcher et à éviter des obstacles

Constatant qu'un robot, pour évoluer dans un monde réel, a besoin d'un grand nombre de capteurs mais que tous ne fournissent pas des informations pertinentes pour la tâche à effectuer, J.M. Porta (Porta, 2000) introduit un nouvel algorithme d'apprentissage, le ' p learning' dont l'objectif est de diminuer le temps nécessaire à l'apprentissage de la marche d'un robot hexapode en découvrant les sous-ensembles de capteurs qui donnent les informations utiles pour la tâche. La méthode a été validée en simulation.

Les travaux de M. M. Svinin (Svinin, 2001) portent sur la commande d'un robot marcheur selon une approche à base de règles et d'apprentissage par renforcement. Ce robot n'a pas de connaissance a priori sur l'environnement et cherche à atteindre un point particulier de l'espace. Une règle associe un vecteur d'entrée (composé de valeurs particulières des capteurs du robot) à une action. L'apprentissage consiste à sélectionner les règles les plus pertinentes pour atteindre l'objectif. Il est montré en simulation que le robot apprend à marcher en direction de l'objectif sans tomber mais qu'il ne suit pas forcément le chemin le plus court.

Les travaux de C. Touzet et A. Johannet ont permis de montrer qu'un robot hexapode avec une architecture de commande distribuée pouvait apprendre à marcher. Cependant l'attribution

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

des pénalités et des récompenses n'est pas locale, les pénalités sont appliquées à tous les contrôleurs de mouvement y compris à ceux qui ont pris une décision correcte. Les résultats montrent l'intérêt de récompenser ou de pénaliser individuellement les acteurs, le modèle de simulation est enrichi d'un modèle géométrique de contrôle de la posture pour apprendre au robot à maîtriser sa trajectoire.

Ceci dit, il convient également de présenter quelques travaux déjà réalisés au Cameroun dans le cadre de la robotique mobile.

Les différents travaux réalisés au Cameroun :

Prototype d'un robot se déplaçant à l'aide d'un télémètre ultrason en vue de l'obtention du DPET1, les étudiants AYEBE, MEFO et NKOULOU de l'ENSET de Douala, du département de génie électrique, ont présenté un rapport de fin d'étude sur le thème « PROTOTYPE D'UN ROBOT SE DEPLACANT A L'AIDE D'UN TELEMETRE ULTRASON » L'objectif de leur travail était de réaliser un robot mobile esquivant les obstacles dans l'environnement de l'évolution. La figure ci-dessous présente une simulation sur PROTEUS.

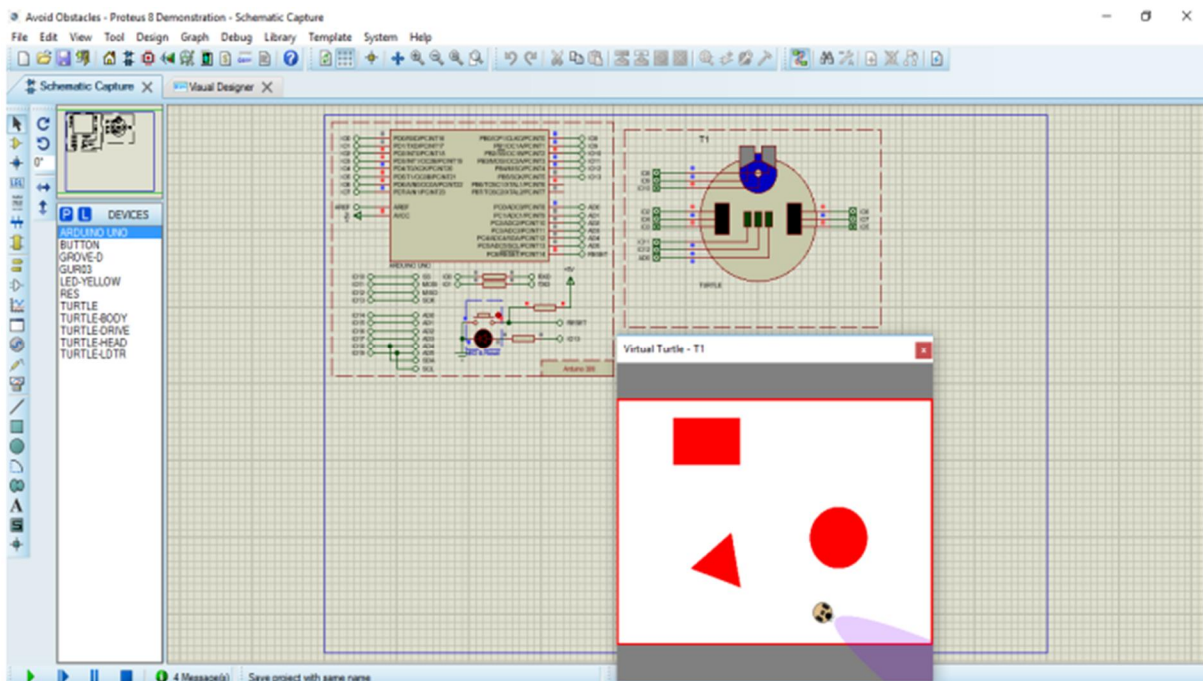


Figure I.17: Simulation d'un robot avec évitement d'obstacles[7]

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

Bien que ce travail soit remarquable, il présente néanmoins quelques insuffisances telles que l'incapacité à suivre une direction définie et l'absence des dispositifs de signalisation sur le fonctionnement du robot.

Etude et réalisation d'un robot suiveur de ligne avec évitement d'obstacles et muni d'une commande Bluetooth, en vue de l'obtention du DPET1, les étudiants DOUANLA, KIFACK et NDE de l'ENSET de Douala, du département de génie électrique, ont présenté un rapport de fin d'étude sur le thème « **Etude et réalisation d'un robot suiveur de ligne avec évitement d'obstacles et muni d'une commande Bluetooth** ». Il s'agit d'un dispositif capable de se déplacer tout seul de façon sûr et rapide suivant une ligne définie au préalable, tout en esquivant des obstacles présents dans son environnement d'évolution sans intervention humaine. Le schéma ci-dessous nous montre la simulation d'un suiveur de ligne.

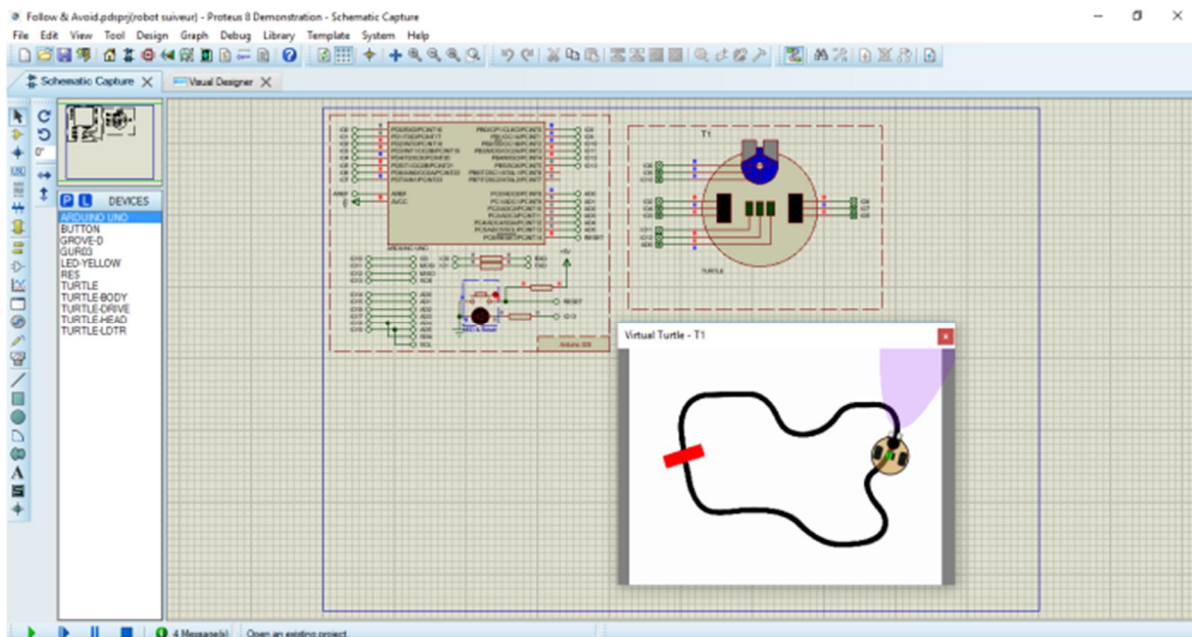


Figure I. 18: Simulation d'un robot suiveur de ligne avec évitement d'obstacles [9]

Apport personnel

Partant des projets cités ci-dessus, nous envisageons leur amélioration par la conception et la réalisation d'un robot hexapode autonome. A leurs travaux, nous utiliserons l'algorithme du Q-Learning qui est une technique d'optimisation le fonctionnement et de la navigation de notre robot. Il est à noter que cette technique n'a pas encore été utilisée au Cameroun à ce jour.

CONCLUSION

La robotique est la branche de l'intelligence artificielle concernée par l'étude des systèmes automatiques capables d'interagir directement avec le monde physique. C'est une automatisation de ses machines, ou l'objectif est d'augmenter les capacités de localisation et de navigation dans son espace de travail. Ce chapitre nous a permis de présenter un exposé sur la généralité de la robotique mobile, en ressortant l'historique des robots mobiles, les types des robots et leur domaine. On a présenté aussi les quelques méthodes utilisé dans l'intelligence artificiel pour la modélisation des robots et les différents algorithmes utilisées pour la commande des robots mobiles en citant les avantages et les inconvénients de chacune.

CHAPITRE II : METHODE DE CONCEPTION ET DE MODELISATION

Résumé : dans cette partie, nous allons présenter le cahier de charges fonctionnel, ainsi que toutes les étapes et éléments nécessaires à la conception la modélisation et la commande de notre système.

Aperçu :

INTRODUCTION

- I. CAHIER DE CHARGES FONCTIONNEL
- II. METHODOLOGIE DE CONCEPTION
- I. MODELISATION
- II. CONCEPTION DU ROBOT
- III. COMMANDE DU ROBOT

CONCLUSION

INTRODUCTION

Nous avons vu dans le chapitre précédant la généralité sur la robotique mobile ainsi que les différents travaux ayant été effectués dans ce domaine. Dans ce chapitre nous présenter un cahier de charges fonctionnel dans le but de mieux formuler le besoin. Par la suite nous présenterons notre méthodologie de conception, la modélisation notamment géométrique et cinématique. Nous allons également concevoir notre robot tout en présentant le choix des outils matériels et logiciels nécessaires à la réalisation de notre dispositif. Nous terminerons par le choix de la méthode de commande de notre système.

I. CAHIER DE CHARGES FONCTIONNEL

Le cahier des charges fonctionnel est un document formulant le besoin, au moyen de fonctions détaillant les services rendus par un produit et les contraintes auxquelles il est soumis. Le cahier des charges vise à définir et à faire valider les spécifications d'un produit ou d'un service à réaliser. Ici il s'agit de cahier des charges fonctionnel, formulant le besoin au moyen de fonctions détaillant les services attendus et les contraintes auxquelles le produit à fournir est soumis.

I.1 Le concept général et les principaux services attendus

I.1.1 Formulation du besoin

Economiser la vie des soldats, préserver la vie de la population en situation d'attaques terroristes et prévenir des attaques grâce à un dispositif (robot) intelligent, autonome et à moindre coût.

I.2.2 Les clients, utilisateurs et usagers potentiels

- Les clients : c'est toute personne qui réalise l'achat, on peut citer les hommes, les femmes, sociétés industrielles, forces armées et bien d'autres ;
- Les utilisateurs et usagers : ce système est destiné à toutes les industries ayants besoin d'assurer la sécurité des biens et des personnes.

I.2 Contexte du projet

Depuis le début des exactions de Boko Haram au Cameroun en 2014, l'armée a subi 336 attaque du groupe islamique nigérien dans la région de l'extrême-nord [3]. Pour faire face à ces combattants parfois invisibles, l'armée a opté pour l'utilisation des drones qui permet ainsi de

réduire l'effort humain. À ce jour l'on estime à 18000 le nombre de personnes tués lors des attentats perpétrés par Boko Haram depuis 2009 selon des sources sécuritaires.

I.2.1 La nécessité de concevoir notre dispositif

Comme nous l'avons dit plus haut, Au Cameroun, près de 20000 personnes tués jusqu'à ce jour des différentes attaques terroristes au sein de la population et au sein de l'armée laissant ainsi plusieurs familles meurtries. Le pays perd sa ressource humaine et des multiples conséquences économiques et sociales qui en résultent. En plus notre système est adéquat à l'environnement sécuritaire camerounais car peut être mis sur pieds et fonctionnel.

I.2.2 Limite de l'étude

L'étude effectuée sera limitée aux conditions atmosphériques, intempéries, et à certaines zones géographiques du Cameroun ; mais le système sera conforme aux normes scientifiques en vigueur.

I.3 Énoncé fonctionnel du besoin : comprendre et analyser le besoin

Le besoin peut être énoncé de la manière suivante : naviguer dans des zones difficilement accessibles, éviter les obstacles, enregistrer des images. Il s'agit donc d'agir en amont mais aussi en aval de la sécurité des biens et des personnes dans nos frontières et à l'intérieur du territoire Camerounais.

I.3.1 Analyse fonctionnelle du besoin (bête à corne)

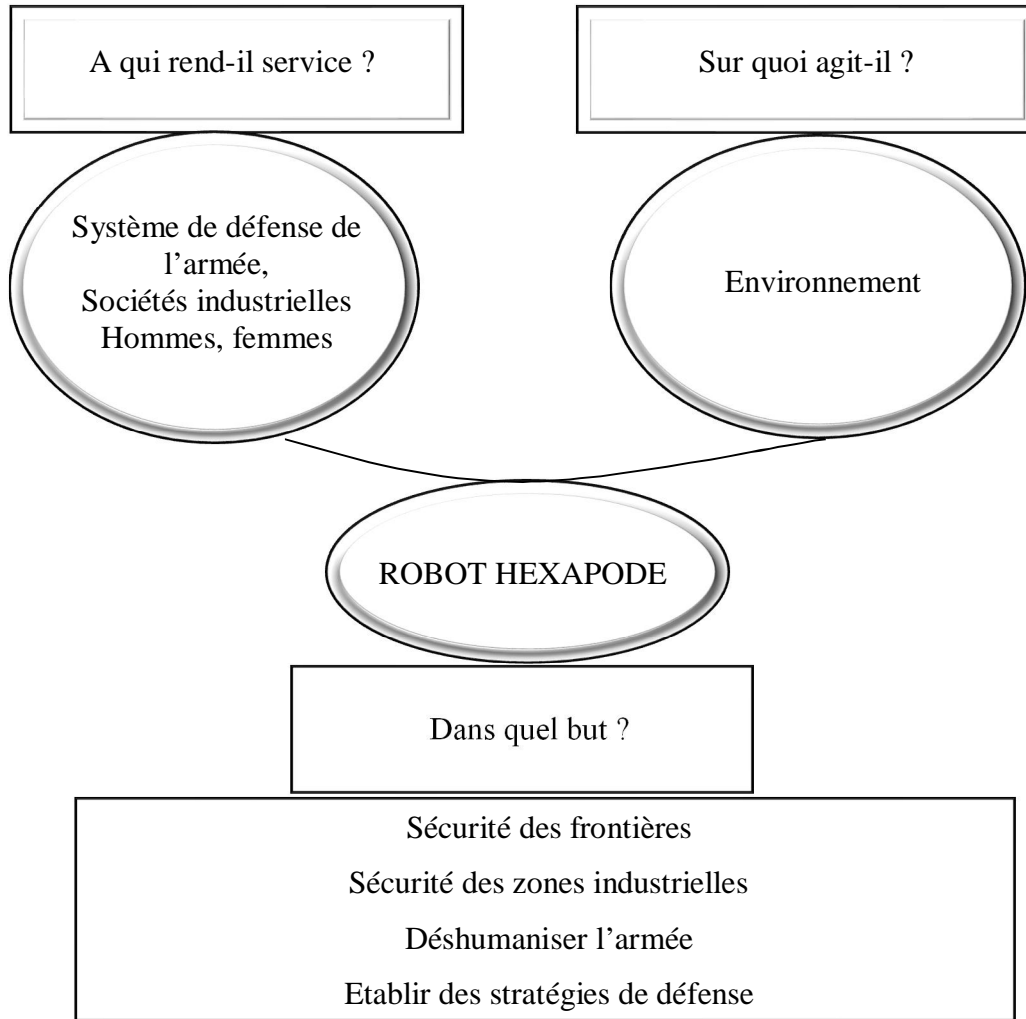


Figure I.19: le diagramme bête à corne exprimant le besoin du robot

I.3.2 Validation du besoin

– À qui le produit rend-il service ?

