الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

جامعة الدكتور يحي فارس بالمدية

Université Yahia Farès de Médéa

FACULTÉ DE LA TECHNOLOGIE

DÉPARTEMENT DE GÉNIE MÉCANIQUE

Projet de fin d'études présenté pour l'obtention du déplôme de :

MASTER EN GÉNIE MÉCANIQUE

Spécialité : MÉCATRONIQUE

Théme :

Etude, simulation et réalisation de la commande d'un bras manipulateur

Sous la direction de :

M. TCHOKETCH KEBIR Sofiane.

M. MAAZOUZ Mohamed.

Préparé par :

LADEK MOUSSA.

ALILI AHMED.

ANNÉE UNIVERSITAIRE 2016-2017

Dédicaces

Louanges à ALLAH
le tout puissant de m'avoir donnée la
force et le courage pour accomplir ce modeste travail.
J'ai l'honneur de dédier ce modeste travail à : A mes très chers parents, ma chere épouse et notre petit chérubin Amir Abd Errazak qui m'ont soutenu et encouragé durant toute la période de mes études et à qui je souhaite une longue et heureuse vie, que dieu les protège et me donne la force pour que je puisse leur rendre un petit peu de leur bienfait et leur bonté. A mes très chers frères et mes très chères sœurs. A toute les personnes de ma famille. La famille LADEK et La famille HAMDOUCHE
Je remercie mon ami et binôme Alili Ahmed.Ainsi que tous mes autres amis, HAMADI Ahmed, LAGHOUATI Youcef, FEKIRINI Ahemd et AKAK Smain et de prés ou de loin ceux et celles que je n'ait pas cité. A toute ma promotion de Mécatronique Et enfin,
à tous ceux qui m'aiment
et mes futures
épreuves
♡

Moussa

Dédicaces

Louanges à ALLAH le tout puissant de m'avoir donnée la force et le courage pour accomplir ce modeste travail. J'ai l'honneur de dédier ce modeste travail à : A mes très chers parents, ma chere épouse et nos trois petites adourables filles ROMAISSA, RITEDJ et RIMES NOURE ELHOUDA qui m'ont soutenu et encouragé durant toute la période de mes études et à qui je souhaite une longue et heureuse vie, que dieu les protège et me donne la force pour que je puisse leur rendre un petit peu de leur bienfait et leur bonté. A mes très chers frères et mes très chères soeurs. A toute les personnes de ma famille. La famille ALILI et La famille MOUKAH Je remercie mon ami et binôme LADEK Moussa.Ainsi que tous mes autres amis, HAKIM KEROUANI et de prés ou de loin ceux et celles que je n'ait pas cité. A toute ma promotion de Mécatronique Et enfin à la brise lointaine de mon cœur

♡

Ahmed

Remerciements

Nous remercions notre Dieu, le tout puissant, de nous avoir donné la santé et la volonté pour compléter ce modeste travail.

Ainsi que l'ensemble des membres du jury qui ont examiné cette thèse pour leurs observations enrichissantes et leurs conseils. Et plus spécialement Mr. TCHOKETCH KEBIR Sofiane et M. MAAZOUZ Mohamed qui ont acceptés d'être les promoteurs de notre thèse.

Nous remercions vivement Mr. BOUMAHDI, Mr. BACHEN, Mr. ABOUDERA, Mr. YAHIA.

Nous tenons à exprimer notre profonde gratitude au Mr. DILMI Ahmed et Mr. GOTIT Mahmoud qui nous ont permis d'apporter certaines améliorations au travail qu'ils ont produits précédemment.

Résumé

Table des matières

Introduction général 1

1 Des Généralités sur les Robots. 3

1.1 Introduction 4

1.2 Définition générale 4

1.2.1 Le robot industriel 4

1.2.2 La robotique 4

1.3 L'histoire de la robotique industrielle 5

1.4 Les Types des Robots 6

1.4.1 Robots mobiles 6

1.4.2 Robots manipulateurs 7

1.5 Les éléments constitutifs d'un robot 8

1.5.1 Unité opérationnelle 9

1.5.2 Unité informationnelle 9

1.6 Structure mécanique articulée 9

1.6.1 Structure mécanique articulée á chaîne cinématique simple 10

1.6.2 Structure mécanique articulée á chaîne cinématique fermée 10

1.7 Architecture des robots 11

1.7.1 La base 11

1.7.2 Le porteur 12

1.8 Classification des robots 14

1.8.1 Classification fonctionnelle 14

1.8.2 Classification géométrique 17

1.9 Les Domaines d'applications 20

1.9.1 Les Tâches simples . . . 20
1.9.2 Les Tâches complexes . . . 20
1.10 Conclusion . . . 21

2 La Modélisation des robots manipulateurs 22

2.1 Introduction . . . 23
2.2 Structure mécanique . . . 23
2.3 Les différentes méthodes de modélisation . . . 24
2.3.1 Modèle géométrique direct . . . 24
2.3.2 Modèle géométrique inverse . . . 26
2.3.3 Le model cinématique direct . . . 28
2.3.4 Le model cinématique inverse . . . 30
2.3.5 La Modélisation dynamique . . . 30
2.4 Conclusion . . . 31

3 Arduino et Servomoteur 32

3.1 Introduction . . . 33
3.2 Arduino . . . 33
3.2.1 Définition . . . 33
3.2.2 Microcontrôleur . . . 33
3.2.3 Les différentes cartes Arduino . . . 34
3.2.4 La carte Arduino Uno . . . 35
3.2.5 Les modes de communication . . . 37
3.2.6 L'alimentation électrique . . . 38
3.2.7 Valeurs de courant ou de tension . . . 40
3.2.8 La Programmation de l'Arduino . . . 40
3.3 Les Servomoteurs . . . 42
3.3.1 Définition . . . 42
3.3.2 Le Principe de l'asservissement d'un Servomoteur . . . 43
3.3.3 Le rôle des servomoteurs . . . 45
3.3.4 Fonctionnement d'un servomoteur standard . . . 45

3.3.5 Les composants d'un servomoteur 46
3.3.6 La théorie de la commande des servomoteurs 46
3.3.7 l'implimentation de la commande théorique 47
3.3.8 Les servomoteurs digital et analogique 47
3.3.9 Le pilotage des servomoteurs 48
3.4 Conclusion 49

4 La Réalisation de la Commande d'un Bras Manipulateur, Résultats et Discussions 51

4.1 Introduction 52
4.2 Simulation, Repérage et paramétrage du bras manipulateur plan de type 3R dans le plans 52
4.2.1 Le Repérage 52
4.2.2 Le Paramétrage 54
4.2.3 L'etablissement des relations géométriques 55
4.2.4 L'extraction du modèle géométrique direct 55
4.3 Le matériale utilisée (Partie hardware) 56
4.3.1 Bref définition hardware 56
4.3.2 Bras manipulateur 56
4.3.3 Arduino Uno 58
4.3.4 ArduinoWi-FiShield 59
4.3.5 Alimentation stabilisée (à partir d'une alimentation de PC) 60
4.3.6 Le Moteur à courant continu (M.C.C) 61
4.3.7 Le Routeur Wi-Fi 62
4.4 Partie software 63
4.4.1 MATLAB 63
4.4.2 Le langage de programmation ARDUINO 64
4.5 La Réalisation De La Commande 65
4.5.1 Méthodologie de l'approche proposée 65

4.5.2 Réalisation, Tests, Résultats et discussion 66
4.6 Conclusion 68
Conclusion général 70

A Fichier technique **72**

B Programme arduino de commandé de notre robot **73**

Bibliographie 78

Table des figures

1.1 Robotique industrielle 5

1.2 Spirit 7

1.3 Robot fanuc 8

1.4 Structure fonctionnelle d'un robot 8

1.5 La Structure sérielle 10

1.6 La Structure fermée 11

1.7 Architecture d'un robot 11

1.8 Représentation d'une articulation rotoîde 12

1.9 Représentation d'une articulation prismatique 13

1.10 Manipulateur á commande manuelle 15

1.11 Manipulateur à cycle préréglé 15

1.12 Robot programmable 16

1.13 Robot intelligent 17

1.14 Robot cartésien 17

1.15 Robot cylindrique 18

1.16 Robot sphérique 18

1.17 Robot SCARA 19

1.18 Robot 3R 20

2.1 Systèmes de coordonnées et les paramètres de Denavit - Hardenberg 25

2.2 Transformations entre l'organe terminal et le repère atelier 28

2.3 Exemple d'un robot plan à trois degrés de liberté 29

3.1 Microcontrôleur ATmega328 34

3.2 La carte à microcontrôleur 35
3.3 les composants de la carte Arduino 36
3.4 Entré et Sortie numérique 38
3.5 Alimentation et entrée Analogiques 38
3.6 Alimentation de la carte Arduino 39
3.7 les différentes prises 40
3.8 Coloration Syntaxique dans IDE de l'Arduino 41
3.9 les différents types de servomoteur 42
3.10 Les Trois Files 43
3.11 Le Principe de l'asservissement d'un Servomoteur 43
3.12 Les composants d'un servomoteur 46
3.13 Chronogramme de commande de Servomoteur 47
3.14 Présentation d'un servomoteur avec Arduino 48
3.15 Exemple de piloter des servomoteurs avec l'Arduino 49

4.1 Chaîne cinématique d'un bras manipulateur série 52
4.2 Paramètres de Denavit-Hartenberg modifiés 53
4.3 Repérage de notre mécanisme 54
4.4 Bras manipulateur 57
4.5 carte ARDUINO 58
4.6 ArduinoWi-FiShield 59
4.7 Fonctions des connecteurs ATX 24 pins 60
4.8 Le bloc diagramme d'une alimentation 60
4.9 Circuit commande M.C.C à base de transistor 62
4.10 point d'accées sans fil 63
4.11 interface Matlab2016b 64
4.12 langage de programmation ARDUINO 65
4.13 Synoptique du système conçu 65
4.14 Représentation final de notre projet 66

4.15 Espace de travail 67

4.16 Interface de la génération de la commande de notre Robot 67

4.17 Espace de travail 68

A.1 liaisons mécaniques 72

Liste des tableaux

3.1 Valeurs de courant ou de tension 40
3.2 Les connaissances nécessaires au pilotage d'un servomoteur 49

4.1 Paramétrage de notre mécanisme 54

Liste des abréviations

Abréviations	**Définitions**
(ddl)	Degrés de liberté
(ddm)	Degrés de mobilité
(DHM)	Denavit-Hartenberg Modifiée
(MCD)	Modèle Cinématique Direct
(MCI)	Modèle Cinématique Inverse
(MDD)	Modèle Dynamique Direct
(MDI)	Modèle Dynamique Inverse
(MGD)	Modèle Géométrique Direct
(MGI)	Modèle Géométrique Inverse
(MCC)	Moteur à Courant Continu
(MLI)	Modulation de largeur d'impulsion
(PWM)	Pulse Width Modulation
(IEEE)	Institute of Electrical and Electronics Engineers
(WI-FI)	Wireless Internet for Frequent Interface

Introduction général

Il est incontestable de dire que les robots sont devenus des outils impératifs dans le paysage de fabrication. La robotique a connu d'énormes pas dans le sens de l'évolution pour être à la fois plus précise et pratique.

La robotique se substitue à l'homme ou prolonge son action en apportant précision, rapidité ou capacité à appliquer d'importants efforts. Elle est généralement constituée d'un bras manipulateur plus ou moins anthropomorphe mis en mouvement par des actionneurs électriques, hydrauliques ou pneumatiques et dont la position est repérée par une série de capteurs.

Cependant, le problème des performances est l'une des raisons qui empêchent la robotisation de certains procédés industriels. Par ailleurs, en production industrielle, il est très difficile de changer les éléments mécaniques ou le système de commande du robot.

Notre travail est la continuité d'un travail qui a été élaborée durant l'année 2015-2016 par un groupe d'étudiants appartenant à l'université de Médéa. Spécialité : MÉCATRONIQUE.

Le travail consiste à donner plus de performance au projet d'une commande d'un bras manipulateur en matière de précision.

Notre projet est une amélioration sur les volets réalisation :

— Nous avons introduit une nouvelle carte électronique qui assemble les parties commande et puissances des servomoteurs et un moteur à courant continu.

— Introduire un nouvel organe terminal, à savoir une perceuse fixée sur le bras manipulateur.

— Un autre une plate-forme qui sert comme support de fixation et un espace de travail.

Sur le volet programmation :

— Sur ce volet, nous avons introduit un nouvel algorithme qui a pour tâche d'identification

l'espace de travail du robot. Cet à l'aide de MATLAB ; un interface graphique a été réalisé pour faciliter de la commande la tache de la commande aux opérateurs.

Ce mémoire est organisé de la manière suivante :

— Dans le premier chapitre, en commençant par une généralité sur les robots, (les différents types des robots existants dans l'industrie).

— Le deuxième chapitre présentera avec plus d'approfondissement les méthodes les plus importantes pour la modélisation des robots, et nous enchainons ici à l'étude de notre réalisation par la détermination des équations qui gèrent les mouvements de notre robot.

— Le troisième chapitre traite la présentation de la carte ARDUINO avec les servomoteurs.

— Enfin le contenu du quatrième chapitre présente les performances et tests obtenus avec les applications qu'on a réalisées.

— Nous terminons ce mémoire par une conclusion et quelques perspectives.

Chapitre 1

Des Généralités sur les Robots.

1.1 Introduction

L'utilisation des systèmes robotiques apparait aujourd'hui dans plusieurs domaines d'activités la médecine, la défense, la recherche scientifique etc. Les robots sont utilisés de manière privilégiée pour des missions où les objectifs sont quantifiables et clairement définis. Ils sont destinés à faciliter les tâches pour l'homme et à amplifier le rendement .

Dans ce premier chapitre, on va donner un aperçu non exhaustif sur les robots, un bref historique sur l'évolution de la robotique industrielle et présenter les différents types de robots et les éléments constitutifs de ces derniers.

1.2 Définition générale

1.2.1 Le robot industriel

Le terme robot a été introduit pour la premiére fois par l'auteur Tchéque Capek en 1920 dans sa pièce de théâtre R.U.R. (Rossum's Universal Robot) et il est dérivé du mot robota qui signifie travailleur (de force)[1].

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) accomplissant automatiquement des tâches diverses. C'est une machine intelligente fonctionnelle qui nécessite une autonomie de mouvements.

L'Organisation Internationale de Normalisation définit le robot comme étant un manipulateur á plusieurs degrés de liberté, á commande automatique, reprogrammable, multi-applications. mobile ou non, destiné à être utilisé dans les applications d'automatisation industrielle [2].

1.2.2 La robotique

La robotique est une science qui s'intéresse aux robots. En fait, il s'agit d'un ensemble de disciplines techniques (mécanique, électronique, automatique, informatique) articulées autour d'un objectif et d'un objet communs. Cet objectif est l'automatisation flexible de nombreux secteurs de l'activité humaine réputés jusqu'á très récemment comme ne pouvant se passer de la présence de l'homme, et l'objet est le robot, sorte de machine universelle dont l'homme rêve

depuis toujours pour le remplacer dans les tâches difficiles.

1.3 L'histoire de la robotique industrielle

La robotique industrielle a connu un essor entre 1950-1970. Elle a vu le jour en 1954 lorsque Georges DEVOL a pu réaliser son brevet sur la robotique. Dans ce brevet Devol a conçu un robot qu'il a intitulé Unimate.

En 1961, le premier Unimate fut utilisé dans les usines de GENERAL MOTORS.

En 1966 , l'entreprise Unimation continue de développer des robots et élaborent notamment des robots permettant de faire d'autres tâches, comme des robots de manipulation matérielle ou encore des robots conçus pour la soudure ou pour d'autres applications de ce genre.

En 1978 un nouveau robot est conçu par Unimation Inc avec l'aide de General Motors. Ensemble ils conçurent le robot PUMA 500. Le robot PUMA (Programmable Universal Machine for Assembly) a été conçu par Vic Schienman et fut financé par General Motors et par The Massachussets Institute of Technology au milieu des années 70. Le système de ce robot est composé d'un bras manipulateur permettant d'assembler des composants industriels et de son ordinateur de commande. Ce robot est le robot d'assemblage le plus rependu dans l'industrie des années 70.



FIGURE 1.1 – **Robotique industrielle,(unimate, puma : années 1950-1970)**

En 1985, Reymond Clavel a imaginé le Robot Delta qui possède un bras de manipulation

formé de 3 parallélogrammes. Son brevet tombe dans le domaine public en 2007 et différents constructeurs devraient alors sortir leur propre robot delta.

Le Jet Propulsion Laboratory (JPL) développe un robot industriel hexapode (á 6 pattes) du nom de Lemur. Lemur aura pour mission de monter, assembler et réparer des installations spatiales. Pesant moins de 5 kg, il offre la possibilité innovante d'adapter différents outils sur chacun de ses membres.

Selon l'étude robotique de la Fédération Internationale de Robotique (IFR) en 2012, il y a au moins 1 153 000 robots industriels opérationnels fin 2011 dans le monde.

Avec l'apparition de la robotique industrielle, les robots étaient conçus pour remplacer les ouvriers dans les tâches pénibles, répétitives ou dangereuses (peinture, soudure etc...). Aujourd'hui avec le développement de l'électronique, de l'informatique, de la mécanique et aussi de l'automatique, la technologie robotique a progressé. La recherche dans le domaine de la robotique est dirigée vers le développement de robots dévoués â des tâches bien différentes que celles demandées par l'industrie. Par exemple des robots travaillant en mode automatique ou semi-automatique et qui ont souvent pour objectif d'interagir avec des humains et de les aider dans leurs tâches (surveillance, manutention d'objets lourds etc...). Ils sont dotés d'une intelligence qui leur donne une certaine autonomie.

Ainsi donc, le développement important de l'intelligence artificielle et de la robotique font que de nouveaux robots apparaissent constamment et l'utilisation de systèmes robotiques apparait aujourd'hui dans plusieurs domaines d'activité : la médecine, la défense, la recherche etc....

1.4 Les Types des Robots

Il existe deux types de robots : robots mobiles et robots manipulateurs.

1.4.1 Robots mobiles

Ce sont des robots capables de se déplacer dans un environnement comme le montre la figure I.2. Ils sont équipés ou non de manipulateurs suivant leur utilisation, (les robots explorateurs,

les robots de services etc...).



FIGURE 1.2 – **Spirit,(NASA,2003 sur Mars)**

1.4.2 Robots manipulateurs

Des robots ancrés physiquement à leur place de travail et généralement mis en place pour réaliser une tâche précise répétitive, (tels que les robots industriels, médicaux etc...).

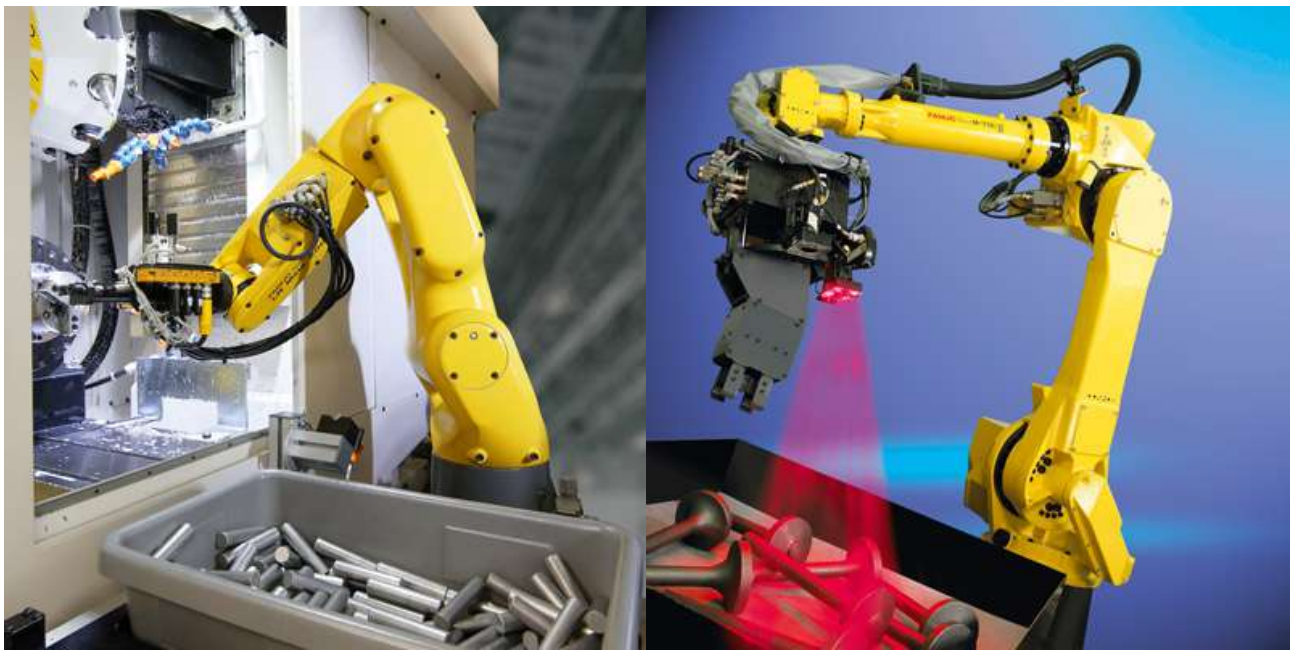


FIGURE 1.3 – **Robot fanuc**

1.5 Les éléments constitutifs d'un robot

Dans un robot en fonctionnement, on peut distinguer plusieurs ensembles interactifs (figure 1.4).

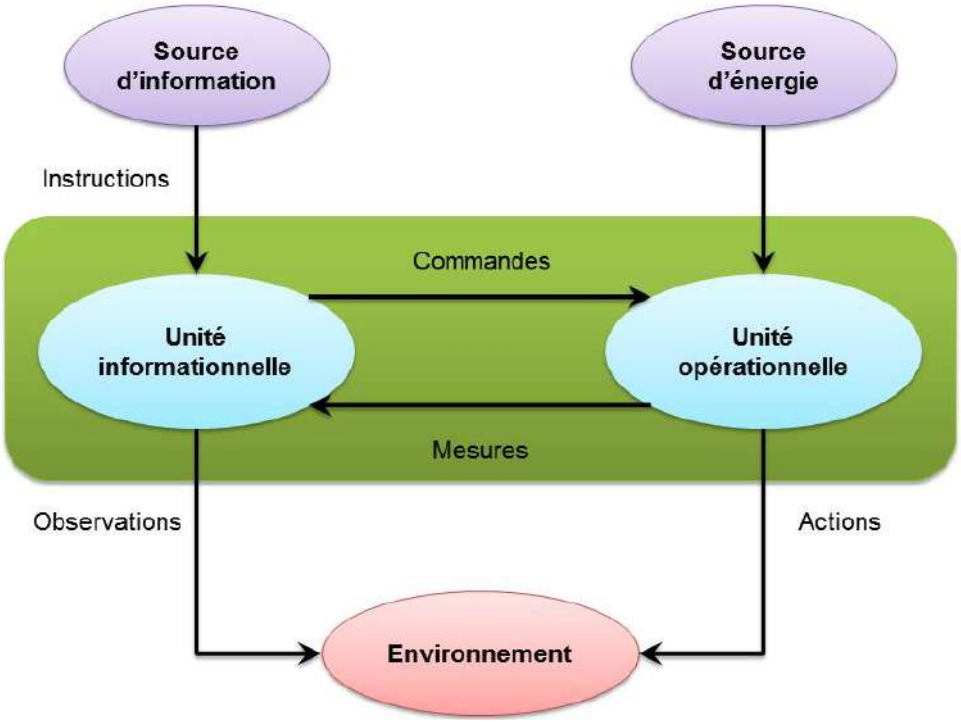


FIGURE 1.4 – **Structure fonctionnelle d'un robot**

1.5.1 Unité opérationnelle

Exerce les actions commandées en empruntant la puissance nécessaire à la source d'énergie. Cette partie, qui constitue le robot physique, intègre la structure mécanique (segments, articulations, architecture, etc...), les modules d'énergie (amplificateurs, variateurs, servovalves etc...), les convertisseurs d'énergie (moteurs, vérins etc...), les chaînes cinématiques de transmission mécanique (réducteurs, vis á billes, courroies crantées etc...), les capteurs de proprioceptifs placés sur chaque axe pour mesurer en permanence leur position et leur vitesse, et enfin l'effecteur, ou organe terminal, qui est en interaction avec l'environnement [3].

1.5.2 Unité informationnelle

Reçoit les instructions décrivant la tâche à accomplir, les mesures relatives à l'état interne de la structure mécanique qui constitue le bras manipulateur et les observations concernant son environnement. Elle élabore en conséquence les commandes de ses différentes articulations en vue de l'exécution de ses tâches. Les systèmes actuels fonctionnent en interaction permanente selon le cycle information-décision-action [4].

1.6 Structure mécanique articulée

Un robot manipulateur est constitué généralement par deux sous-ensembles distincts : un organe terminal qui est le dispositif destiné à manipuler des objets, et une structure mécanique articulée (SMA), constituée d'un ensemble de solides reliés entre eux, généralement les uns á la suite des autres où chaque solide est mobile par rapport au précédent. Cette mobilité s'exprime en termes de degrés de liberté (d.d.l) qui est par définition le nombre de mouvements indépendants possibles d'un solide par rapport au solide qui lui est directement relié.

Une structure mécanique articulée peut être représentée par une architecture composée de plusieurs chaînes de corps rigides assemblés par des liaisons appelées articulations. Les chaînes peuvent être dites soit ouvertes ou en série dans les quelles tous les corps ont au plus deux liaisons, ou bien arborescentes oú au moins l'un des corps a plus de deux liaisons. Les chaînes

peuvent aussi être fermées dans les quelles l'organe terminal est relié à la base du mécanisme par l'intermédiaire de plusieurs chaînes [3].

1.6.1 Structure mécanique articulée á chaîne cinématique simple

C'est une chaîne cinématique dont chaque membre possède un degré de connexion (nombre de liaisons mécaniques) inférieur ou égal á deux. Un robot sériel est formé d'une chaîne cinématique simple dont la base et l'organe effecteur possèdent un degré de connexion de un (c'est-à-dire qu'il n'est relié qu'à un seul corps) et les autres éléments un degré de connexion de deux.

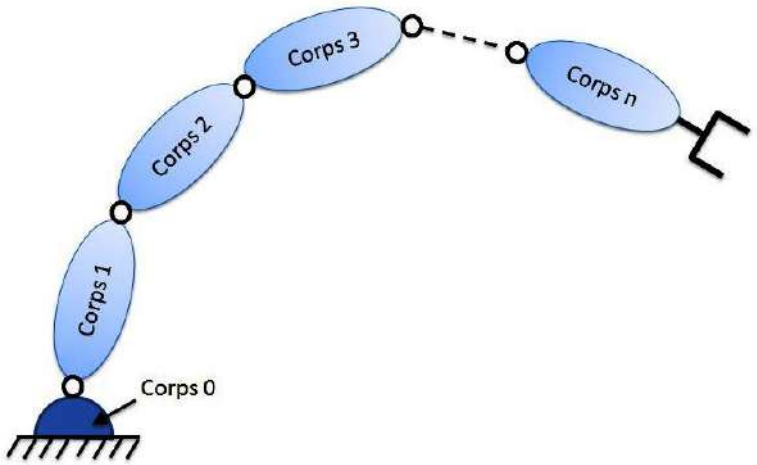


FIGURE 1.5 – **La Structure sérielle**

1.6.2 Structure mécanique articulée á chaîne cinématique fermée

C'est une chaîne cinématique qu'il existe un retour mécanique d'un ou plusieurs segments à un autre dans la chaîne.

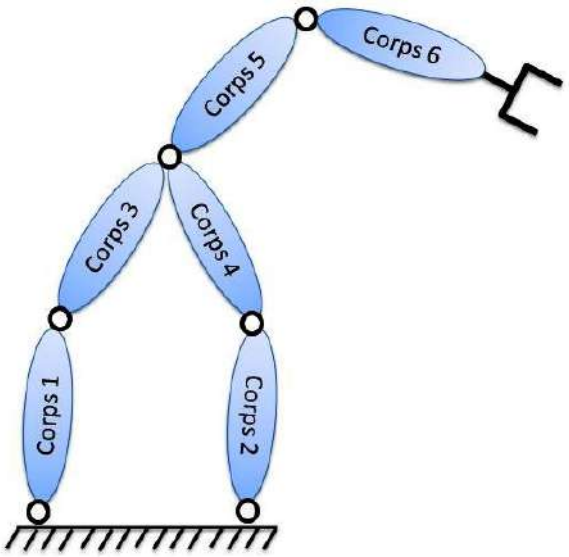


FIGURE 1.6 – **La Structure fermée**

1.7 Architecture des robots

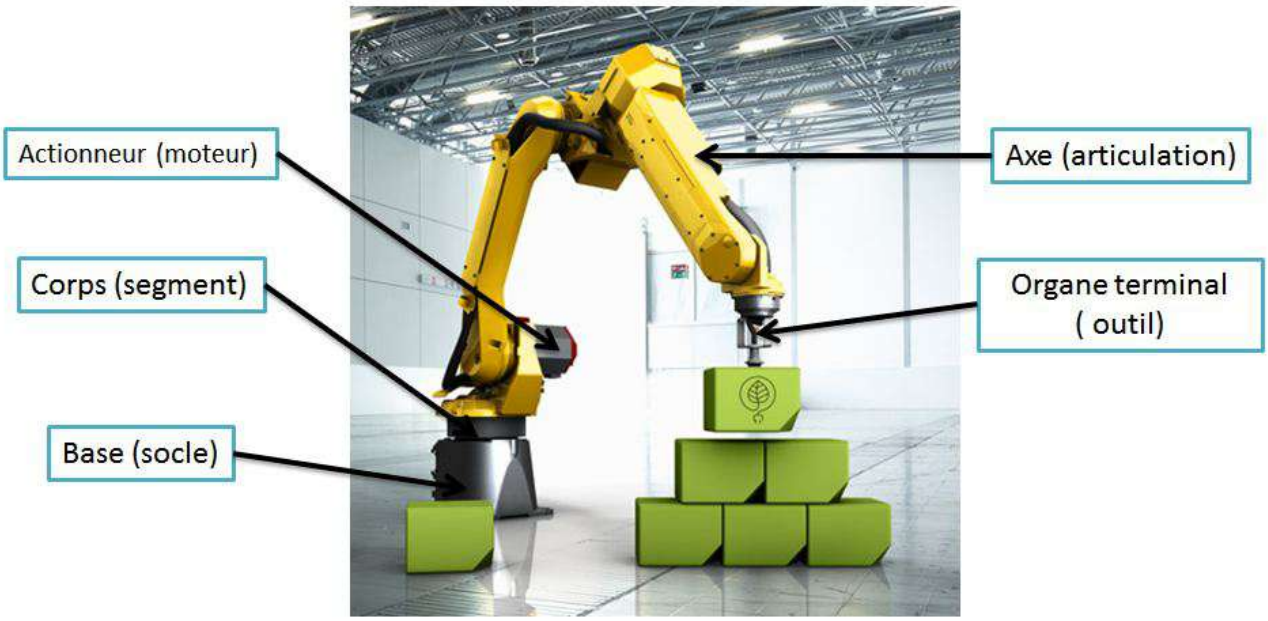


FIGURE 1.7 – **Architecture d'un robot,(Robot fanuc)**

1.7.1 La base

La base du manipulateur est fixée sur le lieu du travail. Ceci est le cas de la quasitotalité des robots industriels.

1.7.2 Le porteur

Le porteur représente l'essentiel du système mécanique articulé (segment, articulation, actionneur, l'organe terminal), il a pour rôle d'amener l'organe terminal dans une situation donnée imposée par la tâche. Il est constitué de :

a. Segment :

Corps solides rigides susceptibles d'être en mouvement par rapport à la base du porteur, et les uns par rapport aux autres,

b. Articulation

Une articulation lie deux corps successifs en limitant le nombre de degré de liberté, de l'un par rapport à l'autre :

- Articulation rotoîde :

Il s'agit d'une articulation de type pivot, notée R, réduisant le mouvement entre deux corps à une rotation autour d'un axe commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe (Voir la figure ci-dessus).

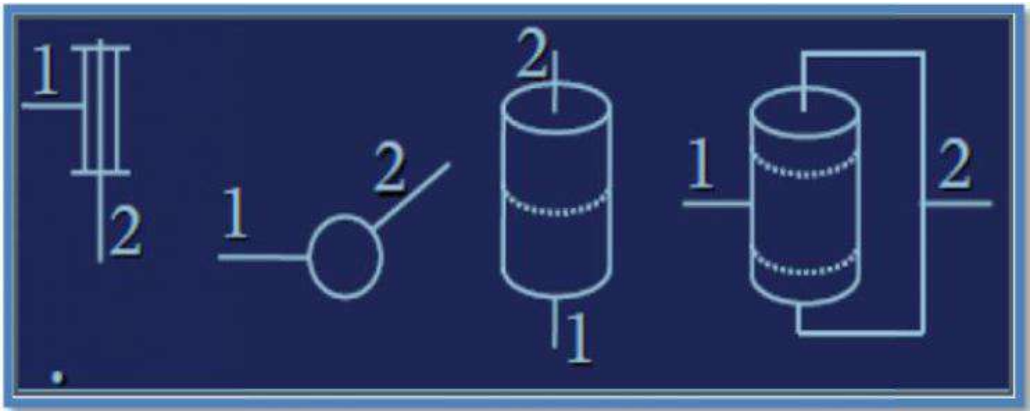


FIGURE 1.8 – **Représentation d'une articulation rotoîde**

- Articulation prismatique :

Il s'agit d'une articulation de type glissière, notée P, réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe (Voir la figure ci-dessus).

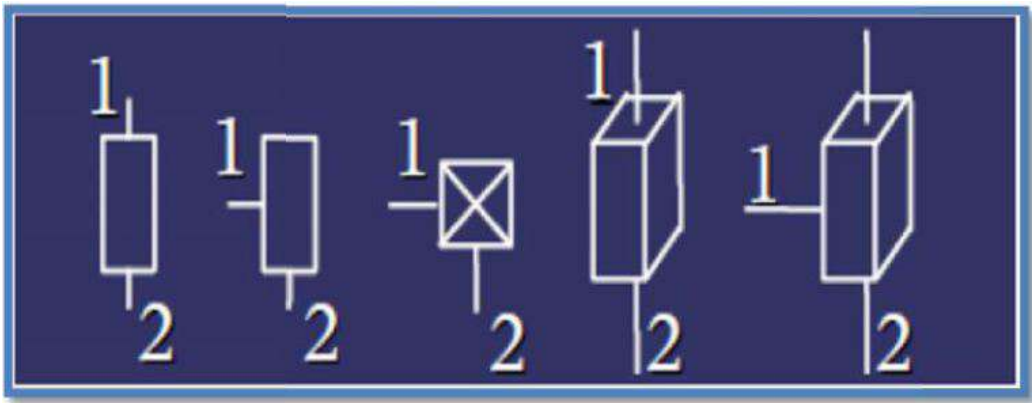


FIGURE 1.9 – **Représentation d'une articulation prismatique**

c. L'actionneur

Pour être animé, la structure mécanique articulée comporte des moteurs le plus souvent associés à des transmissions (courroies crantées),l'ensemble constitue les actionneurs. Les actionneurs utilisent fréquemment des moteurs électriques à aimant permanent, à courant continu, à commande par l'induit. On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas.

Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique). (Les actionneurs pneumatiques sont d'un usage général pour les manipulateurs á cycles (robots tout ou rien). Un manipulateur à cycles est une structure mécanique articulée avec un nombre limité de degrés de liberté permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air) [3].

d. L'organe terminal

On regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, ...), ou á les transformer (outils, torche de soudage, pistolet de peinture, ...). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, au

sens oú il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. Un robot, enfin, peut-être multi-bras, chacun des bras portant un organe terminal différent. On utilisera indifféremment le terme organe terminal, préhenseur, outil ou effecteur pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique, exemple : pistolet pour la soudure dans les robots industriels [3].

1.8 Classification des robots

On peut classer les robots d'un point de vue fonctionnel ou d'après leur structure géométrique.

1.8.1 Classification fonctionnelle

Le nombre de classe et les distinctions entre celles-ci varient de pays à pays (6 classes au Japon, 4 en France). L'A.F.R.I. distingue 4 classes illustrées ci-dessous :

1.8.1.1 Manipulateur à commande manuelle

La Figure I.10 représente les manipulateurs à commande manuelle :



FIGURE 1.10 – **Manipulateur á commande manuelle**

1.8.1.2. Manipulateur automatique

La figure montre un bras manipulateur qui exerce des mouvements de soudure sans l'intervention de l'homme.

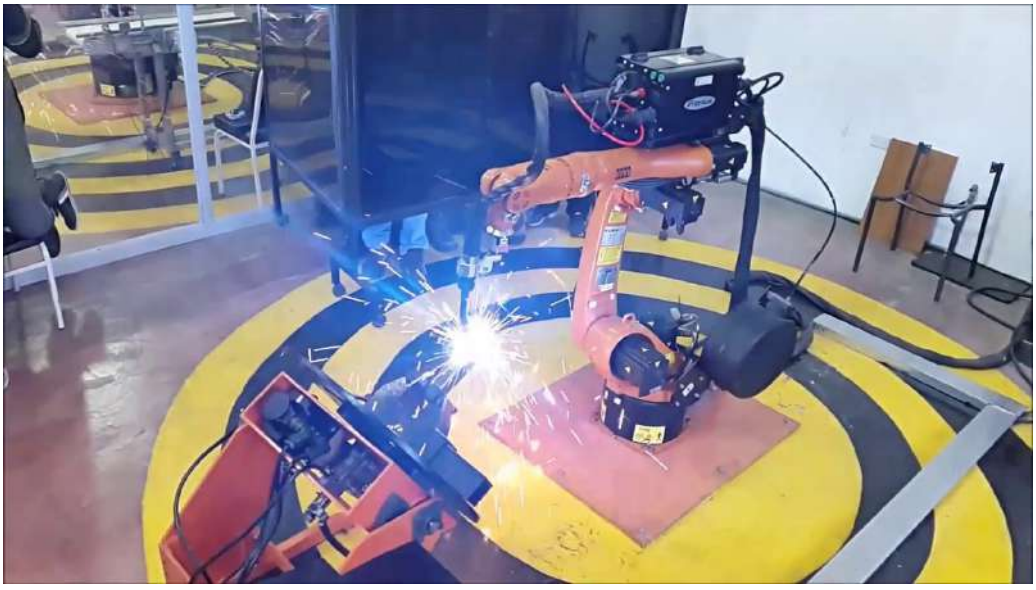


FIGURE 1.11 – **Manipulateur à cycle préréglé**

I.8.1.3. Robots programmables

Ils répètent les mouvements qu'on leur a appris ou programmés sans informations sur l'environnement ou la tâche effectuée. On peut aussi faire la distinction entre robots (playback) qui reproduisent la tâche apprise et robots à commande numérique qui peuvent être programmés hors-ligne.

Pour certains robots, par exemple les robots de peinture, qui doivent suivre une trajectoire complexe qu'il est difficile d'exprimer mathématiquement, un opérateur humain spécialiste de la tâche effectue la trajectoire en guidant le bras du robot à l'aide d'un (pantin), et l'entièreté de la trajectoire est mémorisée par le robot et la figure I.12 représente les robots programmables [3].

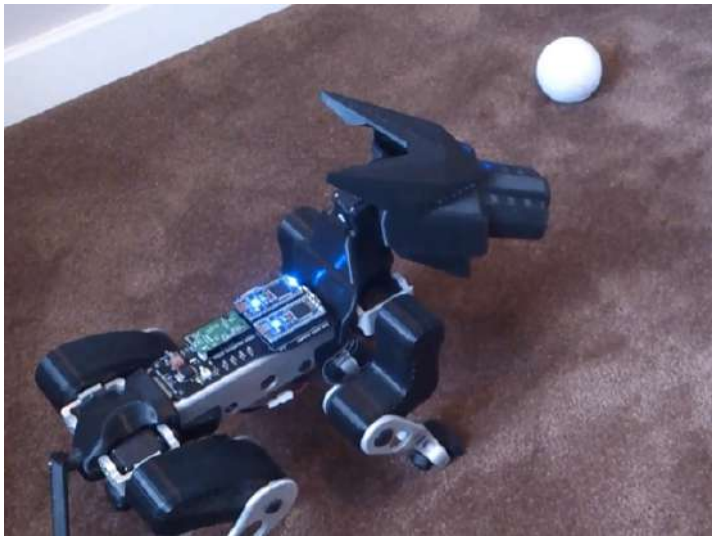


FIGURE 1.12 – **Robot programmable**

I.8.1.4. Robots intelligents

On trouve actuellement des robots de seconde génération qui sont capables d'acquérir et d'utiliser certaines informations sur leur environnement (systèmes de vision, détecteurs de proximité, capteurs d'efforts,...) comme le montre la Figure I.13. Les robots de troisième génération sont capables de comprendre un langage oral proche du langage naturel et de se débrouiller de façon autonome dans un environnement complexe grâce á l'utilisation de l'intelligence artificielle [3].



