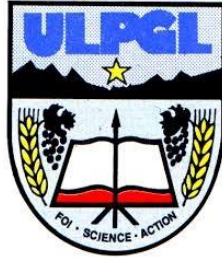


UNIVERSITÉ LIBRE DES PAYS DES GRANDS LACS

ULPGL/GOMA



BP. 368 GOMA

www.ulpgl.net

FACULTÉ DE SCIENCES ET TECHNOLOGIES APPLIQUÉES

Département de Génie Électrique et Informatique

Conception et réalisation d'un bras robotisé à commande manuelle et par application mobile.

Par : KAYIHEMBAKO MUTHETHE Robert

Travail présenté en vue de l'obtention du Diplôme de d'Ingénieur en Sciences et Technologies Appliquées dans le département de Génie Electrique et Informatique

Option : Electronique

Directeur : Prof. Ing. Tech. Olivier BARAKA MUSHAGE

Encadreur: CT. Ing. Raoul IRENGE BAGUMA

Année académique 2022 – 2023

Epigraphe

« Un robot n'est pas tout à fait une machine. Un robot est une machine fabriquée pour imiter de son mieux l'être humain. »

Isaac Asimov

Dédicaces

A nos très chers parents Edmond et Suzanne

Remerciements

Au cours de ce projet, nous avons bénéficié de la précieuse supervision du Professeur Olivier BARAKA MUSHAGE, notre Directeur, et nous souhaitons exprimer notre profonde gratitude envers lui pour son encadrement, le partage de ses connaissances, et les précieux conseils qu'il nous a prodigués tout au long de cette recherche et de notre parcours académique. Nous tenons également à remercier chaleureusement le Chef des Travaux Raoul IRENGE BAGUMA pour ses orientations essentielles lors de l'élaboration de ce travail.

Nous aimerions adresser nos remerciements les plus sincères à nos parents, PALUKU MALIRO Edmond et KAHINDO KASONGYA Suzanne, pour leur inestimable soutien. Leur dévouement et leurs sacrifices inlassables ont été déterminants pour nous offrir toutes les conditions nécessaires afin que nous puissions atteindre nos objectifs, réaliser nos rêves et devenir la personne que nous sommes aujourd'hui.

Enfin, nous voulons exprimer notre profonde reconnaissance à tous nos amis pour leur soutien et leur amitié inestimable.

KAYIHEMBAKO MUTHETHE Robert

Résumé

Dans l'optique de l'automatisation des systèmes de production pour une meilleure efficacité dans le tri et la mise en caisse des bouteilles, ce travail a pour but de concevoir et de modéliser la commande d'un bras robotique. Cela pour permettre l'automatisation du tri d'objets dans les entreprises locales de fabrication de boissons embouteillées. Nous avons réalisé un bras robotisé à commande manuelle et par application mobile. Ce bras robotisé est capable, en fonctionnement automatique, de faire le tri de bouteille en fonction de leurs couleurs. En fonctionnement automatique le bras est doté de capacité d'évitement d'obstacles basées sur la logique floue. Cela est le fruit d'un travail qui combine une modélisation du bras robotisé, la conception d'une application Android avec MIT app inventor pour la commande. Nous avons aussi fait la conception d'un régulateur flou avec MATLAB pour l'évitement d'obstacles ainsi que d'un programme permettant à la fois une commande manuelle, une commande automatique et une commande par application mobile. Tout ce système fonctionne sur base d'une carte Arduino MEGA 2560 avec un ensemble de capteurs de distances et de couleurs, des potentiomètres et un module Bluetooth pour assurer non seulement la commande manuelle mais aussi la commande par application mobile.

Mots clés : bras robotisé, tri d'objets, évitement d'obstacles, appinventor, logique floue

Abstract

In order to have automating production systems for better efficiency in the sorting and crating of bottles, this work aims to design and model the control of a robotic arm. The aim of the work is to enable the automation of object sorting in local bottled beverage manufacturing companies. We have created a robotic arm with manual control and a mobile application. . This robotic arm is capable of sorting bottles according to their colors. In automatic operation, the arm is equipped with obstacle avoidance capability based on fuzzy logic. This is the result of a work that combines modeling of the robotic arm and the design of an Android application with MIT appinventor for the control. We designed also a fuzzy controller with Matlab for obstacle avoidance as well as we developed a program for allowing both manual control, automatic control and control via mobile application. This whole system runs on an arduino-mega 2560 microcontroller with a set of distance and color sensors, potentiometers and a Bluetooth module to provide not only manual control but also control via mobile app.