Le système que nous concevons s'adresse aux utilisateurs désirant assurer la sécurité et la sûreté des biens et des personnes peu importe l'environnement dans lequel l'on se situe. Le consommateur doit avoir un système répondant à toutes ses attentes : il doit être léger, utilisable en tout lieu, transportable, discret et efficace.

- **Sur quoi agit-il ?**

Il agit directement sur l'environnement notamment les zones à risque d'attaques terroristes et sur la surveillance de nos industries.

- **Quelles sont les origines du besoin ?**

Le taux de mortalité très élevé tant du côté de l'armée et de la population dans la localité de la partie Ouest de l'extrême-Nord du Cameroun notamment les départements du Mayo-Tsanaga, Mayo Sava et le logon et Chari tous frontaliers du Nigeria qui subissent depuis février 2015 jusqu'à nos jours des attaques terroristes du groupe Boko Haram.

- **Le besoin peut-il évoluer ?**

Etant donné que la technologie est ascendante, il n'est pas possible qu'un système conçu ne puisse pas évoluer dans le temps et dans l'espace.

- **Le besoin peut-il disparaître ?**

Le besoin peut disparaître si :

- Les conditions atmosphériques tel que l'humidité, la pression atmosphérique rendent notre système défaillant ;
- L'ennemie détecte le dispositif mis en place.

Il est donc peu probable que le besoin disparaisse totalement.

I.3.3 Enoncé des fonctions, des services et des contraintes

Il s'agit ici d'énoncer les fonctions principales, les fonctions secondaires et les contraintes pour mieux visualiser l'ensemble des éléments plus ou moins concret interagissant avec notre système.

Fonctions principales

- FP1 : être capable de naviguer ;
- FP2 : être capable de détecter les obstacles ;
- FP3 : être capable de se déplacer dans un environnement prédéfini ;
- FP4 : être autonome ;
- FP5 : pratique d'utilisation ;

- FS1 : Etre léger ;
- FS2: Etre facilement manipulable ;
- FS3: Etre peu encombrant.

Fonctions contraintes

- FC1 : Doit respecter les normes (homologation)
- FC2 : avoir un déplacement le plus léger possible (distance entre les capteurs) ;
- FC3 : s'adapter à une source d'énergie (tension, autonomie)
- FC4 : s'adapter au tracé de l'environnement (calcul des pattes dans les virages, vitesse) ;
- FC5 : Doit être résistant en conditions normales d'utilisation et face aux intempéries ;
- FC6 : Doit fonctionner pendant un temps suffisamment long ;
- FC7 : Doit être facile d'utilisation ;
- FC8 : doit être discret (bruit).

Fonctions d'estime

- FE1 : Etre innovant ;
- FE3 : Avoir un aspect robuste.

I.4 Organisation des fonctions : méthode APTE

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING



Figure II.20: Diagramme Pieuvre

I.5 Caractérisation des fonctions

Les fonctions énumérées précédemment sont caractérisées par leur flexibilité² et leur niveau d'acceptation, c'est-à-dire : tout niveau de critère d'appréciation doit être assorti d'une indication de flexibilité. On a défini 4 classes de flexibilité :

La classe F0 signifiant flexibilité nulle, niveau impératif ;

La classe F1 signifiant flexibilité faible, niveau peu négociable ;

La classe F2 signifiant flexibilité bonne, niveau négociable ;

La classe F3 signifiant flexibilité forte, niveau très négociable.

Limites des niveaux : niveau de critère d'appréciation au-delà duquel, ou en deçà suivant le cas, le besoin est jugé non satisfait.

Le tableau ci-dessous permet de donner la flexibilité de chaque fonction, ses limites et le niveau de réalisation.

Tableau II.3:Caractérisation des fonctions

Fonctions	Critères d'appréciation	Niveau d'exigence	Limites	Flexibilité
Fonctions principales				
FP1 Etre capable de naviguer	Vitesse 5cm/s	test	6m	F0
FP2 être capable de détecter les obstacles	distance entre les capteurs	test	+/- 30 cm	F0
FP3 être capable de se déplacer dans un	Stabilité et posture	test	-	F1

² Flexibilité des niveaux : la flexibilité est un ensemble d'indications exprimées par le demandeur sur les possibilités de moduler un niveau recherché pour un critère d'appréciation.

**CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE
COMMANDE PAR L'ALGORITHME DU Q-LEARNING**

environnement prédéfini				
FP4 être autonome	fonctionnement	test		
FP5 pratique d'utilisation	Adaptabilité	test	-	F1
FS1 Être léger	poids	500g	+/- 300g	F1
FS2 Être facilement manipulable	Maniabilité, ergonomie	test	-	F1
FS3 Être peu encombrant	volume	250cm ³	+ 10 %	F1
Fonctions de contraintes				
FC1 Doit respecter les normes	Normes d'homologation	test	-	F0
FC2 avoir un déplacement le plus léger possible	Masse, moins élevé possible	test	$M \leq 300g$	F1
FC3 s'adapter à une source d'énergie	énergie électrique	Pile ou batterie rechargeable	9 volts	F0
FC4 s'adapter au tracé de l'environnement	calcul des pattes dans les virages, vitesse	test	0.5°	F1

**CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE
COMMANDE PAR L'ALGORITHME DU Q-LEARNING**

FC5 Doit être résistant en conditions normales d'utilisation et face aux intempéries	Résistance aux contraintes exercées par l'environnement	test	-	F1
FC6 Doit fonctionner pendant un temps suffisamment long	Alimentation autonome	30 min	+/-15min	F1
FC7 Doit être facile d'utilisation	adaptabilité	test	-	F2
FC8 doit être discret	Fonctionnement	test	-	F3
Fonctions d'estime				
FE1 Etre innovant	Enquête d'opinion/ technologie	test	-	F0
FE2 Avoir un aspect robuste	Aspect visuel	test	-	F2

I.6 Le diagramme FAST

Nous utilisons Le diagramme FAST (Functional Analysis system Technique) pour présenter une décomposition hiérarchisée des fonctions du système allant des fonctions de service (fonctions en liens avec le milieu extérieur) et passant par les fonctions techniques (fonctions internes au système) jusqu'à l'énoncé des solutions technologiques employés ou prévues pour remplir les fonctions techniques.

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

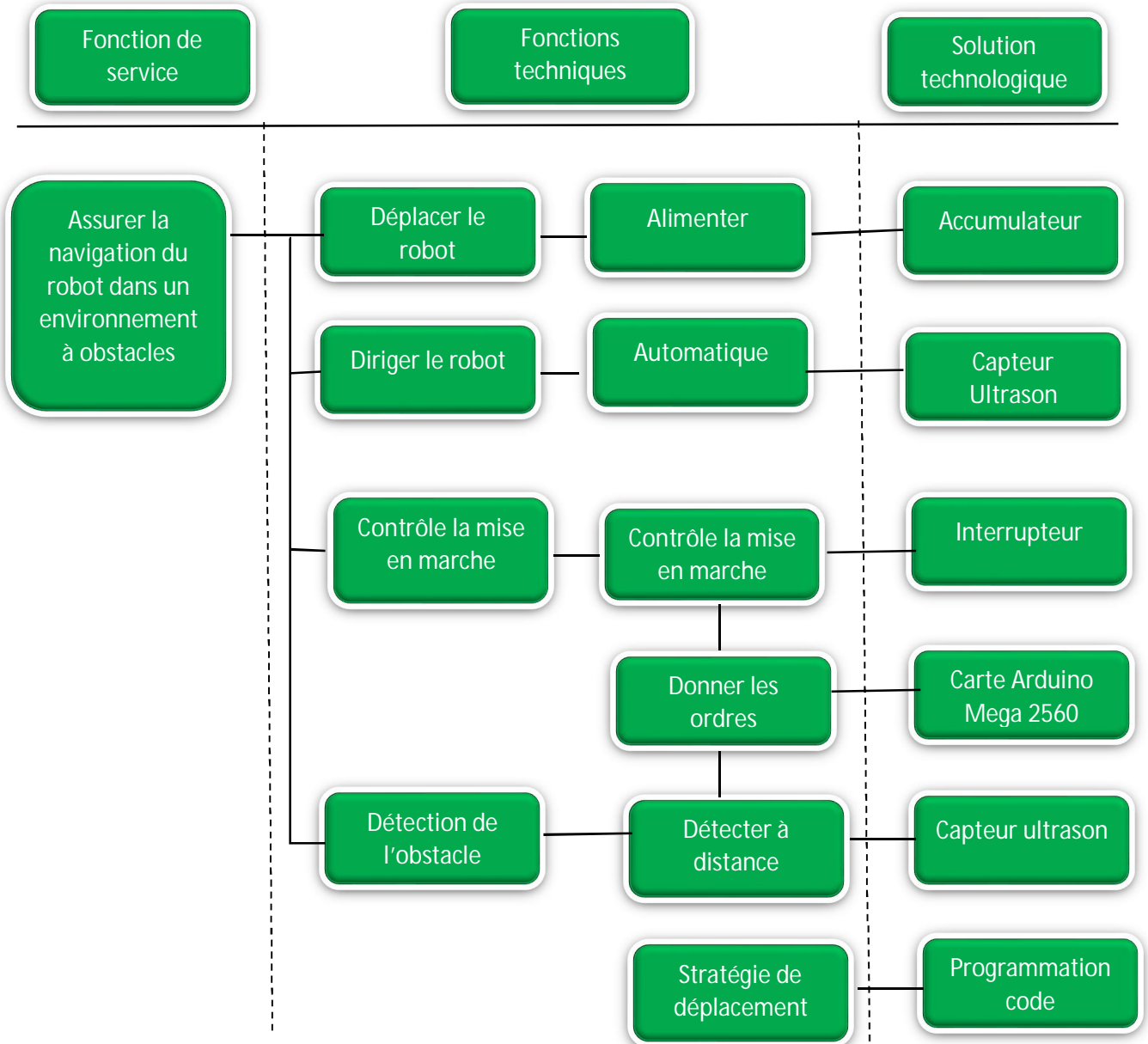


Figure II.22: Le diagramme FAST du robot

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

Représentation générale de la chaine fonctionnelle Nous pouvons représenter la chaine fonctionnelle.