FIGURE 1.13 – **Robot intelligent**

1.8.2 Classification géométrique

On peut aussi classer les robots suivant leur configuration géométrique, autrement dit l'architecture de leur porteur.

- La structure cartésienne (PPP)

C'est une structure à trois liaisons prismatiques et est la plus ancienne. Historiquement, elle découle logiquement de la conception traditionnelle d'une machine-outil à trois axes, type rectifieuse ou fraiseuse par exemple. Cette structure est relativement peu utilisée sauf dans quelques applications particulières telles que robots pratiques, robots de magasinage [3].

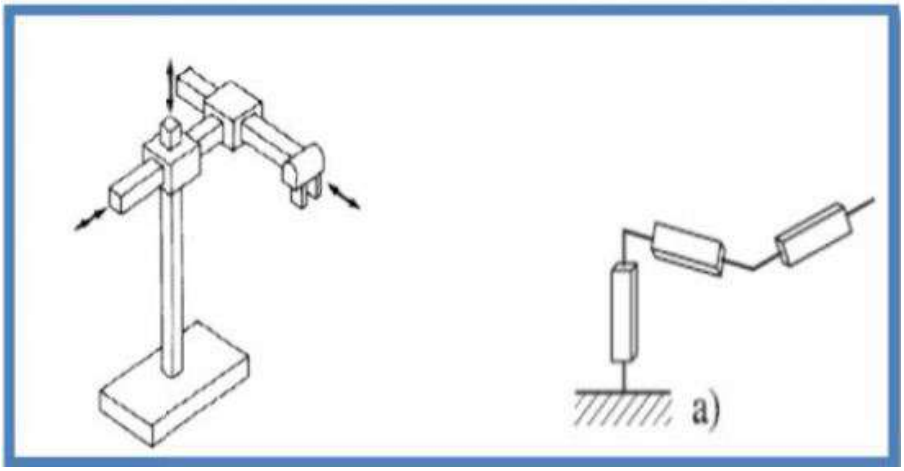


FIGURE 1.14 – **Robot cartésien**

- La structure cylindrique (RPP) ou (PRP)

Cette structure associe une rotation et deux translations. Elle présente l'inconvénient d'offrir un volume de travail faible devant un encombrement total important. Elle n'est pratiquement plus utilisée [3].

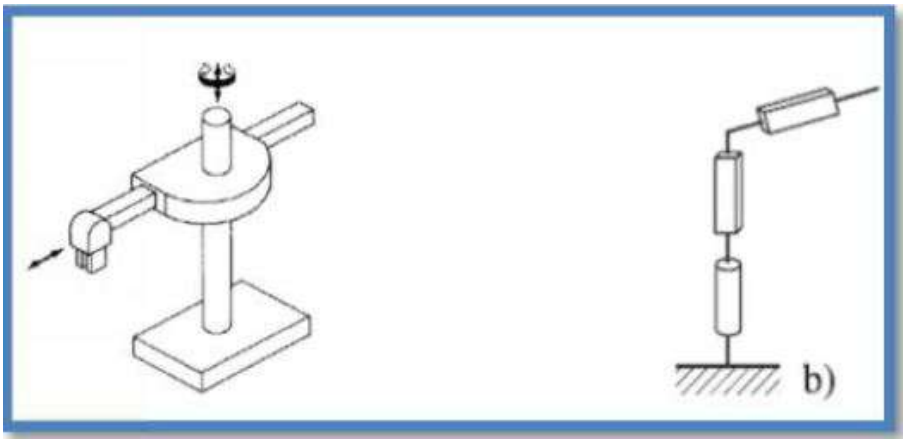


FIGURE 1.15 – **Robot cylindrique**

- La structure sphérique ou polaire

C'est une structure quasiment abandonnée pour des raisons similaires à l'abandon de la structure cylindrique.

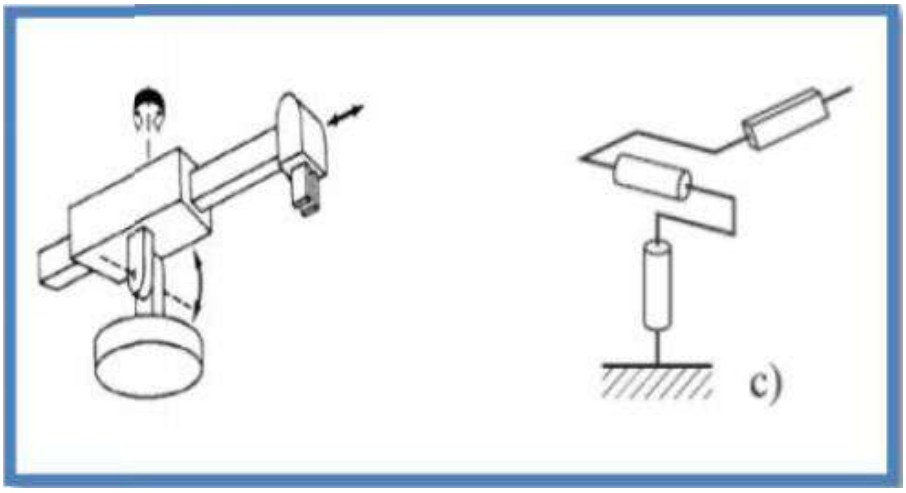


FIGURE 1.16 – **Robot sphérique**

- La structure dite SCARA (RRP)

A axes de rotation parallèles, elle est l'une des plus utilisées en particulier pour des tâches de manutention ou d'assemblages très fréquents dans l'industrie. Ce succès commercial est lié au fait que le ratio entre le volume de travail et l'encombrement est très favorable et aussi au fait que la structure SCARA est très adaptée à ce type de tâches [3].

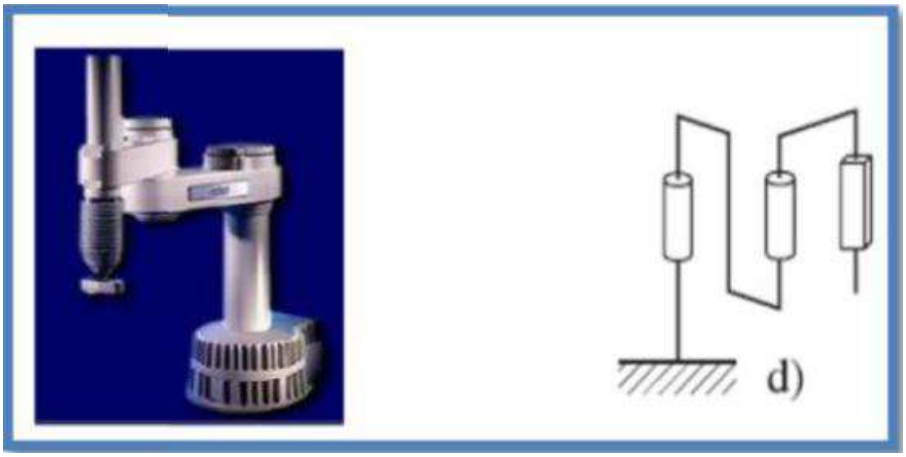


FIGURE 1.17 – **Robot SCARA**

- La structure 3R (anthropomorphe)

Elle permet d'amener un solide en un point de l'espace par trois rotations, généralement une à axe vertical et deux à axes horizontaux et parallèles. C'est le porteur (généraliste) par excellence pouvant se programmer facilement pour différents types de tâches et disposant d'un volume de travail conséquent [3].

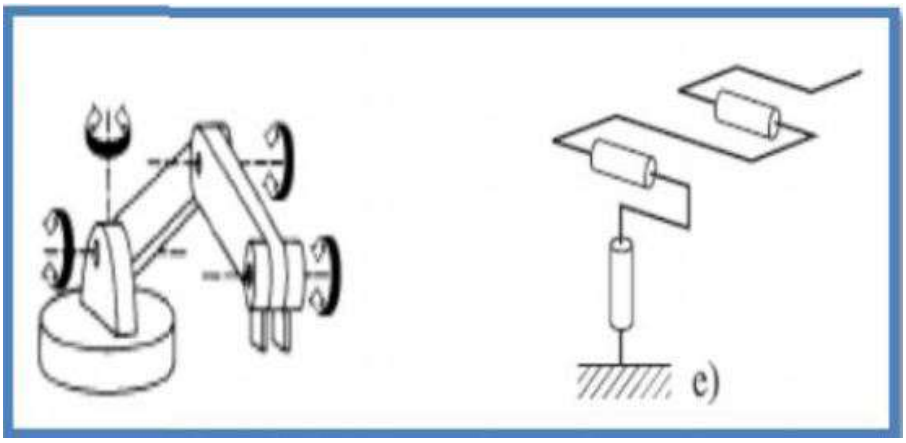


FIGURE 1.18 – **Robot 3R**

1.9 Les Domaines d'applications

Les robots industriels peuvent être utilisés dans plusieurs domaines et peuvent accomplir différentes tâches.

1.9.1 Les Tâches simples

La grande majorité des robots est utilisée pour des tâches simples et répétitives. Ils sont programmés une fois pour toute au cours de la procédure d'apprentissage. Les critères de choix de la solution robotique sont :

- La tâche est assez simple pour être robotisée.
- Les critères de qualité sur la tâche sont importants.
- Pénibilités de la tâche (peinture, charge lourde, environnement hostile, ...)

Exemples de robots utilisés pour des tâches simples : robots soudeurs par points et robots soudeurs à l'arc.

1.9.2 Les Tâches complexes

Exemples robotique de service :

- Robot pompiste.
- Robot de construction.

- Robot Computer motion.

- Robot Assistance aux personnes Handicapées.

1.10 Conclusion

Dans ce chapitre, nous avons donner une idée générale sur la robotique, l'historique des robots, leurs structures, leurs utilisations et les différents types de robots ainsi que leurs classifications et leurs domaines d'applications.

Chapitre 2

La Modélisation des robots manipulateurs

2.1 Introduction

La modélisation des robots, comme tout procédé industriel que l'on peut commander est une étape que l'on peut qualifier d'intermédiaire, puisqu'il s'agit de définir les équations qui décrivent le comportement dynamique d'un robot avec assez de précision pour que leur utilisation soit suffisamment simple au moment de la conception. La modélisation est la base de la commande, à partir de la bonne connaissance du comportement du système étudié, il est plus facile de choisir les lois et stratégies des commandes adaptées pour aboutir aux performances souhaitées [5].

2.2 Structure mécanique

La structure mécanique du robot manipulateur peut être divisée en deux parties [6] :

— Organe terminal : les tâches qui sont dévolues aux robots sont très variées. Pour chaque opération ou travail spécifique, l'organe terminal prend un aspect particulier.

— Elément porteur : il est composé d'un ensemble de corps souples ou rigides liés par des articulations, servant à déplacer l'organe terminal d'une configuration à une autre.

Avant de décrire les relations géométriques entre les différents corps du robot, on définit les notions suivantes :

— Degré de liberté : nombre de paramètres utilisés pour spécifier la configuration d'un élément de la chaîne cinématique par rapport à un autre.

— Degré de mobilité : toute articulation est caractérisée par son degré de mobilité (DDM), c'est-à-dire le nombre de degrés de liberté entre deux corps successifs de la chaîne cinématique ($0 \leq m \leq 6$).

— Espace articulaire : représente l'état des corps composant le robot en fonction des variables articulaires, sa dimension n'est égale au nombre de degrés de liberté du robot.

— Espace opérationnel : décrit la position et l'orientation de l'organe terminal du robot par rapport à un repère de référence, sa dimension ($m \leq 6$).

2.3 Les différentes méthodes de modélisation

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques tels que les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal), et l'espace articulaire (dans lequel est définie la configuration du robot) [7].

On distingue :

— La modélisation géométrique : (directe et inverse).

— La modélisation cinématique (différentielle) : (directe et inverse).

— La modélisation dynamique :(directe et inverse).

2.3.1 Modèle géométrique direct

Le Modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaires [8].

Le Modèle géométrique direct du robot peut représenter par la relation :

$$X = f(q) \tag{2.3.1}$$

Où :

q étant le vecteur des variables articulaires tel que :

$$\mathrm{q} = [q_1\ q_2\ q_3\ \ldots\ldots\ldots\ldots\ q_n]^T \in R^n$$

Les coordonnées opérationnelles sont définies par :

$$\mathrm{X} = [X_1\ X_2\ X_3\ \ldots\ldots\ldots\ldots\ X_n]^T \in R^m : m \leq n.$$

Pour les rotations, d'autres représentations sont couramment utilisées comme les angles d'Euler, les angles de Roulis-Tangage-Lacet (RTL) ou les paramètres d'Euler (quaternions). On note la convention de Denaviet-Hartenberg et les matrices $4 * 4$:

Considérons la (figure II.1) deux articulations successives pour passer de O_{i-1} à O_i et de Z_{i-1} à Z_i

On peut toujours suivre le chemin suivant :

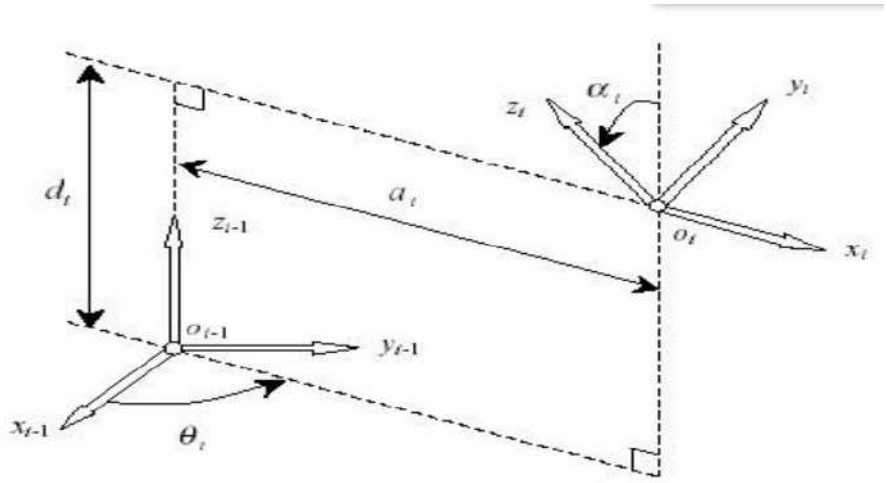


FIGURE 2.1 – **Systèmes de coordonnées et les paramètres de Denavit - Hardenberg**

1. une rotation autour de Z_{i-1} d'un angle θ_i : Rot (z, θ).
2. une translation le long de Z_{i-1} d'une longueur d_i : Trans (z, d).
3. une translation le long de X_i d'une longueur a_i : Trans (x, a).
4. une rotation autour de X_i d'un angle α_i : Rot (x, α).

Les grandeurs sont définies de la façon suivante :

— d_i : distance de O_{i-1} à la projection de O_i sur l'axe Z_{i-1}.

— a_i : distance de O_{i-1} à la projection de O_i sur le plan Y_i, O_i et Z_i.

— θ_i : angle compris entre X_{i-1} et le plan défini par l'axe Z_{i-1} et le point O_i.

— α_i : angle forme par la projection de Z_{i-1} sur le plan normal au plan (Z_{i-1}, O_i) et l'axe Z_i.

La forme générale de la matrice faisant passer d'une articulation à la suivante devra donc toujours :

La matrice de transformation homogène est donnée comme suit : [9]

$$^{i-1}T_i = \text{Trans}\ (x_{i-1}, a_{i-1})\ \text{Rot}\ (x_{i-1}, \alpha_{i-1})\ \text{Trans}\ (z_i, d_i)\ \text{Rot}\ (z_i, \bar{\omega}_i)$$

$$ {}^{i-1}T_i = \begin{bmatrix} 1 & 0 & 0 & \alpha_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\bar{\omega}_i & s\bar{\omega}_i & 0 & 0 \\ c\bar{\omega}_i & c\bar{\omega}_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{i-1}T_i = \begin{bmatrix} c\omega_i & -s\bar{\omega}_i & 0 & d_{i-1} \\ c\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1}s\bar{\omega}_i & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1} & -c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{i-1}T_i = \begin{bmatrix} {}^{i-1}A_i & {}^{i-1}P_i \\ 0_3 & 1 \end{bmatrix} .....(II.2) $$

Avec :

$$ {}^{i-1}A_i = \begin{bmatrix} c\omega_i & -s\bar{\omega}_i & 0 \\ c\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1}s\bar{\omega}_i & -s\alpha_{i-1} \\ s\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1} \end{bmatrix}, {}^{i-1}P_i = \begin{bmatrix} d_{i-1} \\ -s\alpha_{i-1}d_i \\ -c\alpha_{i-1}d_i \end{bmatrix} $$

2.3.2 Modèle géométrique inverse

On a vu que le modèle géométrique direct d'un robot permettait de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires [9].

Le model inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse (MGI). On peut distinguer trois méthodes de calcul du MGI :

— la méthode de Paul qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels ;

— la méthode de Peiper qui permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques.

— la méthode générale de Raghavan et Roth donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

Le modèle géométrique inverse peut s'écrit :

$$q = f^1(X) \quad \text{..... (II .3)}$$

Dans le calcul du MGI trois cas se présentent :

1. absence de solution lorsque la situation désirée est en dehors de la zone accessible du robot. Celle-ci est limitée par le nombre de degrés de liberté, les débattements articulaires et la dimension des segments.
2. infinité de solutions lorsque :
   — le robot est redondant vis-à-vis de la tâche.
   — le robot se trouve dans certaines configurations singulières.
3. solutions en nombre fini, exprimées par un ensemble de vecteurs $[q_1,...,q_i]$.

   On dit qu'un robot manipulateur est résoluble lorsqu'il est possible de calculer toutes les configurations permettant d'atteindre une situation donnée. Aujourd'hui, tous les manipulateurs série ayant jusqu'à six degrés de liberté et qui ne sont pas redondants peuvent être considérés comme résolubles.
4. Le nombre de solutions dépend de l'architecture du robot manipulateur.

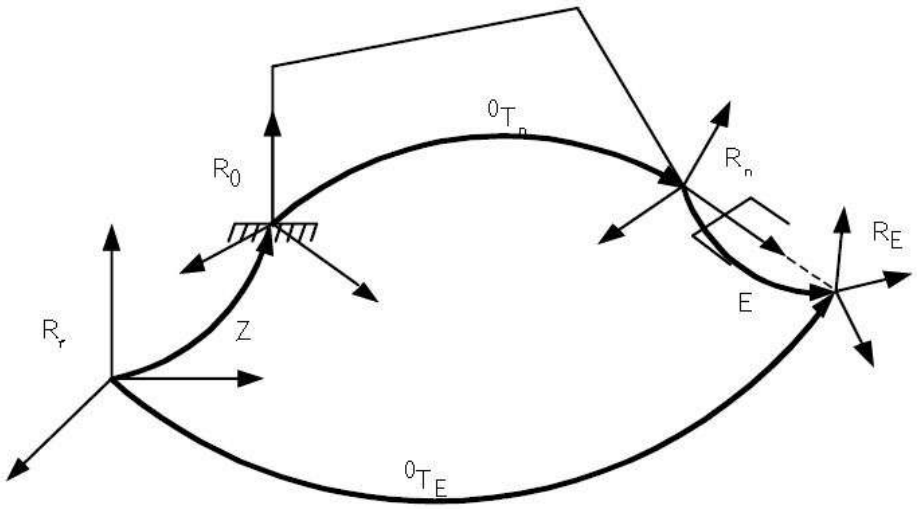


FIGURE 2.2 – **Transformations entre l'organe terminal et le repère atelier**

2.3.3 Le model cinématique direct

Nous venons de constater que le model géométrique posait deux catégories principales de problème dans son utilisation pour la commande des robots [9] :

1. la non linéarité de l'équation X=f (θ) entraine des difficultés d'obtention du transformateur de coordonnées inverse .le robot est soumis à la restriction solvable.
2. avec la commandes géométrique on ne contrôle pas (ou très mal) la vitesse de déplacement du robot. Dans les applications courantes il est très important de connaitre la dure d'exécution d'une tache ou plus exactement d'en être maitre, d'où la nécessite de contrôler la vitesse.

L'utilisation du model cinématique permet dans une certaine mesure de contourner ces deux difficultés d'où son intérêt. Le model cinématique direct d'un robot manipulateur décrit les variations élémentaire des coordonnées opérationnels en fonction des variations élémentaire des coordonnes articulaires il est noté :

$$\dot{X} = J(q)\dot{q} \quad \text{..... (II .4)}$$

Où J(q) désigne la matrice jacobéenne de dimension $(m \times n)$ du mécanisme, égale à $\partial X/\partial q$ et fonction de la configuration articulaire q. La même matrice jacobéenne intervient dans le

calcul du modèle différentiel direct qui donne les variations élémentaires d_x des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires d_q soit :

$$\text{dx} = \text{J(q) dq} \quad \text{..... (II .5)}$$

2.3.3.1 Matrices jacobéenne cinématique

On peut obtenir la matrice jacobéenne par une méthode de calcul direct, fondée sur la relation entre les vecteurs des vitesses de translation et de rotation V_n et ω_n du repère R_n, et les vitesses articulaires $\dot{q}$ [9].

$$\begin{bmatrix} V_n \\ \omega_n \end{bmatrix} = J(q)\dot{q} \quad \text{.....} (II.6)$$

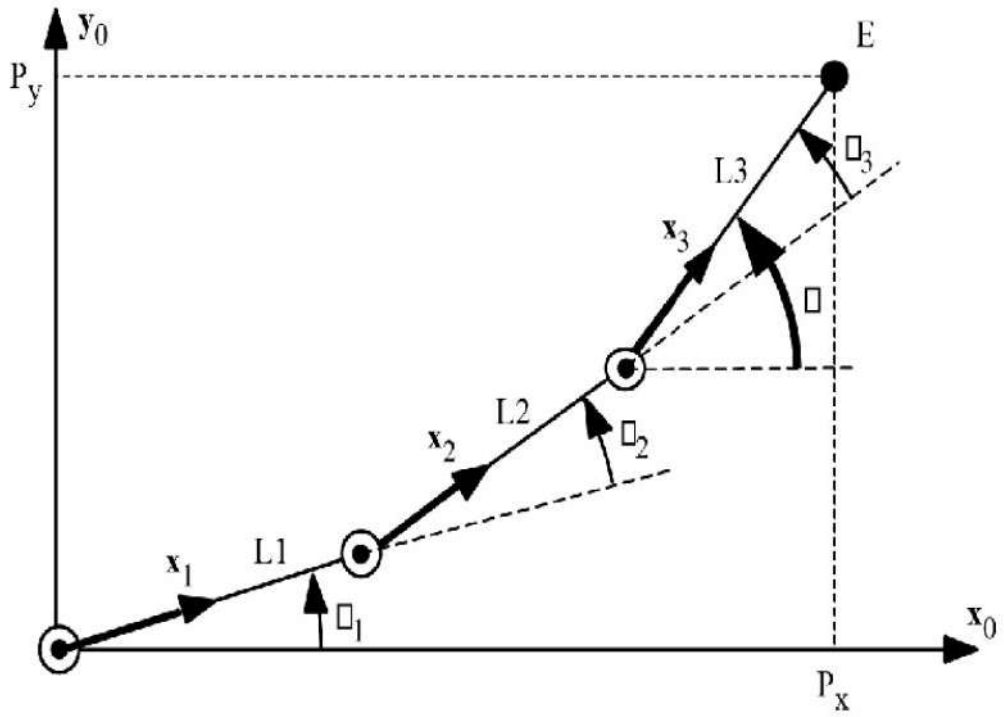


FIGURE 2.3 – **Exemple d'un robot plan à trois degrés de liberté**

On note que Vn est la dérivée par rapport au temps du vecteur P_n. En revanche, ω_n n'est pas la dérivée d'une représentation quelconque de l'orientation.

2.3.4 Le model cinématique inverse

L'objectif du modèle cinématique inverse est de calculer, à partir d'une configuration q donnée, les vitesses articulaires qui assurent au repère terminal une vitesse opérationnelle.

Cette définition est analogue à celle du modèle différentiel inverse : ce dernier permet de déterminer la différentielle articulaire dq correspondant à une différentielle des coordonnées opérationnelles dx spécifiée. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires.

Les algorithmes traitent de façon unifiée les cas réguliers, singuliers et redondants. Elles nécessitent un temps de calcul relativement important. Nous présentons dans ce paragraphe les techniques à mettre en oeuvre pour établir un modèle cinématique inverse dans les cas réguliers, singuliers et redondants [9].

2.3.5 La Modélisation dynamique

Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. On représente le modèle dynamique par une relation de la forme [9] :

$$\Gamma = \mathrm{f}(\mathrm{q}, \dot{q}, \ddot{q}, f_e) \quad \text{..... (II .7)}$$

Avec :

— Γ : vecteur des couples/forces des actionneurs, selon que l'articulation est rotoîde ou prismatique.

— q : Vecteur des positions articulaires.

— $\dot{q}$: Vecteur des vitesses articulaires.

— $\ddot{q}$: Vecteur des accélérations articulaires.

— f_e : vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

Le modèle dynamique direct est celui qui exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulations. Il est alors représenté par la relation :

$$\ddot{q} = g(q, \dot{q}, \Gamma, f_e) \quad \text{..... (II .8)}$$

Parmi les applications du modèle dynamique, on peut citer :

— la simulation, qui utilise le modèle dynamique direct.

— le dimensionnement des actionneurs.

— l'identification des paramètres inertiels et des paramètres de frottement du robot.

— la commande, qui utilise le modèle dynamique inverse.

Plusieurs formalismes ont été utilisés pour obtenir le modèle dynamique des robots, les formalismes les plus souvent utilisés sont :

— le formalisme de Lagrange pour le calcul du model dynamique direct.

— le formalisme de Newton-Euler, leur méthodes sont les mieux adaptés pour le calcul du model dynamique inverse.

2.4 Conclusion

Le chapitre présent fait une description sur les différents modèles de la modélisation mécanique et on s'est intéressé surtout par notre modèle du bras manipulateur.

Á propos de la présentation de la partie commande qui réunit la carte de commande et le type d'actionneur utilisé feront l'objet du chapitre suivant.

Chapitre 3

Arduino et Servomoteur

3.1 Introduction

Le système Arduino est un projet créé par une équipe de développeurs, composée de six individus. Ce système est un outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes. Le système Arduino nous permet de réaliser un grand nombre de choses, qui ont une application dans tous les domaines.

Nous voulons aussi présenter dans ce chapitre les servomoteurs récents. Aujourd'hui dans le monde robotique on trouve beaucoup d'application qui sont réalisées à l'aide des actionneurs électriques de type servomoteur car ils sont facile pour la commande avec une précision élevée et délivre un couple considérable.

3.2 Arduino

3.2.1 Définition

L'Arduino est une plateforme open source d'électronique programmée qui est basée sur une carte à microcontrôleur et un logiciel. Plus simplement, on peut dire que l'Arduino est un module électronique, doté d'un microcontrôleur programmable, il peut être utilisé pour développer des objets interactifs, munis d'interrupteurs ou de capteurs, et peut contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielle.

3.2.2 Microcontrôleur

Un microcontrôleur est un circuit intégré (ou IC, integrated Circuit), qui rassemble sur une puce plusieurs élément complexes dans un espace réduit, au temps des pionniers de l'électronique, on soudait un grand nombre de composants encombrants, tels que les transistors, les résistances ou les condensateurs sur des cartes plus ou moins grandes. Aujourd'hui, tout peut loger dans un petit boîtier en plastique noir muni d'un certain nombre de broches. Ces dernières sont les connexions du circuit intégré un moyen desquelles s'effectue la communication.

La figure montre un microcontrôleur ATmega328, qu'on trouve sur la carte Arduino [10].

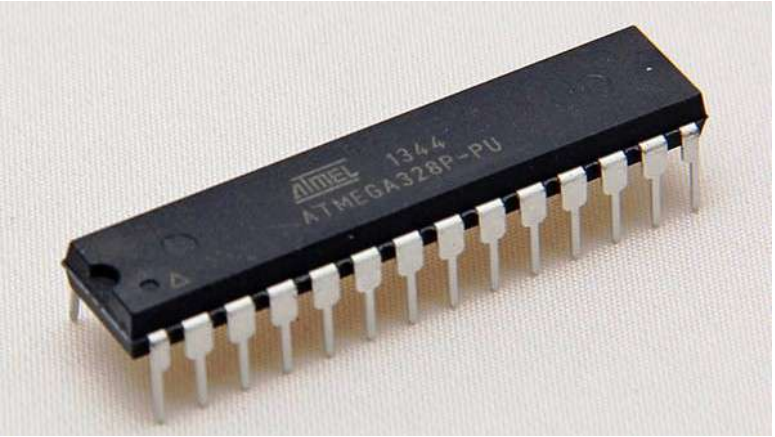


FIGURE 3.1 – **Microcontrôleur ATmega328,(source : Atmel)**

3.2.3 Les différentes cartes Arduino

Les cartes Arduino doivent satisfaire des exigences diverses et variées. Certains utilisateurs souhaiteront effectuer du prototypage et tester de nouveaux montages ou idées, la carte Arduino Uno disposant d'assez d'entrées-sorties pour les projets d'envergure raisonnable. De part sa taille, la carte devrait aussi pouvoir être utilisée à l'avenir dans des projets plus ambitieux. D'autres auront besoin d'un grand nombre de ports afin de pouvoir raccorder de nombreux capteurs ou actionneurs. Une troisième catégorie d'utilisateurs ne cherchera qu'à attirer l'attention et à transmettre des signaux à leurs semblables à l'aide de diodes clignotantes.

La forme, la taille ou les possibilités de connexion jouent un rôle décisif dans le choix de la carte adaptée. C'est pourquoi les développeurs d'Arduino ont mis au point un vaste choix de cartes à microcontrôleur afin que chacun trouve le modèle qui répondre à ses besoin :

- Arduino uno
- Arduino Leonardo
- Arduino Mega 2560
- Arduino Esplora
- Boarduino V2.0
- Arduino Nano
- Arduino LilyPad
- Arduino Due
- Arduino Yûn

3.2.4 La carte Arduino Uno

La pièce maîtresse de tous les montages est : la carte à microcontrôleur Arduino

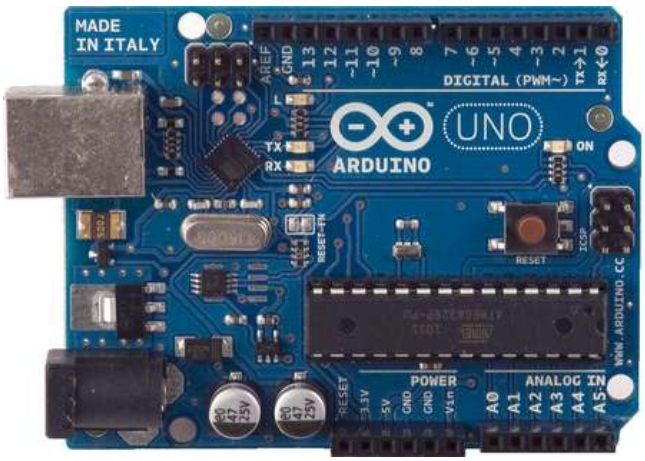


FIGURE 3.2 – **La carte à microcontrôleur,(Arduino)**

Sur cette image, vous ne pouvez évidemment pas vous rendre compte à quel point les dimensions de la carte Arduino sont réduites (environ 7 cm de large et 5 cm de long) ; elle est vraiment très maniable, tient sans problème dans une main et s'avère vraiment compacte.

Au fil des années, plusieurs cartes Arduino ont été développées. Nous avons choisi de prendre l'Uno, car c'est la plus populaire.

Les composants les plus divers y sont reconnaissable (voir FIGURE III.3)

Certains penseront certainement qu'on ne peut pas concevoir quelque chose de sérieux sur une surface aussi réduite. Mais grâce à la miniaturisation des composants ces dernières années, ce qui nécessitait auparavant cinq puces électroniques n'en requiert aujourd'hui plus qu'une.

Le plus gros élément qui saute directement aux yeux est le microcontrôleur proprement dit. Il est de type ATmega328.

Elle convient tout particulièrement à ceux qui veulent faire leurs premiers pas dans le monde des microcontrôleurs.

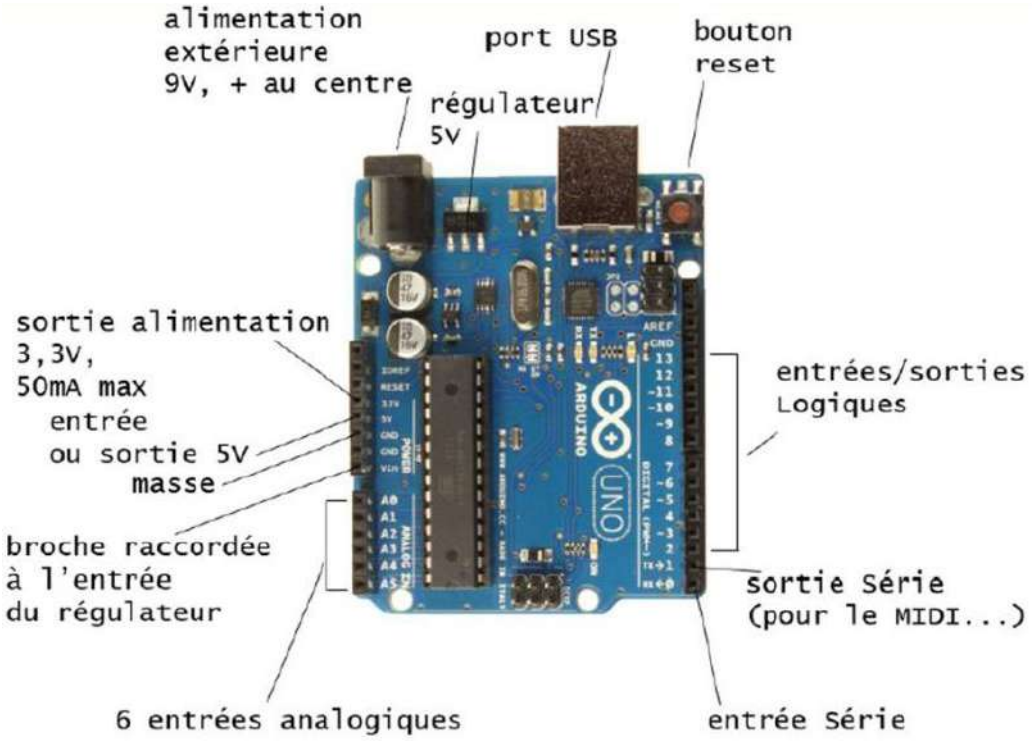


FIGURE 3.3 – **les composants de la carte Arduino**

Ces éléments sont les plus importants de la carte Arduino mais, bien entendu, cela ne veut pas dire que les autres sont à négliger.

Voici les principales caractéristiques de la carte Arduino :

- Microcontrôleur Atmega328.
- Tension de service 5V.
- 14 entrées et sortis numériques (6 sorties commutables en MLI).
- 6 entrées analogiques (résolution 10 bit).
- 32 Ko de mémoire flash (0.5 Ko occupé par le chargeur d'amorçage ou bootloarder).
- 2 Ko de SRAM.
- 1 Ko d'EEPROM.
- Fréquences d'horloge 16 MHz.
- Interface USB.

3.2.5 Les modes de communication

Un microcontrôleur de type Arduino a déjà beaucoup de connexions qu'il s'agit de bien distinguer.

3.2.5.1. Le port USB

Sans le port USB, vous ne seriez pas en mesure d'initialiser une communication.

Le travail avec la carte Arduino peut se diviser en deux étapes : le temps consacré à la mise en oeuvre du montage et celui dédié à la programmation, appelé phase de développement (design time).

La programmation s'effectue dans un environnement de développement que vous allez apprendre à connaître très rapidement. C'est dans cet environnement que vous allez saisir le programme créé par vos soins pour le transmettre au microcontrôleur. Si tout s'est bien passé, le temps de l'exécution (run time) commence. Vous n'avez pas besoin de dire explicitement au microcontrôleur, il démarre en effet immédiatement après avoir reçu toutes les instructions de votre part. Vous pouvez en outre échanger des données avec votre ordinateur via le port USB.

3.2.5.2. Les ports d'entrée ou de sortie (E/S)

Les ports E/S représentent l'interface du microcontrôleur. Il existe plusieurs chemins ou canaux pour échanger des données, comme chez l'homme avec les yeux, les oreilles et la bouche. Il se produit, Grâce et par le biais de ces canaux de communication, une interaction avec l'environnement.

3.2.5.2.1. les ports d'entrée et de sortie sur la carte Arduino

Quand vous la tenez de manière à pouvoir lire l'inscription (UNO) vous verrez les ports d'entrée et de sortie sur le bord supérieur (un bloc de dix broches et un autre de huit broches).

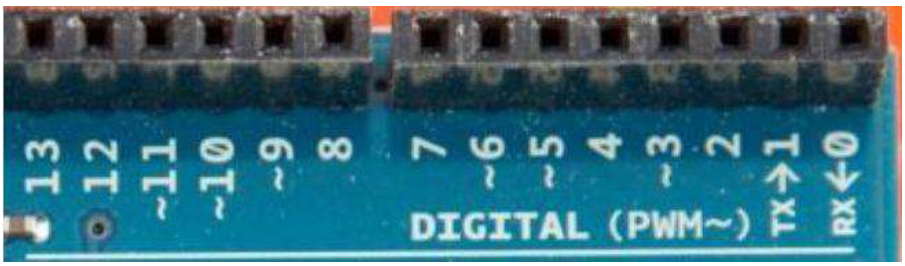


FIGURE 3.4 – **Entré et Sortie numérique**

Il est bien entendu important de savoir pour chaque port quelle est son adresse afin de pouvoir communiquer avec lui au cours de la programmation. C'est pour cela que chaque broche possède un numéro.

Sous certains chiffres se trouve un tilde ( ) indiquant que la broche peut être commutée en sortie analogique. Il s'agit là d'une broche MLI (rappelez-vous la modulation de largeur d'impulsion, on y reviendra un peu plus loin).

Á l'extrémité inférieure de la carte se situent les ports d'alimentation (à gauche) et les ports d'entrée analogique (un bloc de huit broches et un autre de six broches).

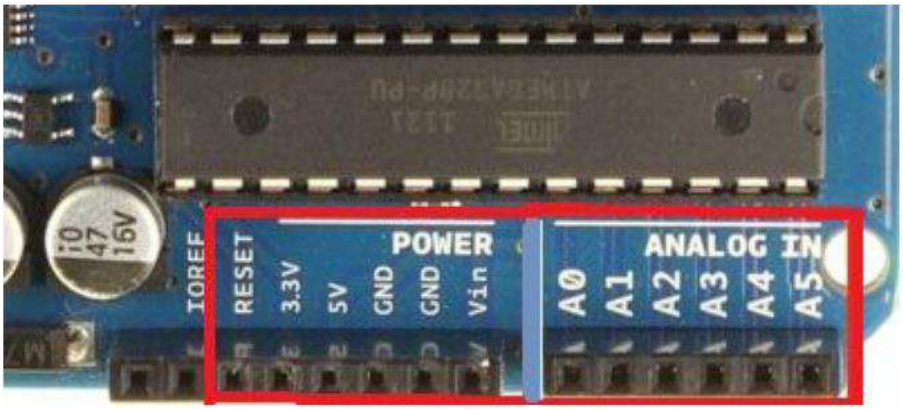


FIGURE 3.5 – **Alimentation et entrée Analogiques**

3.2.6 L'alimentation électrique

La carte Arduino Uno doit être alimentée en énergie pour pouvoir travailler.

Cette alimentation peut s'effectuer tout d'abord via l'interface USB qui relie la carte à l'ordinateur-ce chemin sert aussi à l'échange de données entre la carte et l'ordinateur. En phase de développement avec votre Arduino, la connexion USB va servir d'alimentation primaire de la carte.

La seconde possibilité consiste à brancher une batterie ou bloc secteur au connecteur, appelé prise jack. Vous pouvez, par exemple, employer cette variante si vous avez construit un engin manoeuvrable, commandé par la carte Arduino. Le Véhicule doit pouvoir évoluer librement dans l'espace, sans câble.