Keys words: Robotic arm, Object Sorting, Obstacle Avoidance, Appinventor

Table des matières

Epigraphe	i
Dédicaces	ii
Remerciements.....	iii
Résumé.....	iv
Abstract	v
Table des matières	vi
Liste des abréviations.....	ix
Liste des tableaux.....	x
Liste des figures	xi
0. INTRODUCTION GÉNÉRALE	1
0.1. Contexte	1
0.2. Identification et formulation du problème	2
0.3. Questions de recherche.....	3
0.4. Formulation des hypothèses	3
0.5. Justification du choix du sujet et motivations	4
0.6. Énoncé des objectifs de recherche	5
0.7. Méthodologie	5
0.8. Structure du mémoire	6
Chapitre 1 : GÉNÉRALITÉS LES BRAS ROBOTISÉS	7
1.1 INTRODUCTION.....	7
1.2 LA ROBOTIQUE	7
1.2.1 GENERALITES SUR LES ROBOTS.....	7

1.2.2	TYPES DE ROBOTS	9
1.2.3	CLASSIFICATION DE ROBOTS MANIPULATEURS INDUSTRIELS	10
1.3	LES BRAS ROBOTIQUES	15
1.3.1	LES ROBOTS MANIPULATEURS.....	15
1.3.2	DEGRÉS DE LIBERTÉ	15
1.3.3	CONSTITUANTS D'UN BRAS ROBOTISÉS	16
1.3.4	COMMANDES DES BRAS ROBOTISÉS.....	19
1.4	CONCLUSION PARTIELLE.....	22
Chapitre 2 MODÉLISATION DU BRAS MANIPULATEUR		23
2.1	INTRODUCTION.....	23
2.2	TYPES DE MODELES DE TRANSFORMATION	23
2.3	CONSTITUTION GÉOMÉTRIQUE DES ROBOTS MANIPULATEURS	24
2.4	MODELE GEOMETRIQUE DIRECT.....	27
2.5	MODELE GEOMETRIQUE INVERSE	30
2.5.1	LA MÉTHODE DE PAUL.....	31
2.6	MODELE CINEMATIQUE DIRECT.....	32
2.7	MODELE CINEMATIQUE INVERSE	35
2.8	MODELE DYNAMIQUE.....	36
2.8.1	FORMALISME DE LAGRANGE.....	37
2.8.2	L'ÉNERGIE CINÉTIQUE	38
2.8.3	L'ÉNERGIE POTENTIEL	38
2.8.4	EQUATION DU MOUVEMENT D'UN ROBOT MANIPULATEUR.....	39
2.9	MODÉLISATION DU ROBOT ÉTUDIÉ.....	41
2.9.1	MODELE GEOMETRIQUE DIRECTE	42
2.9.2	MODELE GEOMETRIQUE INVERSE	45
2.9.3	MODELE CINEMATIQUE DIRECT.....	47
2.9.4	MODELE CINEMATIQUE INVERSE	48
2.9.5	MODELE DYNAMIQUE	49
2.10	CONCLUSION PARTIELLE	49

Chapitre 3 : CONCEPTION DU BRAS ROBOTISÉ	51
3.1 INTRODUCTION.....	51
3.2 CONSTITUTION DU BRAS ROBOTISÉ.....	51
3.2.1 LA CARTE ARDUINO.....	51
3.2.2 LES ACTIONNEURS	53
3.2.3 LE SYSTEME MECANIQUE ARTICULÉ.....	55
3.2.4 ORGANE TERMINALE.....	57
3.2.5 LES CAPTEURS	58
3.3 TYPE DE COMMANDE DU BRAS ROBOTISÉ.....	62
3.3.1 COMMANDE MANUELLE.....	62
3.3.2 COMMANDE PAR APPLICATION MOBILE	63
3.3.3 COMMANDE AUTOMATIQUE	70
3.4 REGULATEUR A LOGIQUE FLOU	70
3.4.1 PRINCIPE D'UN REGULATEUR FLOU.....	71
3.4.2 CONCEPTION DU REGULATEUR FLOU	72
3.5 CONCLUSION PARTIELLE.....	76
Chapitre 4 : REALISATION DU BRAS ROBOTISÉ	77
4.1 INTRODUCTION.....	77
4.2 STRUCTURE MECANIQUE DU BRAS ROBOTISÉ.....	77
4.3 MONTAGE ELECTRIQUE DU BRAS ROBOTISÉ	82
4.4 APPLICATION MOBILE	84
4.5 COMMANDE DU BRAS ROBOTISÉ	87
4.6 CONCLUSION PARTIELLE.....	88
CONCLUSION GENERALE	89
Bibliographie.....	92
ANNEXE1: Code arduino du programme.....	95
ANNEXE2: COUT DES MATERIAUX.	111