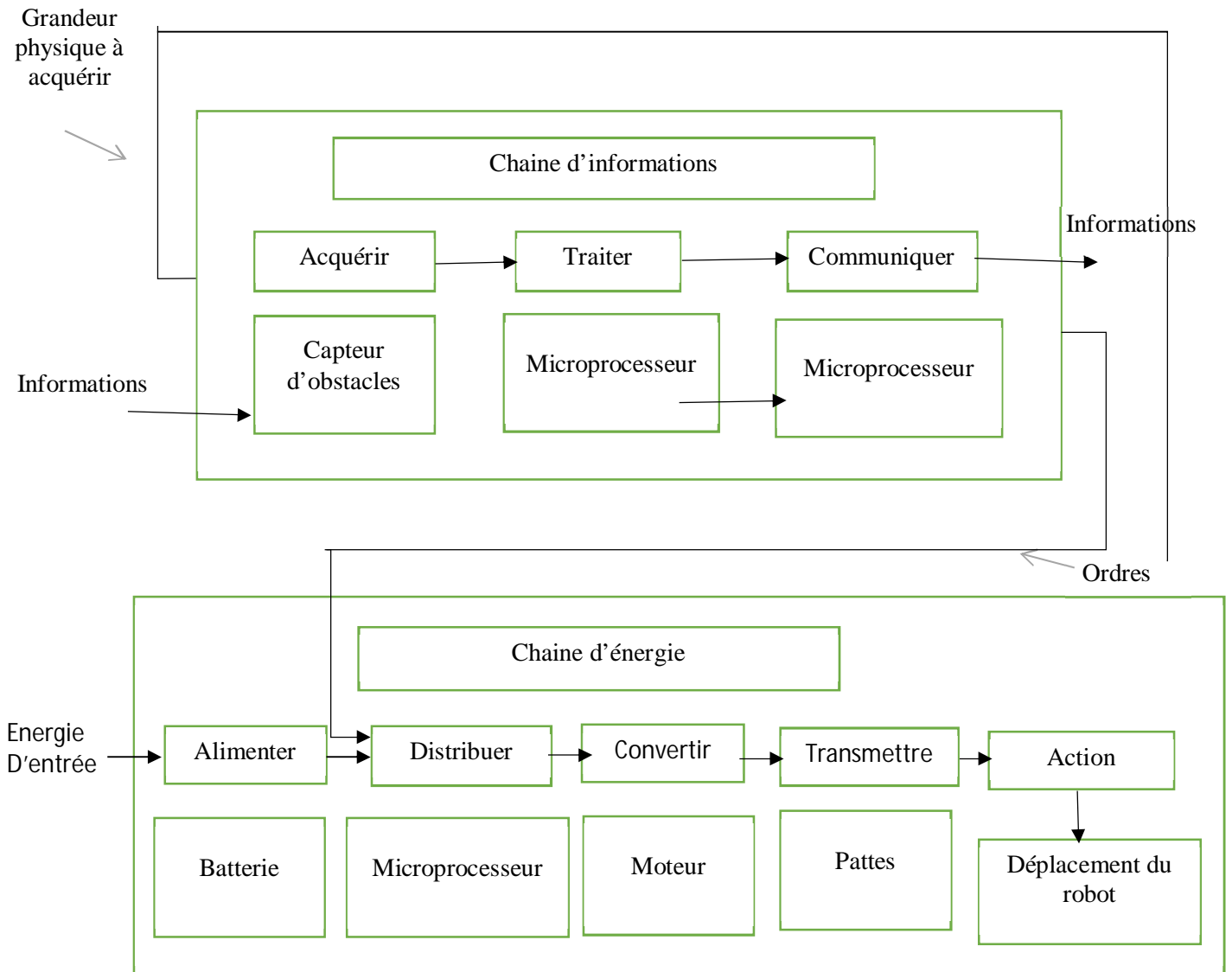


Figure II.23: Schéma de la chaîne fonctionnel du robot

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

L'organigramme déduit de la chaine fonctionnelle de notre robot est le suivant

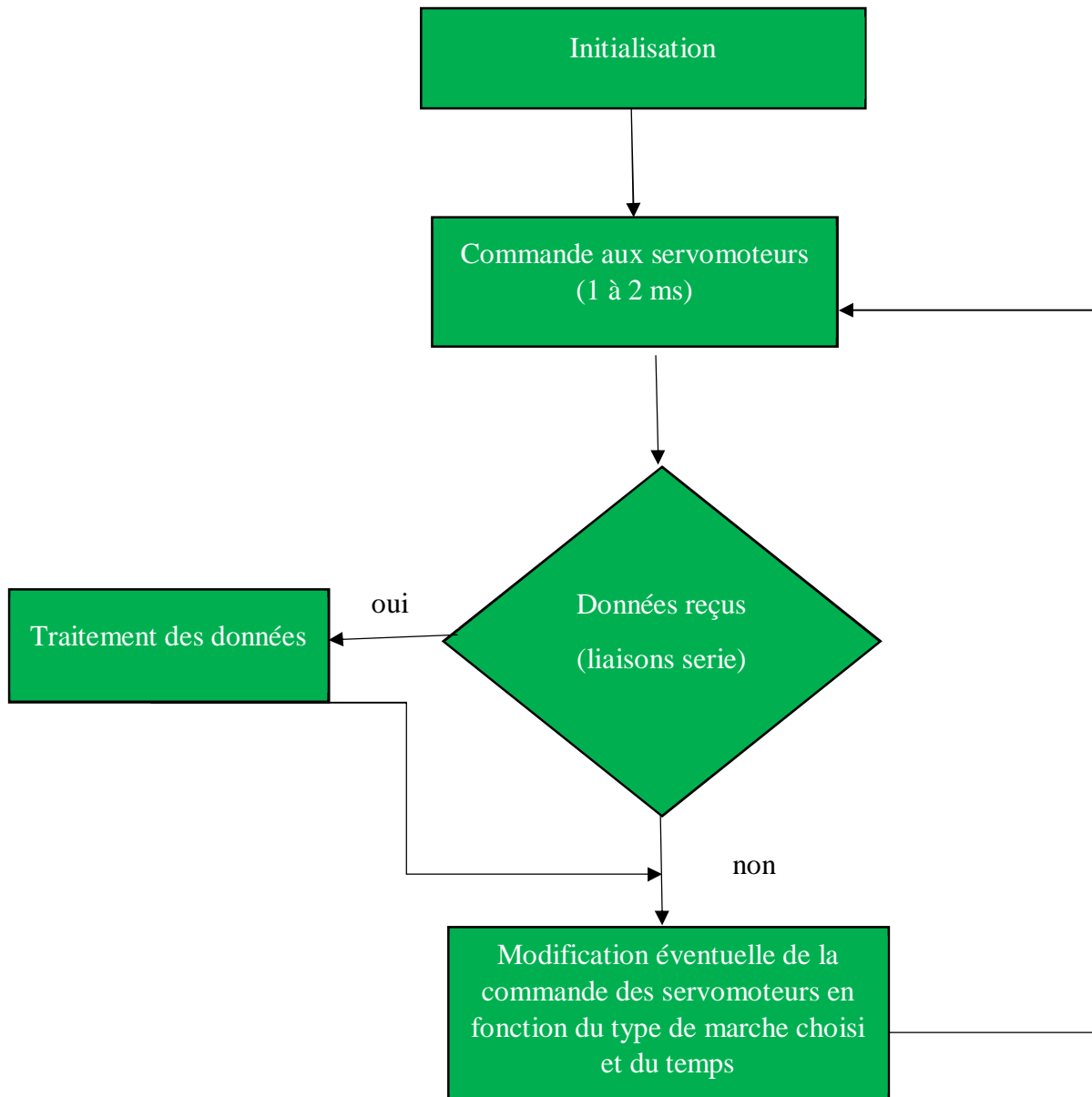


Figure II.24 : Organigramme de la chaine fonctionnel [9]

II. METHODOLOGIE DE CONCEPTION

Comme énoncé à l'introduction générale, l'objectif principal est de concevoir un système mécanique à six pattes en appliquant des techniques de l'intelligence artificielle pour la navigation autonome dans un environnement inconnu, hostile ou inaccessible. Dans cette étude nous avons choisi un robot marcheur hexapode à cause de sa stabilité de marche. Nous allons faire la modélisation géométrique et cinématique pour la patte, pour cela, en déterminant une modélisation géométrique directe pour trouver les angles correspondants aux positions de chaque patte de robot. Cette modélisation nous permettra ainsi de mieux concevoir notre robot.

En outre, il est à noter que la structure des membres et le contrôle des mouvements chez les insectes et les arthropodes, sont menées depuis des décennies. Parmi de nombreux robots à plusieurs pattes, le robot hexapode est l'un des robots les plus utilisés pour un large éventail de tâches.

La plupart de ces tâches sont dangereuses et s'accompagnent généralement d'environnements dociles, non compatibles avec les opérations humaines.

Dès lors, Il convient de rappeler les objectifs spécifiques de notre système :

- Naviguer dans un environnement hostile ;
- Détecter les obstacles et les contourner ;
- Enregistrer des images.

Lorsque le système est mis en marche il doit se déplacer en suivant une trajectoire bien définie. Lorsqu'il détecte l'obstacle grâce au capteur ultrason, il change de direction soit à droite soit par la gauche pour éviter l'obstacle et il poursuit sa trajectoire en effectuant les tâches pour lesquelles il a été conçu. Ceci dit, comme suit : notre système devra fonctionner comme suit :

**CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE
COMMANDE PAR L'ALGORITHME DU Q-LEARNING**

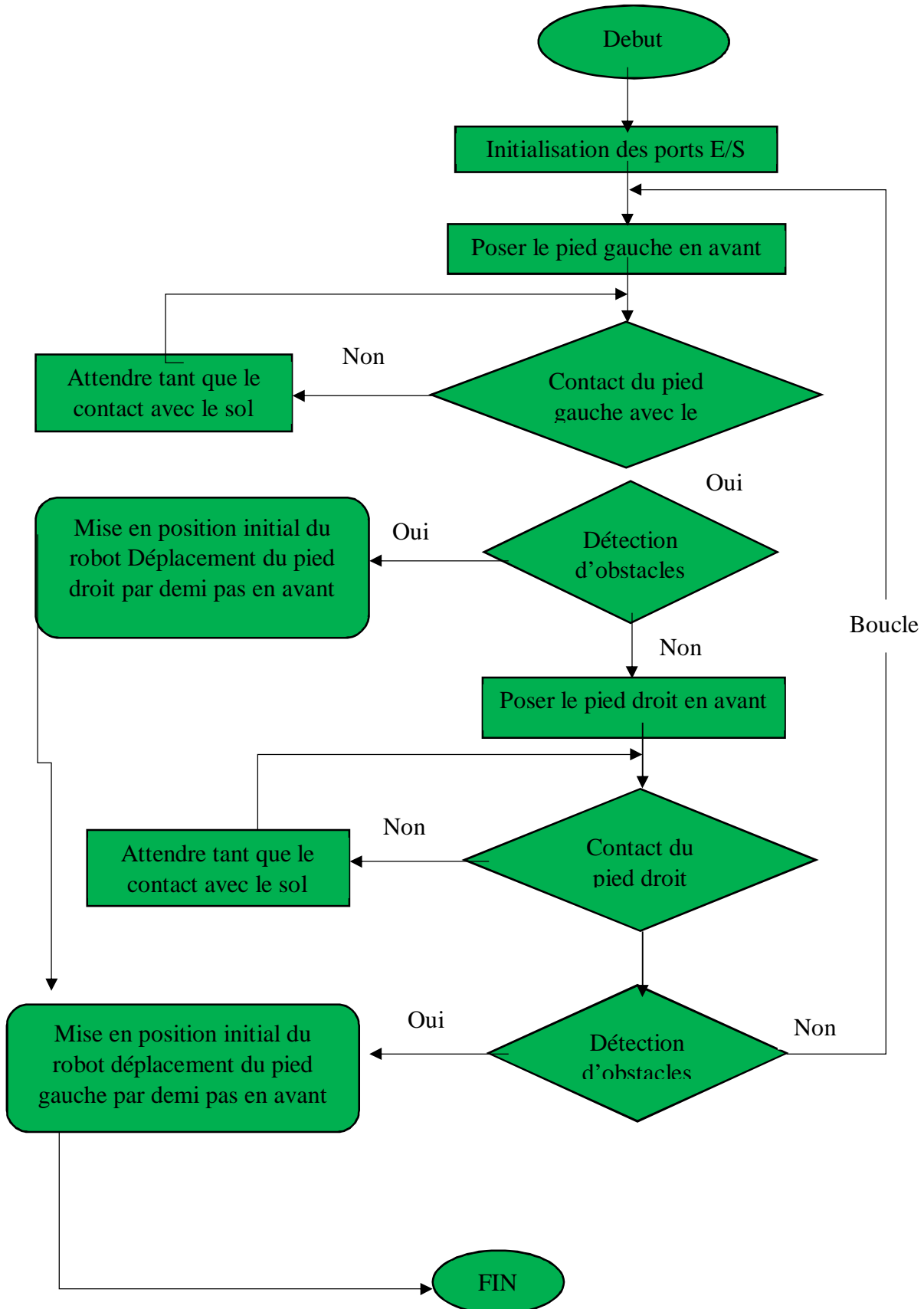


Figure II. 26: Schéma fonctionnel du mouvement du robot

III MODÉLISATION DU ROBOT HEXAPODE

Pour faire La conception de nôtre robot nous avons fait recours aux calculs des modèles mathématiques suivants :

- Les modèles géométriques (MGD) qui expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme ;
- Les modèles cinématique (MCD) qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires ;

Le succès dans la conception d'un robot hexapode réside fondamentalement dans la structure des jambes choisies. Les principaux aspects du déplacement d'un robot hexapode sont régis par les limitations physiques de ses jambes. Il est primordial de choisir une jambe dont la conception orne la plus grande amplitude de mouvements possible et qui ne pose pas de contraintes inutiles pouvant affecter le mouvement du robot. Le schéma suivant en est illustratif :

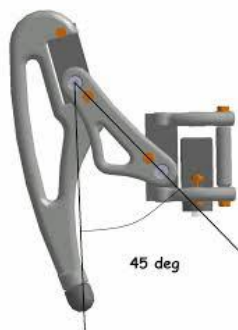


Figure II.27:choix de la patte du robot hexapode [5]

III.1 Modélisation géométrique et cinématique

Dans cette partie, On décrit la méthode de Khalil et Kleinfinger pour le calcul du modèle géométrique direct de l'hexapode robot à structure ouverte simple et arborescente [11]. la méthode de Paul pour trouver les solutions du modèle inverse et on calcule le modèle cinématique de l'hexapode en utilisant le la matrice jacobienne de base utilisant des Ensuite nous présentons la modélisation géométrique de la patte et du robot hexapode.

III.1.1 Modélisation Géométrique Directe des robots

Pour le chercheur, la première interrogation face à un robot est : comment calculer sa position ? Plus précisément : où va se trouver son organe terminal si chaque articulation tourne d'un certain angle ou se déplace d'une certaine distance ? Il faut donc trouver l'ensemble des relations qui permettent d'exprimer les coordonnées opérationnelles de son organe terminal en fonction de ses coordonnées articulaires (**Figure II.27**) : c'est le MGD

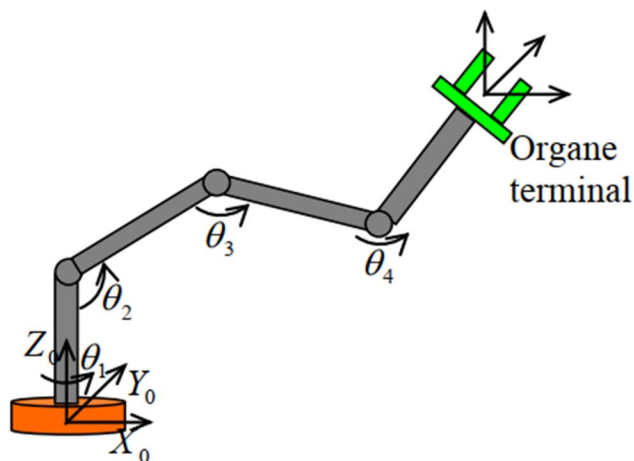


Figure II.28 : Définition du MGD [8]

Le calcul de ce MGD de façon automatique exige une méthode générale pour la description de la morphologie des robots. Plusieurs méthodes et notations ont été proposées dans ce domaine, les deux méthodes les plus répandues sont :

- **La méthode de Denavit-Hartenberg** : qui est développée pour le paramétrage des robots à structures ouvertes simples, mais présente des ambiguïtés lorsqu'elle est appliquée sur des robots ayant des structures arborescentes ou fermées ;
- **La méthode de Khalil et Kleinfinger** : qui permet la description des robots à architectures : simple, arborescente et fermée.

La méthode de Khalil-Kleinfinger pour les robots à structure arborescente

Une structure arborescente est constituée de n articulations, de $n + 1$ corps rigides notés C_0, \dots, C_n , et éventuellement de plusieurs organes terminaux. Par convention, les corps et les articulations sont numérotés (*Figure II.27*) de la manière suivante :

- La base est fixe et constitue le corps C_0 ;
- Les numéros des corps et des articulations sont croissants sur chaque branche en partant de la base vers un organe terminal ;

- Le corps C_j est articulé par l'articulation j ; autrement dit, l'articulation j connecte le corps au corps $C_{a(j)}$. Le corps $C_{a(j)}$ est le corps antécédent du corps C_j lorsque l'on parcourt la chaîne depuis la base. La topologie du système est complètement définie par la donnée des $a(j)$ pour $j = 1, \dots, n$.

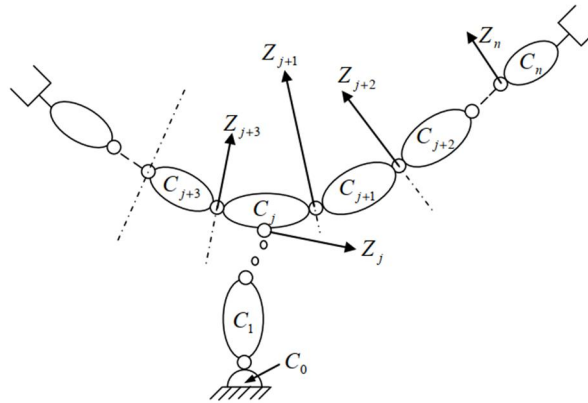


Figure II.29 : Notation associé à une chaîne arborescente [8]

Pour déterminer les paramètres géométriques nécessaires à la définition des transformations entre les différents repères liés aux corps, on place les repères de la manière suivante :

- R_i est fixe par rapport au corps C_i ;
- Z_i est porté par l'axe de l'articulation i ;
- X_i est porté par la perpendiculaire commune à Z_i et à l'axe Z de l'un des corps aval porté par le corps C_i .

Jusque-là ces notations sont parfaitement compatibles avec la description d'une chaîne ouverte simple (méthode de Denavit-Hartenberg).