En effet, l'utilisation d'un câble USB, généralement trop court, limiterait alors la mobilité de l'engin. L'emploi d'une batterie rend le dispositif autonome.

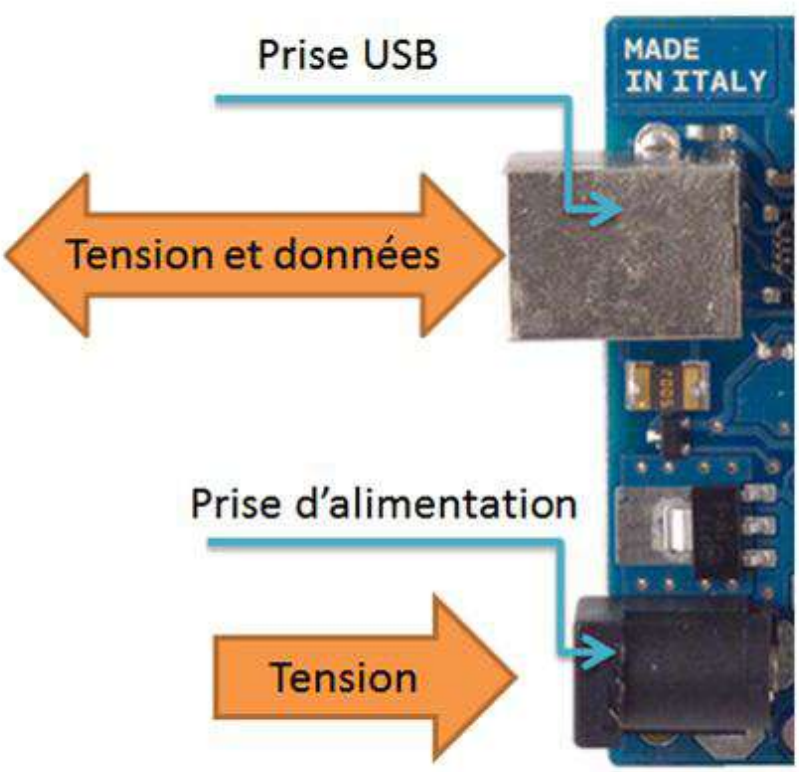


FIGURE 3.6 – **Alimentation de la carte Arduino**

Nous montrons ici les différentes prises. Attention, elles ne peuvent pas être interverties, car elles ont des formes et des fonctions différentes.

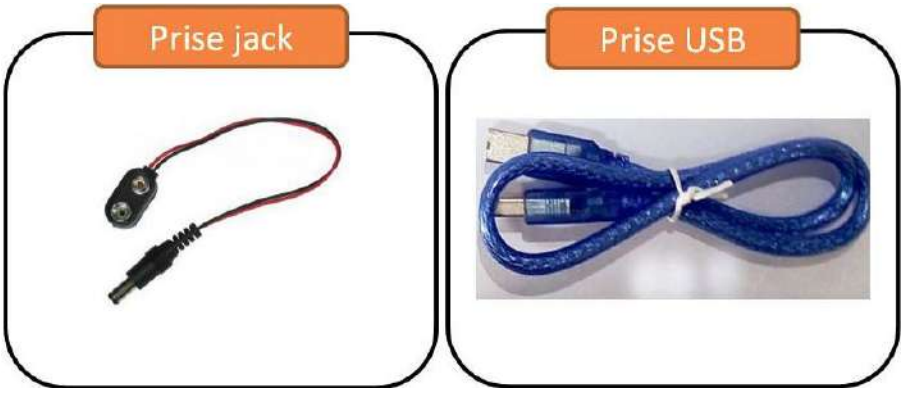


FIGURE 3.7 – **les différentes prises**

Dès qu'il s'agit de courant ou de tension, il convient de consulter le tableau.

3.2.7 Valeurs de courant ou de tension

Catégorie	Valeur
Tension de service	5 V (DC)
Alimentation depuis l'extérieur (recommandée)	7-12 V(DC)
Alimentation depuis l'extérieur (valeur limité)	6-20 V(DC)
Courant continu par broche (maximal)	40 mA

TABLE 3.1 – **Valeurs de courant ou de tension**

l'interface USB peut fournir un courant maximal de 500 mA ; c'est en principe suffisant pour réaliser la plupart des circuits d'essai. Elle est protégée contre le court-circuit et les courants forts grâce à un poly fusible. Mais attention, cela ne doit pas vous empêcher de construire votre circuit avec le plus grand soin.

3.2.8 La Programmation de l'Arduino

3.2.8.1 Le Langage De Programmation

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme Arduino s'effectue

de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres.

3.2.8.2 La Structure D'un Programme Arduino

Comme tous autres langages de programmation de bas niveau, un programme Arduino comporte trois parties :

1) la partie déclaration des variables (optionnelle).

2) la partie initialisation et configuration des entrées/sorties : la fonction setup ().

3) la partie principale qui s'exécute en boucle : la fonction loop ().

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino.



FIGURE 3.8 – **Coloration Syntaxique dans IDE de l'Arduino**

Dans l'interface de programmation de l'Arduino, certains mots apparaissent en différentes couleurs qui clarifient le statut des différents éléments :

- En orange, apparaissent les mots-clés reconnus par le langage Arduino comme des fonctions existantes. Lorsqu'on sélectionne un mot coloré en orange et qu'on effectue un clic avec le bouton

droit de la souris, l'on a la possibilité de choisir (Find in reference) : cette commande ouvre directement la documentation de la fonction sélectionnée.

- En bleu, apparaissent les mots-clés reconnus par le langage Arduino comme des constantes.

- En gris, apparaissent les commentaires qui ne seront pas exécutés dans le programme. Il est utile de bien commenter son code pour s'y retrouver facilement ou pour le transmettre à d'autres personnes. L'on peut déclarer un commentaire de deux manières différentes : dans une ligne de code, tout ce qui se trouve après (//) sera un commentaire. l'on peut encadrer des commentaires sur plusieurs lignes entre (/*) et (*/).

3.3 Les Servomoteurs

3.3.1 Définition

Les servomoteurs sont des actionneurs très utilisés en modélisme et dans l'industrie, ils ont comme caractéristique principale leur (couple), c'est-à-dire la force de rotation qu'ils peuvent exercer. Plus un servomoteur aura de couple et plus il pourra actionner des (membres) lourds comme déplacer un bras qui porte une charge.

Pour la robotique de loisirs, les servomoteurs ont en général peu de couple et sont de taille réduite, bien adaptée à un encombrement minimal et à une énergie disponible limitée.



FIGURE 3.9 – **les différents types de servomoteur**

Les servomoteurs sont pilotés par un fil de commande et alimentés par deux autres fils. Habituellement, ces 3 fils sont rassemblés dans une prise au format.

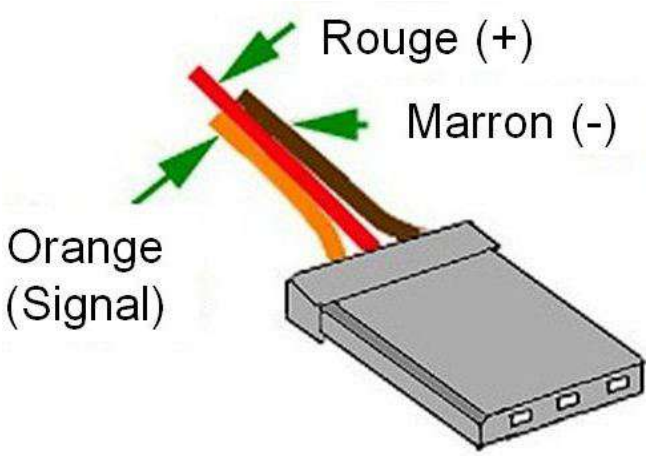


FIGURE 3.10 – **Les Trois Files**

Un fil rouge est relié à l'alimentation positive (+5 ou +6 V selon le servo), le fil noir est relié à la masse (GND) et le fil jaune est utilisé pour la commande (PWM). Il y aurait beaucoup à dire sur le fonctionnement d'un servomoteur, ses composants, son moteur et le petit potentiomètre qui permet de connaître sa position mais cette fiche va droit au but et se limitera à son utilisation avec l'Arduino [11].

3.3.2 Le Principe de l'asservissement d'un Servomoteur

À présent voici le schéma de principe qui a pour fonctionnement celui expliqué par le synoptique précédent :

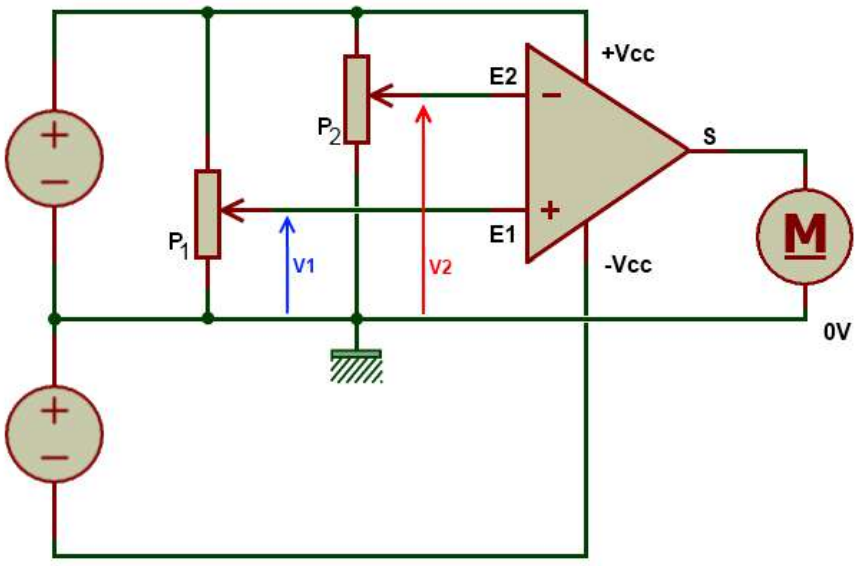


FIGURE 3.11 – **Le Principe de l'asservissement d'un Servomoteur**

De gauche à droite on a : les alimentations qui fournissent la tension positive et négative ; les potentiomètres P1 et P2 ; le comparateur (oui c'est ce gros triangle avec un plus et un moins) : enfin le moteur à courant continu.

Un comparateur est un composant électronique de la famille des circuits intégrés car, il contient en vérité d'autres composants, essentiellement des semi-conducteurs (diodes, transistors) et des résistances. Ce composant a toujours besoin d'une alimentation externe pour fonctionner, c'est à dire qu'on ne peut lui mettre des signaux à son entrée que s'il est alimenté. Autrement il pourrait être endommagé (c'est pas souvent le cas, mais mieux vaut être prudent). Vous le constatez par vous-même, le comparateur est un composant qui possède deux entrées et une sortie. Et, de la manière la plus simple qui soit, en fait il n'y a rien de plus simple qui puisse exister, son fonctionnement réside sur le principe suivant :

Si la tension (je me base par rapport au schéma) $V1$ qui arrive sur l'entrée $E1$ du comparateur est supérieure à la tension $V2$ qui entre sur l'entrée $E2$ du comparateur, alors la tension en sortie S du comparateur est égale à $+Vcc$ (l'alimentation du comparateur).

Tandis que dans le cas opposé où la tension $V2$ va être supérieure à $V1$, la sortie S du comparateur aura une tension égale à $-Vcc$.

En transposant mes dires sous une forme mathématique, cela donnerait ceci :

— Si $V1 > V2$, alors $Vs = +Vcc$.

— Si $V1 > V2$, alors $Vs = -Vcc$.

Comment s'en rappeler. Eh bien grâce aux petits symboles $(+)$ et $(-)$ présents dans le triangle représentant le comparateur.

La sortie du comparateur prendra +Vcc si la tension sur l'entré $(+)$ du comparateur est supérieure à celle sur l'entrée $(-)$ et inversement. Voyez, j'vous z'avais dis que c'était ultra simple.

Il y a encore quelque chose à savoir : Il est impossible que les tensions V1 et V2 soient égales ! Oui, car le comparateur ne peut pas fournir une tension positive ET une tension négative en sa sortie, c'est pourquoi, même si vous reliez E1 et E2 avec un fil, la tension en sortie du comparateur sera toujours OU +Vcc OU -Vcc.

3.3.3 Le rôle des servomoteurs

Il permet de réaliser des déplacements en translation ou en rotation sur des modèles réduits.

Il existe une grande variété et qualité de servomoteur se différenciant par leur vitesse de rotation/seconde, par leur course en degré et par leur couple exprimé en kg/cm.

Ils se distinguent également par la matière utilisée et le montage éventuel sur roulements[11].

3.3.4 Fonctionnement d'un servomoteur standard

Mécaniquement, il s'agit d'un moteur avec réducteur. Le mouvement de sortie est une rotation. Lorsque le moteur tourne, l'axe du servomoteur change de position, ce qui modifie la résistance du potentiomètre. Le rôle de l'électronique est alors de commander le moteur pour que la position de l'axe de sortie soit conforme à la consigne reçue.

La commande se fait via une petite impulsion dont la durée est comprise entre 0,5 ms et 2,5 ms. La valeur de 1,5 ms donne au servomoteur la position centrale.

Les ordres de positions sont transmis sous forme d'un signal codé en largeur d'impulsion, répété périodiquement, en général toutes les 50 ms, ce qui permet à l'électronique de contrôle et de corriger continuellement la position angulaire de l'axe de sortie (en général, course limitée à 90°ou 120°).

L'augmentation de l'amplitude des impulsions agit sur le contrôle de la vitesse jusqu'au moment où le bras du servomoteur se trouve dans la position souhaitée.

Durant la rotation, le potentiomètre indique au circuit électronique le moment où la position désirée est obtenue. Les impulsions diminuent l'amplitude jusqu'à ce que plus aucune tension ne soit appliquée au moteur du Servomoteur tout en conservant le bras dans sa nouvelle position.

3.3.4.1 les Inconvénients des servomoteurs

L'impulsion est forte et instantanée, entrainant aucune continuité de la rotation initiée comme pourrait le faire une mise sous tension continue. Les petites impulsions données au moyen de la manette de commande ne produisent que de faibles effets. Ils peuvent se révéler inefficaces et sont les causes de l'importante zone neutre qui caractérise les servomoteurs traditionnels.

3.3.5 Les composants d'un servomoteur

1- Un moteur électrique MCC.

2- Une réduction en sortie de ce moteur pour avoir du couple.

3- Un potentiomètre qui induit une résistance variable en fonction de la position angulaire de sortie, donc le potentiomètre est lié à l'axe du moteur.

4- Un asservissement électronique pour contrôler la position angulaire

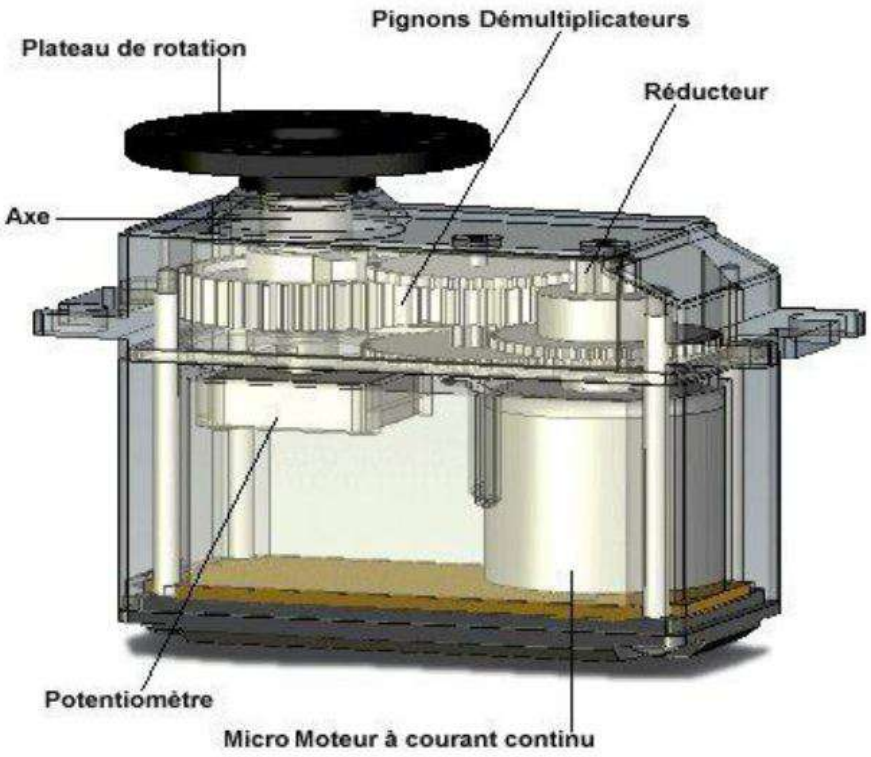


FIGURE 3.12 – **Les composants d'un servomoteur**

3.3.6 La théorie de la commande des servomoteurs

Le signal de commande d'un servomoteur est un signal PWM de fréquence 50 Hz et dont le niveau haut doit être entre 1 et 2 ms (en théorie).

- 1 ms : 0°en angle de sortie.
- 1.5 ms : 90°en angle de sortie.
- 2 ms : 180°en angle de sortie.

Le déplacement angulaire total est de 180°.

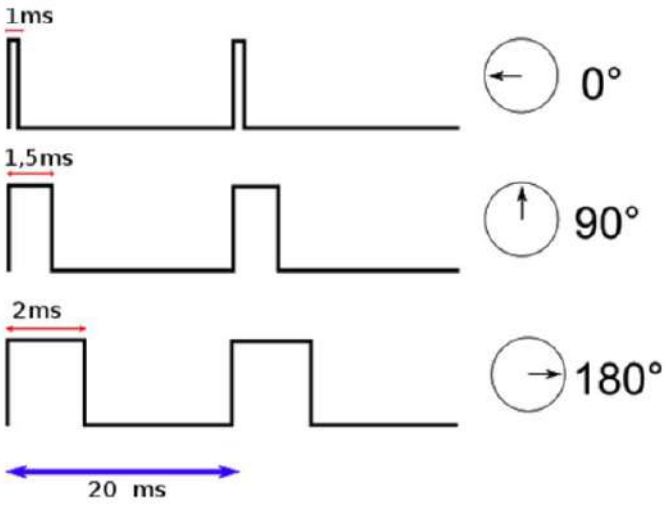


FIGURE 3.13 – **Chronogramme de commande de Servomoteur**

3.3.7 l'implimentation de la commande théorique

La commande d'un servomoteur en pratique est effectuée par un microcontrôleur avec l'utilisation :

- Périphérie PWM.
- Périphérie Timer.
- Ou utilisation des fonctions de type delay ms.

3.3.8 Les servomoteurs digital et analogique

Outre le fait qu'il contient le même moteur, les mêmes engrenages que les analogiques et disposent eux aussi d'un potentiomètre, le servomoteur digital se différencie du servomoteur analogique par le fait qu'il incorpore un microprocesseur qui analyse le signal reçu du récepteur et contrôle le moteur du servomoteur.

La différence principale est donc la façon de traiter l'information reccue du récepteur et le contrôle exercé sur la tension électrique appliquée au moteur, ce qui permet de réduire la zone neutre, augmentant ainsi l'amplitude du mouvement et générant une grande stabilité du positionnement statique.

3.3.8.1 Les Avantages du servomoteur digital

- Des impulsions de plus grande fréquence : 300 impulsions par seconde contre 50 impulsions par seconde dans le cas d'un servomoteur analogique.
- Le moteur réagit mieux à l'ordre donné.
- Une diminution de la zone neutre du servomoteur entrainant une meilleure rapidité de réponse, une accélération rapide, un freinage progressif.
- Une meilleure stabilité dans le positionnement statique.

3.3.8.2 Les Inconvénient du servomoteur digital

- Une plus grande consommation d'énergie.

3.3.9 Le pilotage des servomoteurs

Le mode de commande d'un servomoteur est standardisé : on envoie sur son fil de commande une impulsion dont la durée correspond à l'angle désiré.

Historiquement, cette impulsion était délivrée par un circuit oscillateur. Le circuit intégré NE555 est un exemple vedette du circuit utilisé.

Avec la programmation de l'Arduino, ce circuit n'est plus nécessaire. Une bibliothèque (library) dédiée, la bibliothèque (servo), permet de piloter un servomoteur en lui transmettant simplement l'angle sur lequel il souhaite se positionner.

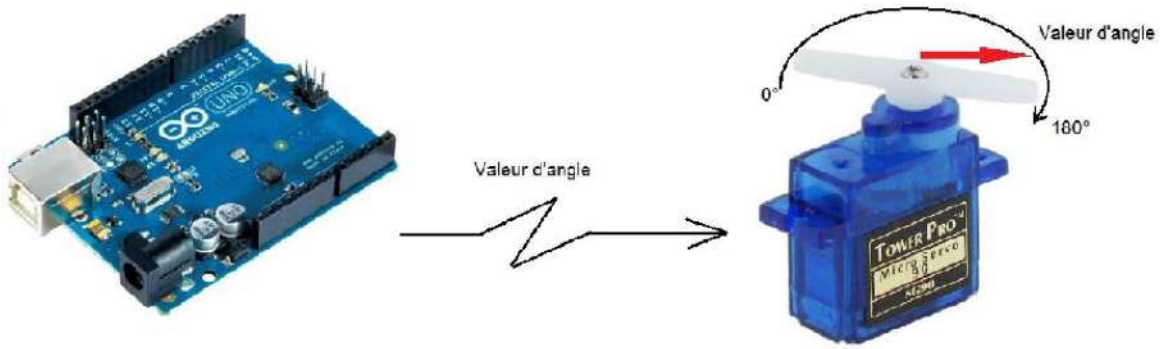


FIGURE 3.14 – **Présentation d'un servomoteur avec Arduino**

Ce qui permet de réaliser simplement des mouvements qui peuvent être complexes et de les

automatiser, voir la figure suivante :



FIGURE 3.15 – **Exemple de piloter des servomoteurs avec l'Arduino**

Les connaissances nécessaires au pilotage d'un servomoteur sont :

TABLE 3.2 – **Les connaissances nécessaires au pilotage d'un servomoteur**

#include <Servo.h>	L'utilisation de la librairie servo.h qui prend en charge la communication de bas niveau entre l'Arduino et le servomoteur. Les commandes précédées par # sont des ordres particuliers donnés au compilateur
Servo myservo ;	La création d'un objet de type Servo, appelé ici myservo
myservo.attach(9) ;	Le fil de commande de ce servo sera connecté au PIN 9 et l'objet myservo pilotera ce PIN
myservo.write(90) ; myservo.write(Angle) ;	Demander au servomoteur de se déplacer á l'angle désiré, soit de façon absolue en lui indiquant une valeur entière (90°) dans le premier cas, soit en lui passant le contenu d'une variable (Angle) compris entre 0 et 180, ce qui peut être utile par exemple pour donner une progressivité au déplacement en faisant varier l'angle d'un pas fixe (quelques degrés) par une boucle
myservo.read(Angle) ;	Pour lire la valeur de l'angle du servomoteur

3.4 Conclusion

En effet ce chapitre est divisé en deux parties, on a exposé dans la 1er partie les différents types de la carte Arduino précisément la carte Arduino Uno avec sa microcontrôleur et ses principales caractéristiques.

La 2ème partie rassemble le fonctionnement du servomoteur et ses deux modes de com-

mandes (analogique, digital) en tenant la méthode de pilotage d'un servomoteur avec la carte Arduino.

Chapitre 4

La Réalisation de la Commande d'un Bras Manipulateur, Résultats et Discussions

4.1 Introduction

Ce chapitre sera consacré à la présentation expérimentale du travail accompli selon différents phases de réalisation, généralement, on peut dire que notre projet est constitué de deux principales parties :

— Partie soft (Partie Programmations).

— Partie hard (Partie Mécatroniques).

Ainsi, l'adaptation entre les deux parties (soft et hard).

4.2 Simulation, Repérage et paramétrage du bras manipulateur plan de type 3R dans le plans

4.2.1 Le Repérage

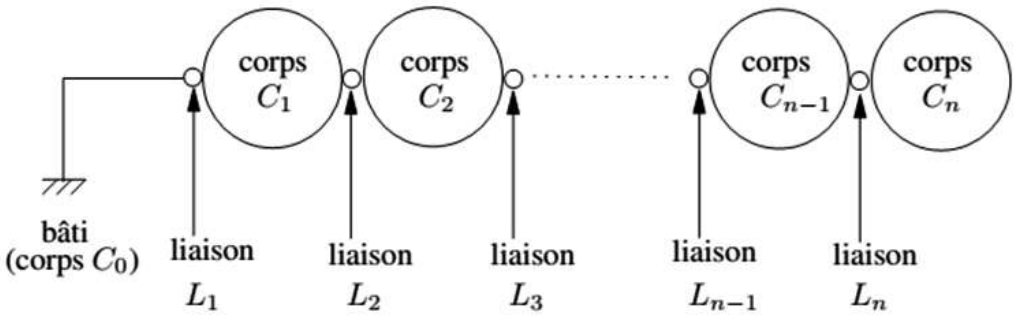


FIGURE 4.1 – **Chaîne cinématique d'un bras manipulateur série**

Selon la méthode des paramètres de Denavit-Hartenberg modifiés, On associe au ière corps le repère $R_i = (O_i, x_i, y_i, z_i)$, avec i = 0, 1, ..., n. Pour placer les repères R_1 à R_n selon la règle des paramètres de Denavit-Hartenberg modifiés, il convient de suivre les règles suivantes (Figure IV.2), pour i = 2, 3, ..., n :

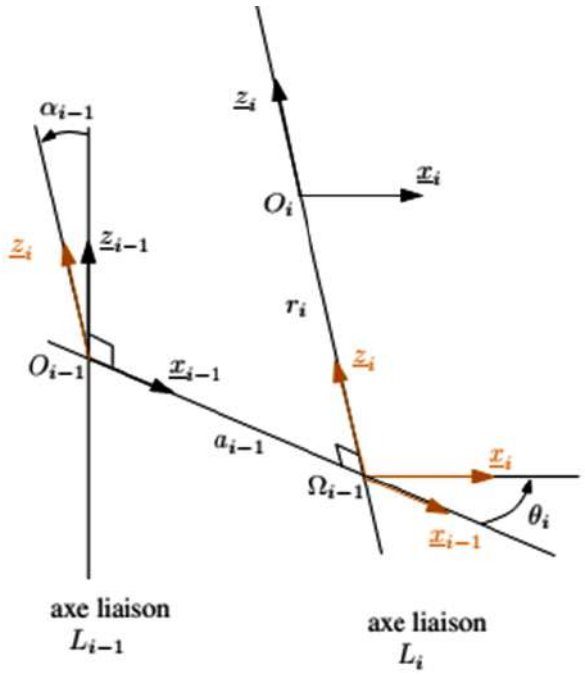


FIGURE 4.2 – **Paramètres de Denavit-Hartenberg modifiés**

— z_{i-1} est l'axe de la liaison L_{i-1}, librement orienté.

— O_{i-1} est le pied de la perpendiculaire commune aux axes des liaisons L_{i-1} et L_i, situé sur L_{i-1}.

— x_{i-1} est le vecteur unitaire de cette perpendiculaire commune, orienté de L_{i-1} vers L_i.

Le mécanisme comporte quatres liaisons rotoîdes.

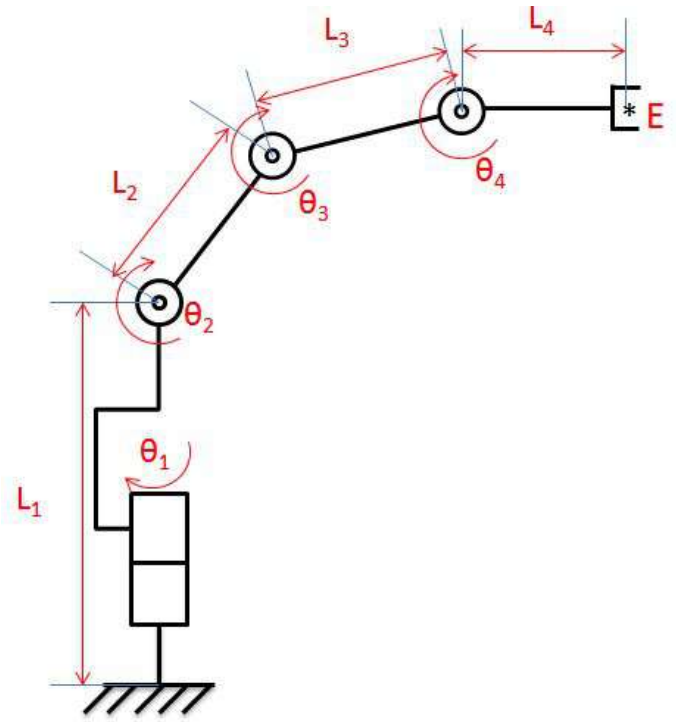


FIGURE 4.3 – **Repérage de notre mécanisme**

4.2.2 Le Paramétrage

selon la méthode des paramètres de Denavit-Hartenberg modifiés Les paramètres, On peut les définir comme suit :

— z_i est l'axe de la liaison i.

— α_{i-1} (twist angle) est l'angle algébrique entre z_{i-1} et z_i, mesuré autour de x_{i-1} ;

— a_{i-1} (link length) est la distance arithmétique de la perpendiculaire commune aux axes des liaisons L_{i-1} et L_i mesurée le long de x_{i-1} ;

— θ_i (joint angle) est l'angle algébrique entre x_{i-1} et x_i, mesuré autour de z_i ;

— (offset length) est la distance algébrique du point O_i à la perpendiculaire, mesurée le long de z_i. c'est la distance de x_{i-1} à x_i, mesurée le long de z_i.

corps / Paramètres	α_{i-1}	a_{i-1}	θ_i	r_{i-1}
L1	0	0	θ_1	0
L2	$\pi/2$	0	θ_2	0
L3	0	L_2	θ_3	0
L4	0	L_3	θ_4	0

TABLE 4.1 – **Paramétrage de notre mécanisme**

4.2.3 L'etablissement des relations géométriques

Conformément au paramétrage de Denavit-Hartenberg modifié, la matrice de transformation entre les corps L_{i-1} et L_i est donc :

$$^{i-1}T_i = Trans(x_{i-1}, a_{i-1})Rot(x_{i-1}, \alpha_{i-1})Trans(z_i, d_i)Rot(z_i, \bar{\omega}_i) \tag{4.2.1}$$

Qui prend la forme :

$$^0T_1 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ;^1T_2 = \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ;$$

$$^2T_3 = \begin{bmatrix} C_3 & -S_3 & 0 & L_2 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ;^3T_4 = \begin{bmatrix} C_4 & -S_4 & 0 & L_3 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

On note : $\cos\theta_i = C_i$; $\sin\theta_i = S_i$;

4.2.4 L'extraction du modèle géométrique direct

Suivant l'étude précédente, on peut poser :

$$^0T_4 = ^0T_1 *^1T_2 *^2T_3 *^3T_4 \tag{4.2.1}$$

Donc :

$$^0T_4 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & L_2 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_4 & -S_4 & 0 & L_3 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$
{}^{0}T_{4} = \begin{bmatrix}
C(t_2+t_3+t_4)C(t_1) & -S(t_2+t_3+t_4)C(t_1) & S(t_1) & C(t_1)(L_3C(t_2+t_3)+L_2C(t_2)) \\
C(t_2+t_3+t_4)S(t_1) & -S(t_2+t_3+t_4)S(t_1) & -C(t_1) & S(t_1)(L_3C(t_2+t_3)+L_2C(t_2)) \\
S(t_2+t_3+t_4) & C(t_2+t_3+t_4) & 0 & L_3S(t_2+t_3)+L_2S(t_2) \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

On note : $\theta_i = t_i$; $\cos(\theta) = C(\theta)$; $\sin(\theta) = S(\theta)$;

La position de l'effecteur final, par rapport au troisième élément, est donnée par :

$$
\begin{bmatrix} E_x \\ E_y \\ E_z \\ 1 \end{bmatrix} = {}^{0}T_{4} * \begin{bmatrix} L_4 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix}
C(t_1)(L_3C(t_2+t_3)+L_2C(t_2))+L_4C(t_2+t_3+t_4)C(t_1) \\
S(t_1)(L_3C(t_2+t_3)+L_2C(t_2))+L_4C(t_2+t_3+t_4)S(t_1) \\
L_3S(t_2+t_3)+L_2S(t_2)+L_4S(t_2+t_3+t_4) \\
1
\end{bmatrix}
$$

4.3 Le matériale utilisée (Partie hardware)

4.3.1 Bref définition hardware

Ensemble des objets, des instruments utilisés dans une usine, un service, une exploitation, etc. Ensemble d'éléments susceptibles d'être exploités, élaborés scientifiquement, d'être analysés (Le matériel d'une enquête).

4.3.2 Bras manipulateur

Ce robot (bras manipulateur) est réalisé par les deux étudiants (Mr. DILMI Ahmed et Mr. GOTIT Mahmoud) et dirigé par (M. TCHOKETCH KEBIR Sofiane), durant l'année universitaire 2015-2016, ils sont intéressés à des robots manipulateurs rigides à chaîne ouverte simple, ce robot est commandé visuellement par la caméra (Kinect) à base d'Arduino Uno.

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) conçu pour accomplir automatiquement des tâches imitant ou reproduisant, dans un domaine précis, des actions humaines. La conception de ces systèmes et l'objet d'une discipline scientifique, branche de l'automatisme nommé robotique.

Un bras manipulateur est le bras d'un robot généralement programmable, avec des fonctions similaires à un bras humain. Les liens de ce manipulateur sont reliés par des axes permettant, soit du mouvement de rotation (comme dans un robot articulé) ou de translation (linéaire) de déplacement.

Il peut être autonome ou contrêlé manuellement et peut être utilisé pour effectuer une variété de tâches avec une grande précision. Les bras manipulateurs peuvent être fixes ou mobiles (c'est-à-dire à roues) et peuvent être conçus pour des applications industrielles.

Les robots articulés sont très utilisés dans le domaine de l'industrie. Ils permettent de réaliser du soudage, de l'assemblage, de la peinture, l'usinage (perçage....) et découpage par laser.



FIGURE 4.4 – **Bras manipulateur**

4.3.3 Arduino Uno

L'Arduino est une carte électronique en Matériel Libre pour la création artistique interactive. Elle peut servir[12] :

— Pour des dispositifs interactifs autonomes simples.

— Comme interface entre capteurs/actionneurs et ordinateur.

— Comme programmateur de certains microcontrôleurs.

Arduino est une plate-forme de prototypage open source basée sur le microcontrôleur ATmega et Arduino uno est l'un des tableaux les plus utilisés dans lesquels la description détaillée est donnée [12].

Le tableau Arduino Uno possède 14 entrées/sorties de broches numériques, 6 entrées analogiques, une horloge 16 MHz, une connexion USB, une prise de courant et un bouton de réinitialisation. Généralement, il contient tout le nécessaire pour supporter le microcontrôleur [12].

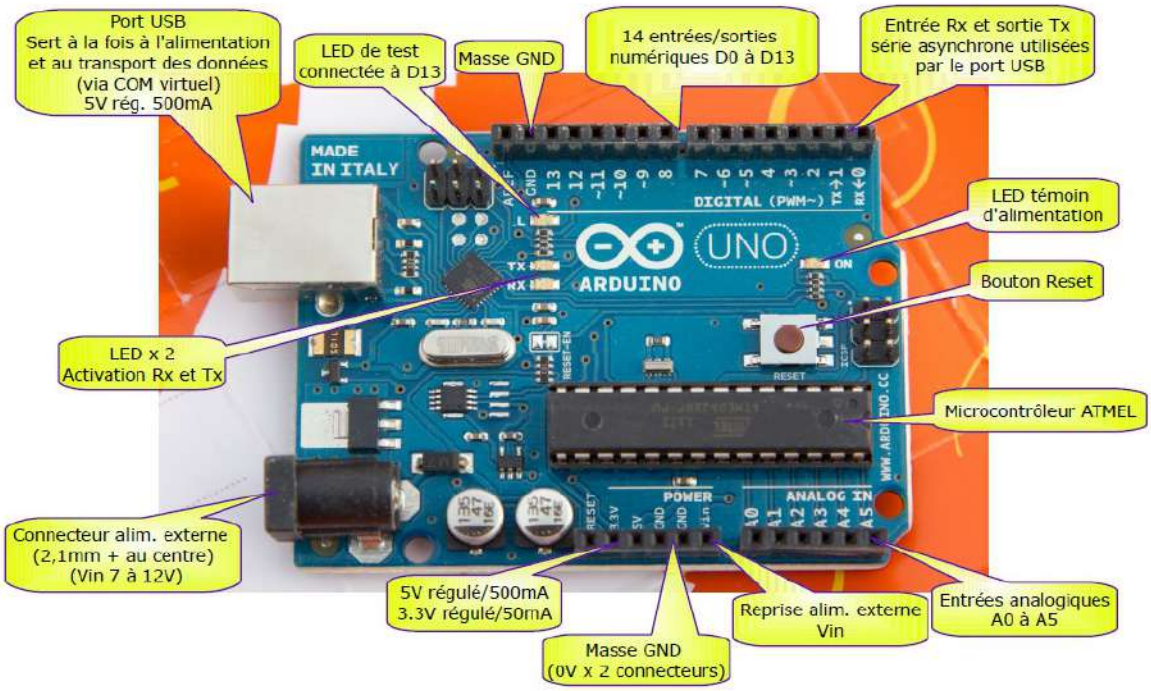


FIGURE 4.5 – **carte ARDUINO**

4.3.4 ArduinoWi-FiShield

Actuellement Arduino est introduit dans les domaines les plus différents pour différentes applications. Grâce à différents types de boucliers de cet appareil. Wi-Fi Shield permet à une carte Arduino de se connecter à Internet en utilisant les spécifications sans fil 802.11 [12].

Le bouclier Wi-Fi est connecté à la carte Arduino à l'aide d'en-têtes longs de câblage qui s'étendent à travers le bouclier. Cela maintient la disposition des broches intact et permet d'empiler un autre bouclier sur le dessus [12].

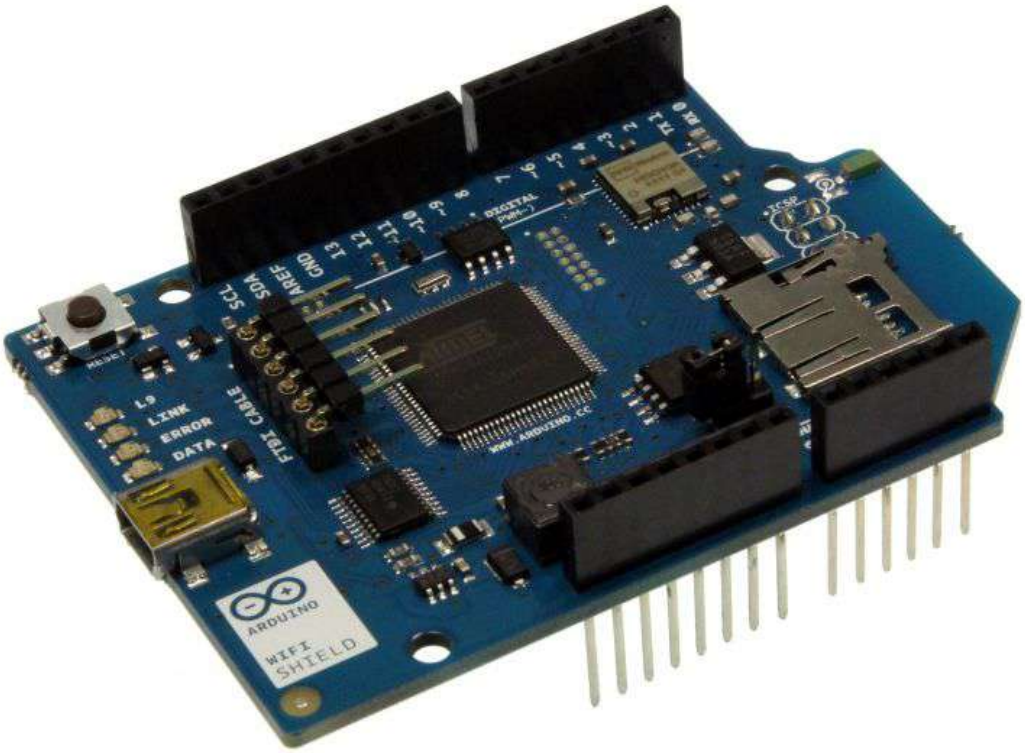


FIGURE 4.6 – **Carte ArduinoWi-FiShield**

4.3.5 Alimentation stabilisée (à partir d'une alimentation de PC)

Une alimentation stabilisée PC est un dispositif électronique, alimentation stabilisée de labo à partir d'une alimentation de PC. Elle pourra fournir des tensions de -12V, 3.3V, 5V et 12V, utilisant une régulation, permettant le réglage de la tension et du courant nécessaires au fonctionnement d'un appareil électrique ou électronique. Il est généralement conçu pour fonctionner en générateur idéal de tension et de courant.