Liste des abréviations

3D = Trois dimensions

Ddl = Degré de liberté

E/S = Entrée / Sortie

MCD = Modèle Cinématique Direct

MCI = Modèle Cinématique Inverse

MDD = Modèle Dynamique Direct

MGD = Modèle Géométrique Direct

MGI = Modèle Géométrique Inverse

MHz : Mégahertz

SCARA : Selective Compliance Articulated Robot for Assemblage

SMA : Système mécanique articulé

P = Prismatique

PWM : Pulse With Modulation

R = Rotoïde

RPP (ou PRP) = Une articulation rotoïde et deux articulations prismatiques

RRP = Deux articulations rotoïdes et une articulation prismatique

RRR = Trois articulations rotoïdes consécutives

Liste des tableaux

<i>Tableau 1 : tableau donnant les paramètres de Denavit-Hartenberg du robot étudié</i>	<i>42</i>
<i>Tableau 2 : caractéristiques de l'arduino-mega</i>	<i>53</i>
<i>Tableau 3 : tableau des caractéristiques des servomoteurs SG90 et MG996R</i>	<i>54</i>
<i>Tableau 4 :tableau regroupant les caractéristiques du capteurs HC SR 04</i>	<i>60</i>

Liste des figures

Figure 1 : images d'illustration des robots humanoïdes	8
Figure 2 : illustration de la structure cartésienne	11
Figure 3 : volume de travail pour une structure cartésienne	11
Figure 4 : illustration de la structure cylindrique	12
Figure 5 : volume de travail de la structure cylindrique	12
Figure 6 : images d'illustration de la structure polaire et de son espace de travail	13
Figure 7 : images d'illustration de la structure SCARA	13
Figure 8 : illustration du volume de travail de la structure SCARA	14
Figure 9 : images d'illustration de la structure anthropomorphe et de son espace de travail	14
Figure 10 : images d'illustration du bras robotique	15
Figure 11 : images d'illustration d'un bras robotique avec ses constituants	16
Figure 12 : image d'illustration de joints Rotoïde	17
Figure 13 : image d'illustration de joints prismatiques	18
Figure 14 : image d'illustration d'un effecteur	18
Figure 15 : structure d'un bras manipulateur série	24
Figure 16 : illustration de repères pour lesquels on fait la transformation	26
Figure 18: illustration du bras robotisés RRR avec ses paramètres	28
Figure 19: illustrations des éléments de la matrice de transformation sur leurs repères	29
Figure 20: illustration du passage du modèle direct vers le modèle inverse	30
Figure 21 : illustration de l'orientation de l'accélération angulaire	33
Figure 22: illustration de l'orientation de l'accélération angulaire	34
Figure 23: illustration du bras robotisés RRR avec ses paramètres	42
Figure 24: illustration de la position de du bras robotique après application des vitesses angulaires de test	44
Figure 25: illustration de la position de du bras robotique après application de deuxièmes valeurs de vitesses angulaires de test	44