Si le corps C_i , $i = a(j)$, n'a pas d'arborescence, X_i est la perpendiculaire commune à Z_i et Z_j . Dans le cas où le corps C_i porte plus d'un corps (*Figure II.29*), par exemple les corps C_j et C_k , il faut alors choisir X_i sur l'une des perpendiculaires communes à Z_i et Z_j ou Z_i et Z_k

- Si X_i est la perpendiculaire commune à Z_i et à un autre axe Z_k (*Figure II.29*), on construit la perpendiculaire commune X_i aux axes Z_i et Z_j . Le passage de R_i (défini par X_i et Z_i) à R_j s'effectue à l'aide des quatre paramètres usuels ($\alpha_j, d_j, \theta_j, r_j$). Pour définir le passage de R_i à R_i' deux paramètres supplémentaires doivent être introduits :

- γ_j : angle entre X_i et X_i' correspondant à une rotation autour de Z_i ;
- b_j : distance entre X_i et X_i' correspondant à une translation le long de Z_i ;

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

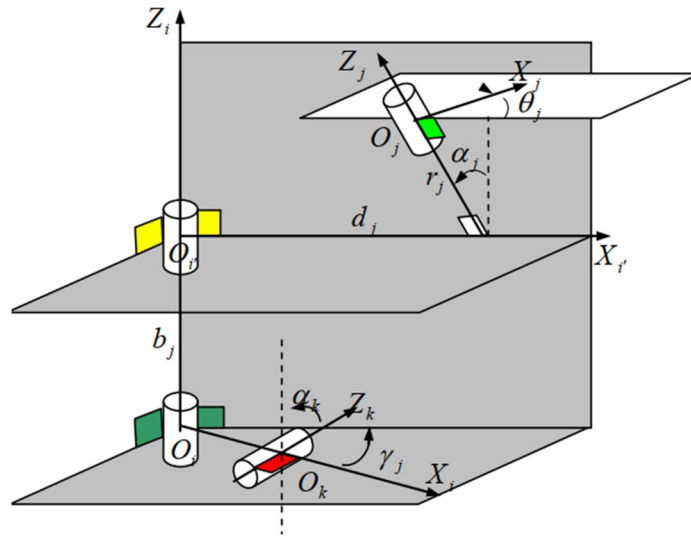


Figure II.30: Paramètres géométrique pour un corps à plus de deux articulations [8]

Nous allons dès à présent résumer les conventions pour l'assignation des repères et la définition des Paramètre dans le tableau suivant :

Tableau II.4: Assignation des repères et définition des paramètres [8]

Règles de construction :	$Z_j =$ axe de l'articulation j supportant le corps C_j $X_i' =$ axe perpendiculaire à $(Z_i \& Z_j)$ / $X_i =$ axe perpendiculaire à $(Z_i \& Z_k)$				
Variable	Axe de référence	De l'axe	A L'axe	Remarque	Type de paramètre
γ_j	Z_i	X_i	X_i'	angle spécial pour les ramifications	paramètres d'embranchement
b_j	Z_i	X_i	X_i'	distance spéciale pour les ramifications	
α_j	X_{j-1}	Z_{j-1}	Z_j	angle entre les axes de deux articulations	Paramètres fixes

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

d_j	X_{j-1}	Z_{j-1}	Z_j	distance entre les axes de deux articulations	Variables d'articulation
θ_j	Z_j	X_{j-1}	X_j	pour les articulations rotatives	
r_j	Z_j	X_{j-1}	X_j	pour les articulations prismatiques	

On aura donc besoin des six paramètres géométriques qui permettent de construire la matrice de passage ${}^i T_j$ du repère R_j au repère R_i , cette matrice est donnée par :

$$\boxed{{}^i T_j = \text{Rot}(Z, \gamma_j) \text{Trans}(Z, b_j) \text{Rot}(X, \alpha_j) \text{Trans}(X, d_j) \text{Rot}(Z, \theta_j) \text{Trans}(Z, r_j)} \quad (\text{II.1})$$

Après son développement, on obtient :

$$T_j = \left(\begin{array}{cccc} C\gamma_j C\theta_j - S\gamma_j C\alpha_j S\theta_j & -C\gamma_j S\theta_j - S\gamma_j C\alpha_j C\theta_j & S\gamma_j S\alpha_j & d_j C\gamma_j + r_j S\gamma_j S\alpha_j \\ S\gamma_j C\theta S\alpha_j + S C\theta \gamma_j C\alpha_j S\theta_j & -S\gamma_j S\theta S\alpha_j + C C\theta \gamma_j C\alpha_j C\theta_j & -C\gamma_j S\alpha_j & d_j S\gamma_j - r_j C\gamma_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j + b \\ 0 & 0 & 0 & 1 \end{array} \right) \quad \text{II.2}$$

C'est cette méthode qu'en va utiliser dans nos calculs car elle représente la forme générale de la matrice de transformation en décrivant les robots à structures : ouverte simple et s'étend aux robots à structures : arborescente et fermée.

La transformation inverse correspondante à la notation de Khalil et Kleinfinger est :

$${}^jT_i = \begin{pmatrix} & & & -b_j S\alpha_j S\theta_j - d_j C\theta_j \\ & & & -b_j S\alpha_j C\theta_j - d_j S\theta_j \\ & & & -b_j C\alpha_j S\theta_j - r_j \\ 0 & & 0 & 1 \end{pmatrix} \quad \text{II.3}$$

Chaîne cinématique d'une patte

La chaîne cinématique choisie pour la patte du robot est représentée sur la (**Figure II.30**), cette patte est constituée de trois tiges rigides de longueurs respectives l_1 , l_2 et l_3 , articulées entre elles par trois articulations rotoïdes θ_1, θ_2 et θ_3 .

L'articulation rotule permet un contact complet du pied de la patte avec le sol, la longueur est considérée comme nulle ($l_4 = 0$) et ($\theta_4 = \theta_5 = \theta_6 = 0$).

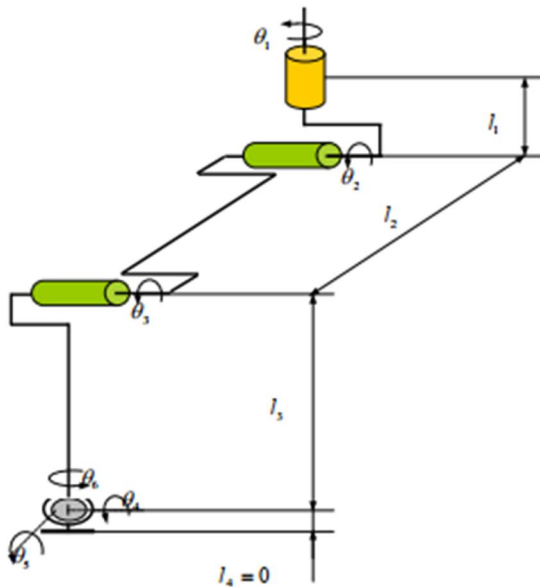


Figure II.31: La chaîne cinématique d'une patte [8]

Description du banc d'essai d'une patte

Pour que cette patte puisse effectuer librement ses différents mouvements comme si elle est rattachée à un robot marcheur réel qui se déplace sur un terrain naturel, on propose le banc d'essai suivant (**Figure II.31**), ce dispositif doit supporter la patte en lui permettant d'effectuer les deux mouvements suivants :

- Coulissement vertical pour changer l'attitude du centre de gravité du robot, ;

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

- Coulissement horizontal pour déplacer et pivoter le centre de gravité du robot. ;

Ce banc d'essai est constitué de deux barres verticales encastrées à une base fixe sur lesquelles coulisse une barre horizontale.

Pendant la phase "patte en l'air", on doit maintenir le centre de gravité du robot à une hauteur voulue grâce au bon choix des 4 ressorts de compression calibrés (R_1, R_2, R_3 et R_4), et au même temps éliminer tout mouvement relatif sur la barre horizontale.

Quand le pied de la patte entre en contact avec le sol pendant la phase " patte posée", le Centre de gravité du robot doit coulisser sans pivoter le long de la barre horizontale. Ce mécanisme (patte et son banc d'essai) compte en tout cinq articulations dont les deux premières sont passives, alors que les trois dernières formant la patte proprement dite sont actives.

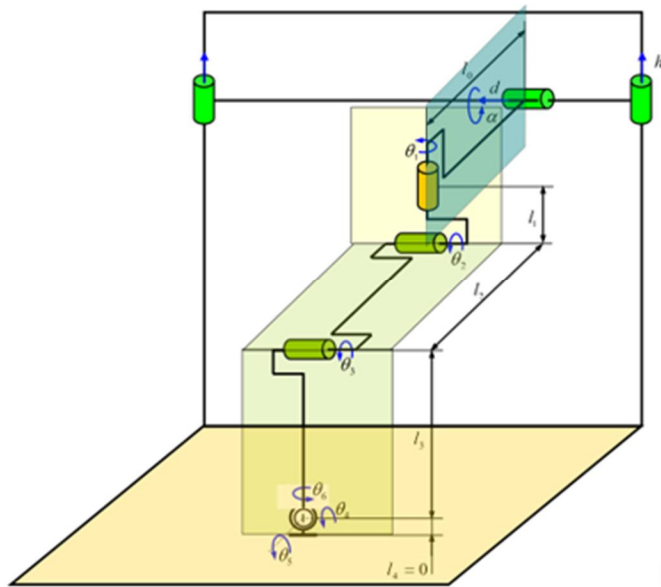


Figure II.32 : Chaîne cinématique d'un banc d'essai de la patte [8]

Repérage du mécanisme : patte et son banc d'essai

On va attribuer à chaque corps appartenant à l'ensemble (atelier et mécanisme) (**Figure II.38**) son repère associé de la façon suivante :

Tableau II.5: Repère de la patte et du banc d'essai

Repère	Lié à (au)
$R_0(X_0, Y_0, Z_0)$	repère atelier ;
$R_1(X_1, Y_1, Z_1)$	centre de gravité du corps du robot;

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

$R_2(X_2, Y_2, Z_2)$	Première articulation de la patte liant la patte au corps du robot ;
$R_3(X_3, Y_3, Z_3)$	deuxième articulation de la patte ;
$R_4(X_4, Y_4, Z_4)$	troisième articulation de la patte ;
$R_5(X_5, Y_5, Z_5)$	pied de la patte du robot.

Pour passer du repère R_0 (repère atelier) au repère R_5 (lié au pied de la patte) selon la notation de Khalil et Kleinfinger, les paramètres géométriques (**Tableau II.6**) valent :

Tableau II.6: Paramètres géométriques relatifs au mécanisme

j	σ_j	γ_j	b_j	α_j	d_j	θ_j	r_j
1	1	β	h	α	d	θ	r
2	0	ϕ	0	0	l_0	θ_1	0
3	0	0	$-l_1$	$-\pi/2$	0	θ_2	0
4	0	0	0	0	l_2	θ_3	0
5	0	0	0	0	l_3	0	0

Calcul du MGD pour une patte

Le MGD de la patte consiste à déterminer les coordonnées opérationnelles du bout de la patte lié à R_5 par rapport au repère de la base R_0 en fonction des coordonnées articulaires $q = [\theta_1 \theta_2 \theta_3]^T$, ce modèle est donnée par le produit matriciel suivant :

$${}^0T_5 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 \quad (\text{II.4})$$

Le calcul des matrices de transformation homogène successives suivantes nous donne :

$${}^0T_1 = \begin{pmatrix} C\beta C\theta_j - S\beta C\alpha_j S\theta & -C\beta S\theta_j - S\beta C\alpha C\theta & S\beta S\alpha & dC\beta + rS\beta S\alpha \\ S\beta C\theta + C\beta C\alpha S\theta & -S\beta S\theta + C\beta C\alpha C\theta & -C\beta S\alpha & d_j S\gamma_j - rC\beta S\alpha \\ S\alpha S\theta & S\alpha C\theta & C\alpha & rC\alpha + h \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{II.5}$$

$${}^1T_2 = \begin{pmatrix} C\varphi C\theta_1 - S\varphi C\alpha S\theta_1 & -C\varphi S\theta_1 - S\varphi C\theta_1 & 0 & l_0 C\varphi \\ S\varphi C\theta_1 + C\varphi S\theta_1 & -C\varphi C\theta_1 + S\varphi S\theta_1 & 0 & l_0 S\varphi \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{II.6}$$

$${}^2T_3 = \begin{pmatrix} C\theta_2 & -S\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S\theta_2 & -C\theta & 0 & -l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{II.7}$$

$${}^3T_4 = \begin{pmatrix} C\theta_3 & -S\theta_3 & 0 & l_2 \\ S\theta_3 & C\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{II.8}$$

$${}^4T_5 = \begin{pmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{II.9}$$

Configuration d'une patte à partir du MGD

A l'aide des paramètres géométriques du mécanisme et aux variables articulaires (θ_1, θ_2 et θ_3) de la patte, nous allons ressortir le MGD de la patte et de l'hexapode :

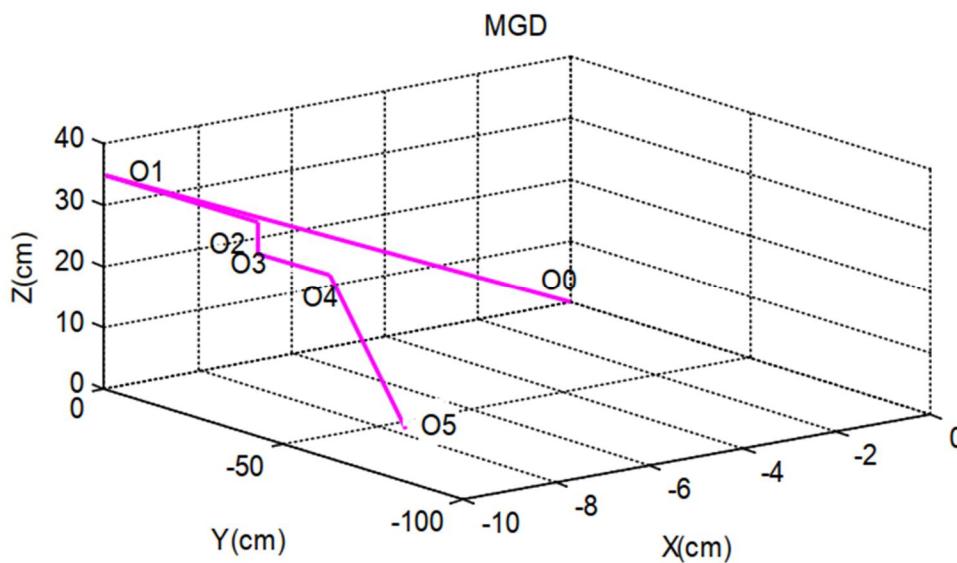


Figure II.33: MGD d'une patte pour $\theta_1 = 0, \theta_2 = 0, \theta_3 = \pi/4$ [8]

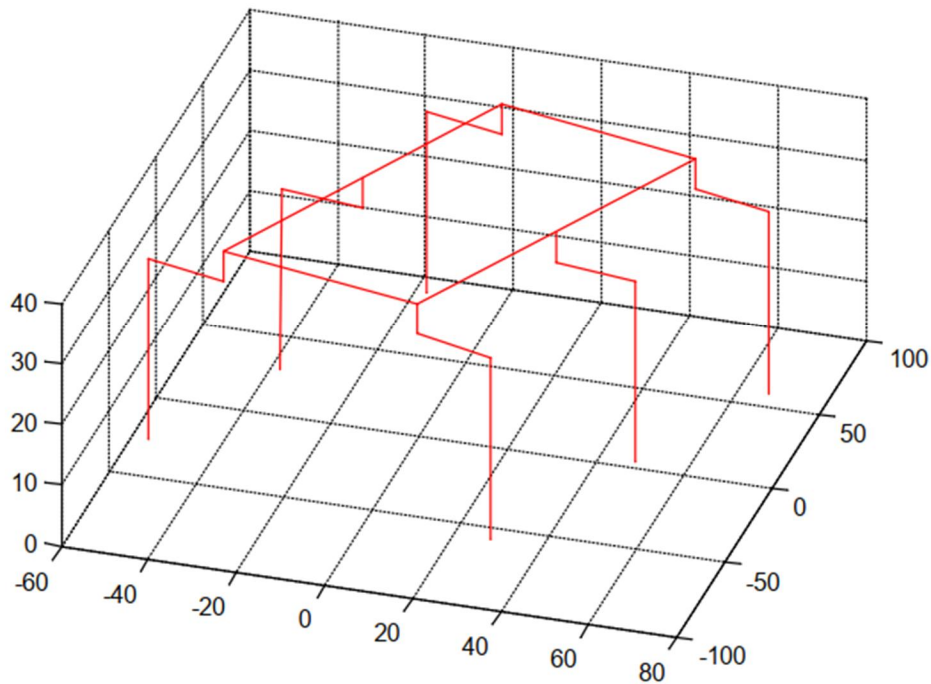


Figure II.34: Schéma du MGD pour l'hexapode pour $\theta_1 = 0, \theta_2 = 0, \theta_3 = \pi/2$ [8]

III.1.2 Modélisation Cinématique du robot

Le modèle cinématique direct (**Figure II.15**) d'un robot décrit les vitesses des coordonnées Opérationnelles en fonction des vitesses articulaires, il est noté :

$$\dot{X} = J(q) \dot{q} \quad (\text{II.10})$$

Où $J(q)$ désigne la matrice Jacobienne de dimension $m \times n$ du mécanisme qui est fonction de la variable articulaire q . Cette matrice est à la base du calcul du modèle cinématique, elle facilite aussi le calcul des singularités et la dimension de l'espace opérationnel accessible du robot.