FIGURE 4.7 – **Fonctions des connecteurs ATX 24 pins**

La relation entre la tension d'entrée V_e et celle de sortie V_s est la suivante : (V_s=$F_0 V_e R_0 I_s$). Avec Is l'intensité de sortie, F_0 le facteur de stabilisation en tension et R_0 la résistance interne de l'alimentation.

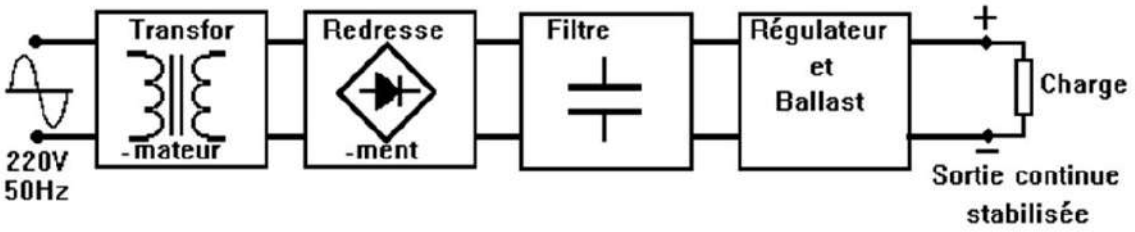


FIGURE 4.8 – **Le bloc diagramme d'une alimentation [12]**

Le transformateur remplit deux fonctions : isolement galvanique entre l'équipement et le secteur ; transformation de la tension alternative du réseau, déterminée par le rapport entre le nombre de spires (tours) du secondaire et du primaire.

Le redresseur convertit la tension alternative du secondaire du transformateur en impulsions unidirectionnelles de courant. On n'utilise plus aujourd'hui qu'un montage : le redressement

double alternances ou onde entière par pont de Graetz. Le pont de Graetz est constitué de quatre diodes disposées en losange. Il existe des ponts de diodes intégrés, c'est à dire que l'on a moulé les quatre diodes dans un seul boîtier. Ce bloc est plus facile à utiliser mais a un inconvénient. Lorsqu'une des diodes moulées est défectueuse, il faut changer l'ensemble.

4.3.6 Le Moteur à courant continu (M.C.C)

Cette partie vous présente le résumé sur les machines à courant continu.

Un moteur à courant continu est une machine qui transforme l'énergie électrique qui lui est appliquée en une énergie mécanique entraînant une charge mécanique.

— Le couple développé par un moteur à courant continu est proportionnel au courant d'induit et au flux magnétique.

— La vitesse de rotation est proportionnelle à la tension d'alimentation et inversement proportionnelle au flux magnétique.

— Le rendement est le rapport entre la puissance développée par le moteur ou puissance utile (puissance mécanique) et celle qui lui est appliquée ou puissance absorbée (puissance électrique).

Le moteur comprend :

— Un circuit magnétique comportant une partie fixe, le stator, une partie tournant, le rotor et l'entrefer l'espace entre les deux parties.

— Une source de champ magnétique nommée l'inducteur (le stator) crée par un bobinage oudes aimants permanents.

— Un circuit électrique induit (le rotor) subit les effets de ce champ magnétiques.

— Le collecteur et les balais permettent d'accéder au circuit électrique rotorique.

Pour commander un moteur à courant continu (M.C.C) , il y'a plusieurs montages ainsi que plusieurs types de commande. Et pour ce projet, on a utilisé Le transistor, il est commandé par une sortie de la carte Arduino via la résistance sur la base. Lorsque l'état de la sortie est au niveau 0, le transistor est bloqué et le courant ne le traverse pas. Le moteur ne tourne pas. Lorsque la sortie vaut 1, le transistor est commandé et devient saturé, c'est-à-dire qu'il laisse

passer le courant et le moteur se met à tourner.

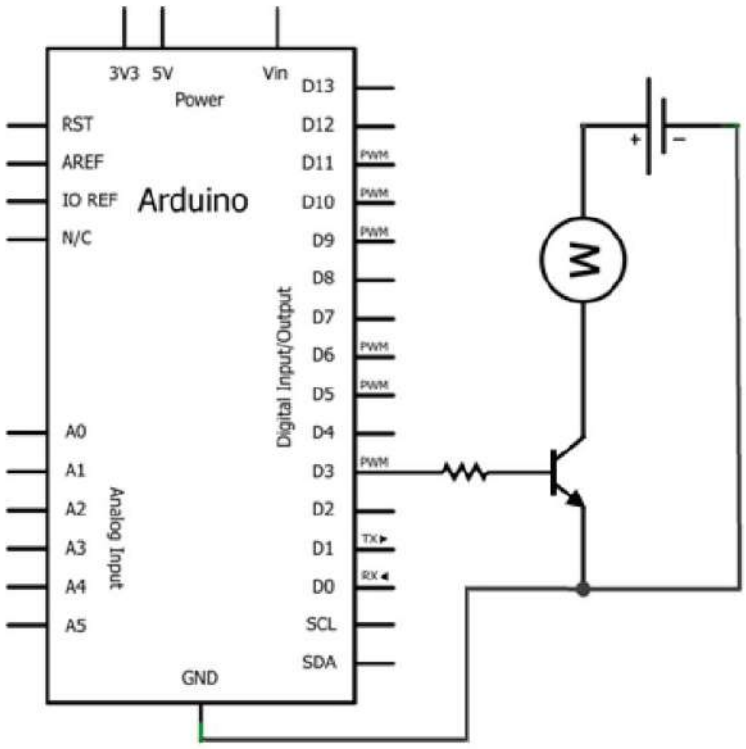


FIGURE 4.9 – **Circuit commande M.C.C à base de transistor**

4.3.7 Le Routeur Wi-Fi

Dans les réseaux informatiques, un point d'accès sans fil (WAP1 pour Wireless Access Point en anglais) est un dispositif qui permet aux périphériques sans fil de se connecter à un réseau câblé ou au réseau Internet à l'aide d'une connexion radio. Le point d'accès en tant que dispositif autonome est habituellement relié à un routeur (par l'intermédiaire d'un réseau câblé), mais il peut aussi faire partie intégrante du routeur lui-même.

Les point d'accès sans fil, les plus répandus, sont de type Wi-Fi (box ou hot-spot Wi-Fi).

Les points d'accès publics ou privés des FAI sont généralement compatibles avec les normes Wi-Fi IEEE 802.11n ou IEEE 802.11ac et offrent des débits supérieurs à 100 Mb/s.

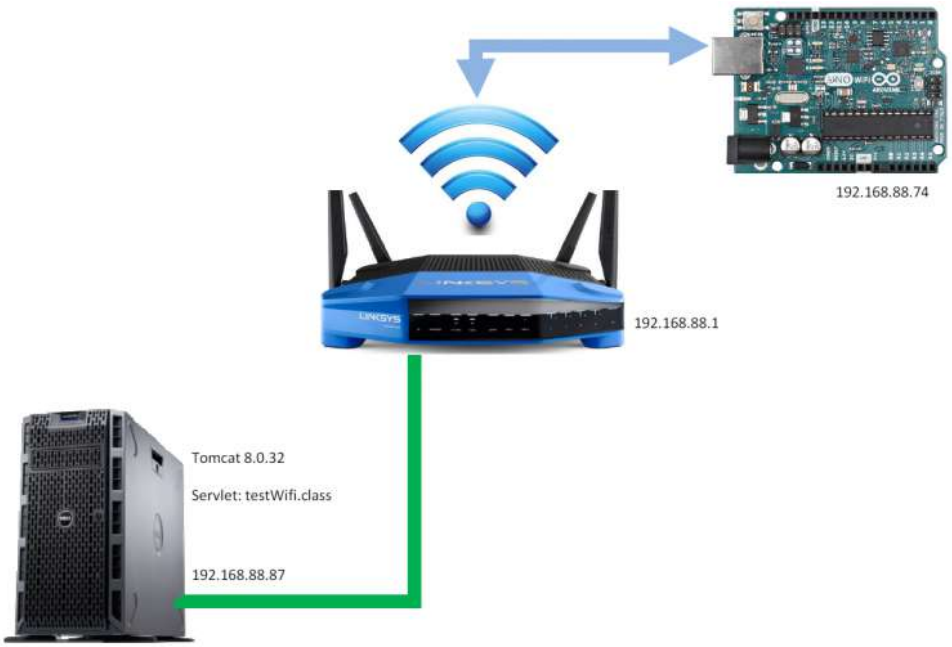


FIGURE 4.10 – **point d'accées sans fil**

4.4 Partie software

4.4.1 MATLAB

En utilisant l'outil de programmation MATLAB, est pour le but de commander des servo-moteur et moteur à courant continu à l'aide d'une commande (carte ARDUINO,...), le langage MATLAB, ou (MATRIX LABORATORY), est un langage de programmation et de développement de quatrième génération.

Développé par la société (The Math Works), MATLAB permet la manipulation de matrice, afficher des courbes et des données, mettre en oeuvre des algorithmes, créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran.

Les utilisateurs de MATLAB sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que pour la recherche. Au cours de notre projet, nous avons utilisé MATLAB R2015b qui est plus développé que les anciennes versions surtout dans le domaine de l'ingénierie.

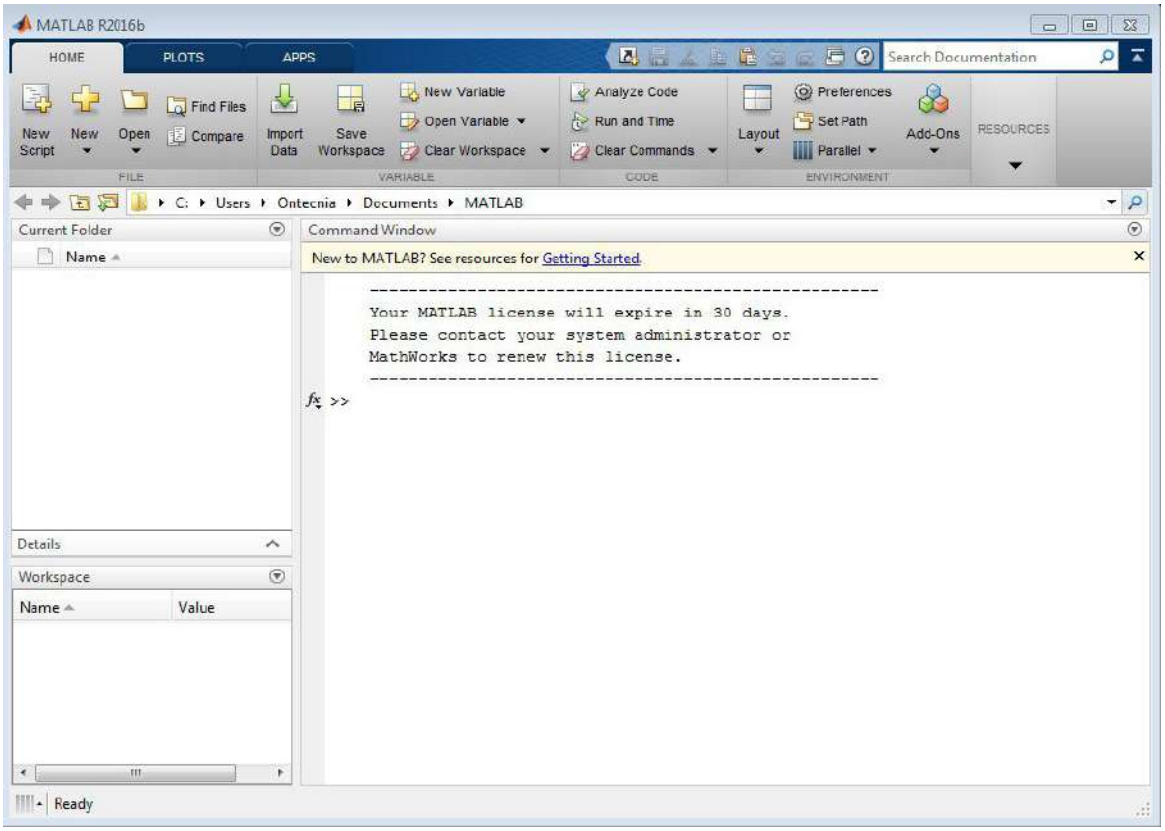


FIGURE 4.11 – **interface Matlab2016b**

4.4.2 Le langage de programmation ARDUINO

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme Arduino s'effectue de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres. Voyons plus en détail la structure d'un programme écrit en Arduino.

Il y a deux façons de programmer les cartes Arduino.

— Soit avec du code (pur) en ligne de code type langage C ;

— Soit sous forme de blocs type puzzle, nommé Ardublock.

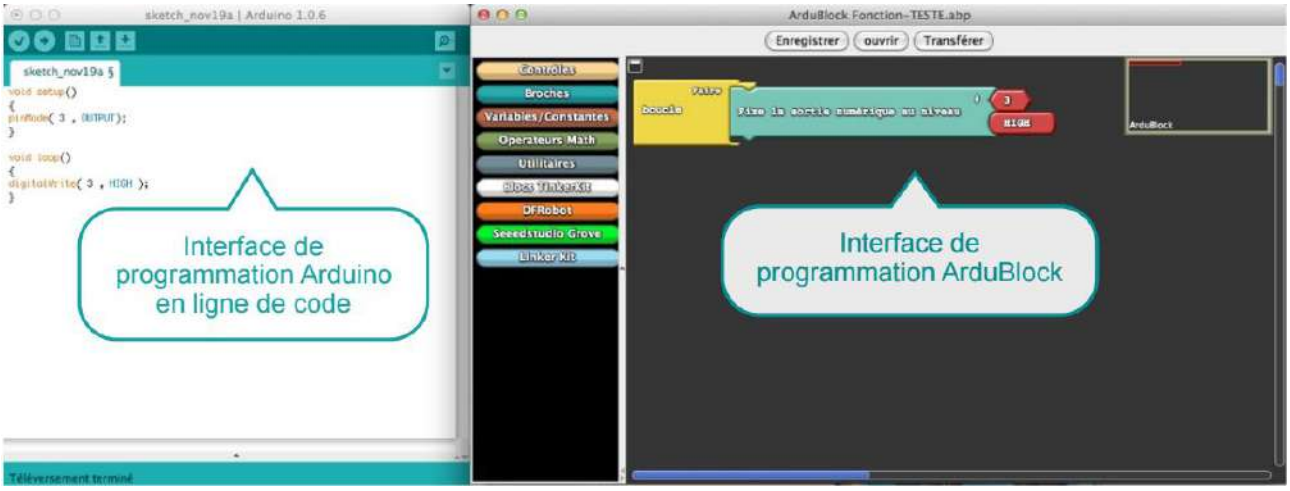


FIGURE 4.12 – **langage de programmation ARDUINO**

4.5 La Réalisation De La Commande

4.5.1 Méthodologie de l'approche proposée

Notre méthode consiste à générer et réaliser la commande d'un bras manipulateur, pour atteindre cet objectif, trois étapes sont nécessaires : simulation de l'objectif, génération et l'envoi de la commande via un réseau Wi-Fi, réception et exécution de commande.

La figure suivante illustre un schéma synoptique de l'approche proposée montrant les différents blocs constituant notre projet.

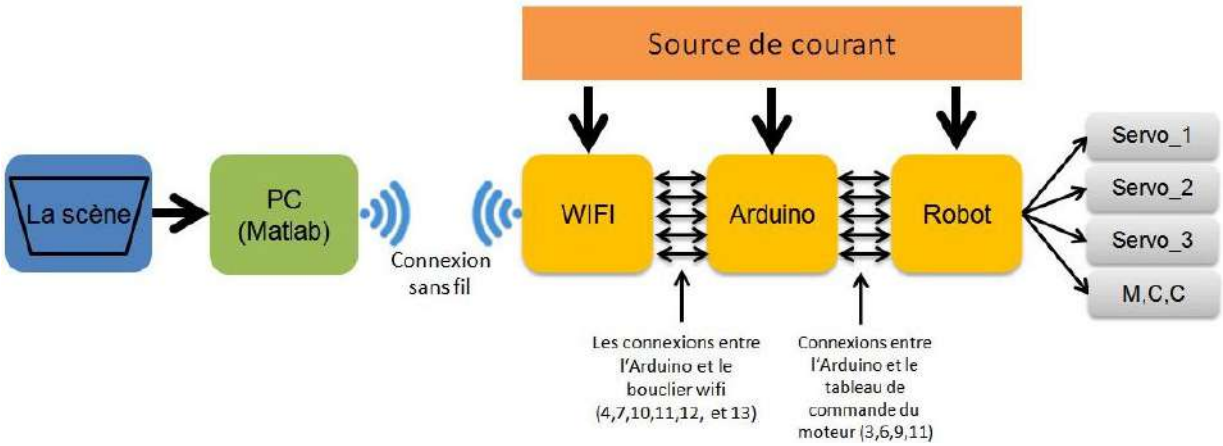


FIGURE 4.13 – **Synoptique du système conçu [12]**

4.5.2 Réalisation, Tests, Résultats et discussion

L'implémentation des déférentes parties (soft et hardes) avec certains niveaux d'adaptation (câblage) conduit vers un produit mécatronique représenté par la figure suivante :

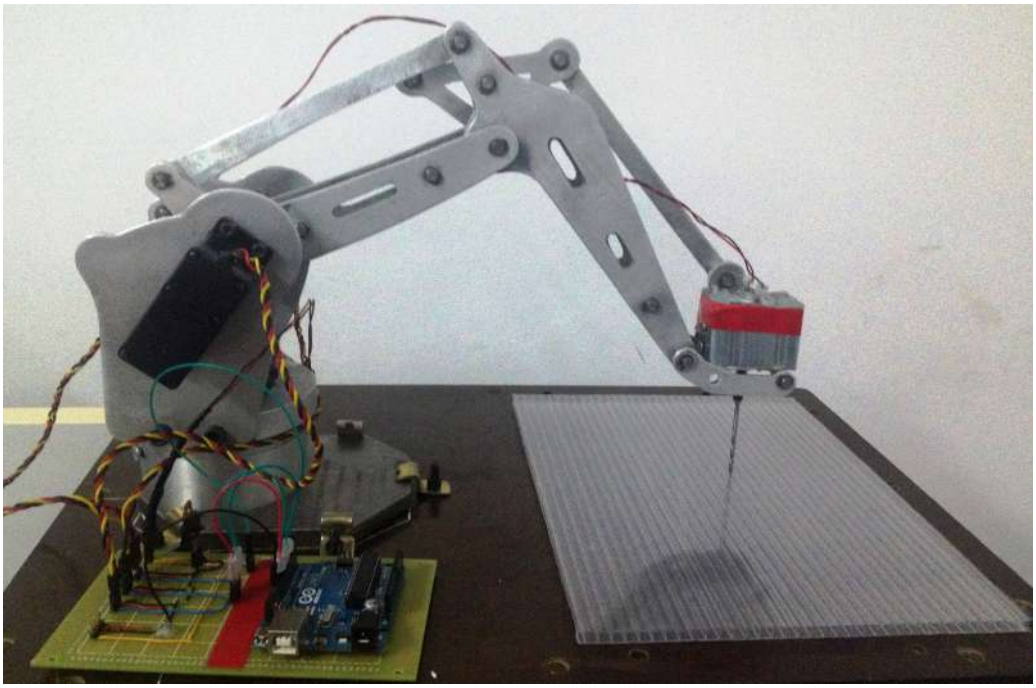


FIGURE 4.14 – **Représentation final de notre projet**

Ce produit capable de faire quelque tâche bien déterminer avec un certain degré d'exigence, de façons général, la liberté de notre bras dépond :

— De la géométrie du robot,

— De la longueur des segments,

— Du débattement des articulations (limité par des butées).

Exemples :

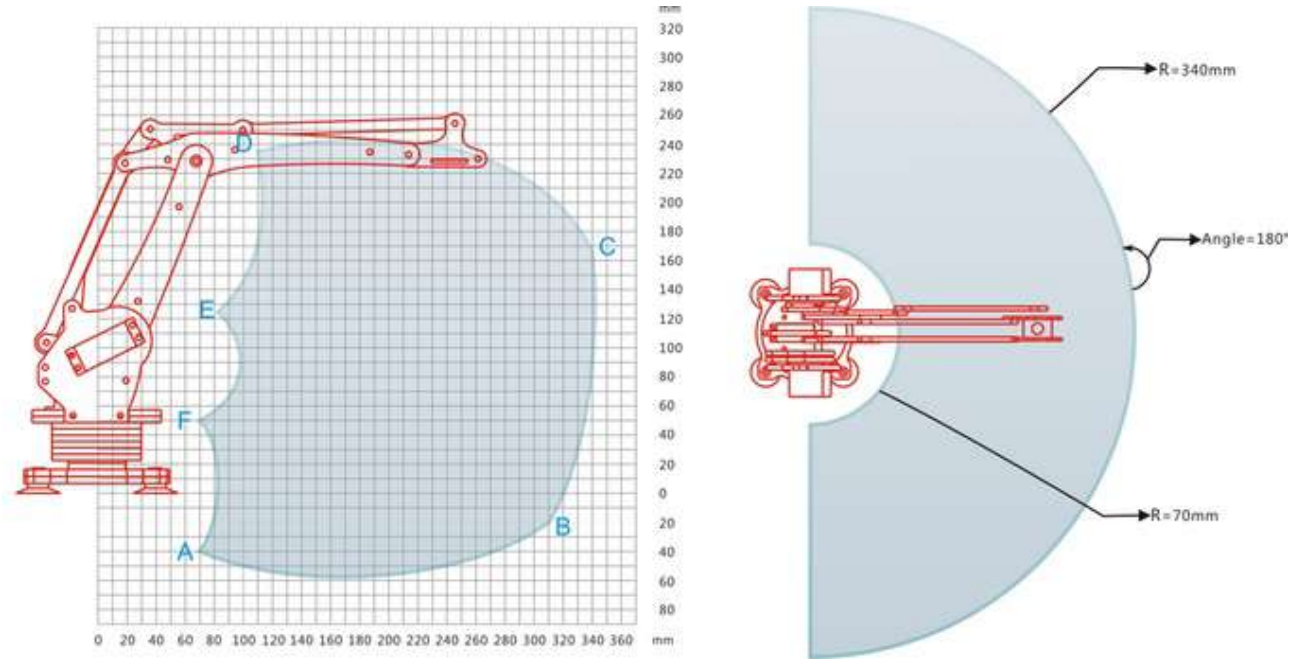


FIGURE 4.15 – **Espace de travail**

Tests, Résultats et discussion

La simulation de l'objectif fait une partie importante dans notre projet, cette simulation est traduite dans une interface Matlab voir la figure suivante :

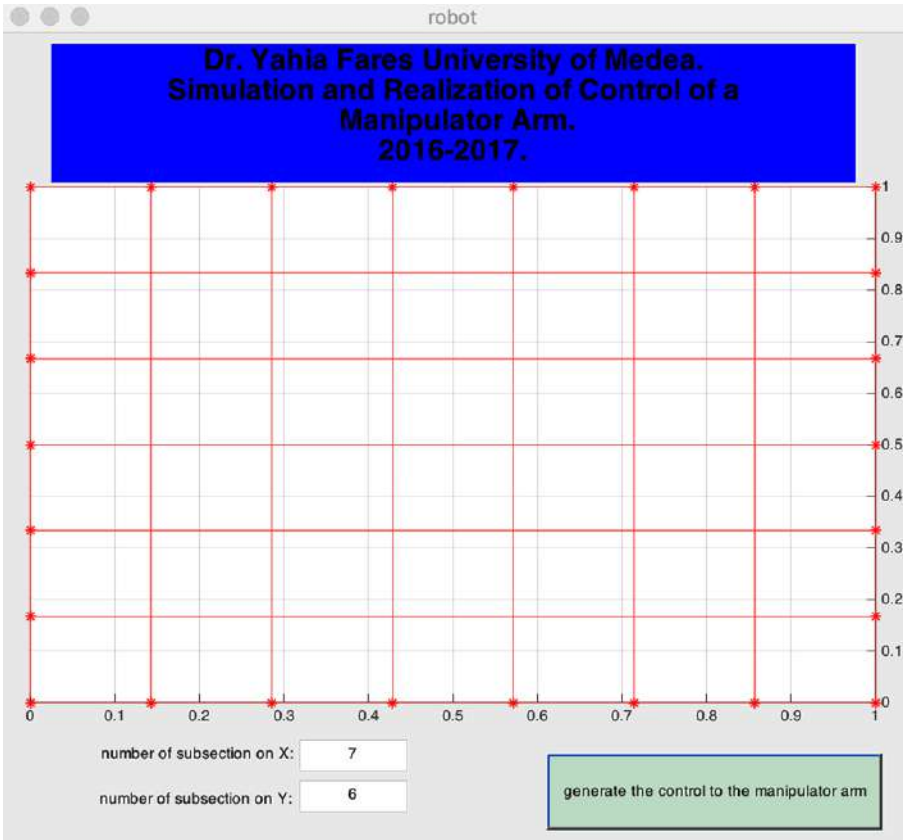


FIGURE 4.16 – **Interface de la génération de la commande de notre Robot**

Cette interface de simulation représente la scène réelle de l'exécution (perçage), une fois que l'utilisateur termine sa simulation, il peut générer l'exécution pour mettre cette simulation sur le plan réel.

L'excusions de dizaines commande simuler sur l'interface de Matlab est représenté par la figure suivante :

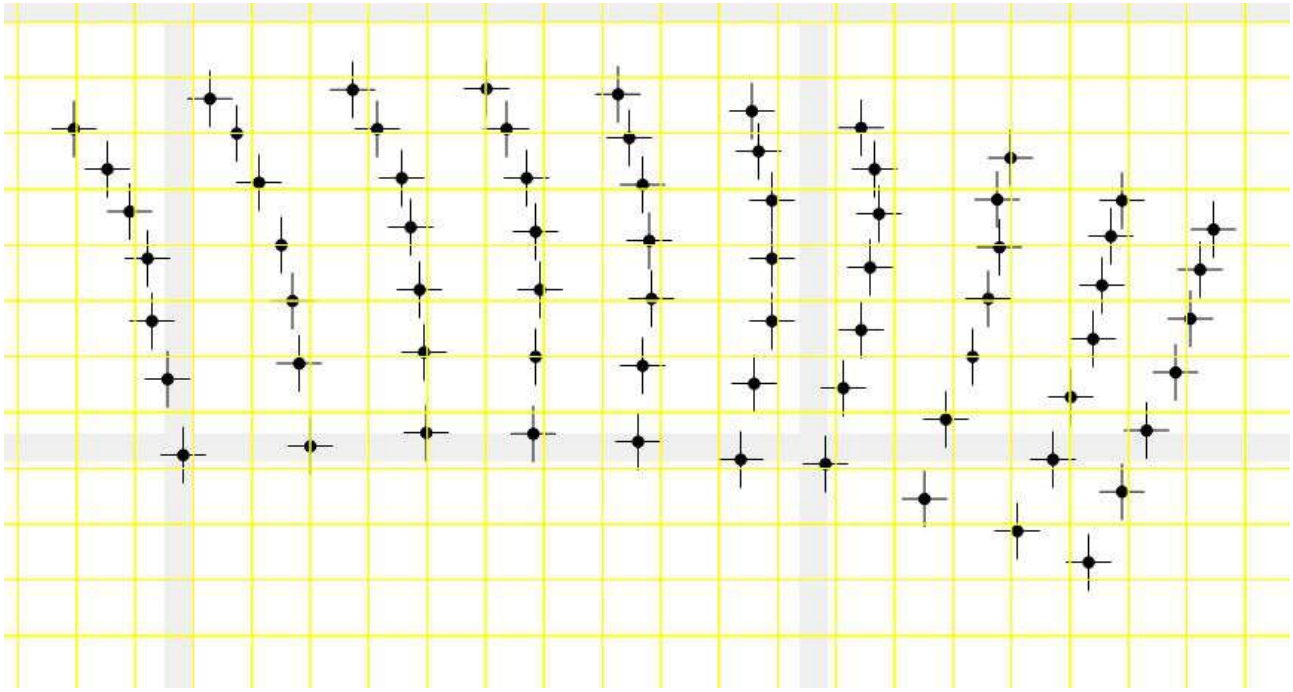


FIGURE 4.17 – **Point percée dans l'espace de travail**

Nous constatons que notre bras suit bien les cibles de perçage établis, et ça signifier que la commande par langage Matlab est réalisé comme nous avons prévu.

Les résultats du test permettant de conclure que le robot manipulateur suit les cibles, ce qui correspond au interface graphique du Matlab, avec une précision importante.

4.6 Conclusion

Nous avons décrit dans ce chapitre les étapes de notre projet, telle que les circuits utilisés ainsi que les schémas jusqu'à l'étude et l'implémentation des logiciels de simulation et programmation d'une part, d'autre part un programmation capable d'adapter à une commande électronique afin de construire un système Mécatronique qui répond à nos besoins.

Enfin on a pu faire des semples tests sur le bras malgré les difficultés qu'on a trouvées dans ce travail.

Conclusion général

Dans ce projet, nous avons apporté une valeur ajoutée en matière de performance d'un autre travail de recherche réalisé antérieurement à l'université de Médéa, dont le sujet est l'étude et réalisation d'un robot manipulateur.

Dans la première partie de ce travail une introduction aux différents modèles des robots utilisés suivie par une étude approfondi et minutieuse du projet réalisé à savoir le travail de base. Dans ce contexte nous avons identifié tous les aspects importants du travail ainsi que les nouvelles pistes qui pourront faire l'objet d'une recherche poussée, à l'instar de l'amélioration de la précision ciblée dans notre travail.

Une deuxième partie a été consacrée à la partie software dans le but de commander le bras manipulateur en utilisant deux logiciels différents : le logiciel Arduino et le logiciel Matlab. Ce dernier est utilisé pour réaliser un interface graphique qui nous a beaucoup aidés à la programmation en exploitant ces deux logiciels.

La programmation via le langage Arduino est une tâche très importante, elle nous a conduit à mieux maitriser le coté software de manière à ce qu'on puisse traduire les mouvements du bras manipulateur en instructions exécutables par le logiciel.

Afin d'atteindre notre objectif de ce projet, qui est l'amélioration de la précision, notre touche s'est articulé sur l'élaboration d'un espace de travail modélisé par Matlab et représenté par un interface graphique lié directement à la carte Arduino responsable de la commande du robot.

En effet ce projet de réalisation nous a permis de traiter des problèmes d'ordre pratique (problèmes mécaniques, programmation) et de vérifier des connaissances théoriques acquises tout au long de notre cursus universitaires. En d'autre partie ce projet nous a permet d'acquérir

une bonne expérience pratique et théorique dans le monde de la robotique.

L'analyse des travaux présentés dans ce mémoire nous permet d'envisager des perspectives ainsi que les pistes de recherche afin d'améliorer les performances de cette œuvre scientifique tel que :

— Introduire les techniques de l'intelligence artificielle dans la commande de ce robot dans le but d'avoir plus de performance.

— Introduire les différentes liaisons sans fil pour la commande.

— Introduire une liaison optique pour la commande.

— Malgré tous les efforts fournis ce bras nécessite quelques améliorations pour atteindre une précision dans les standards.

Annexe A

Fichier technique

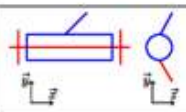
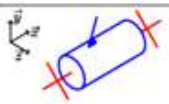
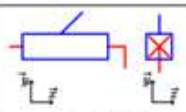
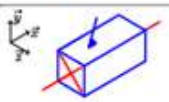
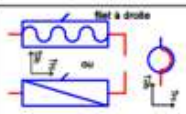
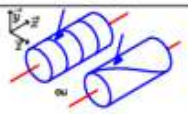
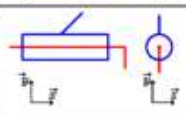
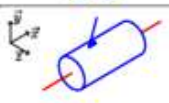
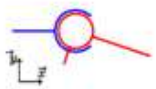
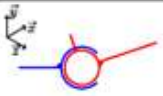
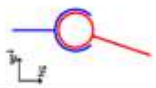
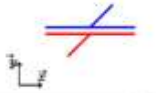
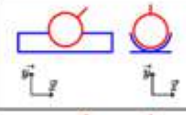
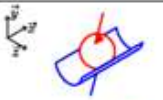
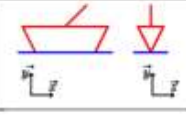
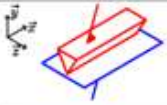
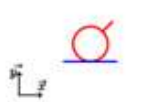
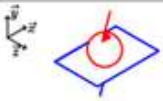
Nom	point d'application	ddl	Représentation plane	Représentation 3D	Torseur cinématique	Torseur des AM
Encastrement	tout point de l'espace	0			$\begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & M_x \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Pivot d'axe $(A, \vec{x})$	tout point de l'axe	1			$\begin{Bmatrix} \omega_x & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & 0 \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Glissière de direction $\vec{x}$	tout point de l'espace	1			$\begin{Bmatrix} 0 & V_x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & M_x \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Hélicoïdale d'axe $(A, \vec{x})$	tout point de l'axe	1			$\begin{Bmatrix} \omega_x & p.\omega_x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & p.F_x \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Pivot glissant d'axe $(A, \vec{x})$	tout point de l'axe	2			$\begin{Bmatrix} \omega_x & V_x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & 0 \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Rotule à doigt de centre A bloquée en $\vec{x}$	centre de la liaison	2			$\begin{Bmatrix} 0 & 0 \\ \omega_y & 0 \\ \omega_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & M_x \\ F_y & 0 \\ F_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$
Rotule de centre A	centre de la liaison	3			$\begin{Bmatrix} \omega_x & 0 \\ \omega_y & 0 \\ \omega_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & 0 \\ F_y & 0 \\ F_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$
Plane de normale $\vec{y}$	tout point de l'espace	3			$\begin{Bmatrix} 0 & V_x \\ \omega_y & 0 \\ 0 & V_z \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & M_x \\ F_y & 0 \\ 0 & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Annulaire en A d'axe $(A, \vec{x})$	au centre de la liaison	4			$\begin{Bmatrix} \omega_x & V_x \\ \omega_y & 0 \\ \omega_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & 0 \\ F_y & 0 \\ F_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$
Rectiligne de ligne $(A, \vec{x})$ et de normale $\vec{y}$	tout point du plan $(A, \vec{x}, \vec{y})$	4			$\begin{Bmatrix} \omega_x & V_x \\ \omega_y & 0 \\ 0 & V_z \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & 0 \\ F_y & 0 \\ 0 & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Ponctuelle en A de normale $\vec{y}$	tout point de la normale au contact	5			$\begin{Bmatrix} \omega_x & V_x \\ \omega_y & 0 \\ \omega_z & V_z \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & 0 \\ F_y & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$

FIGURE A.1 – **Les différentes représentations des liaisons mécaniques**

Annexe B

Programme arduino de commandé de notre robot

Exemple : chaîne d'envoi et de réception UDP WiFi

```
#include <SPI.h>
#include <WiFi.h>
#include <WiFiUdp.h>
#ifndef HAVE_HWSERIAL1
#include "SoftwareSerial.h"
SoftwareSerial Serial1(6, 7); // RX, TX
#endif
char ssid[] = "DJAWEB_15BF4";            // your network SSID (name)
char pass[] = "A2EF6F582D43C";        // your network password
int status = WL_IDLE_STATUS;     // the Wifi radio's status
unsigned int localPort = 10002;  // local port to listen on
char packetBuffer[255];          // buffer to hold incoming packet
char ReplyBuffer[] = "ACK";      // a string to send back
void setup() {
  // initialize serial for debugging
  Serial.begin(115200);
  // initialize serial for ESP module
  Serial1.begin(9600);
```

```
  // initialize ESP module
//  WiFi.init( Serial1);
  // check for the presence of the shield:
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    // don't continue:
    while (true);
  }
  // attempt to connect to WiFi network
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network
    status = WiFi.begin(ssid, pass);
  }
    Serial.println("Connected to wifi");
  printWifiStatus();
  Serial.println(" \ nStarting connection to server...");
  // if you get a connection, report back via serial:
  Udp.begin(localPort);
    Serial.print("Listening on port ");
  Serial.println(localPort);
}
void loop() {
  // if there's data available, read a packet
  int packetSize = Udp.parsePacket();
  if (packetSize) {
    Serial.print("Received packet of size ");
```

```
    Serial.println(packetSize);
    Serial.print("From ");
    IPAddress remoteIp = Udp.remoteIP();
    Serial.print(remoteIp);
    Serial.print(", port ");
    Serial.println(Udp.remotePort());
    // read the packet into packetBufffer
    int len = Udp.read(packetBuffer, 255);
    if (len > 0) {
      packetBuffer[len] = 0;
    }
    Serial.println("Contents:");
    Serial.println(packetBuffer);

    // send a reply, to the IP address and port that sent us the packet we received
    Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
    Udp.write(ReplyBuffer);
    Udp.endPacket();
  }
}
void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  // print your WiFi shield's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
```

```
  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
}
```

Bibliographie

[1] F. Z. BOUZIANE, "Rétro-conception du bras horizontal de robot manipulateur de la cellule flexible (tlemcen)," Ph.D. dissertation, 2014.

[2] M. Priel, *Les Robots industriels : caracteristiques performances et choix.* AFNOR, 1990.

[3] A. D. Kahina, A. Z. Meziane, and A. Hocine, "Conception et réalisation d'un bras manipulateur commandé par api."

[4] R. SAADI and N. SALHI, "Réalisation de carte à microcontrôleur pour le contrôle de bras manipulateur via un pc," Master's thesis, 2014.

[5] A. Liégeois, "Modélisation et commande des robots manipulateurs," *Techniques de l'ingénieur. Informatique industrielle*, vol. 6, no. S7730, pp. 1–16, 2000.

[6] N. Nombrail, "Analyse et commande d'une famille de robots manipulateurs à structure parallele et redondante," Ph.D. dissertation, 1993.

[7] M. Ait-Ahmed, "Contribution à la modélisation géométrique et dynamique des robots," Ph.D. dissertation, 1993.

[8] J. Qin, "Commande hybride position/force robuste d'un robot manipulateur utilisé en usinageet/ou en soudage," Ph.D. dissertation, Paris, ENSAM, 2013.

[9] W. Khalil and E. Dombre, *Modélisation, identification et commande des robots.* Hermès science publ., 1999.

[10] planete sciences, *Le grand livre d'Arduino.* Editions Eyrolles, 2015.

[11] planete sciences.org, *Commander des servomoteurs classiques et a rotation continue*, 2017.

[12] S. T. Kebir, M. Bouhedda, S. Mekaoui, M. Guesmi, and A. Douakh, "Gesture control of mobile robot based arduino microcontroller," in *Modelling, Identification and Control (ICMIC), 2016 8th International Conference on.* IEEE, 2016, pp. 1081–1085.

ملخص:

ذراع المناورة هو روبوت قابل للبرمجة عموما، مع وظائف مماثلة لذراع الإنسان. ترتبط وصلات هذا الذراع المناور بواسطة محاور مما يجعل من الممكن لجعل الحركات التناوب أو متعدية. والهدف الرئيسي من هذه الأطروحة هو محاكاة وتحقيق السيطرة على ذراع مناور مع درجة معينة من المتطلبات والدقة.

كلمات مفتاحية: ذراع مناور، اردوينو، واي فاي شيلد، النمذجة والسيطرة على الروبوتات، ...

Résumé :

Un bras manipulateur est un robot généralement programmable, avec des fonctions similaires à un bras humain. Les liens de ce bras manipulateur sont reliés par des axes permettant de faire des mouvements de rotation ou de translation. L'objectif principal de ce mémoire est de simuler et réaliser commande d'un bras manipulateur avec un certain degré d'exigence et de précision.

Mots clés : bras manipulateur, Arduino, Wi-Fi-Shield, modélisation et commande des robots, ...

Abstract :

A manipulator arm is a generally programmable robot, with functions similar to a human arm. The links of this manipulator arm are connected by axes making it possible to make rotational or translational movements. The main objective of this thesis is to simulate and realize control of a manipulator arm with a certain degree of requirement and precision.

Key words: arm manipulator, Arduino, Wi-Fi-Shield, modeling and control of robots, ...