Calcul de la matrice Jacobienne de base

On calcule la $k^{ième}$ colonne de la matrice Jacobienne, notée ${}^i J_{n,k}$, projetée dans le repère R_i par la formule :

$${}^i j_{n;k} = \begin{pmatrix} \varphi_k {}^i \alpha_k + \sigma k (-{}^k p_{ny} {}^i s_k + {}^k p_{nx} {}^i n_k) \\ \sigma k {}^i \alpha_k \end{pmatrix} \quad i = 0, \dots, n; k = 1, \dots, n \quad (II.11)$$

Où :

- ${}^i s_k, {}^i n_k$ et ${}^i a_k$: sont respectivement le 1^{ère}, 2^{ème} et 3^{ème} vecteurs la matrice ${}^i A_k$;
- ${}^i A_k$: matrice d'orientation de dimension (3×3) du repère R_k dans le repère R_i ;
- ${}^k P_{nx}$ et ${}^k P_{ny}$: sont respectivement la 1^{ère} et 2^{ème} composantes du vecteur ${}^k P_n$ qui est la quatrième colonne de ${}^k T_n$ calculée précédemment par le modèle géométrique direct. .

Le MCD de la patte

Le calcul de la matrice jacobienne notée ${}^2 J_5$ (projetée dans le repère R_2 et de dimensions (6×5)) est effectué en premier lieu pour tout le mécanisme comportant les cinq articulations ($\theta_1, \theta_2, \theta_3$ et θ_4), soit :

$$\begin{pmatrix} {}^2 V_{5,x} \\ {}^2 V_{5,y} \\ {}^2 V_{5,z} \\ {}^2 \omega_{5,1} \\ \dots \end{pmatrix} = \begin{pmatrix} {}^2 J_{5(1,1)} & {}^2 J_{5(1,2)} & {}^2 J_{5(1,3)} & {}^2 J_{5(1,4)} & {}^2 J_{5(1,5)} \\ {}^2 J_{5(2,1)} & {}^2 J_{5(2,2)} & {}^2 J_{5(2,3)} & {}^2 J_{5(2,4)} & {}^2 J_{5(2,5)} \\ {}^2 J_{5(3,1)} & {}^2 J_{5(3,2)} & {}^2 J_{5(3,3)} & {}^2 J_{5(3,4)} & {}^2 J_{5(3,5)} \\ {}^2 J_{5(4,1)} & {}^2 J_{5(4,2)} & {}^2 J_{5(4,3)} & {}^2 J_{5(4,4)} & {}^2 J_{5(4,5)} \\ {}^2 J_{5(5,1)} & {}^2 J_{5(5,2)} & {}^2 J_{5(5,3)} & {}^2 J_{5(5,4)} & {}^2 J_{5(5,5)} \\ {}^2 J_{5(6,1)} & {}^2 J_{5(6,2)} & {}^2 J_{5(6,3)} & {}^2 J_{5(6,4)} & {}^2 J_{5(6,5)} \end{pmatrix} \begin{pmatrix} \Theta \\ \Theta_1 \\ \Theta_2 \\ \Theta_3 \\ \Theta_4 \end{pmatrix} \quad (II. 12)$$

Comme les articulations θ et θ_4 sont inactives alors :

$${}^2V_5 = \begin{pmatrix} {}^2V_{5,x} \\ {}^2V_{5,y} \\ {}^2V_{5,z} \\ {}^2\omega_{5,x} \\ {}^2\omega_{5,y} \\ {}^2\omega_{5,z} \end{pmatrix} = \begin{pmatrix} {}^2J_{5(1,2)} & {}^2J_{5(1,3)} & {}^2J_{5(1,4)} \\ {}^2J_{5(2,2)} & {}^2J_{5(2,3)} & {}^2J_{5(2,4)} \\ {}^2J_{5(3,2)} & {}^2J_{5(3,3)} & {}^2J_{5(3,4)} \\ {}^2J_{5(4,2)} & {}^2J_{5(4,3)} & {}^2J_{5(4,4)} \\ {}^2J_{5(5,2)} & {}^2J_{5(5,3)} & {}^2J_{5(5,4)} \\ {}^2J_{5(6,2)} & {}^2J_{5(6,3)} & {}^2J_{5(6,4)} \end{pmatrix} \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \end{pmatrix} \quad (\text{II. 12})$$

En séparant cette Jacobienne en deux sous matrices : la 1ère en haut, carrée de dimensions (3×3) , nous permet de calculer le vecteur vitesse de translation 2V_5 du pied de la patte dans le repère R_2 :

$$\begin{pmatrix} V_{5,x} \\ V_{5,y} \\ V_{5,z} \end{pmatrix} = \begin{pmatrix} {}^2J_{5(1,2)} & {}^2J_{5(1,3)} & {}^2J_{5(1,4)} \\ {}^2J_{5(2,2)} & {}^2J_{5(2,3)} & {}^2J_{5(2,4)} \\ {}^2J_{5(3,2)} & {}^2J_{5(3,3)} & {}^2J_{5(3,4)} \end{pmatrix} \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \end{pmatrix} \quad (\text{II. 13})$$

$${}^2JV_5 = \begin{pmatrix} 0 & -S\theta_3 l_3 C\theta_2 - S\theta_2 l_3 C\theta_3 & l_3 - S\theta_2 l_2 & -S\theta_3 l_3 C\theta_2 - S\theta_2 l_3 C\theta_3 & l_3 \\ l_3 C\theta_2 C\theta_3 - l_3 S\theta_2 S\theta_3 + C\theta_2 l_3 & 0 & 0 & 0 & 0 \\ 0 & l_3 S\theta_2 S\theta_3 - l_3 C\theta_2 C\theta_2 - C\theta_2 l_2 & l_3 S\theta_2 S\theta_3 - l_3 C\theta_2 C\theta_2 & l_3 S\theta_2 S\theta_3 - l_3 C\theta_2 C\theta_2 & 0 \end{pmatrix} \quad \text{II.14}$$

La 2ème sous matrice de bas, carrée de dimensions (3× 3) , nous permet de calculer le vecteur vitesse de rotation ${}^2\omega_5$ du pied de la patte dans le repère R_2 :

$${}^2J\omega_5 = \begin{pmatrix} {}^2\omega_{5,x} \\ {}^2\omega_{5,y} \\ {}^2\omega_{5,z} \end{pmatrix} = \begin{pmatrix} {}^2J_{5(4,2)} & {}^2J_{5(4,3)} & {}^2J_{5(4,4)} \\ {}^2J_{5(5,2)} & {}^2J_{5(5,3)} & {}^2J_{5(5,4)} \\ {}^2J_{5(6,2)} & {}^2J_{5(6,3)} & {}^2J_{5(6,4)} \end{pmatrix} \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \end{pmatrix} \quad (\text{II. 15})$$

Donc, la vitesse de rotation du pied de la patte est donnée par :

$${}^2\omega_5 = \begin{pmatrix} {}^2\omega_{5,x} \\ {}^2\omega_{5,y} \\ {}^2\omega_{5,z} \end{pmatrix} = \begin{pmatrix} -S23 & 0 & 0 \\ -C23 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \end{pmatrix} \quad (\text{II. 16})$$

Ainsi la modélisation de notre robot achevée, nous passons à la phase de conception de notre robot.

IV. CONCEPTION DU ROBOT

IV.1 Architecture du système

Dans cette section, il est question de mettre en évidence les différents modules en définissant le comportement de chacun d'eux, elle présente l'architecture du système de l'introduction des données d'entrées en passant par le traitement de ces données pour assurer non seulement la communication.

IV.2 Constitution du robot

Il s'agit ici de ressortir la structure mécanique qui est le squelette du robot, pour cela nous allons présenter le châssis du robot qui est le support sur lequel sont fixés tous les éléments du robot (moteurs, roues et circuits électroniques), la chaîne cinématique et le synoptique.

IV.1.1 Le Châssis

Comme il existe de nombreuses plates-formes pour les logiciels de CAO des modules 3D et avec toutes les informations de base et les connaissances sur la commande, tout le monde peut créer ses propres modules 3D. Pour la conception des modules 3D, j'ai utilisé une plateforme en ligne [14] Pour la conception des modules 3D, je dois d'abord créer le compte et me connecter car j'ai créé un compte étudiant, je peux accéder à toutes les fonctionnalités de onshape. Pour la conception des modules 3D, j'ai pris la référence de conception de l'un des projets disponibles sur ces sites inscrutables Référence de ce projet pour la conception du composant de mon hexapode, mais tout le design est fait par moi comme eux.

Le choix de notre châssis s'est porté sur le modèle à six (6) pattes dont les composants sont les suivants :

- partie supérieure du corps x1 ;
- partie inférieure du corps x1 ;
- Coxa gauche x 3 ;
- Droite Coxa x3 ;
- Fémur x6 ;
- tibia gauche x 3 ;
- tibia droit x3 ;
- Sholder x12.

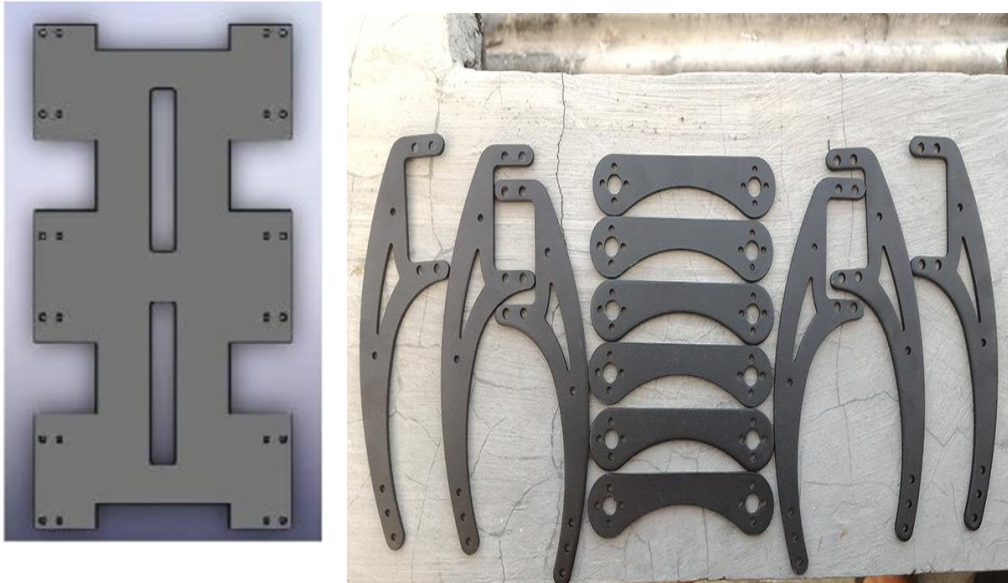


Figure II. 35: Le châssis de notre robot

IV.1.2 Description du matériel

Ici, il est question d'étudier et de choisir les différents composants nécessaires à la réalisation de notre robot. Pour cela nous allons ressortir le schéma synoptique du montage, faire une étude des différents blocs et composants que constitue notre prototype. Il est constitué de plusieurs éléments :

- La batterie permet d'alimenter tous les éléments du montage avec différents niveaux de tension;
- Le capteur ultrason permet au robot d'éviter les obstacles ;
- Le servomoteur permet au capteur ultrason de baliser la zone dans laquelle le robot évolue ;
- Les moteurs à courant continu permettent le déplacement du robot ;
- Le module L298N permet de contrôler le sens de rotation des moteurs ;
- La carte Arduino MEGA 2560 est l'élément mère de notre projet car c'est elle qui fait le traitement et le contrôle de données.

IV.2 Etude des différents blocs

IV.2.1 Alimentation

Elle fournit deux niveaux de tension permettant d'alimenter les éléments du robot en énergie le rendant autonome.

Choix de la batterie

Les tensions d'alimentation des différents modules sont données dans le tableau ci-dessous :

Tableau II. 7: Consommation en énergie de tout le montage

Composants	Alimentation
Carte Arduino Mega	5V/20mA
Capteur ultrasonique	5V/20mA
Moteur à courant continu	5V/1A
L298N	12V/25mA

Il est judicieux d'utiliser une batterie au lithium rechargeable ayant une caractéristique sont adaptées de manière à délivrer deux (02) niveaux de tensions 12V et 7,4V. Le 12V permet d'alimenter le driver (L298N) des moteurs et le capteur magnétique ; le 7,4V sera régulée à 5V par le 7805 pour alimenter la carte Arduino.

IV.2.2 Le microcontrôleur Arduino

Un microcontrôleur est une unité de traitement de l'information de type microprocesseur contenant tous les systèmes informatiques à savoir le microprocesseur, des mémoires et des périphériques. Les microcontrôleurs ont des performances réduites et consomment peu d'énergie, les rendant indispensables dans toute solution d'électronique embarquée (voiture, robots, ...). La carte Arduino n'est pas le microcontrôleur le plus puissant, mais son architecture a été publiée en open-source, et toute sa philosophie s'appuie sur le monde du libre, au sens large. La carte Arduino se relie à un ordinateur par un câble USB. Nous avons plusieurs types de cartes Arduino :

- La carte arduino LENARDO ;
- La carte arduino Mega 2560;
- La carte Arduino Mega ADK ;
- La carte Arduino DUE ;
- La carte Arduino Nano ;
- La carte Arduino Mini Pro ;
- La carte Arduino Yun ;

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

- La carte Arduino UNO.

Le tableau ci-dessous montre les différentes caractéristiques afin de bien choisir notre carte.

Tableau 8: Tableau comparatif Arduino[8]

Nom	Taille (L×l×h) mm	Microcontrôleur	Mémoire flash	SRAM	EEPROM	USB	E/S numérique (PWM)	Entrée analogique	Alimentation en V
LEONARDO	70×54×16	Atmega32u4	32	2,5	1	Alimentation jack, Micro USB	20(7)	12	7-12
MEGA2560	107×53×15	Atmega2560	256	8	4	Alimentation jack, USB	54(15)	16	7-12
MEGA ADK	102×54×16	Atmega2560	256	8	4	Alimentation jack, USB	54(15)	16	7-12
DUE	102×53×16	AT91SAM3X8E	512	96	Alimentation jack, USB	52(12)	12	7-12	
NANO	44×20×18	Atmega168/328	16/32	½	0,5/1	Mini_B USB	14(6)	9	7-12
MINI PRO	33×18×10	Atmega168	16	1	0,5	UART	14(6)	8	3,3/12
YUN	72×53×17	Atmega32u4	32	2,5	1	Micro USB, Ethernet	20(7)	20	5
UNO	74×54×16	Atmega328	32	2	1	Alimentation jack, USB standard-B	14(6)	6	7-12

Notre choix s'est porté sur l'utilisation d'une carte Arduino MEGA 2560 pour car elle est disponible sur le marché, peu coûteux, facile à utiliser et répond à nos besoins

La carte Arduino MEGA 2560

L'Arduino Méga est une carte à microcontrôleur basé sur l'ATmega 2560 (voir fiche technique). Il dispose de 54 broches numériques d'entrée / sortie (dont 14 peuvent être utilisées comme sorties PWM), 16 entrées analogiques, 4 UART (ports série matériels), un oscillateur en cristal de 16 MHz, d'une connexion USB, une prise d'alimentation, d'une embase ICSP et un bouton de réinitialisation. Il contient tout le nécessaire pour soutenir le

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

microcontrôleur, il suffit de le connecter à un ordinateur avec un câble USB ou avec un adaptateur AC-DC ou batterie pour commencer.

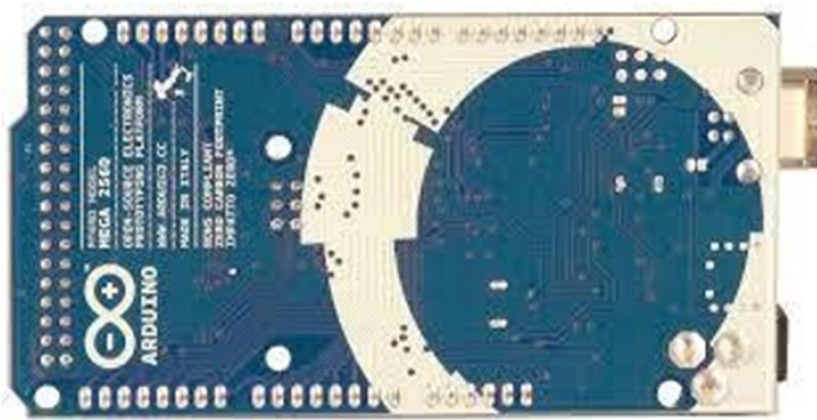
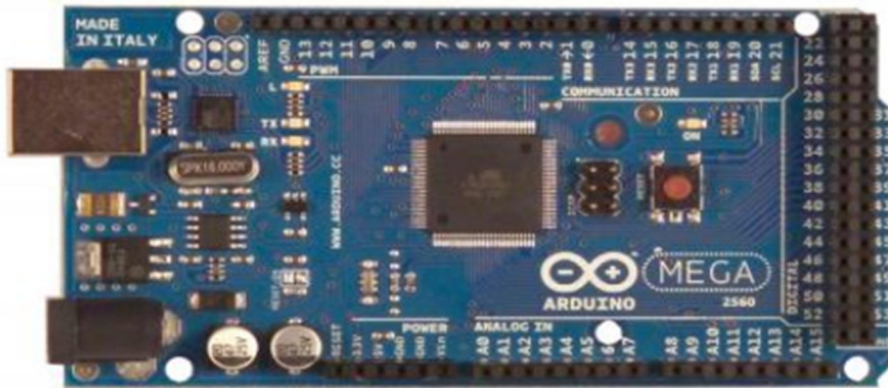


Figure 36 : Vue de la carte Arduino MEGA 2560[8]

Caractéristiques

Le tableau ci-dessous nous montre les caractéristiques détaillées de l'Arduino Mega 2560 :

Tableau II. 9: Caractéristiques de l'Arduino Mega 2560

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Tension d'entrée (limites)	6-20V

E / S numériques Pins	54 (dont 14 fournissent la sortie PWM)
Pins d'entrée analogique	16
DC Courant par E/S Pin	40 mA
Courant DC pour 3.3V Pin	50 mA
Mémoire flash	256 Ko qui 8KB utilisé par bootloader
SRAM	8 KB
EEPROM	4 KB
Vitesse de l'horloge	16 MHz

IV.2.3 Le capteur ultrason (HC-SR04)

Il est constitué d'un émetteur et d'un récepteur qui sont des cellules piézoélectriques. L'émetteur vibre lorsqu'une tension lui est appliquée tandis que le récepteur va produire une tension lorsqu'il reçoit une vibration. Il possède quatre broches : Vcc, GND, Echo et Trigger.



Figure II. 37: module HC-SR04 [5]

- **Le principe de fonctionnement est le suivant :**

Pour déclencher une mesure :

- On envoie une impulsion HIGH de 10 μ s sur la broche TRIGGER du capteur ;
Le capteur envoie alors une série de 8 impulsions à 40KHz (inaudible pour l'être humain) ;

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

- Les ultrasons se propagent dans l'air jusqu'à toucher un obstacle et retournent dans l'autre sens vers le capteur ;
- Le capteur détecte l'écho et clôture la prise de mesure.

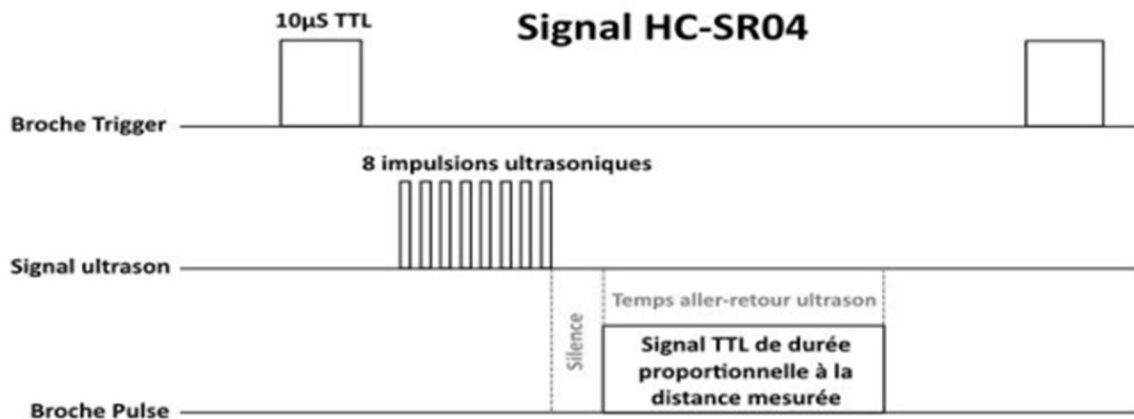


Figure 38: Illustration du signal TRIGGER et ECHO [5]

• Calcul de la distance d'un objet

La distance parcourue par un son se calcule en multipliant la vitesse du son, environ

340 m/s (ou 34000 cm/1000000 µs) par le temps de propagation, soit :

$$2d = c.t$$

- d étant la distance parcourue ;
- c étant la célérité du son dans le vide ;
- t étant le temps mis par l'onde pour aller et revenir.

Le HC-SR04 donne une durée d'impulsion en dizaines de µs. Il faut donc multiplier la valeur obtenue par 10 µs pour obtenir le temps t . On sait aussi que le son fait un aller-retour. La distance vaut donc la moitié. $d = 34.000 \text{ cm}/1000000 \text{ µs} \cdot 10\text{µs} \cdot \text{valeur} / 2$ en simplifiant $d=170000/1000000\text{cm} \cdot \text{valeur}$. Finalement, $d = 17/1000 \text{ cm} \cdot \text{valeur}$. La formule $d = \text{durée}/58 \text{ cm}$ figure aussi dans le manuel d'utilisation du HC-SR04 car la fraction 17/1000 est égale à 1/58.8235. Elle donne cependant des résultats moins précis.

• Les facteurs d'influences

- Le capteur ultrasonique perturbé lorsque les cas suivants se présentent :
Il est très vite inopérant lorsqu'il y a présence d'échos parasites, en présence de poussière ou encore lorsqu'il se trouve dans un endroit trop étroit ;
- Aucun fonctionnement possible dans le vide ;
- Il y a une zone que l'on appelle zone aveugle et qui correspond à une distance minimum dans laquelle le capteur est inopérant

- **Les caractéristiques techniques du HC-SRO4**

Les caractéristiques sont les suivantes :

- Gammes de distance : 2cm à 4m ;
- Alimentation : 5V ;
- Intensité : 15mA ;
- Résolution : 0.3cm ;
- Fréquence d'utilisation : 40Hz
- Angle de mesure : < 15°.
- Dimension : 45×20×15mm

Le sous-programme de détection d'obstacle est illustré (confer annexes).

IV.2.4 Le module (HC-05)

Le HC-05 possède seulement 6 broches :

- Vcc : alimentation qui sera connecté sur le 5V de l'Arduino ;
- GND : la masse ;
- TX/RX : Pour la communication (liaison série) ;
- KEY : broche qui permet d'envoyer des commandes pour la configuration du module.

Le choix du module HC-05 se justifie par ses caractéristiques:

- Sa taille assez réduite, peu encombrant ; coût faible
- La conversion 5V/3.3V car le module fonctionne sous une tension de 3.3V ;
- La diode électroluminescente de signal.

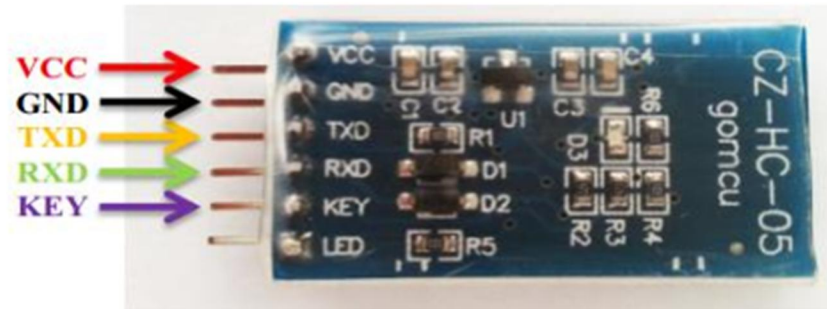


Figure 39: Brochage du HC-05 [6]

IV.2.5 Le servomoteur SG90 9g Micro Servo

Minuscule et léger avec une puissance de sortie élevée. Servo peut tourner d'environ 180 degrés (90 dans chaque sens), et fonctionne exactement comme les types standard, mais plus petites. Vous pouvez utiliser tout Code d'asservissement, le matériel ou la bibliothèque pour contrôler ces servos. Bon pour les débutants qui veulent faire trucs mouvement sans avoir à construire un contrôleur de moteur avec rétroaction et la boîte de vitesses, d'autant qu'il conviendra dans de petits endroits. Il est livré avec une 3 corne (bras) et le matériel.



Figure II. 40 SG90 9g Micro Servo

Les caractéristiques sont les suivantes

- Poids : 9 g ;
- Dimension : 22.2 x 11.8 x 31 mm environ ;
- Couple de décrochage : 1,8 kgf · cm ;
- Vitesse de fonctionnement : / 60 degré de 0,1 ;
- Tension de fonctionnement : 4,8 V (~ 5V) ;
- Mort largeur de bande : 10 us ;
- Plage de température : 0 ° C - 55 ° C ;

Après avoir mené l'étude des différents blocs de notre système, nous allons réaliser le schéma synoptique récapitulant les éléments matériels présentés précédemment :

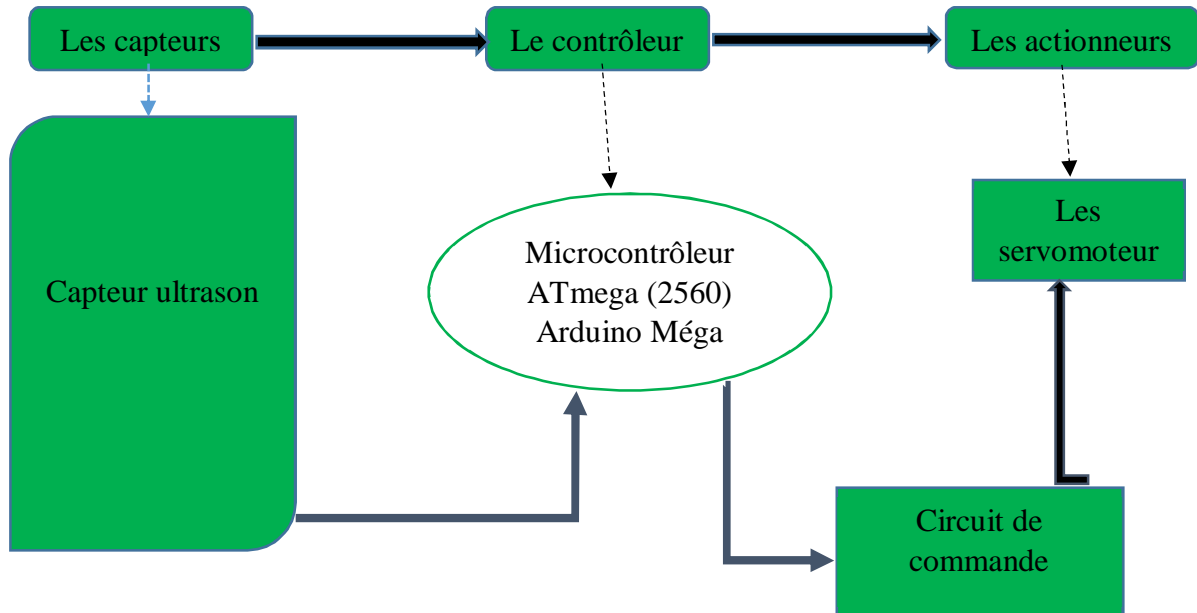


Figure II. 41: Synoptique du robot

V. LA COMMANDE DU ROBOT

Pour la commande de notre robot, nous allons faire recours à l'apprentissage par renforcement, qui est une méthode d'apprentissage par expérience pour définir une action optimale à effectuer dans une situation donnée afin de maximiser les récompenses. C'est également une méthode d'apprentissage orientée objectif qui va conduire à un contrôleur optimal pour la tâche spécifiée par les récompenses. Elle permet aussi de résoudre les problèmes de récompense retardée pour lesquels il faut apprendre à sacrifier une récompense à court terme pour obtenir une plus forte récompense à long terme, et donc apprendre de bonnes séquences d'actions qui permettront de maximiser la récompense à long terme. Les algorithmes d'apprentissage par renforcement sont basés sur les processus markoviens et demandent de discrétiser l'environnement en couples (état-action), c'est cette optique que nous avons choisi l'algorithme du Q Learning.

V.1 Comportement d'évitement d'obstacles

La méthode d'apprentissage Q-learning appartient à la famille des méthodes TD (Temporal Difference learning) permettant d'évaluer la somme des récompenses futures à partir des récompenses immédiates, sans attendre la fin d'un épisode comme pour les méthodes de type Monte Carlo et sans utiliser un modèle de l'environnement comme pour les méthodes de type Programmation Dynamique. L'algorithme de Q-learning a pour objet la construction de la valeur associée au couple (état, action) selon la stratégie optimale. Il procède de façon itérative selon un mécanisme entrelacé d'évaluation et d'amélioration de la stratégie courante. Les conditions de convergence de cet algorithme ont été établies par Watkins et Dayan en 1992. L'actualisation à chaque instant t de la valeur $Q(s, t, a, t)$ associée au couple (état, action) s'écrit :

$$Q(\text{état}, \text{action}) = R(\text{état}, \text{action}) + \text{Gamma} * \text{Max} [Q(\text{état suivant}, \text{toutes les actions})]$$

Le principe de l'algorithme Q-learning, défini formellement par l'algorithme 1.1, est de mettre à jour itérativement, à la suite de chaque transition $(s_n; a_n; s_{n+1}; r_n)$, la fonction de valeur courante Q_n pour le couple $(s_n; a_n)$, où s_n représente l'état courant, a_n l'action sélectionnée et réalisée, s_{n+1} l'état résultant et r_n la récompense immédiate. Cette mise à jour se fait sur la base de l'observation des transitions instantanées et de leur récompense associée :

Algorithme 1.1 : Le Q-learning

```

/*  $\alpha_n$  est un taux d'apprentissage */
Initialiser( $Q_0$ )
pour  $n \leftarrow 0$  jusqu'à  $N_{tot} - 1$  faire
     $s_n \leftarrow$  ChoixEtat
     $a_n \leftarrow$  ChoixAction
     $(s'_n, r_n) \leftarrow$  Simuler( $s_n, a_n$ )
    { mise à jour de  $Q_n$  :}
    début
         $Q_{n+1} \leftarrow Q_n$ 
         $\delta_n \leftarrow r_n + \gamma \max_b Q_n(s'_n, b) - Q_n(s_n, a_n)$ 
         $Q_{n+1}(s_n, a_n) \leftarrow Q_n(s_n, a_n) + \alpha_n(s_n, a_n)\delta_n$ 
    fin
retourner  $Q_{N_{tot}}$ 

```

Dans cet algorithme, N_{tot} est un paramètre initial fixant le nombre d'itérations. Le taux d'apprentissage $\alpha_n(s; a)$ est propre à chaque couple $(s; a)$, et décroît vers 0 à

chaque passage. La fonction *Simuler* retourne un nouvel état et la récompense associée selon la dynamique du système. Le choix de l'état courant et de l'action à exécuter est effectué par les fonctions *ChoixEtat* et *ChoixAction* et sera discuté plus loin. La fonction *Initialiser* revient la plupart du temps à initialiser les composantes de Q_0 à 0, mais il existe des initialisations plus efficaces.