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

جامعة الدكتور يحي فارس بالمدية

Université Yahia Farès de Médéa

FACULTÉ DE LA TECHNOLOGIE

DÉPARTEMENT DE GÉNIE MÉCANIQUE

Projet de fin d'études présenté pour l'obtention du déplôme de :

MASTER EN GÉNIE MÉCANIQUE

Spécialité : MÉCATRONIQUE

Théme :

Etude, simulation et réalisation de la commande d'un bras manipulateur

Sous la direction de :

M. TCHOKETCH KEBIR Sofiane.

M. MAAZOUZ Mohamed.

Préparé par :

LADEK MOUSSA.

ALILI AHMED.

ANNÉE UNIVERSITAIRE 2016-2017

Dédicaces

Louanges à ALLAH
le tout puissant de m'avoir donnée la
force et le courage pour accomplir ce modeste travail.
J'ai l'honneur de dédier ce modeste travail à : A mes très chers parents, ma chere épouse et notre petit chérubin Amir Abd Errazak qui m'ont soutenu et encouragé durant toute la période de mes études et à qui je souhaite une longue et heureuse vie, que dieu les protège et me donne la force pour que je puisse leur rendre un petit peu de leur bienfait et leur bonté. A mes très chers frères et mes très chères sœurs. A toute les personnes de ma famille. La famille LADEK et La famille HAMDOUCHE
Je remercie mon ami et binôme Alili Ahmed.Ainsi que tous mes autres amis, HAMADI Ahmed, LAGHOUATI Youcef, FEKIRINI Ahemd et AKAK Smain et de prés ou de loin ceux et celles que je n'ait pas cité. A toute ma promotion de Mécatronique Et enfin,
à tous ceux qui m'aiment
et mes futures
épreuves
♡

Moussa

Dédicaces

Louanges à ALLAH le tout puissant de m'avoir donnée la force et le courage pour accomplir ce modeste travail. J'ai l'honneur de dédier ce modeste travail à : A mes très chers parents, ma chere épouse et nos trois petites adourables filles ROMAISSA, RITEDJ et RIMES NOURE ELHOUDA qui m'ont soutenu et encouragé durant toute la période de mes études et à qui je souhaite une longue et heureuse vie, que dieu les protège et me donne la force pour que je puisse leur rendre un petit peu de leur bienfait et leur bonté. A mes très chers frères et mes très chères soeurs. A toute les personnes de ma famille. La famille ALILI et La famille MOUKAH Je remercie mon ami et binôme LADEK Moussa.Ainsi que tous mes autres amis, HAKIM KEROUANI et de prés ou de loin ceux et celles que je n'ait pas cité. A toute ma promotion de Mécatronique Et enfin à la brise lointaine de mon cœur

♡

Ahmed

Remerciements

Nous remercions notre Dieu, le tout puissant, de nous avoir donné la santé et la volonté pour compléter ce modeste travail.

Ainsi que l'ensemble des membres du jury qui ont examiné cette thèse pour leurs observations enrichissantes et leurs conseils. Et plus spécialement Mr. TCHOKETCH KEBIR Sofiane et M. MAAZOUZ Mohamed qui ont acceptés d'être les promoteurs de notre thèse.

Nous remercions vivement Mr. BOUMAHDI, Mr. BACHEN, Mr. ABOUDERA, Mr. YAHIA.

Nous tenons à exprimer notre profonde gratitude au Mr. DILMI Ahmed et Mr. GOTIT Mahmoud qui nous ont permis d'apporter certaines améliorations au travail qu'ils ont produits précédemment.

Résumé

Table des matières

Introduction général 1

1 Des Généralités sur les Robots. 3

1.1 Introduction 4

1.2 Définition générale 4

1.2.1 Le robot industriel 4

1.2.2 La robotique 4

1.3 L'histoire de la robotique industrielle 5

1.4 Les Types des Robots 6

1.4.1 Robots mobiles 6

1.4.2 Robots manipulateurs 7

1.5 Les éléments constitutifs d'un robot 8

1.5.1 Unité opérationnelle 9

1.5.2 Unité informationnelle 9

1.6 Structure mécanique articulée 9

1.6.1 Structure mécanique articulée á chaîne cinématique simple 10

1.6.2 Structure mécanique articulée á chaîne cinématique fermée 10

1.7 Architecture des robots 11

1.7.1 La base 11

1.7.2 Le porteur 12

1.8 Classification des robots 14

1.8.1 Classification fonctionnelle 14

1.8.2 Classification géométrique 17

1.9 Les Domaines d'applications 20

1.9.1 Les Tâches simples . . . 20
1.9.2 Les Tâches complexes . . . 20
1.10 Conclusion . . . 21

2 La Modélisation des robots manipulateurs 22

2.1 Introduction . . . 23
2.2 Structure mécanique . . . 23
2.3 Les différentes méthodes de modélisation . . . 24
2.3.1 Modèle géométrique direct . . . 24
2.3.2 Modèle géométrique inverse . . . 26
2.3.3 Le model cinématique direct . . . 28
2.3.4 Le model cinématique inverse . . . 30
2.3.5 La Modélisation dynamique . . . 30
2.4 Conclusion . . . 31

3 Arduino et Servomoteur 32

3.1 Introduction . . . 33
3.2 Arduino . . . 33
3.2.1 Définition . . . 33
3.2.2 Microcontrôleur . . . 33
3.2.3 Les différentes cartes Arduino . . . 34
3.2.4 La carte Arduino Uno . . . 35
3.2.5 Les modes de communication . . . 37
3.2.6 L'alimentation électrique . . . 38
3.2.7 Valeurs de courant ou de tension . . . 40
3.2.8 La Programmation de l'Arduino . . . 40
3.3 Les Servomoteurs . . . 42
3.3.1 Définition . . . 42
3.3.2 Le Principe de l'asservissement d'un Servomoteur . . . 43
3.3.3 Le rôle des servomoteurs . . . 45
3.3.4 Fonctionnement d'un servomoteur standard . . . 45

3.3.5 Les composants d'un servomoteur 46
3.3.6 La théorie de la commande des servomoteurs 46
3.3.7 l'implimentation de la commande théorique 47
3.3.8 Les servomoteurs digital et analogique 47
3.3.9 Le pilotage des servomoteurs 48
3.4 Conclusion 49

4 La Réalisation de la Commande d'un Bras Manipulateur, Résultats et Discussions 51

4.1 Introduction 52
4.2 Simulation, Repérage et paramétrage du bras manipulateur plan de type 3R dans le plans 52
4.2.1 Le Repérage 52
4.2.2 Le Paramétrage 54
4.2.3 L'etablissement des relations géométriques 55
4.2.4 L'extraction du modèle géométrique direct 55
4.3 Le matériale utilisée (Partie hardware) 56
4.3.1 Bref définition hardware 56
4.3.2 Bras manipulateur 56
4.3.3 Arduino Uno 58
4.3.4 ArduinoWi-FiShield 59
4.3.5 Alimentation stabilisée (à partir d'une alimentation de PC) 60
4.3.6 Le Moteur à courant continu (M.C.C) 61
4.3.7 Le Routeur Wi-Fi 62
4.4 Partie software 63
4.4.1 MATLAB 63
4.4.2 Le langage de programmation ARDUINO 64
4.5 La Réalisation De La Commande 65
4.5.1 Méthodologie de l'approche proposée 65

4.5.2 Réalisation, Tests, Résultats et discussion 66
4.6 Conclusion 68
Conclusion général 70

A Fichier technique **72**

B Programme arduino de commandé de notre robot **73**

Bibliographie 78

Table des figures

1.1 Robotique industrielle 5

1.2 Spirit 7

1.3 Robot fanuc 8

1.4 Structure fonctionnelle d'un robot 8

1.5 La Structure sérielle 10

1.6 La Structure fermée 11

1.7 Architecture d'un robot 11

1.8 Représentation d'une articulation rotoîde 12

1.9 Représentation d'une articulation prismatique 13

1.10 Manipulateur á commande manuelle 15

1.11 Manipulateur à cycle préréglé 15

1.12 Robot programmable 16

1.13 Robot intelligent 17

1.14 Robot cartésien 17

1.15 Robot cylindrique 18

1.16 Robot sphérique 18

1.17 Robot SCARA 19

1.18 Robot 3R 20

2.1 Systèmes de coordonnées et les paramètres de Denavit - Hardenberg 25

2.2 Transformations entre l'organe terminal et le repère atelier 28

2.3 Exemple d'un robot plan à trois degrés de liberté 29

3.1 Microcontrôleur ATmega328 34

3.2 La carte à microcontrôleur 35
3.3 les composants de la carte Arduino 36
3.4 Entré et Sortie numérique 38
3.5 Alimentation et entrée Analogiques 38
3.6 Alimentation de la carte Arduino 39
3.7 les différentes prises 40
3.8 Coloration Syntaxique dans IDE de l'Arduino 41
3.9 les différents types de servomoteur 42
3.10 Les Trois Files 43
3.11 Le Principe de l'asservissement d'un Servomoteur 43
3.12 Les composants d'un servomoteur 46
3.13 Chronogramme de commande de Servomoteur 47
3.14 Présentation d'un servomoteur avec Arduino 48
3.15 Exemple de piloter des servomoteurs avec l'Arduino 49

4.1 Chaîne cinématique d'un bras manipulateur série 52
4.2 Paramètres de Denavit-Hartenberg modifiés 53
4.3 Repérage de notre mécanisme 54
4.4 Bras manipulateur 57
4.5 carte ARDUINO 58
4.6 ArduinoWi-FiShield 59
4.7 Fonctions des connecteurs ATX 24 pins 60
4.8 Le bloc diagramme d'une alimentation 60
4.9 Circuit commande M.C.C à base de transistor 62
4.10 point d'accées sans fil 63
4.11 interface Matlab2016b 64
4.12 langage de programmation ARDUINO 65
4.13 Synoptique du système conçu 65
4.14 Représentation final de notre projet 66

4.15 Espace de travail 67

4.16 Interface de la génération de la commande de notre Robot 67

4.17 Espace de travail 68

A.1 liaisons mécaniques 72

Liste des tableaux

3.1 Valeurs de courant ou de tension 40
3.2 Les connaissances nécessaires au pilotage d'un servomoteur 49

4.1 Paramétrage de notre mécanisme 54

Liste des abréviations

Abréviations	**Définitions**
(ddl)	Degrés de liberté
(ddm)	Degrés de mobilité
(DHM)	Denavit-Hartenberg Modifiée
(MCD)	Modèle Cinématique Direct
(MCI)	Modèle Cinématique Inverse
(MDD)	Modèle Dynamique Direct
(MDI)	Modèle Dynamique Inverse
(MGD)	Modèle Géométrique Direct
(MGI)	Modèle Géométrique Inverse
(MCC)	Moteur à Courant Continu
(MLI)	Modulation de largeur d'impulsion
(PWM)	Pulse Width Modulation
(IEEE)	Institute of Electrical and Electronics Engineers
(WI-FI)	Wireless Internet for Frequent Interface

Introduction général

Il est incontestable de dire que les robots sont devenus des outils impératifs dans le paysage de fabrication. La robotique a connu d'énormes pas dans le sens de l'évolution pour être à la fois plus précise et pratique.

La robotique se substitue à l'homme ou prolonge son action en apportant précision, rapidité ou capacité à appliquer d'importants efforts. Elle est généralement constituée d'un bras manipulateur plus ou moins anthropomorphe mis en mouvement par des actionneurs électriques, hydrauliques ou pneumatiques et dont la position est repérée par une série de capteurs.

Cependant, le problème des performances est l'une des raisons qui empêchent la robotisation de certains procédés industriels. Par ailleurs, en production industrielle, il est très difficile de changer les éléments mécaniques ou le système de commande du robot.

Notre travail est la continuité d'un travail qui a été élaborée durant l'année 2015-2016 par un groupe d'étudiants appartenant à l'université de Médéa. Spécialité : MÉCATRONIQUE.

Le travail consiste à donner plus de performance au projet d'une commande d'un bras manipulateur en matière de précision.

Notre projet est une amélioration sur les volets réalisation :

— Nous avons introduit une nouvelle carte électronique qui assemble les parties commande et puissances des servomoteurs et un moteur à courant continu.

— Introduire un nouvel organe terminal, à savoir une perceuse fixée sur le bras manipulateur.

— Un autre une plate-forme qui sert comme support de fixation et un espace de travail.

Sur le volet programmation :

— Sur ce volet, nous avons introduit un nouvel algorithme qui a pour tâche d'identification

l'espace de travail du robot. Cet à l'aide de MATLAB ; un interface graphique a été réalisé pour faciliter de la commande la tache de la commande aux opérateurs.

Ce mémoire est organisé de la manière suivante :

— Dans le premier chapitre, en commençant par une généralité sur les robots, (les différents types des robots existants dans l'industrie).

— Le deuxième chapitre présentera avec plus d'approfondissement les méthodes les plus importantes pour la modélisation des robots, et nous enchainons ici à l'étude de notre réalisation par la détermination des équations qui gèrent les mouvements de notre robot.

— Le troisième chapitre traite la présentation de la carte ARDUINO avec les servomoteurs.

— Enfin le contenu du quatrième chapitre présente les performances et tests obtenus avec les applications qu'on a réalisées.

— Nous terminons ce mémoire par une conclusion et quelques perspectives.

Chapitre 1

Des Généralités sur les Robots.

1.1 Introduction

L'utilisation des systèmes robotiques apparait aujourd'hui dans plusieurs domaines d'activités la médecine, la défense, la recherche scientifique etc. Les robots sont utilisés de manière privilégiée pour des missions où les objectifs sont quantifiables et clairement définis. Ils sont destinés à faciliter les tâches pour l'homme et à amplifier le rendement .

Dans ce premier chapitre, on va donner un aperçu non exhaustif sur les robots, un bref historique sur l'évolution de la robotique industrielle et présenter les différents types de robots et les éléments constitutifs de ces derniers.

1.2 Définition générale

1.2.1 Le robot industriel

Le terme robot a été introduit pour la premiére fois par l'auteur Tchéque Capek en 1920 dans sa pièce de théâtre R.U.R. (Rossum's Universal Robot) et il est dérivé du mot robota qui signifie travailleur (de force)[1].

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) accomplissant automatiquement des tâches diverses. C'est une machine intelligente fonctionnelle qui nécessite une autonomie de mouvements.

L'Organisation Internationale de Normalisation définit le robot comme étant un manipulateur á plusieurs degrés de liberté, á commande automatique, reprogrammable, multi-applications. mobile ou non, destiné à être utilisé dans les applications d'automatisation industrielle [2].

1.2.2 La robotique

La robotique est une science qui s'intéresse aux robots. En fait, il s'agit d'un ensemble de disciplines techniques (mécanique, électronique, automatique, informatique) articulées autour d'un objectif et d'un objet communs. Cet objectif est l'automatisation flexible de nombreux secteurs de l'activité humaine réputés jusqu'á très récemment comme ne pouvant se passer de la présence de l'homme, et l'objet est le robot, sorte de machine universelle dont l'homme rêve

depuis toujours pour le remplacer dans les tâches difficiles.

1.3 L'histoire de la robotique industrielle

La robotique industrielle a connu un essor entre 1950-1970. Elle a vu le jour en 1954 lorsque Georges DEVOL a pu réaliser son brevet sur la robotique. Dans ce brevet Devol a conçu un robot qu'il a intitulé Unimate.

En 1961, le premier Unimate fut utilisé dans les usines de GENERAL MOTORS.

En 1966 , l'entreprise Unimation continue de développer des robots et élaborent notamment des robots permettant de faire d'autres tâches, comme des robots de manipulation matérielle ou encore des robots conçus pour la soudure ou pour d'autres applications de ce genre.

En 1978 un nouveau robot est conçu par Unimation Inc avec l'aide de General Motors. Ensemble ils conçurent le robot PUMA 500. Le robot PUMA (Programmable Universal Machine for Assembly) a été conçu par Vic Schienman et fut financé par General Motors et par The Massachussets Institute of Technology au milieu des années 70. Le système de ce robot est composé d'un bras manipulateur permettant d'assembler des composants industriels et de son ordinateur de commande. Ce robot est le robot d'assemblage le plus rependu dans l'industrie des années 70.



FIGURE 1.1 – **Robotique industrielle,(unimate, puma : années 1950-1970)**

En 1985, Reymond Clavel a imaginé le Robot Delta qui possède un bras de manipulation

formé de 3 parallélogrammes. Son brevet tombe dans le domaine public en 2007 et différents constructeurs devraient alors sortir leur propre robot delta.

Le Jet Propulsion Laboratory (JPL) développe un robot industriel hexapode (á 6 pattes) du nom de Lemur. Lemur aura pour mission de monter, assembler et réparer des installations spatiales. Pesant moins de 5 kg, il offre la possibilité innovante d'adapter différents outils sur chacun de ses membres.

Selon l'étude robotique de la Fédération Internationale de Robotique (IFR) en 2012, il y a au moins 1 153 000 robots industriels opérationnels fin 2011 dans le monde.

Avec l'apparition de la robotique industrielle, les robots étaient conçus pour remplacer les ouvriers dans les tâches pénibles, répétitives ou dangereuses (peinture, soudure etc...). Aujourd'hui avec le développement de l'électronique, de l'informatique, de la mécanique et aussi de l'automatique, la technologie robotique a progressé. La recherche dans le domaine de la robotique est dirigée vers le développement de robots dévoués â des tâches bien différentes que celles demandées par l'industrie. Par exemple des robots travaillant en mode automatique ou semi-automatique et qui ont souvent pour objectif d'interagir avec des humains et de les aider dans leurs tâches (surveillance, manutention d'objets lourds etc...). Ils sont dotés d'une intelligence qui leur donne une certaine autonomie.

Ainsi donc, le développement important de l'intelligence artificielle et de la robotique font que de nouveaux robots apparaissent constamment et l'utilisation de systèmes robotiques apparait aujourd'hui dans plusieurs domaines d'activité : la médecine, la défense, la recherche etc....

1.4 Les Types des Robots

Il existe deux types de robots : robots mobiles et robots manipulateurs.

1.4.1 Robots mobiles

Ce sont des robots capables de se déplacer dans un environnement comme le montre la figure I.2. Ils sont équipés ou non de manipulateurs suivant leur utilisation, (les robots explorateurs,

les robots de services etc...).



FIGURE 1.2 – **Spirit,(NASA,2003 sur Mars)**

1.4.2 Robots manipulateurs

Des robots ancrés physiquement à leur place de travail et généralement mis en place pour réaliser une tâche précise répétitive, (tels que les robots industriels, médicaux etc...).

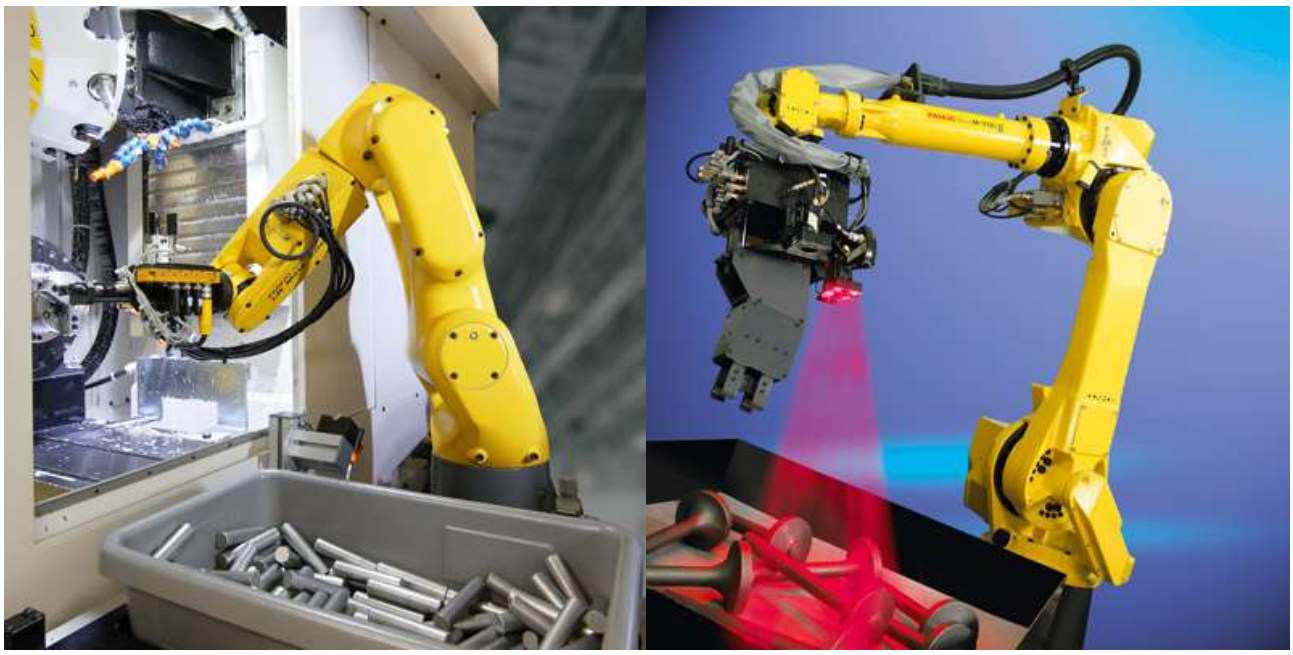


FIGURE 1.3 – **Robot fanuc**

1.5 Les éléments constitutifs d'un robot

Dans un robot en fonctionnement, on peut distinguer plusieurs ensembles interactifs (figure 1.4).

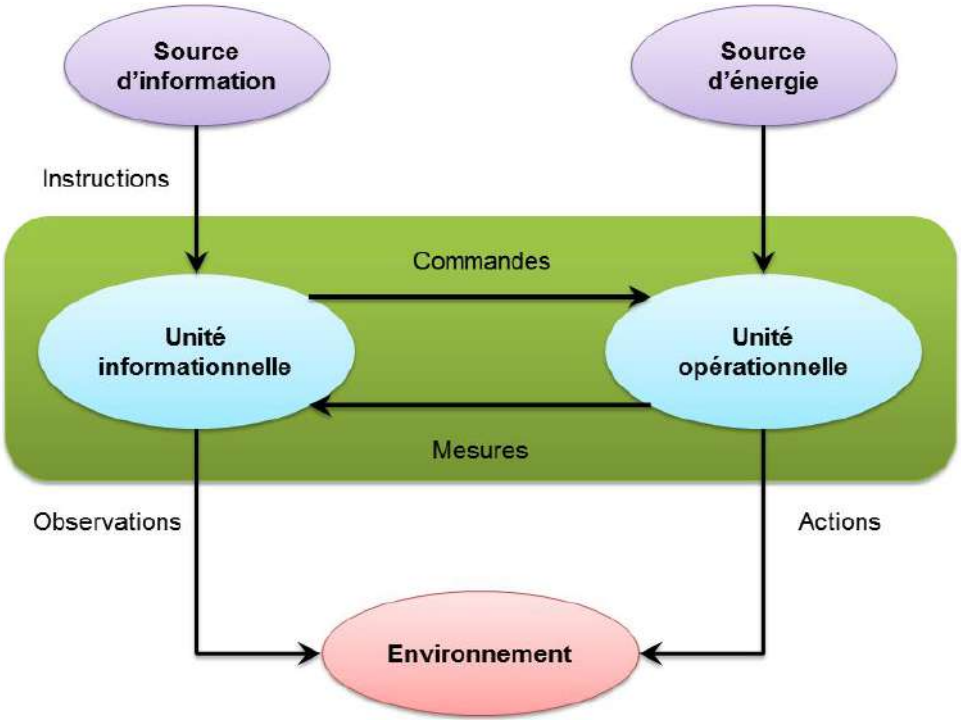


FIGURE 1.4 – **Structure fonctionnelle d'un robot**

1.5.1 Unité opérationnelle

Exerce les actions commandées en empruntant la puissance nécessaire à la source d'énergie. Cette partie, qui constitue le robot physique, intègre la structure mécanique (segments, articulations, architecture, etc...), les modules d'énergie (amplificateurs, variateurs, servovalves etc...), les convertisseurs d'énergie (moteurs, vérins etc...), les chaînes cinématiques de transmission mécanique (réducteurs, vis á billes, courroies crantées etc...), les capteurs de proprioceptifs placés sur chaque axe pour mesurer en permanence leur position et leur vitesse, et enfin l'effecteur, ou organe terminal, qui est en interaction avec l'environnement [3].

1.5.2 Unité informationnelle

Reçoit les instructions décrivant la tâche à accomplir, les mesures relatives à l'état interne de la structure mécanique qui constitue le bras manipulateur et les observations concernant son environnement. Elle élabore en conséquence les commandes de ses différentes articulations en vue de l'exécution de ses tâches. Les systèmes actuels fonctionnent en interaction permanente selon le cycle information-décision-action [4].

1.6 Structure mécanique articulée

Un robot manipulateur est constitué généralement par deux sous-ensembles distincts : un organe terminal qui est le dispositif destiné à manipuler des objets, et une structure mécanique articulée (SMA), constituée d'un ensemble de solides reliés entre eux, généralement les uns á la suite des autres où chaque solide est mobile par rapport au précédent. Cette mobilité s'exprime en termes de degrés de liberté (d.d.l) qui est par définition le nombre de mouvements indépendants possibles d'un solide par rapport au solide qui lui est directement relié.

Une structure mécanique articulée peut être représentée par une architecture composée de plusieurs chaînes de corps rigides assemblés par des liaisons appelées articulations. Les chaînes peuvent être dites soit ouvertes ou en série dans les quelles tous les corps ont au plus deux liaisons, ou bien arborescentes oú au moins l'un des corps a plus de deux liaisons. Les chaînes

peuvent aussi être fermées dans les quelles l'organe terminal est relié à la base du mécanisme par l'intermédiaire de plusieurs chaînes [3].

1.6.1 Structure mécanique articulée á chaîne cinématique simple

C'est une chaîne cinématique dont chaque membre possède un degré de connexion (nombre de liaisons mécaniques) inférieur ou égal á deux. Un robot sériel est formé d'une chaîne cinématique simple dont la base et l'organe effecteur possèdent un degré de connexion de un (c'est-à-dire qu'il n'est relié qu'à un seul corps) et les autres éléments un degré de connexion de deux.

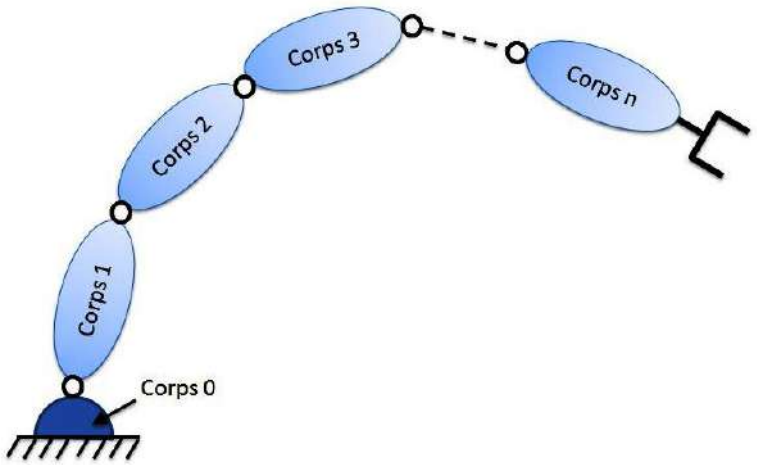


FIGURE 1.5 – **La Structure sérielle**

1.6.2 Structure mécanique articulée á chaîne cinématique fermée

C'est une chaîne cinématique qu'il existe un retour mécanique d'un ou plusieurs segments à un autre dans la chaîne.

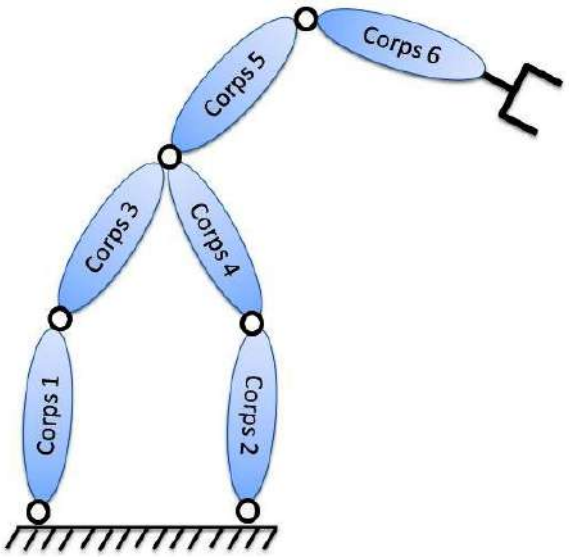


FIGURE 1.6 – **La Structure fermée**

1.7 Architecture des robots

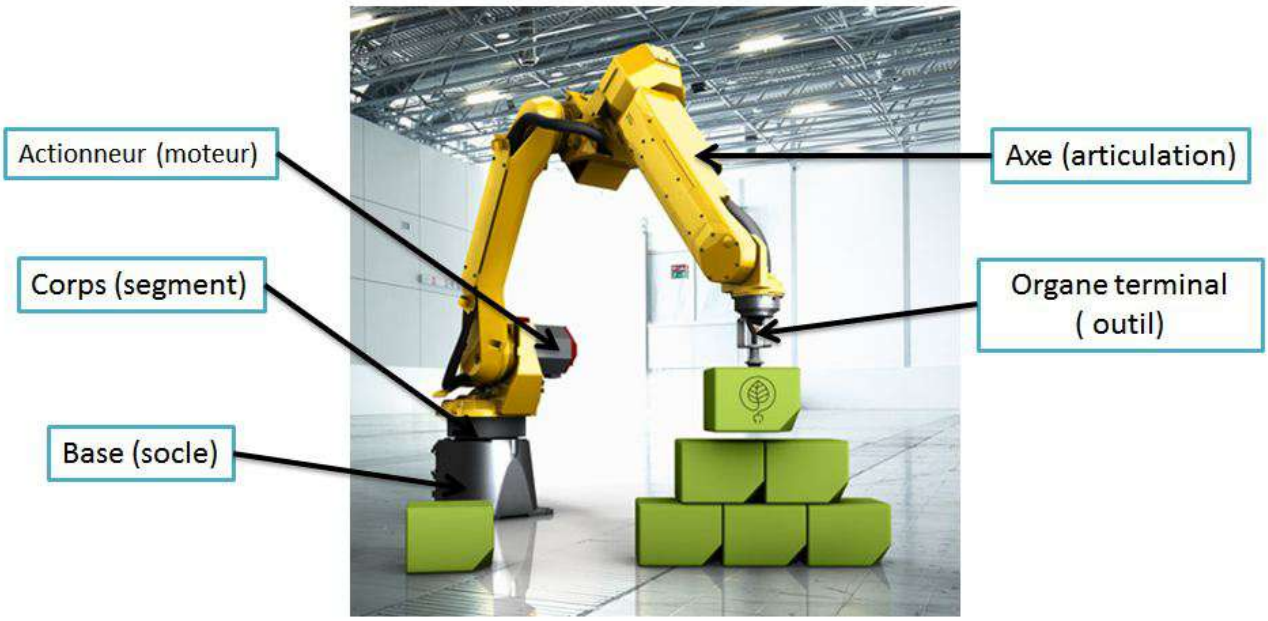


FIGURE 1.7 – **Architecture d'un robot,(Robot fanuc)**

1.7.1 La base

La base du manipulateur est fixée sur le lieu du travail. Ceci est le cas de la quasitotalité des robots industriels.

1.7.2 Le porteur

Le porteur représente l'essentiel du système mécanique articulé (segment, articulation, actionneur, l'organe terminal), il a pour rôle d'amener l'organe terminal dans une situation donnée imposée par la tâche. Il est constitué de :

a. Segment :

Corps solides rigides susceptibles d'être en mouvement par rapport à la base du porteur, et les uns par rapport aux autres,

b. Articulation

Une articulation lie deux corps successifs en limitant le nombre de degré de liberté, de l'un par rapport à l'autre :

- Articulation rotoîde :

Il s'agit d'une articulation de type pivot, notée R, réduisant le mouvement entre deux corps à une rotation autour d'un axe commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe (Voir la figure ci-dessus).

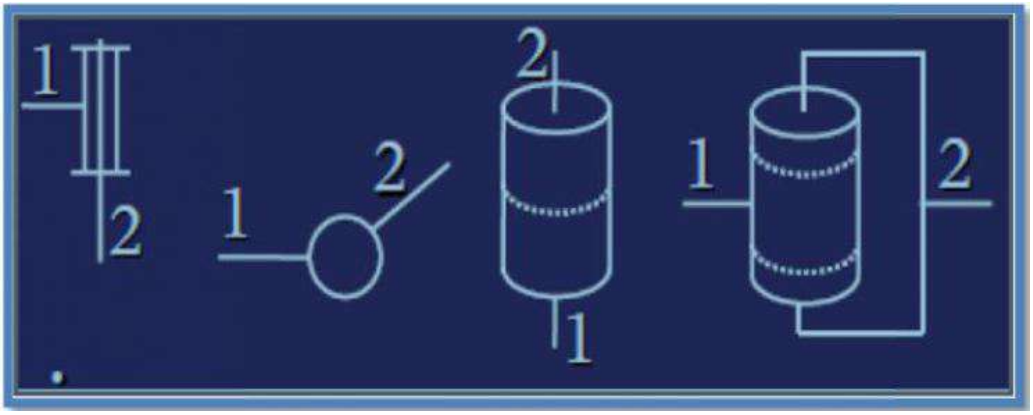


FIGURE 1.8 – **Représentation d'une articulation rotoîde**

- Articulation prismatique :

Il s'agit d'une articulation de type glissière, notée P, réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe (Voir la figure ci-dessus).

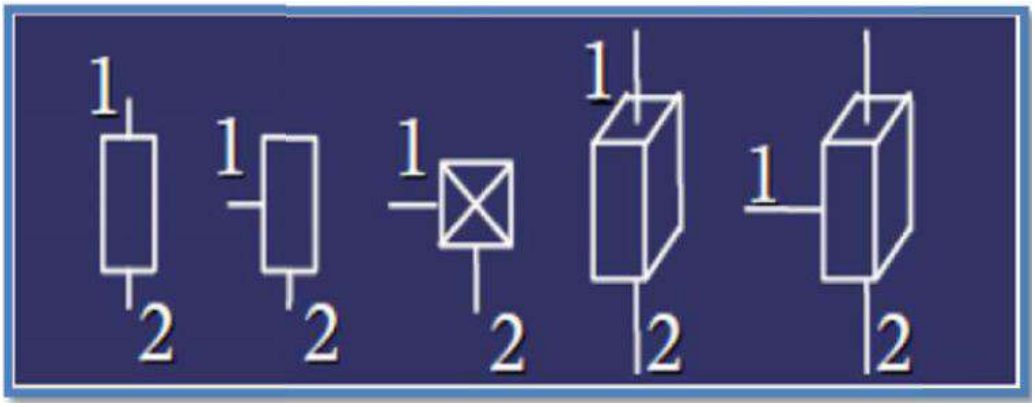


FIGURE 1.9 – **Représentation d'une articulation prismatique**

c. L'actionneur

Pour être animé, la structure mécanique articulée comporte des moteurs le plus souvent associés à des transmissions (courroies crantées),l'ensemble constitue les actionneurs. Les actionneurs utilisent fréquemment des moteurs électriques à aimant permanent, à courant continu, à commande par l'induit. On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas.

Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique). (Les actionneurs pneumatiques sont d'un usage général pour les manipulateurs á cycles (robots tout ou rien). Un manipulateur à cycles est une structure mécanique articulée avec un nombre limité de degrés de liberté permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air) [3].

d. L'organe terminal

On regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, ...), ou á les transformer (outils, torche de soudage, pistolet de peinture, ...). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, au

sens oú il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. Un robot, enfin, peut-être multi-bras, chacun des bras portant un organe terminal différent. On utilisera indifféremment le terme organe terminal, préhenseur, outil ou effecteur pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique, exemple : pistolet pour la soudure dans les robots industriels [3].

1.8 Classification des robots

On peut classer les robots d'un point de vue fonctionnel ou d'après leur structure géométrique.

1.8.1 Classification fonctionnelle

Le nombre de classe et les distinctions entre celles-ci varient de pays à pays (6 classes au Japon, 4 en France). L'A.F.R.I. distingue 4 classes illustrées ci-dessous :

1.8.1.1 Manipulateur à commande manuelle

La Figure I.10 représente les manipulateurs à commande manuelle :



FIGURE 1.10 – **Manipulateur á commande manuelle**

1.8.1.2. Manipulateur automatique

La figure montre un bras manipulateur qui exerce des mouvements de soudure sans l'intervention de l'homme.

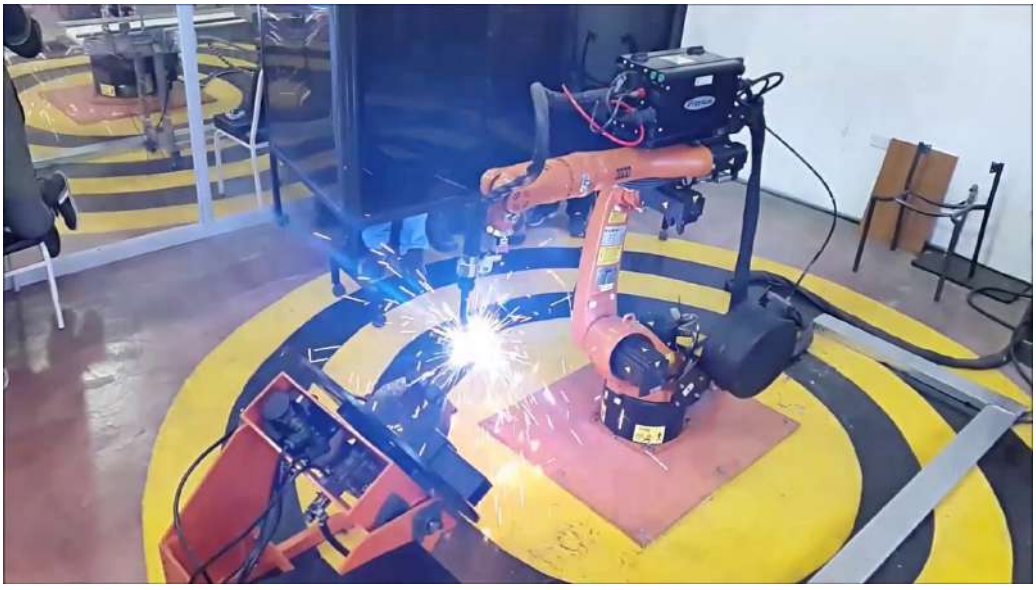


FIGURE 1.11 – **Manipulateur à cycle préréglé**

I.8.1.3. Robots programmables

Ils répètent les mouvements qu'on leur a appris ou programmés sans informations sur l'environnement ou la tâche effectuée. On peut aussi faire la distinction entre robots (playback) qui reproduisent la tâche apprise et robots à commande numérique qui peuvent être programmés hors-ligne.

Pour certains robots, par exemple les robots de peinture, qui doivent suivre une trajectoire complexe qu'il est difficile d'exprimer mathématiquement, un opérateur humain spécialiste de la tâche effectue la trajectoire en guidant le bras du robot à l'aide d'un (pantin), et l'entièreté de la trajectoire est mémorisée par le robot et la figure I.12 représente les robots programmables [3].

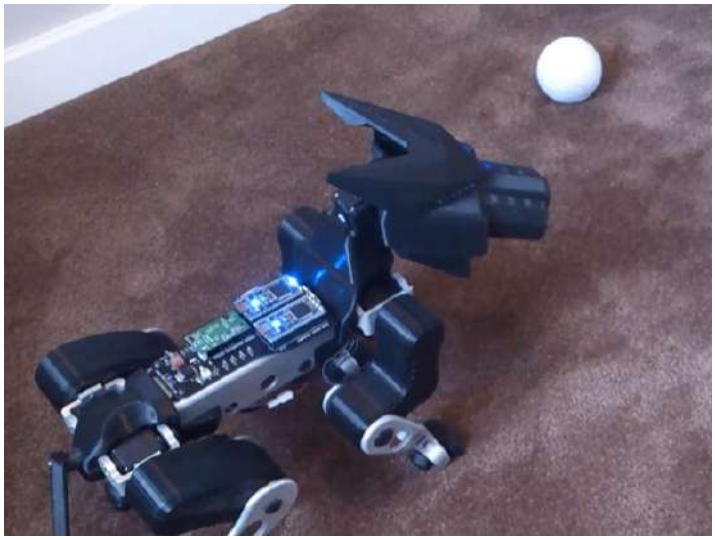


FIGURE 1.12 – **Robot programmable**

I.8.1.4. Robots intelligents

On trouve actuellement des robots de seconde génération qui sont capables d'acquérir et d'utiliser certaines informations sur leur environnement (systèmes de vision, détecteurs de proximité, capteurs d'efforts,...) comme le montre la Figure I.13. Les robots de troisième génération sont capables de comprendre un langage oral proche du langage naturel et de se débrouiller de façon autonome dans un environnement complexe grâce á l'utilisation de l'intelligence artificielle [3].