Chaque acteur (contrôleur de mouvement de patte) mène son apprentissage de type Q-learning (selon une stratégie individuelle) sur la base de l'actualisation d'une table de 64×2 valeurs $Q(s, a)$.

La fonction de renforcement pour chaque patte est la suivante :

Pénalité : r = - 1 si	Récompense : r = +1 si
Chute lors d'une proaction	Pas de chute lors de la proaction
L'ordre de proaction est répété deux fois de suite.	Le mouvement de rétraction a dépassé la durée minimale D_{min}
Le mouvement de rétraction a une durée trop longue ($>D_{max}$) ou trop courte ($<D_{min}$)	
Toutes les pattes sont au sol	
Les mouvements de proaction se propagent de l'avant vers l'arrière	

Pour toute politique, il existe un état absorbant de récompense nulle. Dès lors la suite consistera à présenter un comportement d'évitement d'obstacles.

V.2 Comportement d'évitement d'obstacles

Si l'environnement du robot contient un ou plusieurs obstacles, le robot doit pouvoir éviter les collisions avec ces objets. Comme il est présenté précédemment, le système de navigation autonome contient deux comportements élémentaires: comportement de recherche de but et autre pour l'évitement d'obstacle. Le robot exécute l'action adéquate pour atteindre la destination finale en toute sécurité sans risque de collision avec les objets en déclenchant l'un des deux comportements selon la situation perçue.

Le robot doit atteindre le but (**G1**) avec un point de départ (**S1**). Le robot commence à exécuter l'action suivant le comportement actif ; en fonction du contexte actuel. Tout d'abord, le robot se déplace vers l'objectif en activant le comportement (contrôleur) de convergence vers

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

le but avec une vitesse maximale jusqu'à ce qu'il détecte l'obstacle 1 (Obs 1) sur son côté droit, puis il change son comportement à l'évitement d'obstacles au point **A** jusqu'au point **B** au cours de lequel le robot se situe dans un espace libre et il se déplace vers l'objectif jusqu'à le point **C**. A ce moment, le robot détecte en avant l'obstacle 2 (Obs 2) et exécute le comportement d'évitement d'obstacle jusqu'au point **D** en tournant vers la droite. Les points **E**, **F**, **G** et **H** subissent aux mêmes actions. Enfin, le comportement de convergence à l'objectif est activé pour atteindre la configuration finale avec la diminution de la vitesse de déplacement quand il se rapproche vers le but. La coordination des deux comportements se fait par le module présenté ci-dessous :

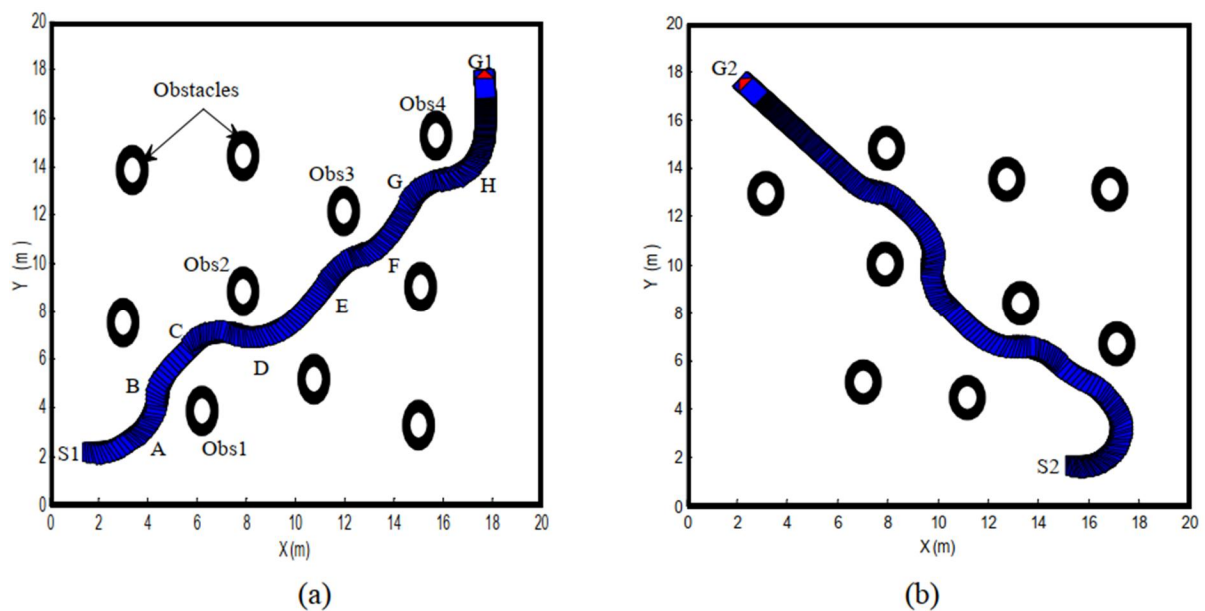


Figure 42: Navigation avec évitement d'obstacles

Conclusion

Il était question dans cette partie de présenter le cahier de charges fonctionnel, la modélisation et concevoir la partie matérielle de notre système en passant de la mise en place de la méthodologie. Dans la suite il sera question de monter le schéma électronique et de présenter la simulation sous Matlab.

CHAPITRE III : RESULTATS ET INTERPRETATIONS

Résumé : dans cette partie, nous allons présenter les différents résultats de nos différentes simulations et les logiciels utilisés.

Aperçu :

INTRODUCTION

- I. PLATEFORME DE PROGRAMMATION
ARDUINO
- II. MONTAGE ELECTRONIQUE
- III. SIMULATION DE LA COMMANDE
- IV. PROTOTYPAGE

CONCLUSION

Introduction

Après avoir finalisé l'étape de conception, nous passons dans ce chapitre aux résultats d'implémentation de notre système robotique qui nous ont permis de réaliser nos programmes de commande. Nous commençons tout d'abord par la présentation des environnements de simulation, ensuite nous enchaînons par la présentation des résultats dans ces plateformes et nous terminons par des discussions

I. PLATEFORME DE PROGRAMMATION ARDUINO

L'interface de l'IDE Arduino est plutôt simple, il offre une interface minimale et épurée pour développer un programme sur les cartes Arduino. Il est doté d'un éditeur de code avec coloration syntaxique et d'une barre d'outils rapide. Ce sont les deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le plus souvent. On retrouve aussi une barre de menus, plus classique qui est utilisé pour accéder aux fonctions avancées de l'IDE. Enfin, une console affichant les résultats de la compilation du code source, des opérations sur la carte, etc...



Figure I. 43: Vue de l'environnement IDE Arduino

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le C, le C++, le Java et le Processing. Le langage impose une structure particulière typique de l'informatique embarquée.

- La fonction « **setup** » contiendra toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de communications série, etc...) ;
- La fonction « **loop** » elle, est exécutée en boucle après l'exécution de la fonction **setup** ;

II. SCHEMA DU MONTAGE ELECTRONIQUE

Le logiciel ISIS de Proteus est principalement connu pour éditeur des schémas électriques. Par ailleurs, permet le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

Le logiciel comporte trois vues principales :

- La « Platine d'essai », où l'on voit les composants tels qu'ils sont dans la réalité et où l'on construit le montage ;
- La « Vue schématique », représentant le schéma fonctionnel du circuit ;
- Le « Circuit imprimé », représentant la vue du circuit imprimé tel qu'il sera sorti en PDF pour être imprimé.

Pour le câblage de l'hexapode, nous avons conçu le schéma de circuit sur le Proteus et développé le circuit sur la carte matricielle en pvc comme indiqué sur les photos. La connexion du servomoteur est courante comme

La carte de commande est équipée de :

- Une carte Arduino Meg 2560 ;
- Des servomoteurs pour faire bouger le robot ;
- Des résistances ;
- Capteur ultrason.

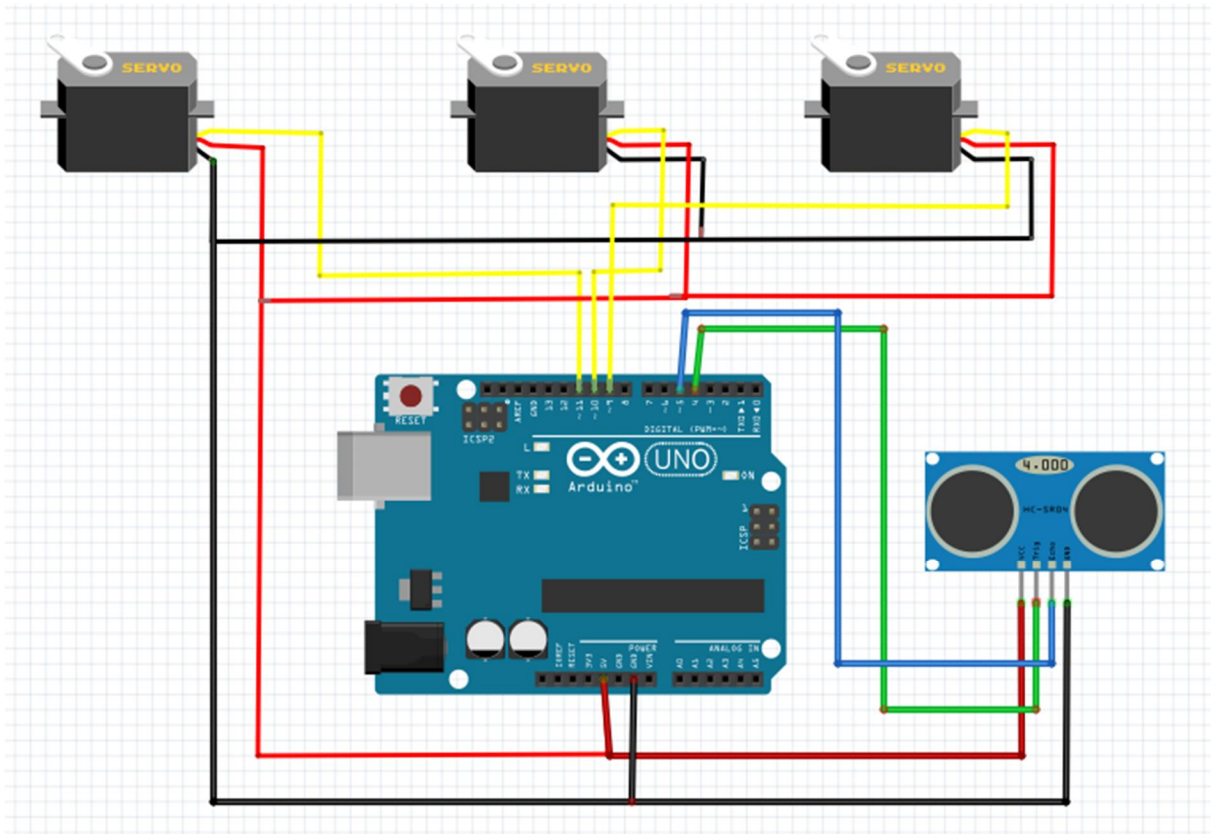


Figure 44; Schéma électronique du robot

III.SIMULATION DE LA COMMANDE SOUS MATLAB

MATLAB est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées, il peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques.

Nous présenterons dans cette section les résultats de simulation de notre système en utilisant MATLAB, nous apercevons la position de l'organe terminal en fonction des différentes valeurs des variables articulaires présenté à la figure ci-dessous nous permettant d'observer le comportement du robot à partir du modèle géométrique direct et inverse (en boucle ouverte) dans un suivi de trajectoire.

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

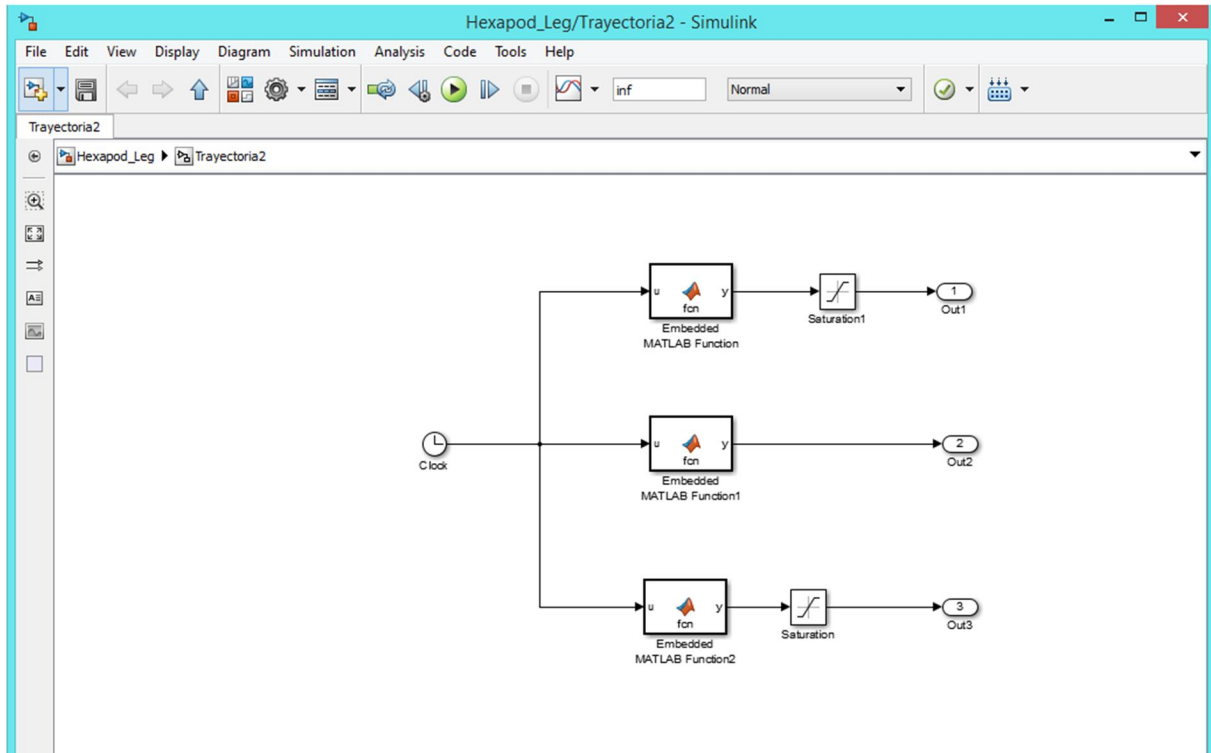
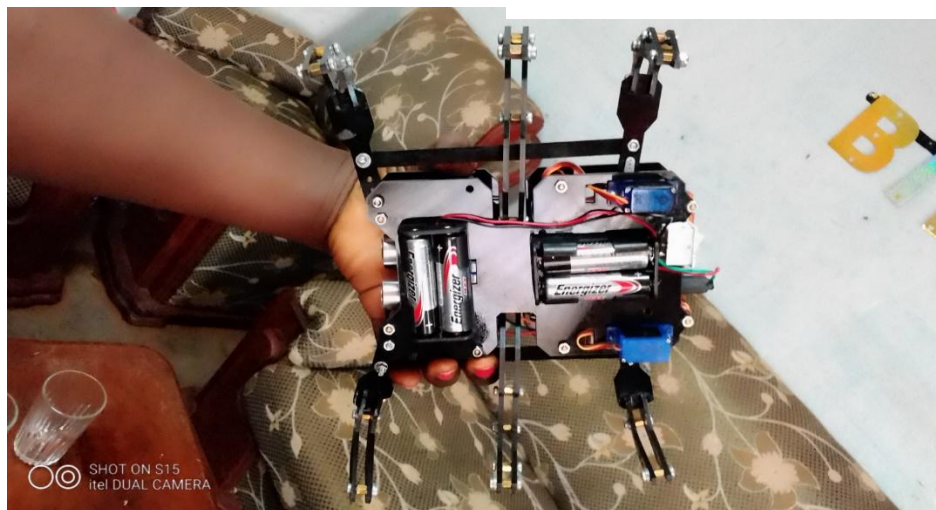
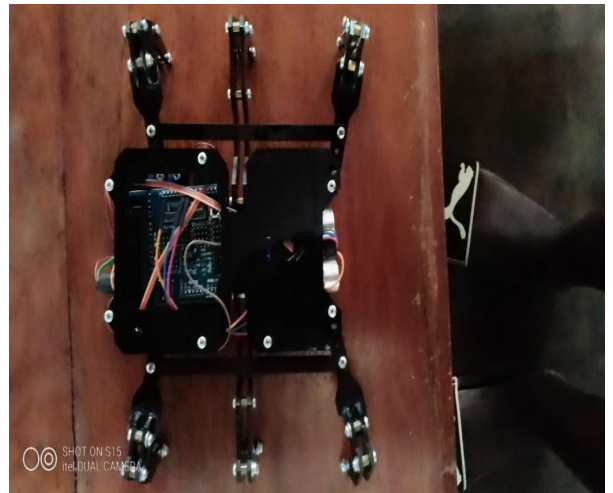


Figure III.45: Schéma d'une patte sur Simulink

Ce schéma représente la trajectoire d'une patte de l'hexapode ou nous avons considéré uniquement le centre de gravité du robot.

IV.PROTOTYPAGE

Prototype final fonctionnel obtenu est le suivant :



Conclusion

Dans ce chapitre nous avons effectué le câblage de l'hexapode, ou nous avons conçu le schéma de circuit, les simulations obtenus nous ont permis de comprendre l'apprentissage de la trajectoire de notre robot à travers la simulation d'une patte sur MATLAB. De même le prototype final obtenu est fonctionnel.

CONCLUSION GENERALE

A partir du thème qui concerne la conception et la réalisation d'un robot hexapode commandé l'algorithme du Q Learning, nous avons commencé le premier chapitre par une étude bibliographique sur les robots mobiles d'une façon générale notamment l'historique des robots mobiles, l'architecture, les types de robots mobiles, leurs classifications, leurs caractéristiques ainsi que les avantages et inconvénients d'utilisation. En outre, nous avons présenté quelques méthodes d'intelligence artificielle et quelques algorithmes de commande de robots.

Ensuite, au deuxième chapitre nous avons présenté en amont le cahier de charges fonctionnel en spécifiant le concept général et les principaux services attendus. L'élaboration de cette partie nous a permis de rappeler le contexte de notre projet et d'énoncer les besoins fonctionnels. Par la suite, nous avons adopté une méthodologie de conception adoptée, nous avons fait la modélisation géométrique et cinématique d'une patte. Cette modélisation nous a permis faire la conception du robot. La conception nous a permis de présenter l'architecture, de décrire la constitution de notre robot de même nous avons présenté le hardware utilisé. Nous avons clôturé notre chapitre en présentant l'algorithme du Q Learning pour la commande du robot ou génération de la trajectoire de marche de l'hexapode,

Au troisième chapitre nous avons réalisé le schéma électronique de notre système et nous avons procéder aux simulations sur MATLAB en intégrant des composants électroniques au cœur du microcontrôleur Arduino. Grâce aux matériels électroniques et aux outils logiciels, nous avons réalisé notre prototype.

Les perspectives restantes ouvertes pour notre travail peuvent concerner les éléments suivants :

- Intégrer un système de localisation avec le module GSM/GPRS dans le but d'avoir les coordonnées géographiques et la position du robot pendant sa navigation dans l'environnement choisi ;
- Les modules émetteur FM pour enregistrer les données dans le cadre de l'espionnage ;
- Implémentation des lois de commande pour simuler sa marche réelle du robot ;
- Faire la génération d'un mouvement de marche en calculant les consignes de référence en position, vitesse et accélération qui sont fonction du temps et assurer le passage des bouts des différentes pattes par des trajectoires imposées, définies selon le mode de la marche du robot.

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

Ce travail nous a permis de connaître les circuits de commande utilisés afin de contrôler les moteurs associés au robot. Ainsi, l'ensemble de système de perception et de déplacement est commandé par un microcontrôleur qui doit en utilisant les informations actuelles, décider l'action à prendre ceci en prenant en compte l'algorithme du Q Learning. Ce thème nous a également permis de traiter des problèmes d'ordre pratique et de vérifier des connaissances théoriques acquises toute le long de notre formation.

REFERENCES BIBLIOGRAPHIQUE

- [1] **Cécile Jean-Jacques**, *'la guerre des robots, les révolutions militaires de demain' parue le 15.09.2006 aux Editions Monde réel*
- [2] **Patrick Reignier**, '' Pilotage réactif d'un robot mobile, Etude de lien entre la perception et l'action''. Thèse de Doctorat, institut national polytechnique de Grenoble (INPG), 1994.
- [3] **VINCENT MAILLE, CYPRIEN ACCARD, BRUNO BRETON** la robotique, apprendre par l'exemple.2016
- [4] **ABAIDI, BALI** réalisation d'un robot mobile avec évitement d'obstacle avec trajectoire programmée. En vue de l'obtention du master.2012
- [5] **AYEBE, NKOULOU,MEFO. PROTOTYPE D'UN ROBOT SE DEPLACANT A L'AIDE D'UN TELEMETRE ULTRASON . 2016.**
- [6] **DOUANLA, KIFACK, NDE. etude et realisation d'un robot suiveur de ligne avec évitement d'obstacles et muni d'une commande bluetooth. memoire de fin d'étude en vue de l'obtention du DPET1. 2017.**
- [7] **ANDRE ZAREMBOWITCH** les ultrasons. Presse universitaire de France.2003
- [8] **Lavi M.Zamstein, A.Antonino Arroyo, Eric M.Schwartz, Sara Keen, Blake C. Sutton, Gorang Gandhi** , '' Koolio : Path planning using Reinforcement Learning on a real robot platform'',.Florida Conference on Recent Advances in Robotics (FCRAR). Miami, 2006.
- [9] **William D. Smart, Leslie Pack Kaelbling** "Effective Reinforcement Learning for Mobile Robots". Proccedings of 2002 IEEE. International Conference on Robotics & Automations, Washington, DC, 2002.
- [10] **Richard S. Sutton and Andrew G. Barto,Ronnald J.Williams**, ''Reinforcement Learning is direct adaptive Optimal Control'', Proc of ACC, Boston, June 1991.
- [11] **Richard S. Sutton and Andrew G. Barto**, '' Reinforcement Learning: An Introduction MIT Press, Cambridge, Massachusetts, 2005.
- [12] **David Filliat**, '' Robotique Mobile'', C10-2 Ecole Nationale Supérieur des Techniques Avancées ENSTA, Octobre 2004.
- [13] **Stephan ten Hagen and Ben Krose** , '' Q-Learning for systems with continuous state and action spaces''. Proc BENELEARN 10th 2000, Belgian-Dutch Conference in Machine Learning, 2000.
- [14] **Claude Touzet**, '' L'apprentissage par Renforcement'', CESAR, USA, Janvier 1998.

[15] Glennec Pierre Yves, " Algorithmes d'Apprentissage Pour Systèmes d'inférence Floue". Editions Hermes, 1999.

REFERENCES WEBOGRAPHIE

[16] www.dfrobot.com visité le 02/04/2020

[17] www.arduino.cc/en/main/arduinoboardmega visité le 15/04/2020

[18] www.carnetdumaker.net/article/controler-un-servomoteur-avec-une-carte-arduino/
visité le 01/05/2020

[19] <https://www.instructables.com/id/DIY-Spider-RobotQuad-robot-Quadruped/> visité le
20/05/2020

[20] www.moxonmotor.com visité le 04/05/2020

ANNEXES

I. Sous-programme détection d'obstacle :

```
Const int trigPin=12;  
Const int echoPin=11;  
Const int buzzer = 13;  
Long duration;  
Int distance;  
Int safetyDistance
```

On commence le code avec quatre constantes : deux constantes pour les broches *TRIGGER* et *ECHO* du capteur, une constante pour le buzzer, une constante qui servira de durée pour la prise de mesure et une constante pour définir la distance qui nous sépare de l'obstacle du son.

```
Void setup () {  
    PinMode (trigPin, OUTPUT);  
    DigitalWrite (trigPin, LOW) ;  
    PinMode (echoPin, INPUT);  
    PinMode (buzzer, OUTPUT);  
    Serial.begin(9600);  
}
```

La fonction *setup ()* initialise le port série, met la broche *TRIGGER* du capteur en sortie et à *LOW*, et met la broche *ECHO* du capteur en entrée.

```
Void loop () {  
    DigitalWrite (trigPin, HIGH);  
    DelayMicroseconds (10);  
    DigitalWrite (trigPin, LOW);  
    Duration = pulseIn (echoPin, HIGH);  
    Distance = duration *0.034/2;  
    SafetyDistance = distance;
```

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

```
If (safetyDistance<=10)
  {DigitalWrite (buzzer, HIGH);
}
Else {
  DigitalWrite (buzzer, LOW);
}
}
```

La fonction *loop* () s'occupe de la mesure et de l'affichage. Elle génère d'abord l'impulsion *HIGH* de 10µs qui déclenche la prise de mesure. Elle mesure ensuite le temps nécessaire pour un aller-retour du signal ultrason avec la fonction *pulseIn* (). Pour finir, elle calcule la distance avant de l'afficher sur le port série.

Tableau des résultats

<u>Distance mm</u>	<u>mesure</u>	<u>Erreur</u>
<u>10</u>	<u>10.20</u>	<u>0.20</u>
<u>20</u>	<u>18.70</u>	<u>1.30</u>
<u>30</u>	<u>23.46</u>	<u>6.54</u>
<u>40</u>	<u>32.64</u>	<u>7.36</u>
<u>50</u>	<u>42.50</u>	<u>7.50</u>
<u>60</u>	<u>52.02</u>	<u>7.98</u>
<u>70</u>	<u>66.30</u>	<u>3.70</u>

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

Tableau 10: devis estimatif du projet

Tableau III. : Devis estimatif du projet N°	Désignation	valeurs/r éférence	PU (FCFA)	Quantité	PT (FCFA)
1	Carte Arduino	Arduino Mega 2560	30 000	1	30 000
2	Capteurs	Ultrason	3 500	1	3 500
3	HC SR 04	7 000		1	7 000
4	Kit robot	2WD	30 000	1	50 000
5	Régulateur	7805	500	1	500
6	Batterie d'alimentation au lithium	3,7V/220 0mAh	3 000	6	18 000
7	Plaque à trous	7×12 (cm)	2 000	1	2 000
8	Connecteurs	Pins	500	2	1 000
9	Câbles de connexion	/	/	4 000	
10	Servo moteur	6000	03	18000	
Prix total des composants			89000		
Divers			30000		
Main d'oeuvre		30%			20000
Prix total de la réalisation			139000		

TABLE DES MATIERES

DEDICACE	i
SOMMAIRE	iii
LISTE DES FIGURES	iv
LISTE DES TABLEAUX	vi
LISTE DES ABREVIATIONS	vii
GLOSSAIRE	viii
AVANT-PROPOS	ix
RESUME	x
ABSTRACT	xi
INTRODUCTION GENERALE	1
CHAPITRE I : REVUE DE LA LITTERATURE SUR LA ROBOTIQUE MOBILE	3
CHAPITRE I : REVUE DE LA LITTERATURE SUR LA ROBOTIQUE MOBILE	3
INTRODUCTION	4
I. Généralités sur la robotique mobile	4
I.1 Définitions	4
I.2 Historique des robots mobiles	5
I.3 Architecture des robots mobiles	8
I.4 Classification des robots mobiles	8
II LES METHODES INTELLIGENTES	17
II.1 LES METHODES D'INTELLIGENCES ARTIFICIELLES	17
II.2 Les algorithmes de commande de robot	19
III. ETAT DE L'ART	20
CONCLUSION	24
CHAPITRE II : METHODE DE CONCEPTION ET MODELISATION	25
CHAPITRE II : METHODE DE CONCEPTION ET DE MODELISATION	25
INTRODUCTION	26
I. CAHIER DE CHARGES FONCTIONNEL	26
I.1 Le concept général et les principaux services attendus	26
I.1.1 Formulation du besoin	26
I.2.2 Les clients, utilisateurs et usagers potentiels	26
I.2 Contexte du projet	26
I.2.1 La nécessité de concevoir notre dispositif	27

CONCEPTION ET REALISATION D'UN ROBOT HEXAPODE COMMANDE PAR L'ALGORITHME DU Q-LEARNING

I.2.2	Limite de l'étude	27
I.3	Enoncé fonctionnel du besoin : comprendre et analyser le besoin	27
I.3.1	Analyse fonctionnelle du besoin (bête à corne)	28
I.3.2	Validation du besoin	28
I.3.3	Enoncé des fonctions, des services et des contraintes	29
I.4	Organisation des fonctions : méthode APTE.....	30
I.5	Caractérisation des fonctions	32
I.6	Le diagramme FAST	35
II.	METHODOLOGIE DE CONCEPTION	39
III	MODÉLISATION DU ROBOT HEXAPODE	41
III.1	Modélisation géométrique et cinématique	41
III.1.1	Modélisation Géométrique Directe des robots	42
III.1.2	Modélisation Cinématique du robot.....	51
IV.	CONCEPTION DU ROBOT.....	54
IV.1	Architecture du système	54
IV.2	Constitution du robot	55
IV.2	Etude des différents blocs	56
V.	LA COMMANDE DU ROBOT	64
V.1	Comportement d'évitement d'obstacles.....	65
V.2	Comportement d'évitement d'obstacles.....	66
	Conclusion.....	67
	CHAPITRE III : RESULTATS ET INTERPRETATIONS.....	68
	CHAPITRE III : RESULTATS ET INTERPRETATION.....	68
I.	PLATEFORME DE PROGRAMMATION ARDUINO.....	69
II.	SCHEMA DU MONTAGE ELECTRONIQUE.....	70
III.	SIMULATION DE LA COMMANDE SOUS MATLAB.....	71
IV.	PROTOTYPAGE	73
	Conclusion.....	73
I.	Sous-programme détection d'obstacle :.....	78
	TABLE DES MATIERES	81