FIGURE 1.13 – **Robot intelligent**

1.8.2 Classification géométrique

On peut aussi classer les robots suivant leur configuration géométrique, autrement dit l'architecture de leur porteur.

- La structure cartésienne (PPP)

C'est une structure à trois liaisons prismatiques et est la plus ancienne. Historiquement, elle découle logiquement de la conception traditionnelle d'une machine-outil à trois axes, type rectifieuse ou fraiseuse par exemple. Cette structure est relativement peu utilisée sauf dans quelques applications particulières telles que robots pratiques, robots de magasinage [3].

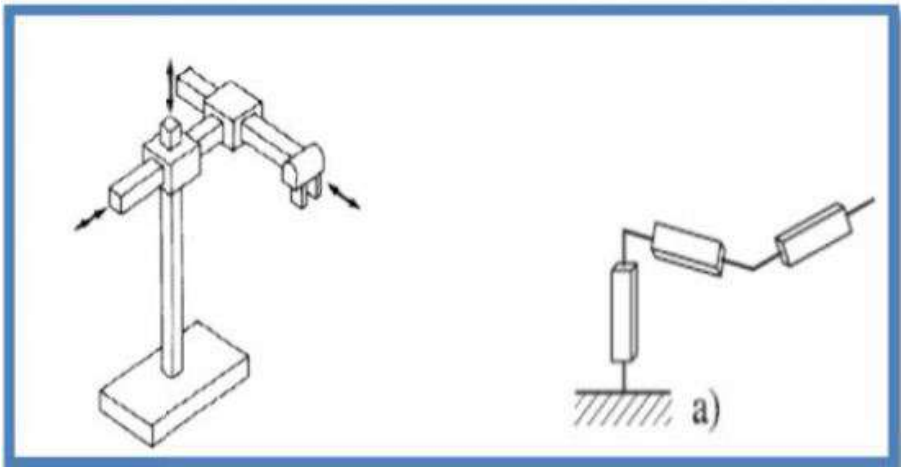


FIGURE 1.14 – **Robot cartésien**

- La structure cylindrique (RPP) ou (PRP)

Cette structure associe une rotation et deux translations. Elle présente l'inconvénient d'offrir un volume de travail faible devant un encombrement total important. Elle n'est pratiquement plus utilisée [3].

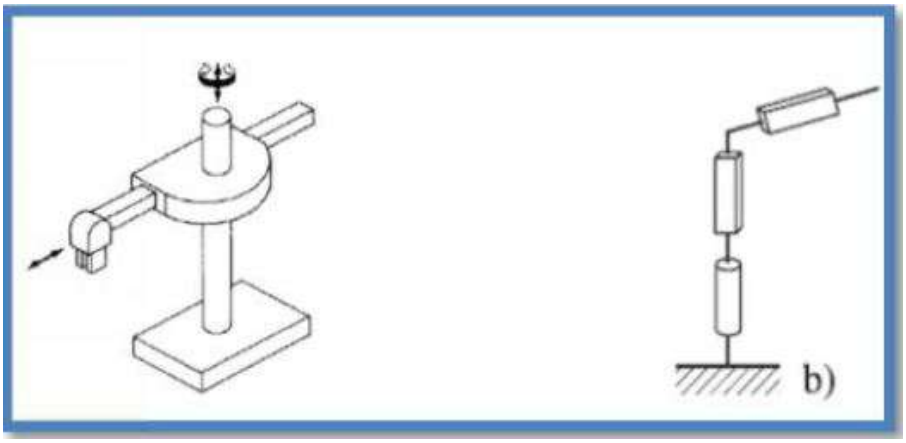


FIGURE 1.15 – **Robot cylindrique**

- La structure sphérique ou polaire

C'est une structure quasiment abandonnée pour des raisons similaires à l'abandon de la structure cylindrique.

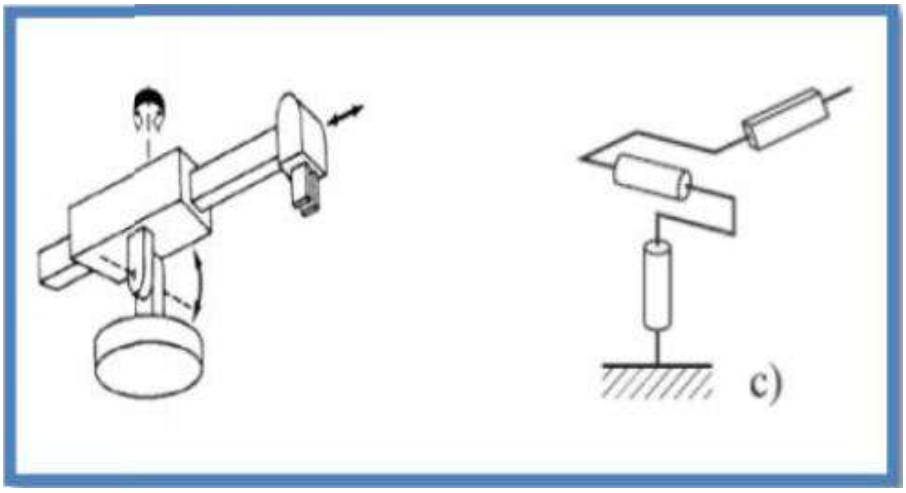


FIGURE 1.16 – **Robot sphérique**

- La structure dite SCARA (RRP)

A axes de rotation parallèles, elle est l'une des plus utilisées en particulier pour des tâches de manutention ou d'assemblages très fréquents dans l'industrie. Ce succès commercial est lié au fait que le ratio entre le volume de travail et l'encombrement est très favorable et aussi au fait que la structure SCARA est très adaptée à ce type de tâches [3].

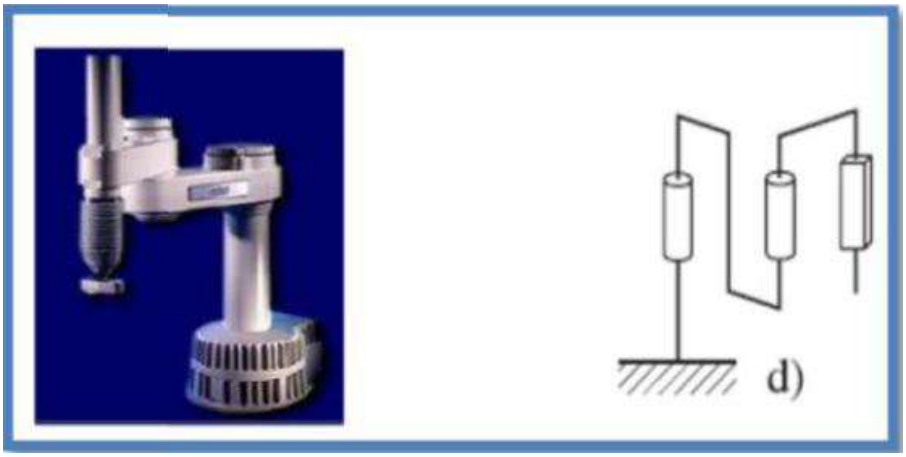


FIGURE 1.17 – **Robot SCARA**

- La structure 3R (anthropomorphe)

Elle permet d'amener un solide en un point de l'espace par trois rotations, généralement une à axe vertical et deux à axes horizontaux et parallèles. C'est le porteur (généraliste) par excellence pouvant se programmer facilement pour différents types de tâches et disposant d'un volume de travail conséquent [3].

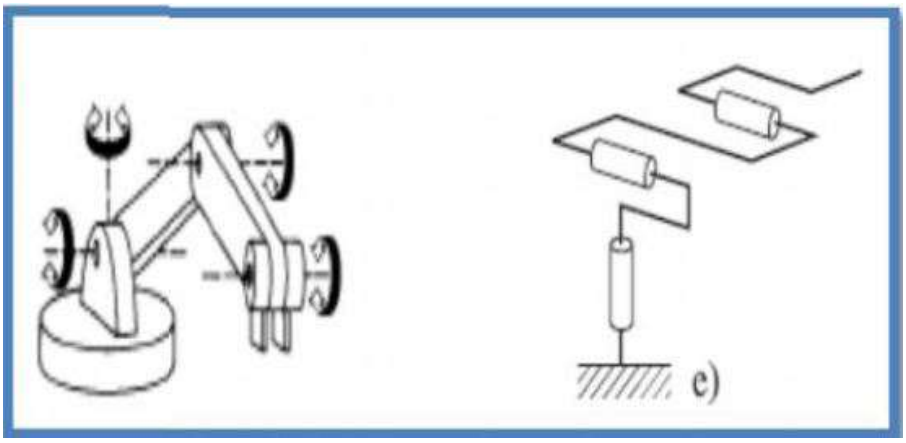


FIGURE 1.18 – **Robot 3R**

1.9 Les Domaines d'applications

Les robots industriels peuvent être utilisés dans plusieurs domaines et peuvent accomplir différentes tâches.

1.9.1 Les Tâches simples

La grande majorité des robots est utilisée pour des tâches simples et répétitives. Ils sont programmés une fois pour toute au cours de la procédure d'apprentissage. Les critères de choix de la solution robotique sont :

- La tâche est assez simple pour être robotisée.
- Les critères de qualité sur la tâche sont importants.
- Pénibilités de la tâche (peinture, charge lourde, environnement hostile, ...)

Exemples de robots utilisés pour des tâches simples : robots soudeurs par points et robots soudeurs à l'arc.

1.9.2 Les Tâches complexes

Exemples robotique de service :

- Robot pompiste.
- Robot de construction.

- Robot Computer motion.

- Robot Assistance aux personnes Handicapées.

1.10 Conclusion

Dans ce chapitre, nous avons donner une idée générale sur la robotique, l'historique des robots, leurs structures, leurs utilisations et les différents types de robots ainsi que leurs classifications et leurs domaines d'applications.

Chapitre 2

La Modélisation des robots manipulateurs

2.1 Introduction

La modélisation des robots, comme tout procédé industriel que l'on peut commander est une étape que l'on peut qualifier d'intermédiaire, puisqu'il s'agit de définir les équations qui décrivent le comportement dynamique d'un robot avec assez de précision pour que leur utilisation soit suffisamment simple au moment de la conception. La modélisation est la base de la commande, à partir de la bonne connaissance du comportement du système étudié, il est plus facile de choisir les lois et stratégies des commandes adaptées pour aboutir aux performances souhaitées [5].

2.2 Structure mécanique

La structure mécanique du robot manipulateur peut être divisée en deux parties [6] :

— Organe terminal : les tâches qui sont dévolues aux robots sont très variées. Pour chaque opération ou travail spécifique, l'organe terminal prend un aspect particulier.

— Elément porteur : il est composé d'un ensemble de corps souples ou rigides liés par des articulations, servant à déplacer l'organe terminal d'une configuration à une autre.

Avant de décrire les relations géométriques entre les différents corps du robot, on définit les notions suivantes :

— Degré de liberté : nombre de paramètres utilisés pour spécifier la configuration d'un élément de la chaîne cinématique par rapport à un autre.

— Degré de mobilité : toute articulation est caractérisée par son degré de mobilité (DDM), c'est-à-dire le nombre de degrés de liberté entre deux corps successifs de la chaîne cinématique ($0 \leq m \leq 6$).

— Espace articulaire : représente l'état des corps composant le robot en fonction des variables articulaires, sa dimension n'est égale au nombre de degrés de liberté du robot.

— Espace opérationnel : décrit la position et l'orientation de l'organe terminal du robot par rapport à un repère de référence, sa dimension ($m \leq 6$).

2.3 Les différentes méthodes de modélisation

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques tels que les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal), et l'espace articulaire (dans lequel est définie la configuration du robot) [7].

On distingue :

— La modélisation géométrique : (directe et inverse).

— La modélisation cinématique (différentielle) : (directe et inverse).

— La modélisation dynamique :(directe et inverse).

2.3.1 Modèle géométrique direct

Le Modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaires [8].

Le Modèle géométrique direct du robot peut représenter par la relation :

$$X = f(q) \tag{2.3.1}$$

Où :

q étant le vecteur des variables articulaires tel que :

$$\mathrm{q} = [q_1\ q_2\ q_3\ \ldots\ldots\ldots\ldots\ q_n]^T \in R^n$$

Les coordonnées opérationnelles sont définies par :

$$\mathrm{X} = [X_1\ X_2\ X_3\ \ldots\ldots\ldots\ldots\ X_n]^T \in R^m : m \leq n.$$

Pour les rotations, d'autres représentations sont couramment utilisées comme les angles d'Euler, les angles de Roulis-Tangage-Lacet (RTL) ou les paramètres d'Euler (quaternions). On note la convention de Denaviet-Hartenberg et les matrices $4 * 4$:

Considérons la (figure II.1) deux articulations successives pour passer de O_{i-1} à O_i et de Z_{i-1} à Z_i

On peut toujours suivre le chemin suivant :

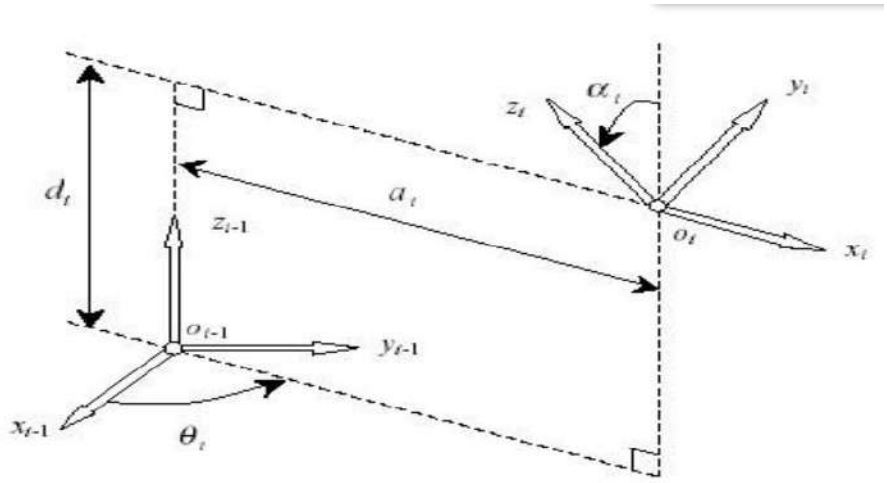


FIGURE 2.1 – **Systèmes de coordonnées et les paramètres de Denavit - Hardenberg**

1. une rotation autour de Z_{i-1} d'un angle θ_i : Rot (z, θ).
2. une translation le long de Z_{i-1} d'une longueur d_i : Trans (z, d).
3. une translation le long de X_i d'une longueur a_i : Trans (x, a).
4. une rotation autour de X_i d'un angle α_i : Rot (x, α).

Les grandeurs sont définies de la façon suivante :

— d_i : distance de O_{i-1} à la projection de O_i sur l'axe Z_{i-1}.

— a_i : distance de O_{i-1} à la projection de O_i sur le plan Y_i, O_i et Z_i.

— θ_i : angle compris entre X_{i-1} et le plan défini par l'axe Z_{i-1} et le point O_i.

— α_i : angle forme par la projection de Z_{i-1} sur le plan normal au plan (Z_{i-1}, O_i) et l'axe Z_i.

La forme générale de la matrice faisant passer d'une articulation à la suivante devra donc toujours :

La matrice de transformation homogène est donnée comme suit : [9]

$$^{i-1}T_i = \text{Trans}\ (x_{i-1}, a_{i-1})\ \text{Rot}\ (x_{i-1}, \alpha_{i-1})\ \text{Trans}\ (z_i, d_i)\ \text{Rot}\ (z_i, \bar{\omega}_i)$$

$$ {}^{i-1}T_i = \begin{bmatrix} 1 & 0 & 0 & \alpha_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\bar{\omega}_i & s\bar{\omega}_i & 0 & 0 \\ c\bar{\omega}_i & c\bar{\omega}_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{i-1}T_i = \begin{bmatrix} c\omega_i & -s\bar{\omega}_i & 0 & d_{i-1} \\ c\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1}s\bar{\omega}_i & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1} & -c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{i-1}T_i = \begin{bmatrix} {}^{i-1}A_i & {}^{i-1}P_i \\ 0_3 & 1 \end{bmatrix} .....(II.2) $$

Avec :

$$ {}^{i-1}A_i = \begin{bmatrix} c\omega_i & -s\bar{\omega}_i & 0 \\ c\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1}s\bar{\omega}_i & -s\alpha_{i-1} \\ s\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1}s\bar{\omega}_i & c\alpha_{i-1} \end{bmatrix}, {}^{i-1}P_i = \begin{bmatrix} d_{i-1} \\ -s\alpha_{i-1}d_i \\ -c\alpha_{i-1}d_i \end{bmatrix} $$

2.3.2 Modèle géométrique inverse

On a vu que le modèle géométrique direct d'un robot permettait de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires [9].

Le model inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse (MGI). On peut distinguer trois méthodes de calcul du MGI :

— la méthode de Paul qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels ;

— la méthode de Peiper qui permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques.

— la méthode générale de Raghavan et Roth donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

Le modèle géométrique inverse peut s'écrit :

$$q = f^1(X) \quad \text{..... (II .3)}$$

Dans le calcul du MGI trois cas se présentent :

1. absence de solution lorsque la situation désirée est en dehors de la zone accessible du robot. Celle-ci est limitée par le nombre de degrés de liberté, les débattements articulaires et la dimension des segments.
2. infinité de solutions lorsque :
   — le robot est redondant vis-à-vis de la tâche.
   — le robot se trouve dans certaines configurations singulières.
3. solutions en nombre fini, exprimées par un ensemble de vecteurs $[q_1,...,q_i]$.

   On dit qu'un robot manipulateur est résoluble lorsqu'il est possible de calculer toutes les configurations permettant d'atteindre une situation donnée. Aujourd'hui, tous les manipulateurs série ayant jusqu'à six degrés de liberté et qui ne sont pas redondants peuvent être considérés comme résolubles.
4. Le nombre de solutions dépend de l'architecture du robot manipulateur.

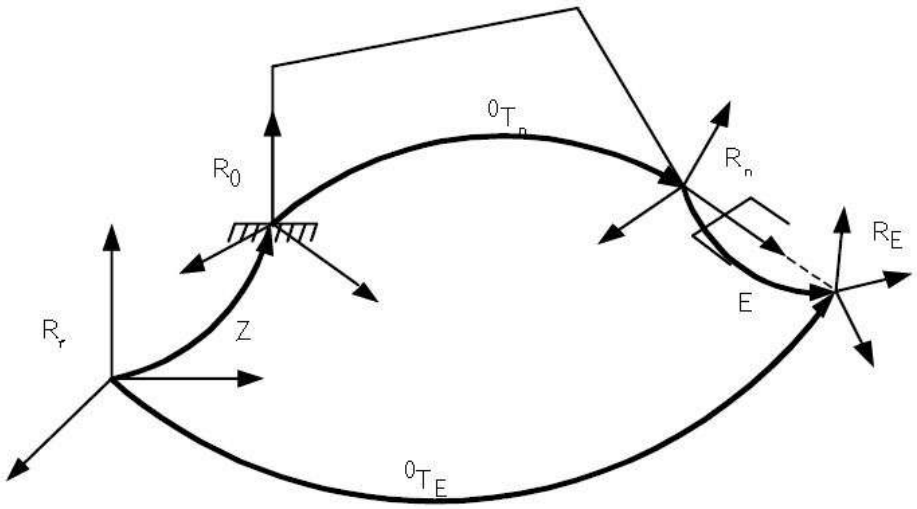


FIGURE 2.2 – **Transformations entre l'organe terminal et le repère atelier**

2.3.3 Le model cinématique direct

Nous venons de constater que le model géométrique posait deux catégories principales de problème dans son utilisation pour la commande des robots [9] :

1. la non linéarité de l'équation X=f (θ) entraine des difficultés d'obtention du transformateur de coordonnées inverse .le robot est soumis à la restriction solvable.
2. avec la commandes géométrique on ne contrôle pas (ou très mal) la vitesse de déplacement du robot. Dans les applications courantes il est très important de connaitre la dure d'exécution d'une tache ou plus exactement d'en être maitre, d'où la nécessite de contrôler la vitesse.

L'utilisation du model cinématique permet dans une certaine mesure de contourner ces deux difficultés d'où son intérêt. Le model cinématique direct d'un robot manipulateur décrit les variations élémentaire des coordonnées opérationnels en fonction des variations élémentaire des coordonnes articulaires il est noté :

$$\dot{X} = J(q)\dot{q} \quad \text{..... (II .4)}$$

Où J(q) désigne la matrice jacobéenne de dimension $(m \times n)$ du mécanisme, égale à $\partial X/\partial q$ et fonction de la configuration articulaire q. La même matrice jacobéenne intervient dans le

calcul du modèle différentiel direct qui donne les variations élémentaires d_x des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires d_q soit :

$$\text{dx} = \text{J(q) dq} \quad \text{..... (II .5)}$$

2.3.3.1 Matrices jacobéenne cinématique

On peut obtenir la matrice jacobéenne par une méthode de calcul direct, fondée sur la relation entre les vecteurs des vitesses de translation et de rotation V_n et ω_n du repère R_n, et les vitesses articulaires $\dot{q}$ [9].

$$\begin{bmatrix} V_n \\ \omega_n \end{bmatrix} = J(q)\dot{q} \quad \text{.....} (II.6)$$

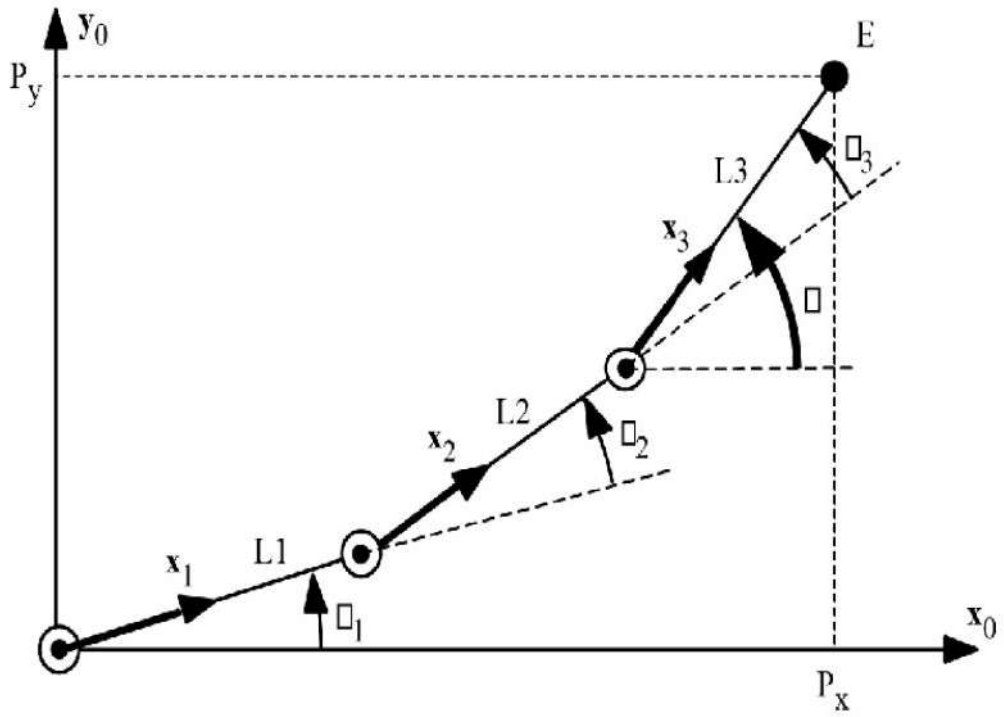


FIGURE 2.3 – **Exemple d'un robot plan à trois degrés de liberté**

On note que Vn est la dérivée par rapport au temps du vecteur P_n. En revanche, ω_n n'est pas la dérivée d'une représentation quelconque de l'orientation.

2.3.4 Le model cinématique inverse

L'objectif du modèle cinématique inverse est de calculer, à partir d'une configuration q donnée, les vitesses articulaires qui assurent au repère terminal une vitesse opérationnelle.

Cette définition est analogue à celle du modèle différentiel inverse : ce dernier permet de déterminer la différentielle articulaire dq correspondant à une différentielle des coordonnées opérationnelles dx spécifiée. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires.

Les algorithmes traitent de façon unifiée les cas réguliers, singuliers et redondants. Elles nécessitent un temps de calcul relativement important. Nous présentons dans ce paragraphe les techniques à mettre en oeuvre pour établir un modèle cinématique inverse dans les cas réguliers, singuliers et redondants [9].

2.3.5 La Modélisation dynamique

Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. On représente le modèle dynamique par une relation de la forme [9] :

$$\Gamma = \mathrm{f}(\mathrm{q}, \dot{q}, \ddot{q}, f_e) \quad \text{..... (II .7)}$$

Avec :

— Γ : vecteur des couples/forces des actionneurs, selon que l'articulation est rotoîde ou prismatique.

— q : Vecteur des positions articulaires.

— $\dot{q}$: Vecteur des vitesses articulaires.

— $\ddot{q}$: Vecteur des accélérations articulaires.

— f_e : vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

Le modèle dynamique direct est celui qui exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulations. Il est alors représenté par la relation :

$$\ddot{q} = g(q, \dot{q}, \Gamma, f_e) \quad \text{..... (II .8)}$$

Parmi les applications du modèle dynamique, on peut citer :

— la simulation, qui utilise le modèle dynamique direct.

— le dimensionnement des actionneurs.

— l'identification des paramètres inertiels et des paramètres de frottement du robot.

— la commande, qui utilise le modèle dynamique inverse.

Plusieurs formalismes ont été utilisés pour obtenir le modèle dynamique des robots, les formalismes les plus souvent utilisés sont :

— le formalisme de Lagrange pour le calcul du model dynamique direct.

— le formalisme de Newton-Euler, leur méthodes sont les mieux adaptés pour le calcul du model dynamique inverse.

2.4 Conclusion

Le chapitre présent fait une description sur les différents modèles de la modélisation mécanique et on s'est intéressé surtout par notre modèle du bras manipulateur.

Á propos de la présentation de la partie commande qui réunit la carte de commande et le type d'actionneur utilisé feront l'objet du chapitre suivant.

Chapitre 3

Arduino et Servomoteur

3.1 Introduction

Le système Arduino est un projet créé par une équipe de développeurs, composée de six individus. Ce système est un outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes. Le système Arduino nous permet de réaliser un grand nombre de choses, qui ont une application dans tous les domaines.

Nous voulons aussi présenter dans ce chapitre les servomoteurs récents. Aujourd'hui dans le monde robotique on trouve beaucoup d'application qui sont réalisées à l'aide des actionneurs électriques de type servomoteur car ils sont facile pour la commande avec une précision élevée et délivre un couple considérable.

3.2 Arduino

3.2.1 Définition

L'Arduino est une plateforme open source d'électronique programmée qui est basée sur une carte à microcontrôleur et un logiciel. Plus simplement, on peut dire que l'Arduino est un module électronique, doté d'un microcontrôleur programmable, il peut être utilisé pour développer des objets interactifs, munis d'interrupteurs ou de capteurs, et peut contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielle.

3.2.2 Microcontrôleur

Un microcontrôleur est un circuit intégré (ou IC, integrated Circuit), qui rassemble sur une puce plusieurs élément complexes dans un espace réduit, au temps des pionniers de l'électronique, on soudait un grand nombre de composants encombrants, tels que les transistors, les résistances ou les condensateurs sur des cartes plus ou moins grandes. Aujourd'hui, tout peut loger dans un petit boîtier en plastique noir muni d'un certain nombre de broches. Ces dernières sont les connexions du circuit intégré un moyen desquelles s'effectue la communication.

La figure montre un microcontrôleur ATmega328, qu'on trouve sur la carte Arduino [10].

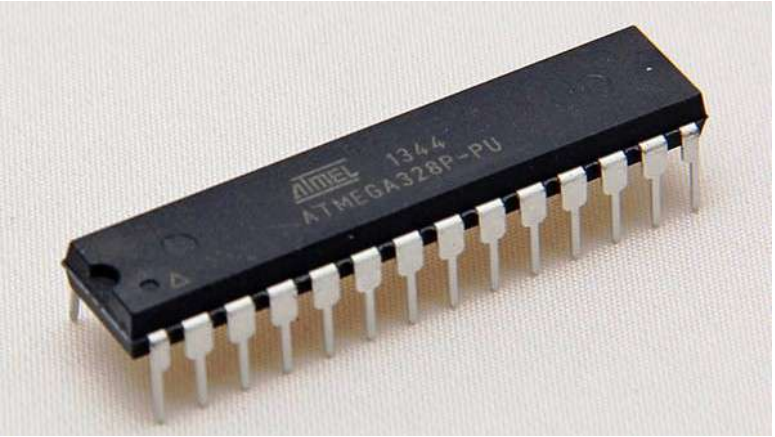


FIGURE 3.1 – **Microcontrôleur ATmega328,(source : Atmel)**

3.2.3 Les différentes cartes Arduino

Les cartes Arduino doivent satisfaire des exigences diverses et variées. Certains utilisateurs souhaiteront effectuer du prototypage et tester de nouveaux montages ou idées, la carte Arduino Uno disposant d'assez d'entrées-sorties pour les projets d'envergure raisonnable. De part sa taille, la carte devrait aussi pouvoir être utilisée à l'avenir dans des projets plus ambitieux. D'autres auront besoin d'un grand nombre de ports afin de pouvoir raccorder de nombreux capteurs ou actionneurs. Une troisième catégorie d'utilisateurs ne cherchera qu'à attirer l'attention et à transmettre des signaux à leurs semblables à l'aide de diodes clignotantes.

La forme, la taille ou les possibilités de connexion jouent un rôle décisif dans le choix de la carte adaptée. C'est pourquoi les développeurs d'Arduino ont mis au point un vaste choix de cartes à microcontrôleur afin que chacun trouve le modèle qui répondre à ses besoin :

- Arduino uno
- Arduino Leonardo
- Arduino Mega 2560
- Arduino Esplora
- Boarduino V2.0
- Arduino Nano
- Arduino LilyPad
- Arduino Due
- Arduino Yûn

3.2.4 La carte Arduino Uno

La pièce maîtresse de tous les montages est : la carte à microcontrôleur Arduino

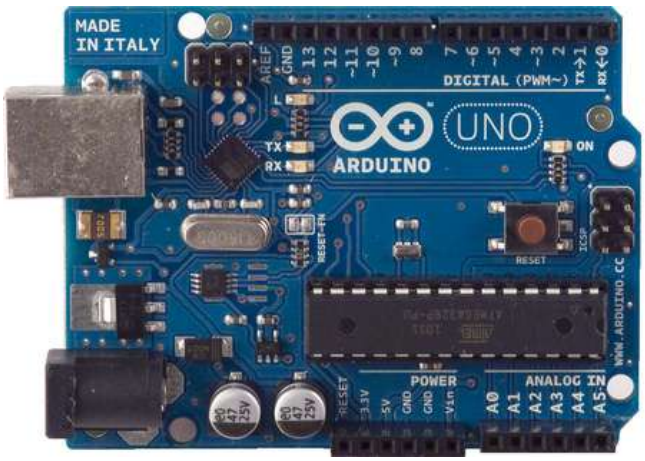


FIGURE 3.2 – **La carte à microcontrôleur,(Arduino)**

Sur cette image, vous ne pouvez évidemment pas vous rendre compte à quel point les dimensions de la carte Arduino sont réduites (environ 7 cm de large et 5 cm de long) ; elle est vraiment très maniable, tient sans problème dans une main et s'avère vraiment compacte.

Au fil des années, plusieurs cartes Arduino ont été développées. Nous avons choisi de prendre l'Uno, car c'est la plus populaire.

Les composants les plus divers y sont reconnaissable (voir FIGURE III.3)

Certains penseront certainement qu'on ne peut pas concevoir quelque chose de sérieux sur une surface aussi réduite. Mais grâce à la miniaturisation des composants ces dernières années, ce qui nécessitait auparavant cinq puces électroniques n'en requiert aujourd'hui plus qu'une.

Le plus gros élément qui saute directement aux yeux est le microcontrôleur proprement dit. Il est de type ATmega328.

Elle convient tout particulièrement à ceux qui veulent faire leurs premiers pas dans le monde des microcontrôleurs.

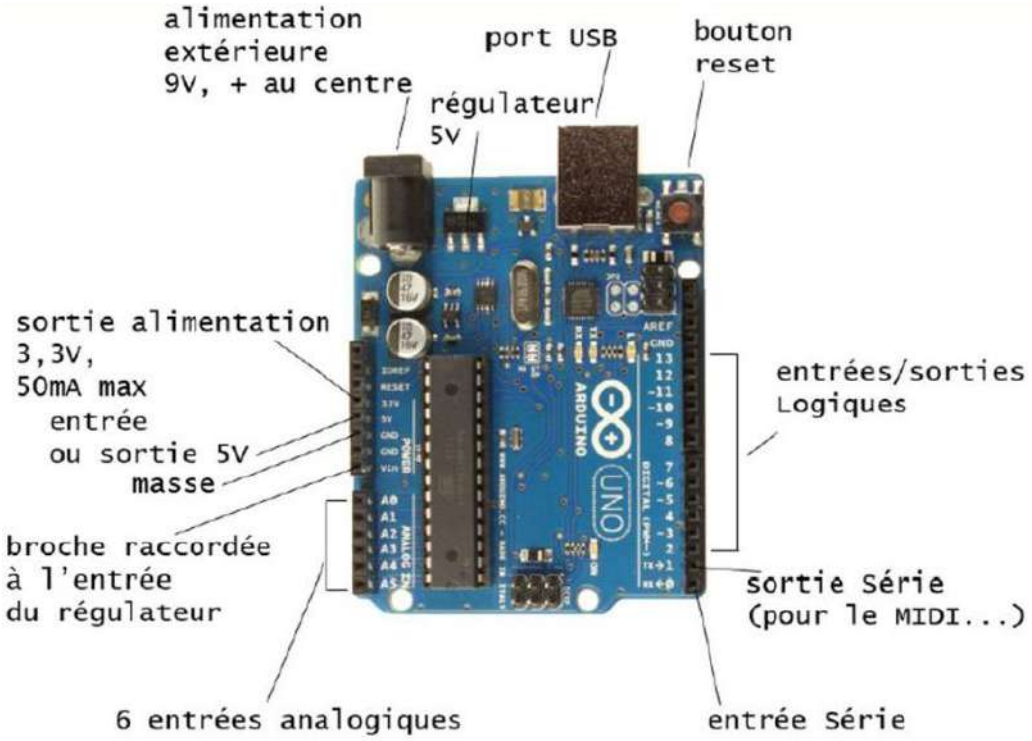


FIGURE 3.3 – **les composants de la carte Arduino**

Ces éléments sont les plus importants de la carte Arduino mais, bien entendu, cela ne veut pas dire que les autres sont à négliger.

Voici les principales caractéristiques de la carte Arduino :

- Microcontrôleur Atmega328.
- Tension de service 5V.
- 14 entrées et sortis numériques (6 sorties commutables en MLI).
- 6 entrées analogiques (résolution 10 bit).
- 32 Ko de mémoire flash (0.5 Ko occupé par le chargeur d'amorçage ou bootloarder).
- 2 Ko de SRAM.
- 1 Ko d'EEPROM.
- Fréquences d'horloge 16 MHz.
- Interface USB.

3.2.5 Les modes de communication

Un microcontrôleur de type Arduino a déjà beaucoup de connexions qu'il s'agit de bien distinguer.

3.2.5.1. Le port USB

Sans le port USB, vous ne seriez pas en mesure d'initialiser une communication.

Le travail avec la carte Arduino peut se diviser en deux étapes : le temps consacré à la mise en oeuvre du montage et celui dédié à la programmation, appelé phase de développement (design time).

La programmation s'effectue dans un environnement de développement que vous allez apprendre à connaître très rapidement. C'est dans cet environnement que vous allez saisir le programme créé par vos soins pour le transmettre au microcontrôleur. Si tout s'est bien passé, le temps de l'exécution (run time) commence. Vous n'avez pas besoin de dire explicitement au microcontrôleur, il démarre en effet immédiatement après avoir reçu toutes les instructions de votre part. Vous pouvez en outre échanger des données avec votre ordinateur via le port USB.

3.2.5.2. Les ports d'entrée ou de sortie (E/S)

Les ports E/S représentent l'interface du microcontrôleur. Il existe plusieurs chemins ou canaux pour échanger des données, comme chez l'homme avec les yeux, les oreilles et la bouche. Il se produit, Grâce et par le biais de ces canaux de communication, une interaction avec l'environnement.

3.2.5.2.1. les ports d'entrée et de sortie sur la carte Arduino

Quand vous la tenez de manière à pouvoir lire l'inscription (UNO) vous verrez les ports d'entrée et de sortie sur le bord supérieur (un bloc de dix broches et un autre de huit broches).

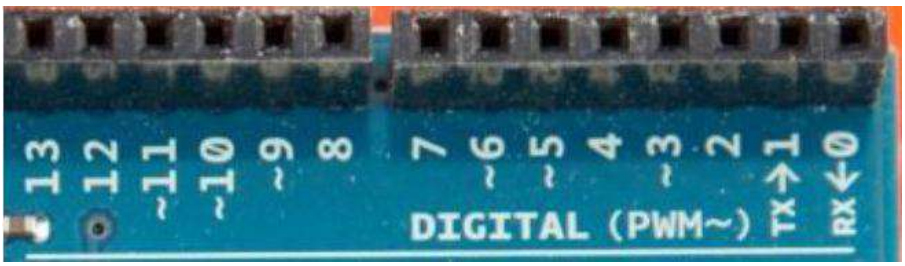


FIGURE 3.4 – **Entré et Sortie numérique**

Il est bien entendu important de savoir pour chaque port quelle est son adresse afin de pouvoir communiquer avec lui au cours de la programmation. C'est pour cela que chaque broche possède un numéro.

Sous certains chiffres se trouve un tilde ( ) indiquant que la broche peut être commutée en sortie analogique. Il s'agit là d'une broche MLI (rappelez-vous la modulation de largeur d'impulsion, on y reviendra un peu plus loin).

Á l'extrémité inférieure de la carte se situent les ports d'alimentation (à gauche) et les ports d'entrée analogique (un bloc de huit broches et un autre de six broches).

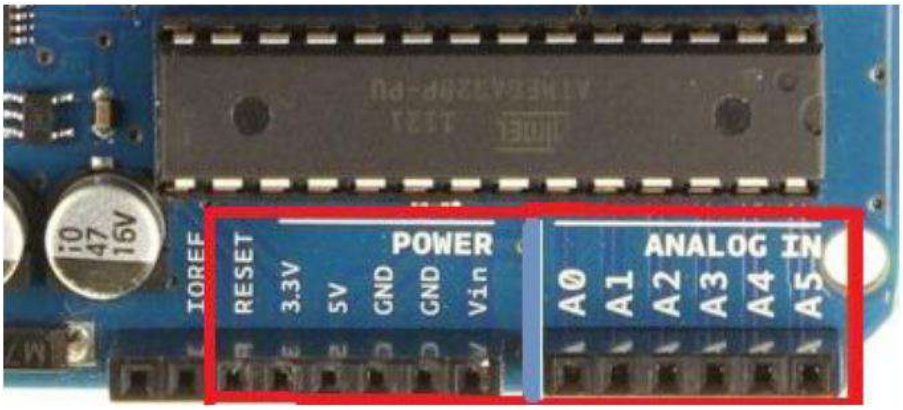


FIGURE 3.5 – **Alimentation et entrée Analogiques**

3.2.6 L'alimentation électrique

La carte Arduino Uno doit être alimentée en énergie pour pouvoir travailler.

Cette alimentation peut s'effectuer tout d'abord via l'interface USB qui relie la carte à l'ordinateur-ce chemin sert aussi à l'échange de données entre la carte et l'ordinateur. En phase de développement avec votre Arduino, la connexion USB va servir d'alimentation primaire de la carte.

La seconde possibilité consiste à brancher une batterie ou bloc secteur au connecteur, appelé prise jack. Vous pouvez, par exemple, employer cette variante si vous avez construit un engin manoeuvrable, commandé par la carte Arduino. Le Véhicule doit pouvoir évoluer librement dans l'espace, sans câble.

En effet, l'utilisation d'un câble USB, généralement trop court, limiterait alors la mobilité de l'engin. L'emploi d'une batterie rend le dispositif autonome.

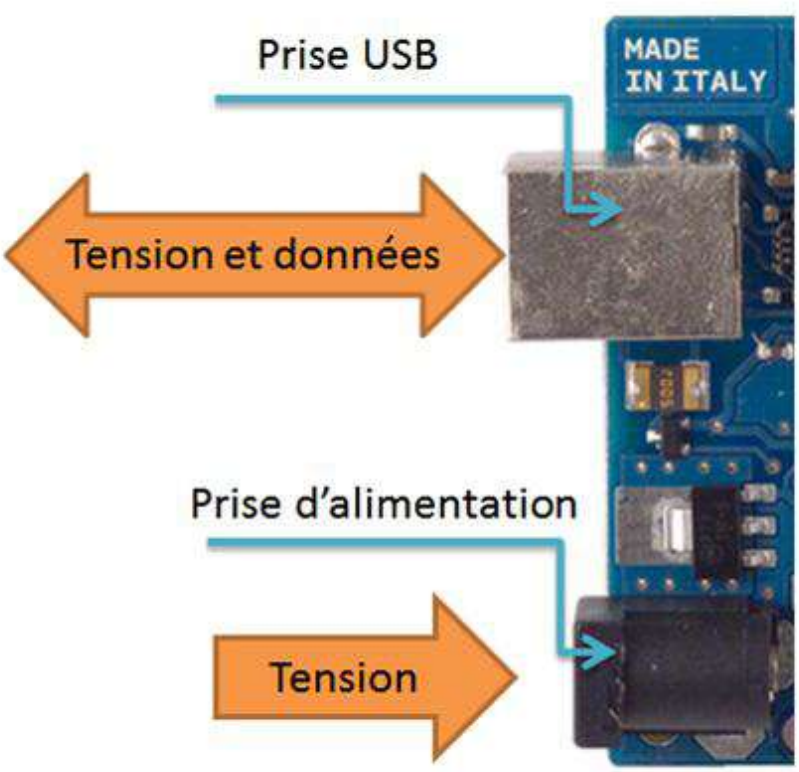


FIGURE 3.6 – **Alimentation de la carte Arduino**

Nous montrons ici les différentes prises. Attention, elles ne peuvent pas être interverties, car elles ont des formes et des fonctions différentes.

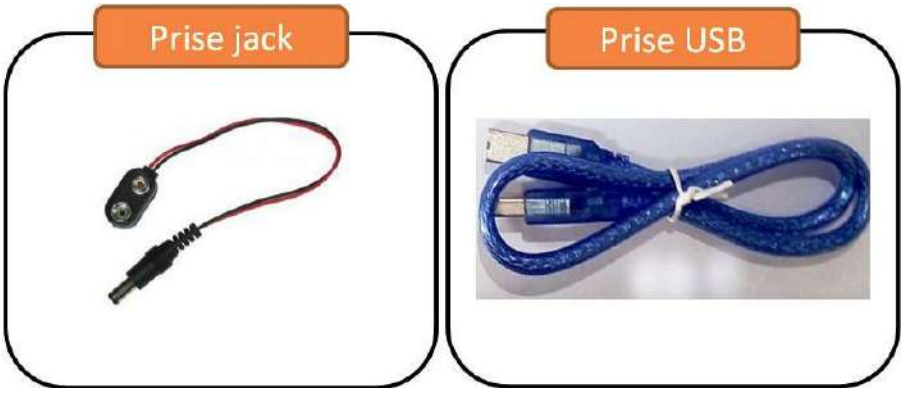


FIGURE 3.7 – **les différentes prises**

Dès qu'il s'agit de courant ou de tension, il convient de consulter le tableau.

3.2.7 Valeurs de courant ou de tension

Catégorie	Valeur
Tension de service	5 V (DC)
Alimentation depuis l'extérieur (recommandée)	7-12 V(DC)
Alimentation depuis l'extérieur (valeur limité)	6-20 V(DC)
Courant continu par broche (maximal)	40 mA

TABLE 3.1 – **Valeurs de courant ou de tension**

l'interface USB peut fournir un courant maximal de 500 mA ; c'est en principe suffisant pour réaliser la plupart des circuits d'essai. Elle est protégée contre le court-circuit et les courants forts grâce à un poly fusible. Mais attention, cela ne doit pas vous empêcher de construire votre circuit avec le plus grand soin.

3.2.8 La Programmation de l'Arduino

3.2.8.1 Le Langage De Programmation

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme Arduino s'effectue

de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres.

3.2.8.2 La Structure D'un Programme Arduino

Comme tous autres langages de programmation de bas niveau, un programme Arduino comporte trois parties :

1) la partie déclaration des variables (optionnelle).

2) la partie initialisation et configuration des entrées/sorties : la fonction setup ().

3) la partie principale qui s'exécute en boucle : la fonction loop ().

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino.



FIGURE 3.8 – **Coloration Syntaxique dans IDE de l'Arduino**

Dans l'interface de programmation de l'Arduino, certains mots apparaissent en différentes couleurs qui clarifient le statut des différents éléments :

- En orange, apparaissent les mots-clés reconnus par le langage Arduino comme des fonctions existantes. Lorsqu'on sélectionne un mot coloré en orange et qu'on effectue un clic avec le bouton

droit de la souris, l'on a la possibilité de choisir (Find in reference) : cette commande ouvre directement la documentation de la fonction sélectionnée.

- En bleu, apparaissent les mots-clés reconnus par le langage Arduino comme des constantes.

- En gris, apparaissent les commentaires qui ne seront pas exécutés dans le programme. Il est utile de bien commenter son code pour s'y retrouver facilement ou pour le transmettre à d'autres personnes. L'on peut déclarer un commentaire de deux manières différentes : dans une ligne de code, tout ce qui se trouve après (//) sera un commentaire. l'on peut encadrer des commentaires sur plusieurs lignes entre (/*) et (*/).

3.3 Les Servomoteurs

3.3.1 Définition

Les servomoteurs sont des actionneurs très utilisés en modélisme et dans l'industrie, ils ont comme caractéristique principale leur (couple), c'est-à-dire la force de rotation qu'ils peuvent exercer. Plus un servomoteur aura de couple et plus il pourra actionner des (membres) lourds comme déplacer un bras qui porte une charge.

Pour la robotique de loisirs, les servomoteurs ont en général peu de couple et sont de taille réduite, bien adaptée à un encombrement minimal et à une énergie disponible limitée.



FIGURE 3.9 – **les différents types de servomoteur**

Les servomoteurs sont pilotés par un fil de commande et alimentés par deux autres fils. Habituellement, ces 3 fils sont rassemblés dans une prise au format.

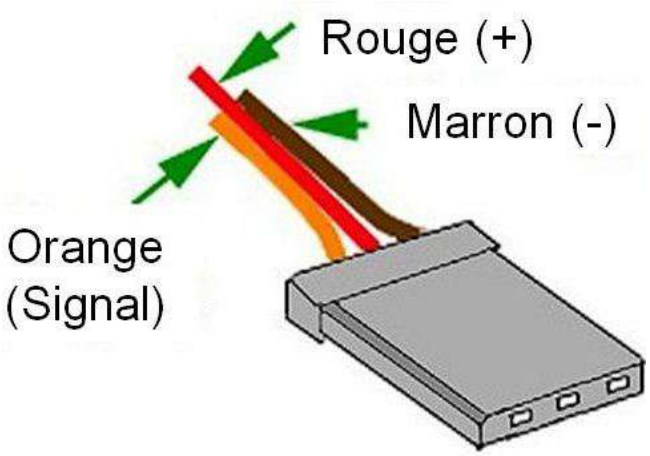


FIGURE 3.10 – **Les Trois Files**

Un fil rouge est relié à l'alimentation positive (+5 ou +6 V selon le servo), le fil noir est relié à la masse (GND) et le fil jaune est utilisé pour la commande (PWM). Il y aurait beaucoup à dire sur le fonctionnement d'un servomoteur, ses composants, son moteur et le petit potentiomètre qui permet de connaître sa position mais cette fiche va droit au but et se limitera à son utilisation avec l'Arduino [11].

3.3.2 Le Principe de l'asservissement d'un Servomoteur

À présent voici le schéma de principe qui a pour fonctionnement celui expliqué par le synoptique précédent :

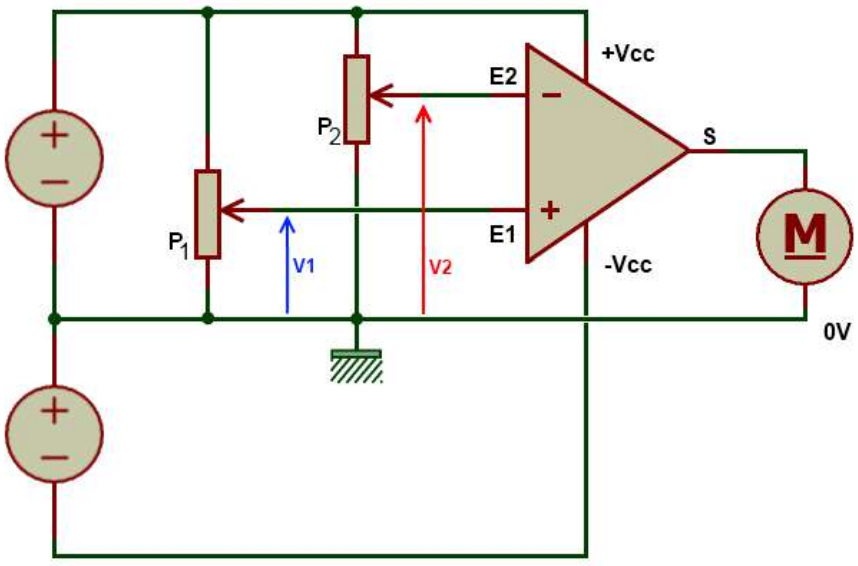


FIGURE 3.11 – **Le Principe de l'asservissement d'un Servomoteur**

De gauche à droite on a : les alimentations qui fournissent la tension positive et négative ; les potentiomètres P1 et P2 ; le comparateur (oui c'est ce gros triangle avec un plus et un moins) : enfin le moteur à courant continu.

Un comparateur est un composant électronique de la famille des circuits intégrés car, il contient en vérité d'autres composants, essentiellement des semi-conducteurs (diodes, transistors) et des résistances. Ce composant a toujours besoin d'une alimentation externe pour fonctionner, c'est à dire qu'on ne peut lui mettre des signaux à son entrée que s'il est alimenté. Autrement il pourrait être endommagé (c'est pas souvent le cas, mais mieux vaut être prudent). Vous le constatez par vous-même, le comparateur est un composant qui possède deux entrées et une sortie. Et, de la manière la plus simple qui soit, en fait il n'y a rien de plus simple qui puisse exister, son fonctionnement réside sur le principe suivant :

Si la tension (je me base par rapport au schéma) $V1$ qui arrive sur l'entrée $E1$ du comparateur est supérieure à la tension $V2$ qui entre sur l'entrée $E2$ du comparateur, alors la tension en sortie S du comparateur est égale à $+Vcc$ (l'alimentation du comparateur).

Tandis que dans le cas opposé où la tension $V2$ va être supérieure à $V1$, la sortie S du comparateur aura une tension égale à $-Vcc$.

En transposant mes dires sous une forme mathématique, cela donnerait ceci :

— Si $V1 > V2$, alors $Vs = +Vcc$.

— Si $V1 > V2$, alors $Vs = -Vcc$.

Comment s'en rappeler. Eh bien grâce aux petits symboles $(+)$ et $(-)$ présents dans le triangle représentant le comparateur.

La sortie du comparateur prendra +Vcc si la tension sur l'entré $(+)$ du comparateur est supérieure à celle sur l'entrée $(-)$ et inversement. Voyez, j'vous z'avais dis que c'était ultra simple.

Il y a encore quelque chose à savoir : Il est impossible que les tensions V1 et V2 soient égales ! Oui, car le comparateur ne peut pas fournir une tension positive ET une tension négative en sa sortie, c'est pourquoi, même si vous reliez E1 et E2 avec un fil, la tension en sortie du comparateur sera toujours OU +Vcc OU -Vcc.

3.3.3 Le rôle des servomoteurs

Il permet de réaliser des déplacements en translation ou en rotation sur des modèles réduits.

Il existe une grande variété et qualité de servomoteur se différenciant par leur vitesse de rotation/seconde, par leur course en degré et par leur couple exprimé en kg/cm.

Ils se distinguent également par la matière utilisée et le montage éventuel sur roulements[11].

3.3.4 Fonctionnement d'un servomoteur standard

Mécaniquement, il s'agit d'un moteur avec réducteur. Le mouvement de sortie est une rotation. Lorsque le moteur tourne, l'axe du servomoteur change de position, ce qui modifie la résistance du potentiomètre. Le rôle de l'électronique est alors de commander le moteur pour que la position de l'axe de sortie soit conforme à la consigne reçue.

La commande se fait via une petite impulsion dont la durée est comprise entre 0,5 ms et 2,5 ms. La valeur de 1,5 ms donne au servomoteur la position centrale.

Les ordres de positions sont transmis sous forme d'un signal codé en largeur d'impulsion, répété périodiquement, en général toutes les 50 ms, ce qui permet à l'électronique de contrôle et de corriger continuellement la position angulaire de l'axe de sortie (en général, course limitée à 90°ou 120°).

L'augmentation de l'amplitude des impulsions agit sur le contrôle de la vitesse jusqu'au moment où le bras du servomoteur se trouve dans la position souhaitée.

Durant la rotation, le potentiomètre indique au circuit électronique le moment où la position désirée est obtenue. Les impulsions diminuent l'amplitude jusqu'à ce que plus aucune tension ne soit appliquée au moteur du Servomoteur tout en conservant le bras dans sa nouvelle position.

3.3.4.1 les Inconvénients des servomoteurs

L'impulsion est forte et instantanée, entrainant aucune continuité de la rotation initiée comme pourrait le faire une mise sous tension continue. Les petites impulsions données au moyen de la manette de commande ne produisent que de faibles effets. Ils peuvent se révéler inefficaces et sont les causes de l'importante zone neutre qui caractérise les servomoteurs traditionnels.

3.3.5 Les composants d'un servomoteur

1- Un moteur électrique MCC.

2- Une réduction en sortie de ce moteur pour avoir du couple.

3- Un potentiomètre qui induit une résistance variable en fonction de la position angulaire de sortie, donc le potentiomètre est lié à l'axe du moteur.

4- Un asservissement électronique pour contrôler la position angulaire

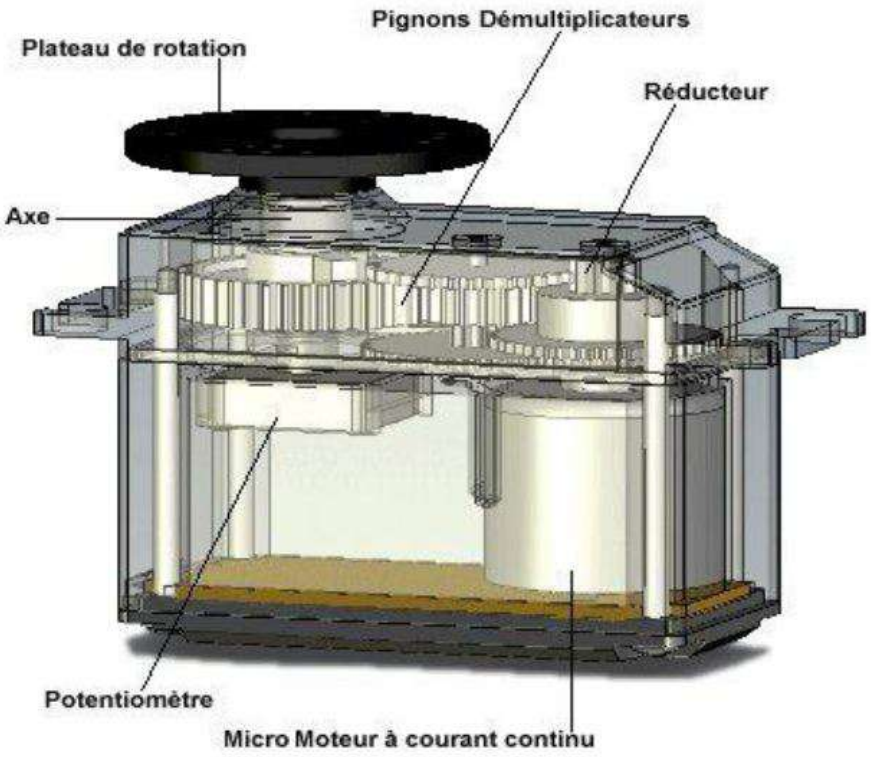


FIGURE 3.12 – **Les composants d'un servomoteur**

3.3.6 La théorie de la commande des servomoteurs

Le signal de commande d'un servomoteur est un signal PWM de fréquence 50 Hz et dont le niveau haut doit être entre 1 et 2 ms (en théorie).

- 1 ms : 0°en angle de sortie.
- 1.5 ms : 90°en angle de sortie.
- 2 ms : 180°en angle de sortie.

Le déplacement angulaire total est de 180°.

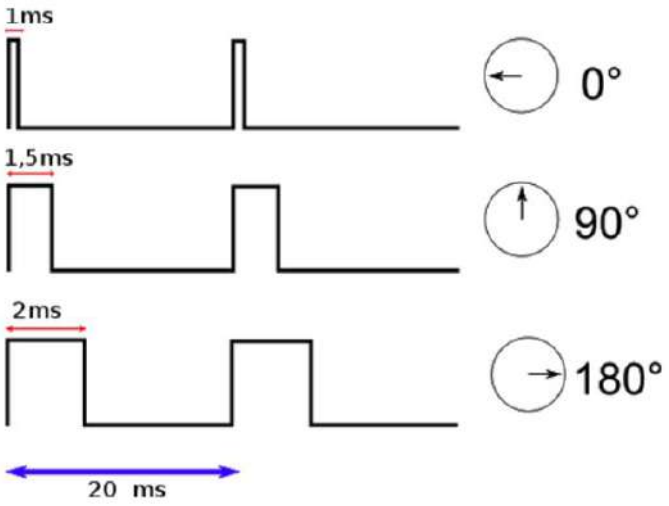


FIGURE 3.13 – **Chronogramme de commande de Servomoteur**

3.3.7 l'implimentation de la commande théorique

La commande d'un servomoteur en pratique est effectuée par un microcontrôleur avec l'utilisation :

- Périphérie PWM.
- Périphérie Timer.
- Ou utilisation des fonctions de type delay ms.

3.3.8 Les servomoteurs digital et analogique

Outre le fait qu'il contient le même moteur, les mêmes engrenages que les analogiques et disposent eux aussi d'un potentiomètre, le servomoteur digital se différencie du servomoteur analogique par le fait qu'il incorpore un microprocesseur qui analyse le signal reçu du récepteur et contrôle le moteur du servomoteur.

La différence principale est donc la façon de traiter l'information reccue du récepteur et le contrôle exercé sur la tension électrique appliquée au moteur, ce qui permet de réduire la zone neutre, augmentant ainsi l'amplitude du mouvement et générant une grande stabilité du positionnement statique.

3.3.8.1 Les Avantages du servomoteur digital

- Des impulsions de plus grande fréquence : 300 impulsions par seconde contre 50 impulsions par seconde dans le cas d'un servomoteur analogique.
- Le moteur réagit mieux à l'ordre donné.
- Une diminution de la zone neutre du servomoteur entrainant une meilleure rapidité de réponse, une accélération rapide, un freinage progressif.
- Une meilleure stabilité dans le positionnement statique.

3.3.8.2 Les Inconvénient du servomoteur digital

- Une plus grande consommation d'énergie.

3.3.9 Le pilotage des servomoteurs

Le mode de commande d'un servomoteur est standardisé : on envoie sur son fil de commande une impulsion dont la durée correspond à l'angle désiré.

Historiquement, cette impulsion était délivrée par un circuit oscillateur. Le circuit intégré NE555 est un exemple vedette du circuit utilisé.

Avec la programmation de l'Arduino, ce circuit n'est plus nécessaire. Une bibliothèque (library) dédiée, la bibliothèque (servo), permet de piloter un servomoteur en lui transmettant simplement l'angle sur lequel il souhaite se positionner.

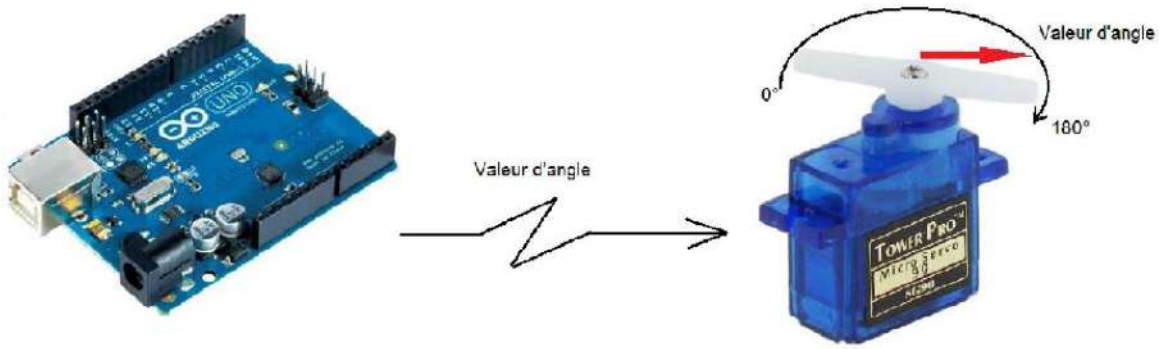


FIGURE 3.14 – **Présentation d'un servomoteur avec Arduino**

Ce qui permet de réaliser simplement des mouvements qui peuvent être complexes et de les

automatiser, voir la figure suivante :



FIGURE 3.15 – **Exemple de piloter des servomoteurs avec l'Arduino**

Les connaissances nécessaires au pilotage d'un servomoteur sont :

TABLE 3.2 – **Les connaissances nécessaires au pilotage d'un servomoteur**

#include <Servo.h>	L'utilisation de la librairie servo.h qui prend en charge la communication de bas niveau entre l'Arduino et le servomoteur. Les commandes précédées par # sont des ordres particuliers donnés au compilateur
Servo myservo ;	La création d'un objet de type Servo, appelé ici myservo
myservo.attach(9) ;	Le fil de commande de ce servo sera connecté au PIN 9 et l'objet myservo pilotera ce PIN
myservo.write(90) ; myservo.write(Angle) ;	Demander au servomoteur de se déplacer á l'angle désiré, soit de façon absolue en lui indiquant une valeur entière (90°) dans le premier cas, soit en lui passant le contenu d'une variable (Angle) compris entre 0 et 180, ce qui peut être utile par exemple pour donner une progressivité au déplacement en faisant varier l'angle d'un pas fixe (quelques degrés) par une boucle
myservo.read(Angle) ;	Pour lire la valeur de l'angle du servomoteur

3.4 Conclusion

En effet ce chapitre est divisé en deux parties, on a exposé dans la 1er partie les différents types de la carte Arduino précisément la carte Arduino Uno avec sa microcontrôleur et ses principales caractéristiques.

La 2ème partie rassemble le fonctionnement du servomoteur et ses deux modes de com-

mandes (analogique, digital) en tenant la méthode de pilotage d'un servomoteur avec la carte Arduino.

Chapitre 4

La Réalisation de la Commande d'un Bras Manipulateur, Résultats et Discussions

4.1 Introduction

Ce chapitre sera consacré à la présentation expérimentale du travail accompli selon différents phases de réalisation, généralement, on peut dire que notre projet est constitué de deux principales parties :

— Partie soft (Partie Programmations).

— Partie hard (Partie Mécatroniques).

Ainsi, l'adaptation entre les deux parties (soft et hard).

4.2 Simulation, Repérage et paramétrage du bras manipulateur plan de type 3R dans le plans

4.2.1 Le Repérage

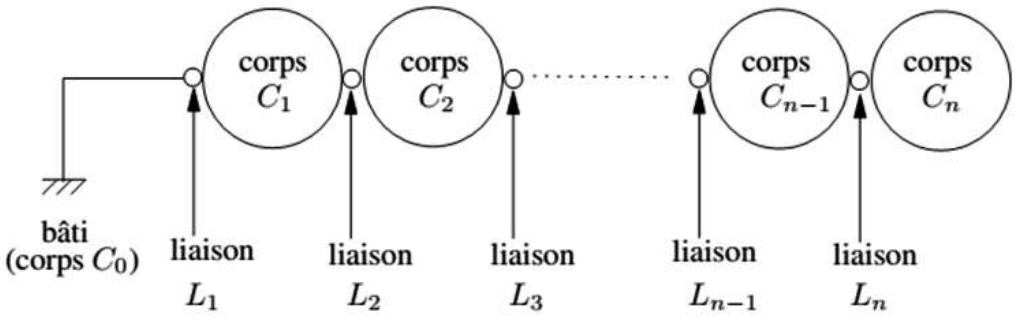


FIGURE 4.1 – **Chaîne cinématique d'un bras manipulateur série**

Selon la méthode des paramètres de Denavit-Hartenberg modifiés, On associe au ière corps le repère $R_i = (O_i, x_i, y_i, z_i)$, avec i = 0, 1, ..., n. Pour placer les repères R_1 à R_n selon la règle des paramètres de Denavit-Hartenberg modifiés, il convient de suivre les règles suivantes (Figure IV.2), pour i = 2, 3, ..., n :

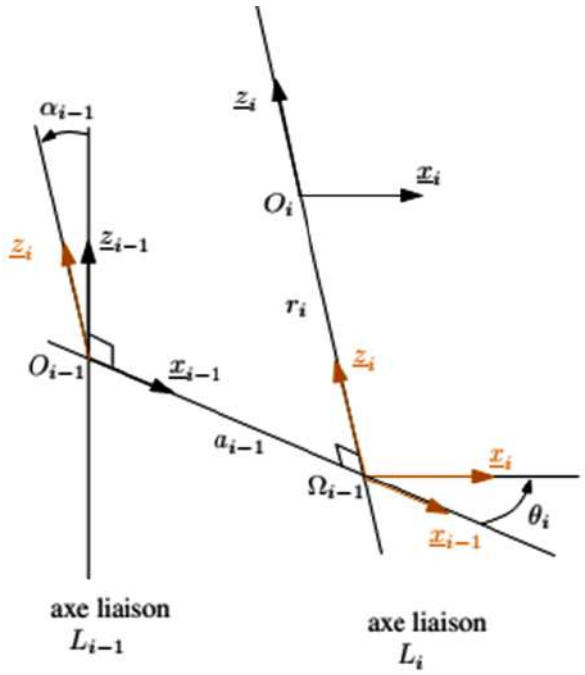


FIGURE 4.2 – **Paramètres de Denavit-Hartenberg modifiés**

— z_{i-1} est l'axe de la liaison L_{i-1}, librement orienté.

— O_{i-1} est le pied de la perpendiculaire commune aux axes des liaisons L_{i-1} et L_i, situé sur L_{i-1}.

— x_{i-1} est le vecteur unitaire de cette perpendiculaire commune, orienté de L_{i-1} vers L_i.

Le mécanisme comporte quatres liaisons rotoîdes.

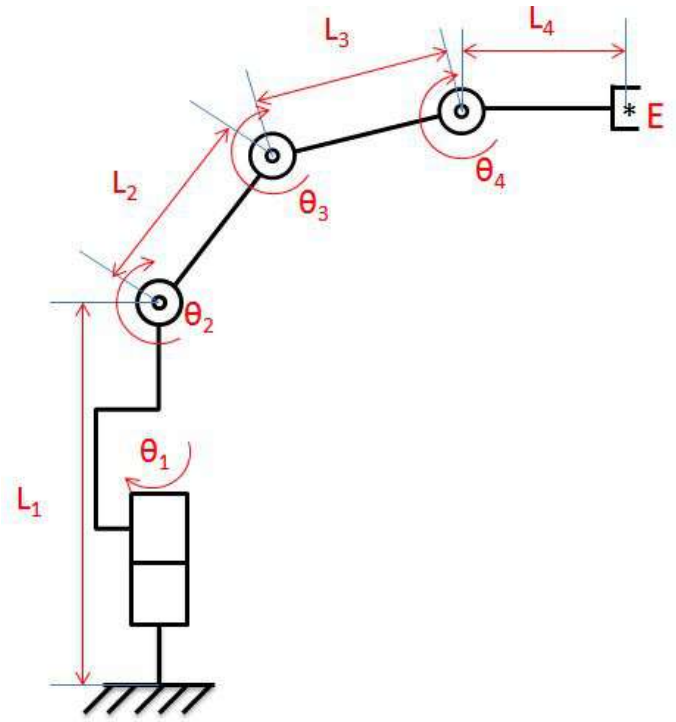


FIGURE 4.3 – **Repérage de notre mécanisme**

4.2.2 Le Paramétrage

selon la méthode des paramètres de Denavit-Hartenberg modifiés Les paramètres, On peut les définir comme suit :

— z_i est l'axe de la liaison i.

— α_{i-1} (twist angle) est l'angle algébrique entre z_{i-1} et z_i, mesuré autour de x_{i-1} ;

— a_{i-1} (link length) est la distance arithmétique de la perpendiculaire commune aux axes des liaisons L_{i-1} et L_i mesurée le long de x_{i-1} ;

— θ_i (joint angle) est l'angle algébrique entre x_{i-1} et x_i, mesuré autour de z_i ;

— (offset length) est la distance algébrique du point O_i à la perpendiculaire, mesurée le long de z_i. c'est la distance de x_{i-1} à x_i, mesurée le long de z_i.

corps / Paramètres	α_{i-1}	a_{i-1}	θ_i	r_{i-1}
L1	0	0	θ_1	0
L2	$\pi/2$	0	θ_2	0
L3	0	L_2	θ_3	0
L4	0	L_3	θ_4	0

TABLE 4.1 – **Paramétrage de notre mécanisme**

4.2.3 L'etablissement des relations géométriques

Conformément au paramétrage de Denavit-Hartenberg modifié, la matrice de transformation entre les corps L_{i-1} et L_i est donc :

$$^{i-1}T_i = Trans(x_{i-1}, a_{i-1})Rot(x_{i-1}, \alpha_{i-1})Trans(z_i, d_i)Rot(z_i, \bar{\omega}_i) \tag{4.2.1}$$

Qui prend la forme :

$$^0T_1 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ;^1T_2 = \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ;$$

$$^2T_3 = \begin{bmatrix} C_3 & -S_3 & 0 & L_2 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ;^3T_4 = \begin{bmatrix} C_4 & -S_4 & 0 & L_3 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

On note : $\cos\theta_i = C_i$; $\sin\theta_i = S_i$;

4.2.4 L'extraction du modèle géométrique direct

Suivant l'étude précédente, on peut poser :

$$^0T_4 = ^0T_1 *^1T_2 *^2T_3 *^3T_4 \tag{4.2.1}$$

Donc :

$$^0T_4 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & L_2 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_4 & -S_4 & 0 & L_3 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$
{}^{0}T_{4} = \begin{bmatrix}
C(t_2+t_3+t_4)C(t_1) & -S(t_2+t_3+t_4)C(t_1) & S(t_1) & C(t_1)(L_3C(t_2+t_3)+L_2C(t_2)) \\
C(t_2+t_3+t_4)S(t_1) & -S(t_2+t_3+t_4)S(t_1) & -C(t_1) & S(t_1)(L_3C(t_2+t_3)+L_2C(t_2)) \\
S(t_2+t_3+t_4) & C(t_2+t_3+t_4) & 0 & L_3S(t_2+t_3)+L_2S(t_2) \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

On note : $\theta_i = t_i$; $\cos(\theta) = C(\theta)$; $\sin(\theta) = S(\theta)$;

La position de l'effecteur final, par rapport au troisième élément, est donnée par :

$$
\begin{bmatrix} E_x \\ E_y \\ E_z \\ 1 \end{bmatrix} = {}^{0}T_{4} * \begin{bmatrix} L_4 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix}
C(t_1)(L_3C(t_2+t_3)+L_2C(t_2))+L_4C(t_2+t_3+t_4)C(t_1) \\
S(t_1)(L_3C(t_2+t_3)+L_2C(t_2))+L_4C(t_2+t_3+t_4)S(t_1) \\
L_3S(t_2+t_3)+L_2S(t_2)+L_4S(t_2+t_3+t_4) \\
1
\end{bmatrix}
$$

4.3 Le matériale utilisée (Partie hardware)

4.3.1 Bref définition hardware

Ensemble des objets, des instruments utilisés dans une usine, un service, une exploitation, etc. Ensemble d'éléments susceptibles d'être exploités, élaborés scientifiquement, d'être analysés (Le matériel d'une enquête).

4.3.2 Bras manipulateur

Ce robot (bras manipulateur) est réalisé par les deux étudiants (Mr. DILMI Ahmed et Mr. GOTIT Mahmoud) et dirigé par (M. TCHOKETCH KEBIR Sofiane), durant l'année universitaire 2015-2016, ils sont intéressés à des robots manipulateurs rigides à chaîne ouverte simple, ce robot est commandé visuellement par la caméra (Kinect) à base d'Arduino Uno.

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) conçu pour accomplir automatiquement des tâches imitant ou reproduisant, dans un domaine précis, des actions humaines. La conception de ces systèmes et l'objet d'une discipline scientifique, branche de l'automatisme nommé robotique.

Un bras manipulateur est le bras d'un robot généralement programmable, avec des fonctions similaires à un bras humain. Les liens de ce manipulateur sont reliés par des axes permettant, soit du mouvement de rotation (comme dans un robot articulé) ou de translation (linéaire) de déplacement.

Il peut être autonome ou contrêlé manuellement et peut être utilisé pour effectuer une variété de tâches avec une grande précision. Les bras manipulateurs peuvent être fixes ou mobiles (c'est-à-dire à roues) et peuvent être conçus pour des applications industrielles.

Les robots articulés sont très utilisés dans le domaine de l'industrie. Ils permettent de réaliser du soudage, de l'assemblage, de la peinture, l'usinage (perçage....) et découpage par laser.



FIGURE 4.4 – **Bras manipulateur**

4.3.3 Arduino Uno

L'Arduino est une carte électronique en Matériel Libre pour la création artistique interactive. Elle peut servir[12] :

— Pour des dispositifs interactifs autonomes simples.

— Comme interface entre capteurs/actionneurs et ordinateur.

— Comme programmateur de certains microcontrôleurs.

Arduino est une plate-forme de prototypage open source basée sur le microcontrôleur ATmega et Arduino uno est l'un des tableaux les plus utilisés dans lesquels la description détaillée est donnée [12].

Le tableau Arduino Uno possède 14 entrées/sorties de broches numériques, 6 entrées analogiques, une horloge 16 MHz, une connexion USB, une prise de courant et un bouton de réinitialisation. Généralement, il contient tout le nécessaire pour supporter le microcontrôleur [12].

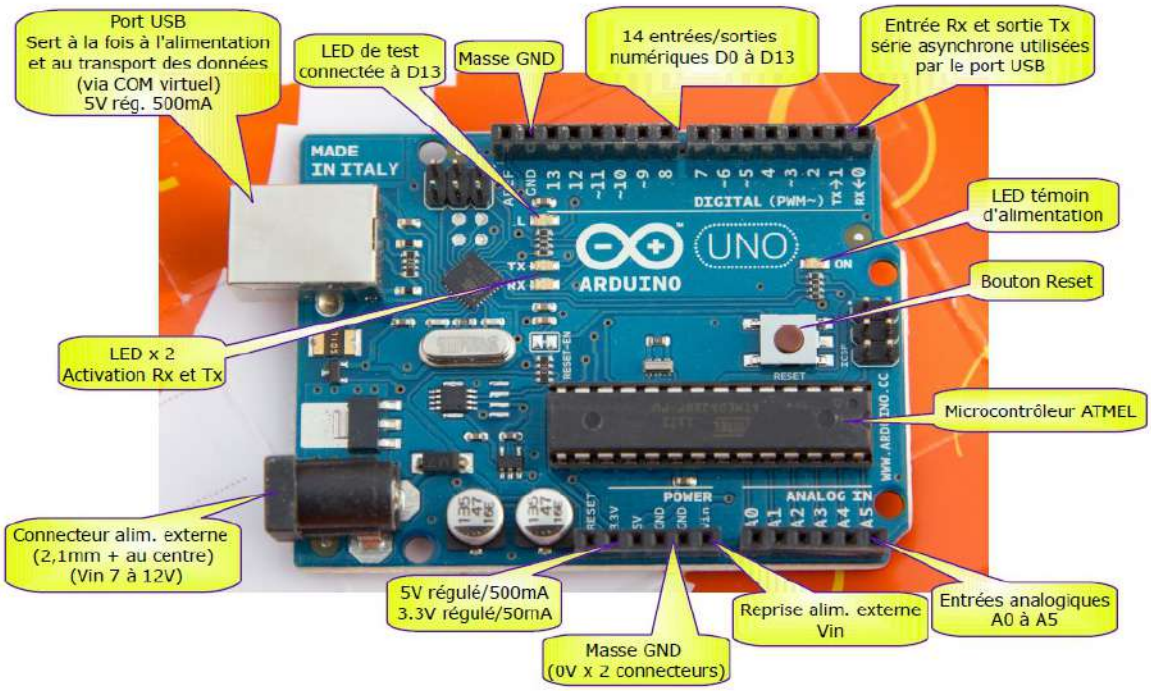


FIGURE 4.5 – **carte ARDUINO**

4.3.4 ArduinoWi-FiShield

Actuellement Arduino est introduit dans les domaines les plus différents pour différentes applications. Grâce à différents types de boucliers de cet appareil. Wi-Fi Shield permet à une carte Arduino de se connecter à Internet en utilisant les spécifications sans fil 802.11 [12].

Le bouclier Wi-Fi est connecté à la carte Arduino à l'aide d'en-têtes longs de câblage qui s'étendent à travers le bouclier. Cela maintient la disposition des broches intact et permet d'empiler un autre bouclier sur le dessus [12].

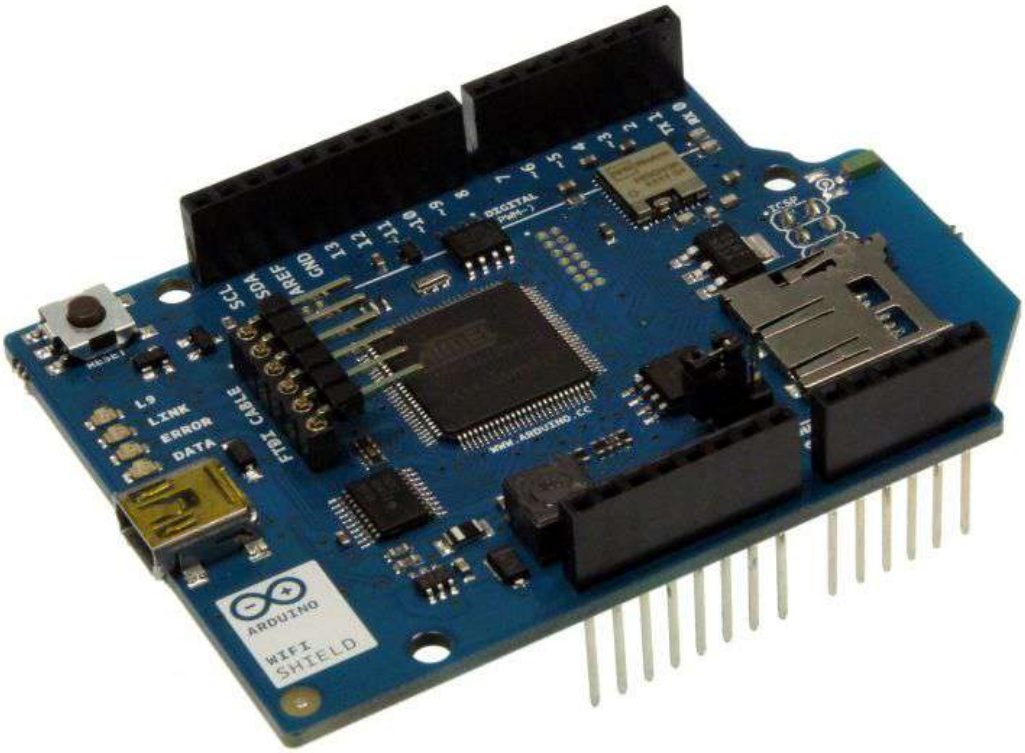


FIGURE 4.6 – **Carte ArduinoWi-FiShield**

4.3.5 Alimentation stabilisée (à partir d'une alimentation de PC)

Une alimentation stabilisée PC est un dispositif électronique, alimentation stabilisée de labo à partir d'une alimentation de PC. Elle pourra fournir des tensions de -12V, 3.3V, 5V et 12V, utilisant une régulation, permettant le réglage de la tension et du courant nécessaires au fonctionnement d'un appareil électrique ou électronique. Il est généralement conçu pour fonctionner en générateur idéal de tension et de courant.



FIGURE 4.7 – **Fonctions des connecteurs ATX 24 pins**

La relation entre la tension d'entrée V_e et celle de sortie V_s est la suivante : (V_s=$F_0 V_e R_0 I_s$). Avec Is l'intensité de sortie, F_0 le facteur de stabilisation en tension et R_0 la résistance interne de l'alimentation.

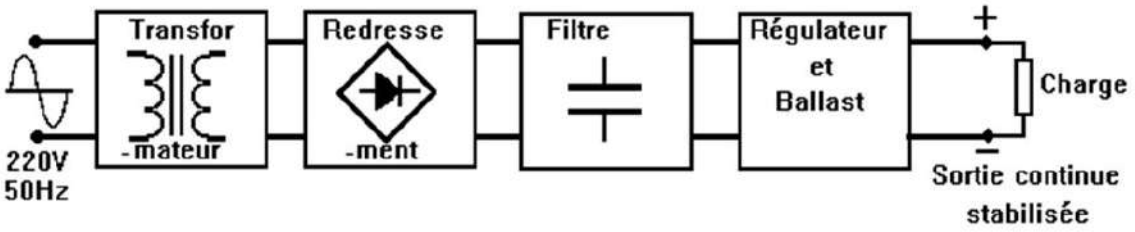


FIGURE 4.8 – **Le bloc diagramme d'une alimentation [12]**

Le transformateur remplit deux fonctions : isolement galvanique entre l'équipement et le secteur ; transformation de la tension alternative du réseau, déterminée par le rapport entre le nombre de spires (tours) du secondaire et du primaire.

Le redresseur convertit la tension alternative du secondaire du transformateur en impulsions unidirectionnelles de courant. On n'utilise plus aujourd'hui qu'un montage : le redressement

double alternances ou onde entière par pont de Graetz. Le pont de Graetz est constitué de quatre diodes disposées en losange. Il existe des ponts de diodes intégrés, c'est à dire que l'on a moulé les quatre diodes dans un seul boîtier. Ce bloc est plus facile à utiliser mais a un inconvénient. Lorsqu'une des diodes moulées est défectueuse, il faut changer l'ensemble.

4.3.6 Le Moteur à courant continu (M.C.C)

Cette partie vous présente le résumé sur les machines à courant continu.

Un moteur à courant continu est une machine qui transforme l'énergie électrique qui lui est appliquée en une énergie mécanique entraînant une charge mécanique.

— Le couple développé par un moteur à courant continu est proportionnel au courant d'induit et au flux magnétique.

— La vitesse de rotation est proportionnelle à la tension d'alimentation et inversement proportionnelle au flux magnétique.

— Le rendement est le rapport entre la puissance développée par le moteur ou puissance utile (puissance mécanique) et celle qui lui est appliquée ou puissance absorbée (puissance électrique).

Le moteur comprend :

— Un circuit magnétique comportant une partie fixe, le stator, une partie tournant, le rotor et l'entrefer l'espace entre les deux parties.

— Une source de champ magnétique nommée l'inducteur (le stator) crée par un bobinage oudes aimants permanents.

— Un circuit électrique induit (le rotor) subit les effets de ce champ magnétiques.

— Le collecteur et les balais permettent d'accéder au circuit électrique rotorique.

Pour commander un moteur à courant continu (M.C.C) , il y'a plusieurs montages ainsi que plusieurs types de commande. Et pour ce projet, on a utilisé Le transistor, il est commandé par une sortie de la carte Arduino via la résistance sur la base. Lorsque l'état de la sortie est au niveau 0, le transistor est bloqué et le courant ne le traverse pas. Le moteur ne tourne pas. Lorsque la sortie vaut 1, le transistor est commandé et devient saturé, c'est-à-dire qu'il laisse

passer le courant et le moteur se met à tourner.

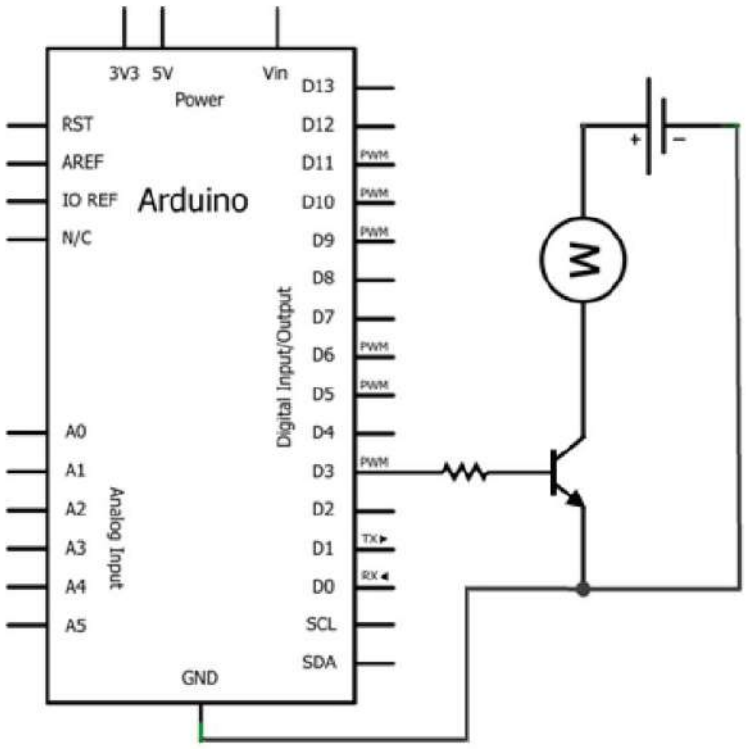


FIGURE 4.9 – **Circuit commande M.C.C à base de transistor**

4.3.7 Le Routeur Wi-Fi

Dans les réseaux informatiques, un point d'accès sans fil (WAP1 pour Wireless Access Point en anglais) est un dispositif qui permet aux périphériques sans fil de se connecter à un réseau câblé ou au réseau Internet à l'aide d'une connexion radio. Le point d'accès en tant que dispositif autonome est habituellement relié à un routeur (par l'intermédiaire d'un réseau câblé), mais il peut aussi faire partie intégrante du routeur lui-même.

Les point d'accès sans fil, les plus répandus, sont de type Wi-Fi (box ou hot-spot Wi-Fi).

Les points d'accès publics ou privés des FAI sont généralement compatibles avec les normes Wi-Fi IEEE 802.11n ou IEEE 802.11ac et offrent des débits supérieurs à 100 Mb/s.

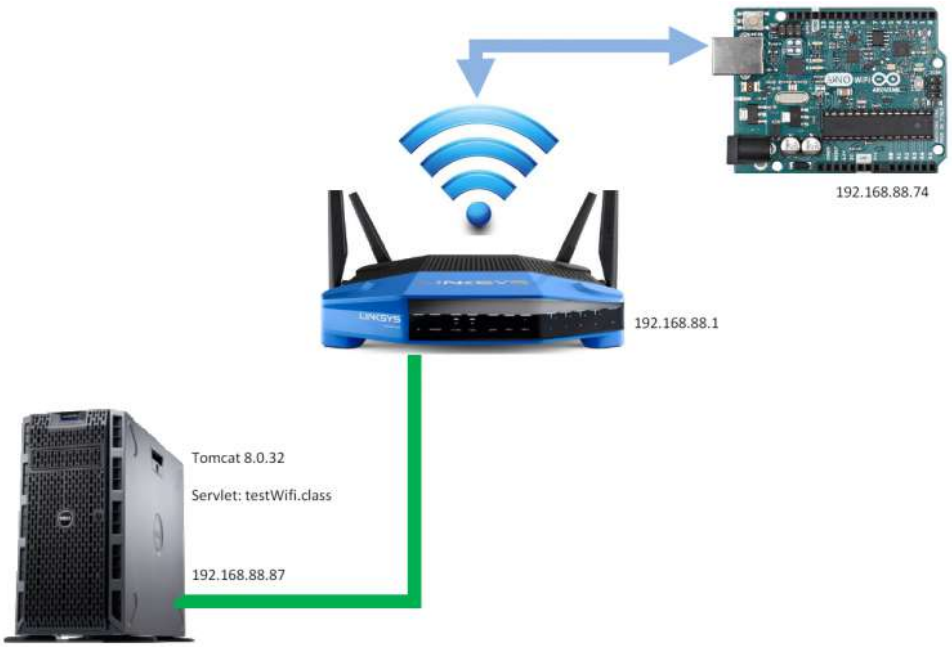


FIGURE 4.10 – **point d'accées sans fil**

4.4 Partie software

4.4.1 MATLAB

En utilisant l'outil de programmation MATLAB, est pour le but de commander des servo-moteur et moteur à courant continu à l'aide d'une commande (carte ARDUINO,...), le langage MATLAB, ou (MATRIX LABORATORY), est un langage de programmation et de développement de quatrième génération.

Développé par la société (The Math Works), MATLAB permet la manipulation de matrice, afficher des courbes et des données, mettre en oeuvre des algorithmes, créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran.

Les utilisateurs de MATLAB sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que pour la recherche. Au cours de notre projet, nous avons utilisé MATLAB R2015b qui est plus développé que les anciennes versions surtout dans le domaine de l'ingénierie.

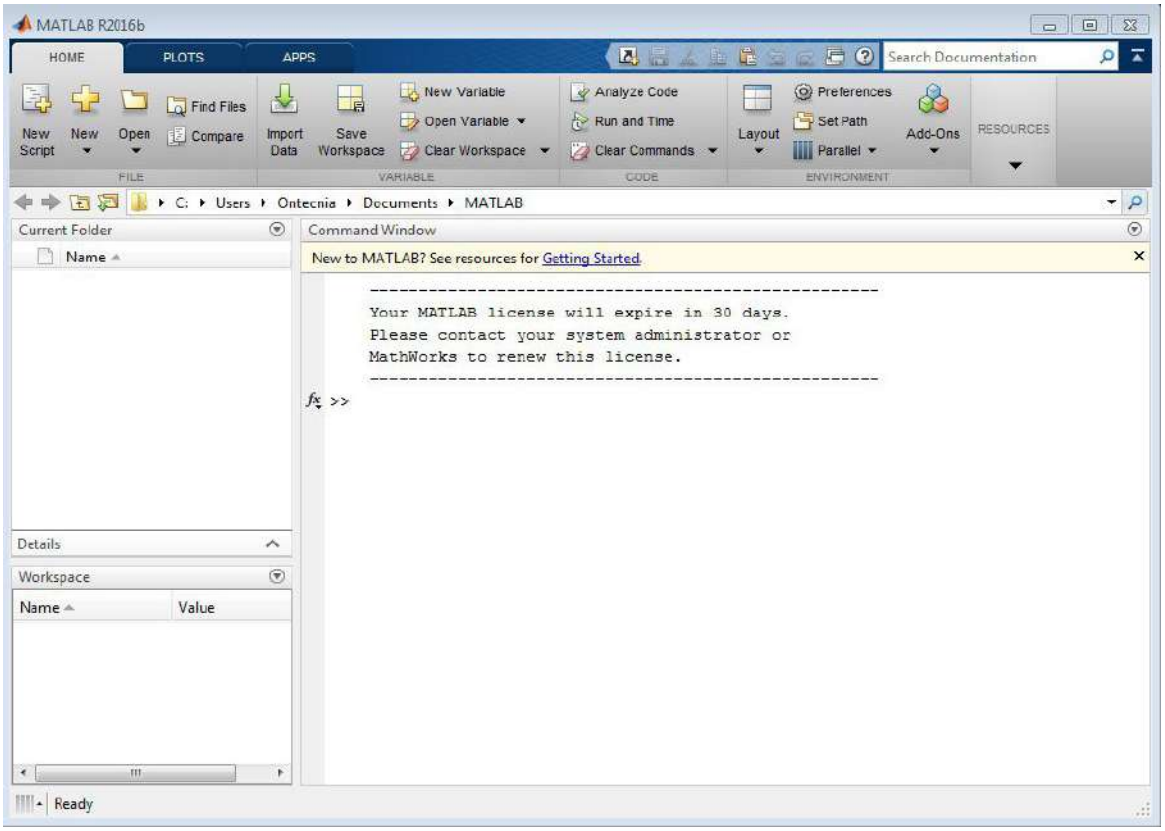


FIGURE 4.11 – **interface Matlab2016b**

4.4.2 Le langage de programmation ARDUINO

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme Arduino s'effectue de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres. Voyons plus en détail la structure d'un programme écrit en Arduino.

Il y a deux façons de programmer les cartes Arduino.

— Soit avec du code (pur) en ligne de code type langage C ;

— Soit sous forme de blocs type puzzle, nommé Ardublock.

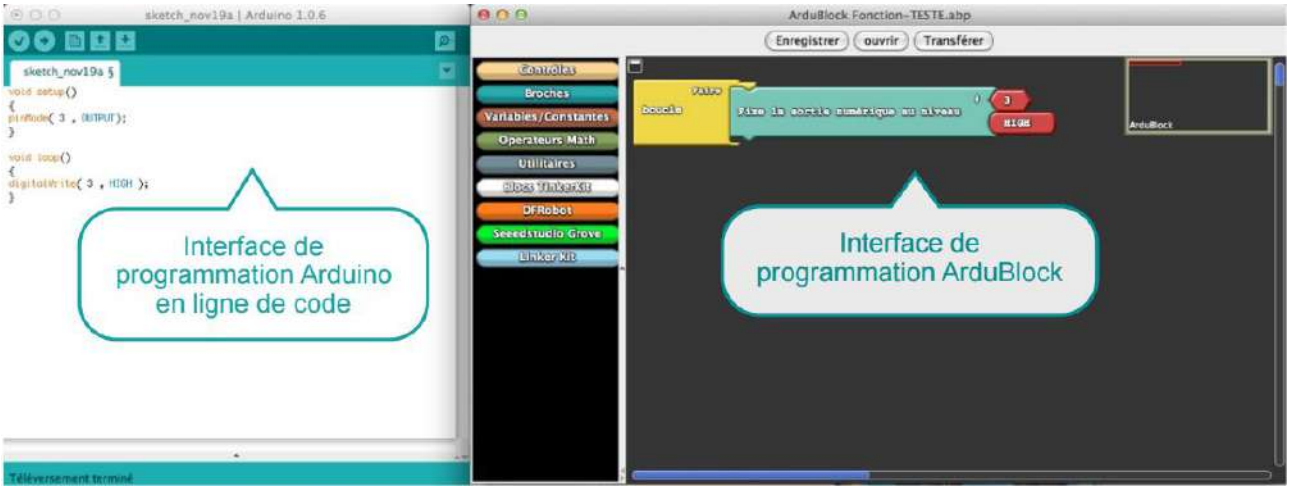


FIGURE 4.12 – **langage de programmation ARDUINO**

4.5 La Réalisation De La Commande

4.5.1 Méthodologie de l'approche proposée

Notre méthode consiste à générer et réaliser la commande d'un bras manipulateur, pour atteindre cet objectif, trois étapes sont nécessaires : simulation de l'objectif, génération et l'envoi de la commande via un réseau Wi-Fi, réception et exécution de commande.

La figure suivante illustre un schéma synoptique de l'approche proposée montrant les différents blocs constituant notre projet.

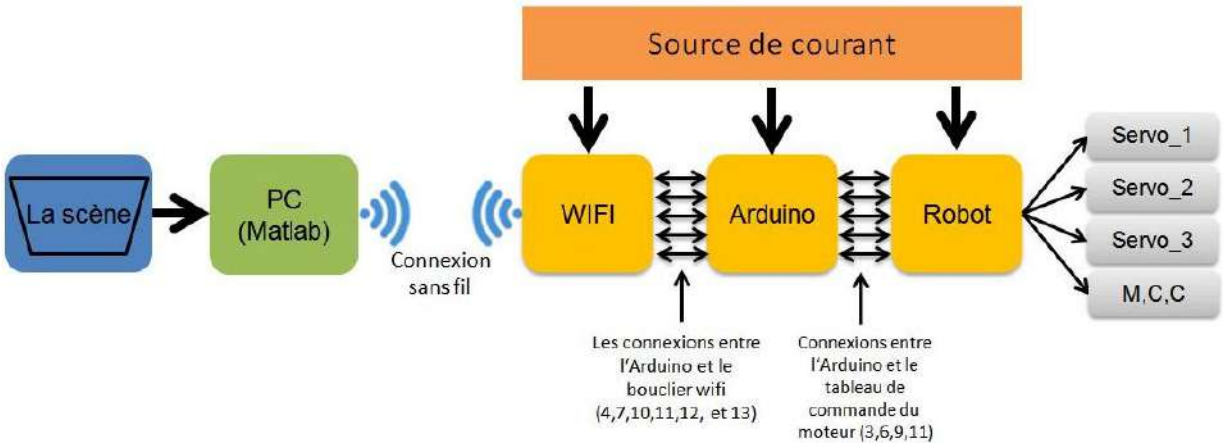


FIGURE 4.13 – **Synoptique du système conçu [12]**

4.5.2 Réalisation, Tests, Résultats et discussion

L'implémentation des déférentes parties (soft et hardes) avec certains niveaux d'adaptation (câblage) conduit vers un produit mécatronique représenté par la figure suivante :

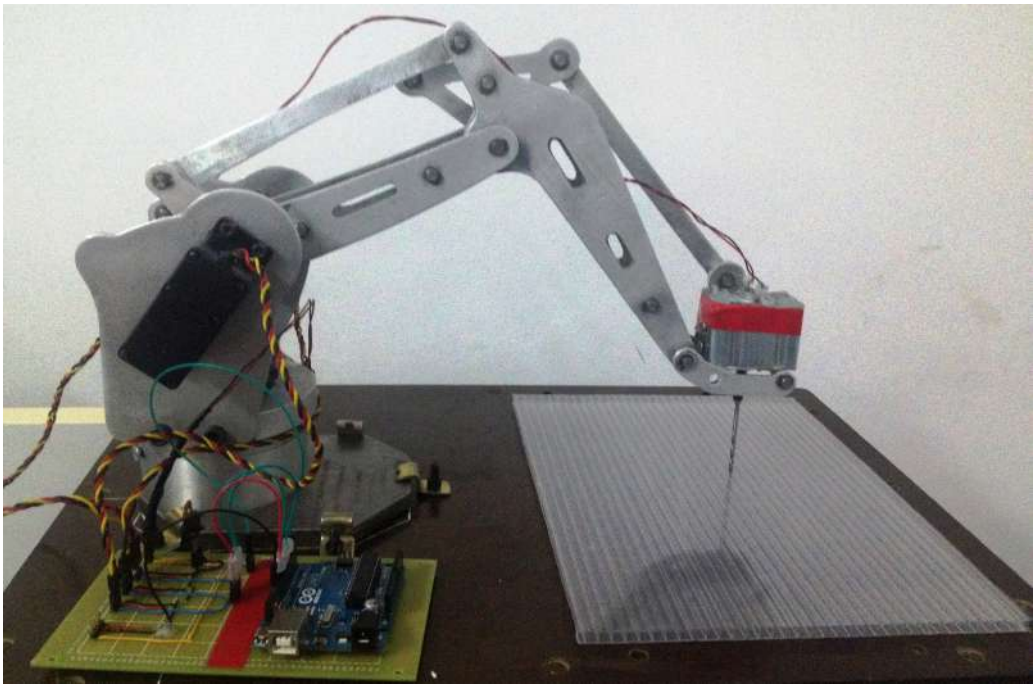


FIGURE 4.14 – **Représentation final de notre projet**

Ce produit capable de faire quelque tâche bien déterminer avec un certain degré d'exigence, de façons général, la liberté de notre bras dépond :

— De la géométrie du robot,

— De la longueur des segments,

— Du débattement des articulations (limité par des butées).

Exemples :

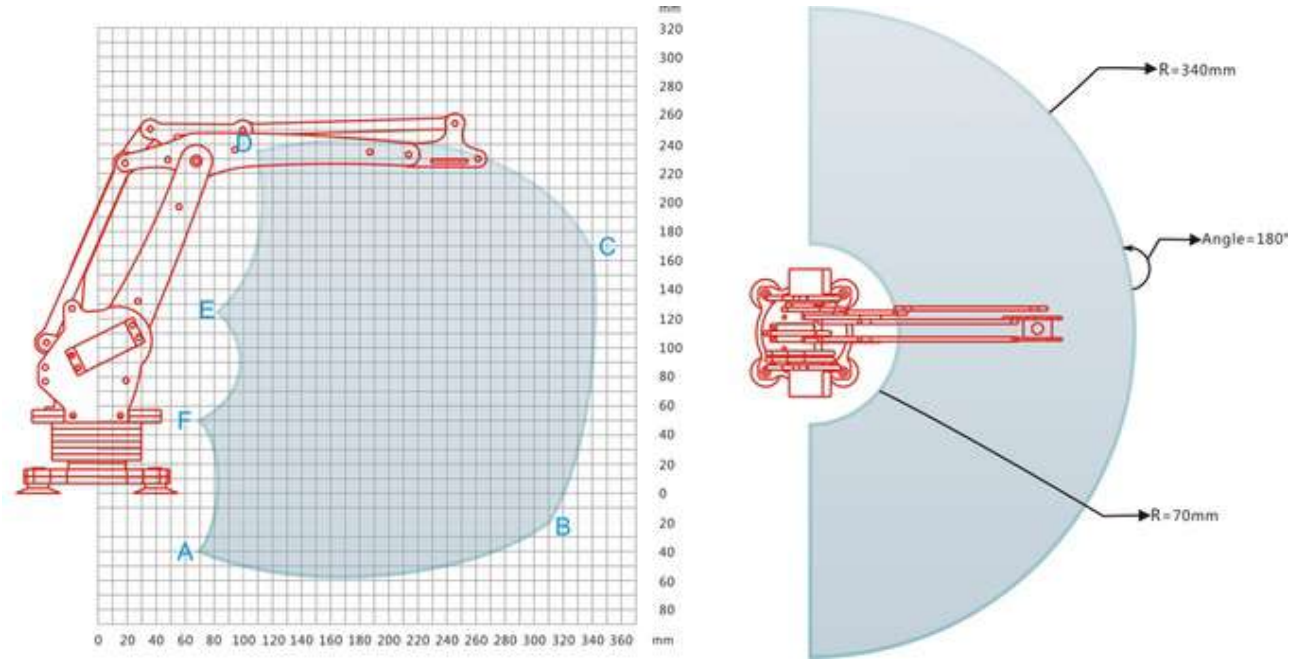


FIGURE 4.15 – **Espace de travail**

Tests, Résultats et discussion

La simulation de l'objectif fait une partie importante dans notre projet, cette simulation est traduite dans une interface Matlab voir la figure suivante :

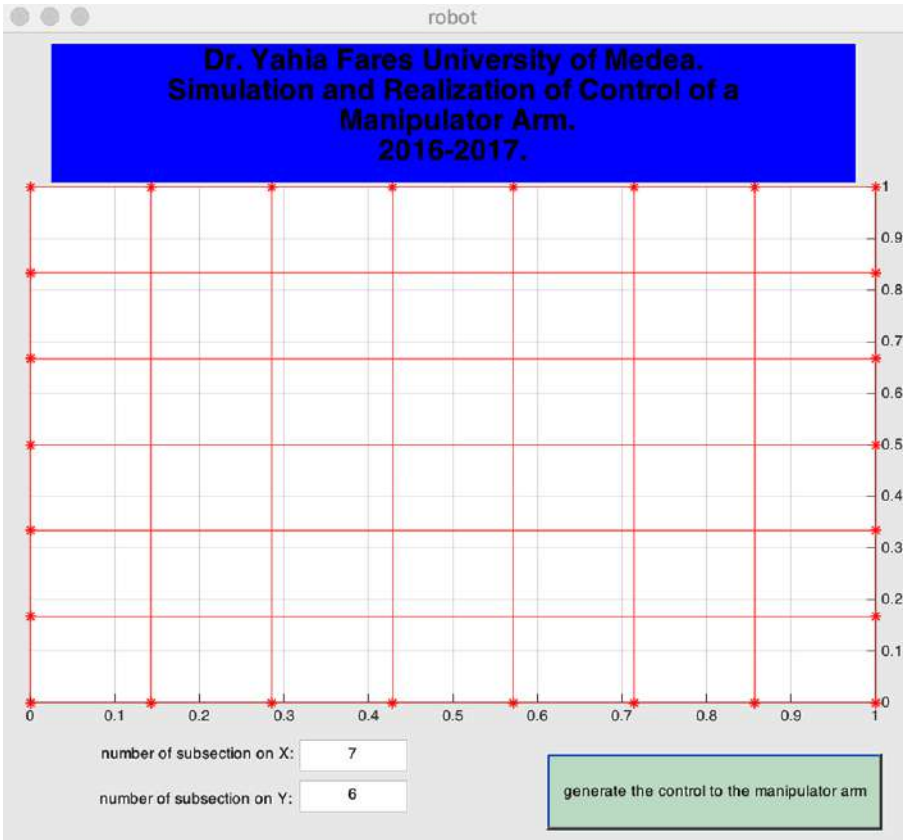


FIGURE 4.16 – **Interface de la génération de la commande de notre Robot**

Cette interface de simulation représente la scène réelle de l'exécution (perçage), une fois que l'utilisateur termine sa simulation, il peut générer l'exécution pour mettre cette simulation sur le plan réel.

L'excusions de dizaines commande simuler sur l'interface de Matlab est représenté par la figure suivante :

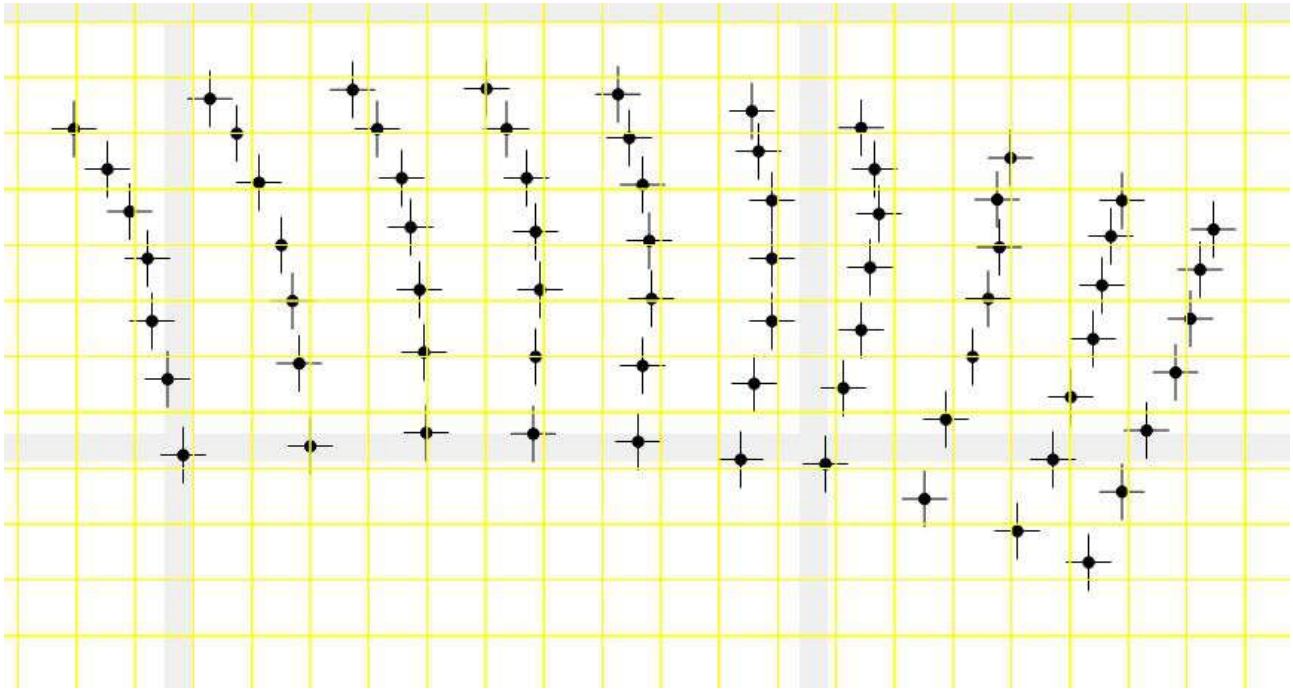


FIGURE 4.17 – **Point percée dans l'espace de travail**

Nous constatons que notre bras suit bien les cibles de perçage établis, et ça signifier que la commande par langage Matlab est réalisé comme nous avons prévu.

Les résultats du test permettant de conclure que le robot manipulateur suit les cibles, ce qui correspond au interface graphique du Matlab, avec une précision importante.

4.6 Conclusion

Nous avons décrit dans ce chapitre les étapes de notre projet, telle que les circuits utilisés ainsi que les schémas jusqu'à l'étude et l'implémentation des logiciels de simulation et programmation d'une part, d'autre part un programmation capable d'adapter à une commande électronique afin de construire un système Mécatronique qui répond à nos besoins.

Enfin on a pu faire des semples tests sur le bras malgré les difficultés qu'on a trouvées dans ce travail.

Conclusion général

Dans ce projet, nous avons apporté une valeur ajoutée en matière de performance d'un autre travail de recherche réalisé antérieurement à l'université de Médéa, dont le sujet est l'étude et réalisation d'un robot manipulateur.

Dans la première partie de ce travail une introduction aux différents modèles des robots utilisés suivie par une étude approfondi et minutieuse du projet réalisé à savoir le travail de base. Dans ce contexte nous avons identifié tous les aspects importants du travail ainsi que les nouvelles pistes qui pourront faire l'objet d'une recherche poussée, à l'instar de l'amélioration de la précision ciblée dans notre travail.

Une deuxième partie a été consacrée à la partie software dans le but de commander le bras manipulateur en utilisant deux logiciels différents : le logiciel Arduino et le logiciel Matlab. Ce dernier est utilisé pour réaliser un interface graphique qui nous a beaucoup aidés à la programmation en exploitant ces deux logiciels.

La programmation via le langage Arduino est une tâche très importante, elle nous a conduit à mieux maitriser le coté software de manière à ce qu'on puisse traduire les mouvements du bras manipulateur en instructions exécutables par le logiciel.

Afin d'atteindre notre objectif de ce projet, qui est l'amélioration de la précision, notre touche s'est articulé sur l'élaboration d'un espace de travail modélisé par Matlab et représenté par un interface graphique lié directement à la carte Arduino responsable de la commande du robot.

En effet ce projet de réalisation nous a permis de traiter des problèmes d'ordre pratique (problèmes mécaniques, programmation) et de vérifier des connaissances théoriques acquises tout au long de notre cursus universitaires. En d'autre partie ce projet nous a permet d'acquérir

une bonne expérience pratique et théorique dans le monde de la robotique.

L'analyse des travaux présentés dans ce mémoire nous permet d'envisager des perspectives ainsi que les pistes de recherche afin d'améliorer les performances de cette œuvre scientifique tel que :

— Introduire les techniques de l'intelligence artificielle dans la commande de ce robot dans le but d'avoir plus de performance.

— Introduire les différentes liaisons sans fil pour la commande.

— Introduire une liaison optique pour la commande.

— Malgré tous les efforts fournis ce bras nécessite quelques améliorations pour atteindre une précision dans les standards.

Annexe A

Fichier technique

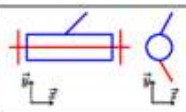
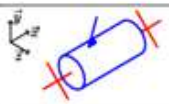
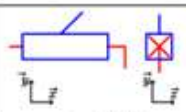
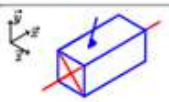
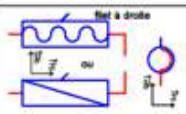
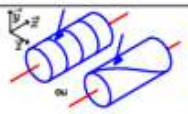
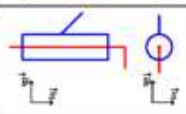
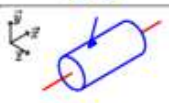
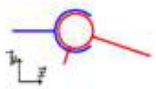
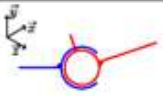
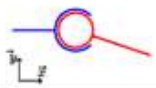
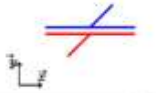
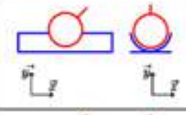
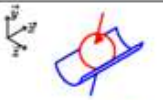
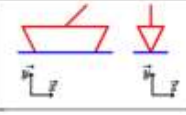
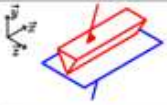
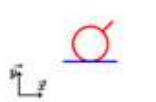
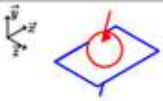
Nom	point d'application	ddl	Représentation plane	Représentation 3D	Torseur cinématique	Torseur des AM
Encastrement	tout point de l'espace	0			$\begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & M_x \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Pivot d'axe $(A, \vec{x})$	tout point de l'axe	1			$\begin{Bmatrix} \omega_x & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & 0 \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Glissière de direction $\vec{x}$	tout point de l'espace	1			$\begin{Bmatrix} 0 & V_x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & M_x \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Hélicoïdale d'axe $(A, \vec{x})$	tout point de l'axe	1			$\begin{Bmatrix} \omega_x & p.\omega_x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & p.F_x \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Pivot glissant d'axe $(A, \vec{x})$	tout point de l'axe	2			$\begin{Bmatrix} \omega_x & V_x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & 0 \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Rotule à doigt de centre A bloquée en $\vec{x}$	centre de la liaison	2			$\begin{Bmatrix} 0 & 0 \\ \omega_y & 0 \\ \omega_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & M_x \\ F_y & 0 \\ F_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$
Rotule de centre A	centre de la liaison	3			$\begin{Bmatrix} \omega_x & 0 \\ \omega_y & 0 \\ \omega_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} F_x & 0 \\ F_y & 0 \\ F_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$
Plane de normale $\vec{y}$	tout point de l'espace	3			$\begin{Bmatrix} 0 & V_x \\ \omega_y & 0 \\ 0 & V_z \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & M_x \\ F_y & 0 \\ 0 & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Annulaire en A d'axe $(A, \vec{x})$	au centre de la liaison	4			$\begin{Bmatrix} \omega_x & V_x \\ \omega_y & 0 \\ \omega_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & 0 \\ F_y & 0 \\ F_z & 0 \end{Bmatrix}_{A,\mathcal{R}}$
Rectiligne de ligne $(A, \vec{x})$ et de normale $\vec{y}$	tout point du plan $(A, \vec{x}, \vec{y})$	4			$\begin{Bmatrix} \omega_x & V_x \\ \omega_y & 0 \\ 0 & V_z \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & 0 \\ F_y & 0 \\ 0 & M_z \end{Bmatrix}_{A,\mathcal{R}}$
Ponctuelle en A de normale $\vec{y}$	tout point de la normale au contact	5			$\begin{Bmatrix} \omega_x & V_x \\ \omega_y & 0 \\ \omega_z & V_z \end{Bmatrix}_{A,\mathcal{R}}$	$\begin{Bmatrix} 0 & 0 \\ F_y & 0 \\ 0 & 0 \end{Bmatrix}_{A,\mathcal{R}}$

FIGURE A.1 – **Les différentes représentations des liaisons mécaniques**

Annexe B

Programme arduino de commandé de notre robot

Exemple : chaîne d'envoi et de réception UDP WiFi

```
#include <SPI.h>
#include <WiFi.h>
#include <WiFiUdp.h>
#ifndef HAVE_HWSERIAL1
#include "SoftwareSerial.h"
SoftwareSerial Serial1(6, 7); // RX, TX
#endif
char ssid[] = "DJAWEB_15BF4";            // your network SSID (name)
char pass[] = "A2EF6F582D43C";        // your network password
int status = WL_IDLE_STATUS;     // the Wifi radio's status
unsigned int localPort = 10002;  // local port to listen on
char packetBuffer[255];          // buffer to hold incoming packet
char ReplyBuffer[] = "ACK";      // a string to send back
void setup() {
  // initialize serial for debugging
  Serial.begin(115200);
  // initialize serial for ESP module
  Serial1.begin(9600);
```

```
  // initialize ESP module
//  WiFi.init( Serial1);
  // check for the presence of the shield:
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    // don't continue:
    while (true);
  }
  // attempt to connect to WiFi network
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network
    status = WiFi.begin(ssid, pass);
  }
    Serial.println("Connected to wifi");
  printWifiStatus();
  Serial.println(" \ nStarting connection to server...");
  // if you get a connection, report back via serial:
  Udp.begin(localPort);
    Serial.print("Listening on port ");
  Serial.println(localPort);
}
void loop() {
  // if there's data available, read a packet
  int packetSize = Udp.parsePacket();
  if (packetSize) {
    Serial.print("Received packet of size ");
```

```
    Serial.println(packetSize);
    Serial.print("From ");
    IPAddress remoteIp = Udp.remoteIP();
    Serial.print(remoteIp);
    Serial.print(", port ");
    Serial.println(Udp.remotePort());
    // read the packet into packetBufffer
    int len = Udp.read(packetBuffer, 255);
    if (len > 0) {
      packetBuffer[len] = 0;
    }
    Serial.println("Contents:");
    Serial.println(packetBuffer);

    // send a reply, to the IP address and port that sent us the packet we received
    Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
    Udp.write(ReplyBuffer);
    Udp.endPacket();
  }
}
void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  // print your WiFi shield's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
```

```
  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
}
```

Bibliographie

[1] F. Z. BOUZIANE, "Rétro-conception du bras horizontal de robot manipulateur de la cellule flexible (tlemcen)," Ph.D. dissertation, 2014.

[2] M. Priel, *Les Robots industriels : caracteristiques performances et choix.* AFNOR, 1990.

[3] A. D. Kahina, A. Z. Meziane, and A. Hocine, "Conception et réalisation d'un bras manipulateur commandé par api."

[4] R. SAADI and N. SALHI, "Réalisation de carte à microcontrôleur pour le contrôle de bras manipulateur via un pc," Master's thesis, 2014.

[5] A. Liégeois, "Modélisation et commande des robots manipulateurs," *Techniques de l'ingénieur. Informatique industrielle*, vol. 6, no. S7730, pp. 1–16, 2000.

[6] N. Nombrail, "Analyse et commande d'une famille de robots manipulateurs à structure parallele et redondante," Ph.D. dissertation, 1993.

[7] M. Ait-Ahmed, "Contribution à la modélisation géométrique et dynamique des robots," Ph.D. dissertation, 1993.

[8] J. Qin, "Commande hybride position/force robuste d'un robot manipulateur utilisé en usinageet/ou en soudage," Ph.D. dissertation, Paris, ENSAM, 2013.

[9] W. Khalil and E. Dombre, *Modélisation, identification et commande des robots.* Hermès science publ., 1999.

[10] planete sciences, *Le grand livre d'Arduino.* Editions Eyrolles, 2015.

[11] planete sciences.org, *Commander des servomoteurs classiques et a rotation continue*, 2017.

[12] S. T. Kebir, M. Bouhedda, S. Mekaoui, M. Guesmi, and A. Douakh, "Gesture control of mobile robot based arduino microcontroller," in *Modelling, Identification and Control (ICMIC), 2016 8th International Conference on.* IEEE, 2016, pp. 1081–1085.

ملخص:

ذراع المناورة هو روبوت قابل للبرمجة عموما، مع وظائف مماثلة لذراع الإنسان. ترتبط وصلات هذا الذراع المناور بواسطة محاور مما يجعل من الممكن لجعل الحركات التناوب أو متعدية. والهدف الرئيسي من هذه الأطروحة هو محاكاة وتحقيق السيطرة على ذراع مناور مع درجة معينة من المتطلبات والدقة.

كلمات مفتاحية: ذراع مناور، اردوينو، واي فاي شيلد، النمذجة والسيطرة على الروبوتات، ...

Résumé :

Un bras manipulateur est un robot généralement programmable, avec des fonctions similaires à un bras humain. Les liens de ce bras manipulateur sont reliés par des axes permettant de faire des mouvements de rotation ou de translation. L'objectif principal de ce mémoire est de simuler et réaliser commande d'un bras manipulateur avec un certain degré d'exigence et de précision.

Mots clés : bras manipulateur, Arduino, Wi-Fi-Shield, modélisation et commande des robots, ...

Abstract :

A manipulator arm is a generally programmable robot, with functions similar to a human arm. The links of this manipulator arm are connected by axes making it possible to make rotational or translational movements. The main objective of this thesis is to simulate and realize control of a manipulator arm with a certain degree of requirement and precision.

Key words: arm manipulator, Arduino, Wi-Fi-Shield, modeling and control of robots, ...