<i>Figure 26: illustration de la position de du bras robotique après application des vitesses angulaires de test</i>	46
<i>Figure 27: illustration de la position de du bras robotique après application de deuxièmes valeurs de vitesses angulaires de test.</i>	46
<i>Figure 28 : la carte arduin- mega 2560 .</i>	52
<i>Figure 29 :illustrations des impulsions et des angles correspondants pour un servomoteur</i>	54
<i>Figure 30 :(a) est le SG90 et (b) est le MG996R</i>	54
<i>Figure 31 : illustration de la base du SMA</i>	56
<i>Figure 32 : illustration de l'avant-bras du SMA</i>	56
<i>Figure 33 :illustration du bras du SMA</i>	57
<i>Figure 34 :illustration de la main de l'organe terminale qui va porter les doigts</i>	57
<i>Figure 35 :illustrations de deux doigts de l'organe terminale du bras</i>	58
<i>Figure 36 : image d'illustration du HC SR 04</i>	59
<i>Figure 37 : illustration du TCS 3200</i>	60
<i>Figure 38 : l'illustration de la structure interne du capteur .</i>	61
<i>Figure 39 : image d'illustration d'un potentiomètre</i>	63
<i>Figure 40 : d'illustration du module Bluetooth HC-06</i>	64
<i>Figure 41 : le diagramme fonctionnel d'un régulateur flou.</i>	71
<i>Figure 42 : le régulateur type Mamdani conçu avec Matlab.</i>	73
<i>Figure 43 :illustrations des sous-ensembles flous de la variable distance1.</i>	74
<i>Figure 44 : illustrations des sous-ensembles flous de la variable distance1.</i>	74
<i>Figure 45 : surface de décision floue.</i>	76
<i>Figure 46 :la base du bras robotisé</i>	77
<i>Figure 47 : l'avant-bras du bras robotisé.</i>	78
<i>Figure 48 : la bras du bras robotisé</i>	79
<i>Figure 49 :la main du bras robotisé.</i>	79
<i>Figure 50 :la structure du bas robotisé complétement assemblée.</i>	80
<i>Figure 51 : la structure du bas robotisé complétement assemblée</i>	81
<i>Figure 52 : image du logiciel Proteus du montage de tout le système.</i>	83

<i>Figure 53 : image de l'interface de l'application</i>	84
<i>Figure 54 : image de l'interface de l'application</i>	85
<i>Figure 55 : le programme dans MIT appinventor</i>	86
<i>Figure 56 : le programme de l'application dans MIT appinventor</i>	86
<i>Figure 57 : le programme de l'application dans MIT appinventor</i>	87
<i>Figure 58 :les potentiomètres pour la commande manuelle</i>	87

0. INTRODUCTION GÉNÉRALE

0.1. Contexte

Dans le contexte économique mondial actuel où la compétitivité est forte, et où les entreprises recherchent toujours plus de productivité, d'efficacité tout en réduisant leurs coûts, les outils d'aide à la gestion de processus de travail sont déterminant pour permettre aux entreprises d'atteindre leurs objectifs. Cependant, cette compétitivité et cette guerre contre le temps n'a jamais été plus significative que dans le secteur de la logistique [1]. Les entreprises sont contraintes aujourd'hui à trouver de moyens de produire toujours plus dans un délai court et d'être capable de gérer d'une manière efficace et rapide la quantité de biens produite. L'organisation des systèmes de production se doit ainsi d'être conforme à ces nouvelles exigences qui sont cruciales pour la rentabilité et la survie de l'entreprise. Le vingtième siècle avait déjà connu une vague d'organisation du travail en chaine mais avec toujours une main d'œuvre humaine qui était incontournable dans beaucoup de processus de travail. Aujourd'hui, la tendance va vers l'automatisation du processus de travail. On cherche à utiliser le moins possible la main d'œuvre humaine car elle est moins efficace, moins rapide et surtout moins rentable sur le long terme. Ainsi donc, les entreprises utilisent de plus en plus des unités de production automatisées. Après la production, nous observons une robotisation de la logistique [2].

Ainsi les systèmes de mise en emballage de produits, de mise en caisse et d'emballage, l'entreposage de produits sont automatisées. L'utilisation de robots pour les tâches précitées ci-haut permet leur automatisation. Il en existe deux types : les robots fixes et les robots mobiles. Parlant des robots fixes, on voit les robots destinés à une tâche spécifique dont la réalisation ne nécessite pas une mobilité de la base de la structure du robot. Ils se chargent, dans la logistique, de la mise en emballage, de la mise en caisse ou de l'emballage de la production [3]. Dans les entrepôts, ils font aussi le chargement et le déchargement des produits.

Les robots mobiles sont quant à eux employés, dans la logistique, pour la préparation des commandes. Grâce à leurs grandes capacités de charge, ils peuvent déplacer des rayonnages entiers et garantir un déplacement efficace de la marchandise jusqu'au magasinier [4].

0.2. Identification et formulation du problème

Les industries qui produisent de boissons peuvent être appelées à gérer une grande quantité de bouteilles de boisson produites et cela chaque jour. La rapidité du processus de mise en bouteille et celui de mise en caisse sont déterminants pour avoir un stock prêt à la vente. Au niveau mondial l'automatisation dans cette industrie a débuté depuis bien longtemps et des solutions innovantes ont été proposées. Des systèmes industriels permettant de faire la mise en bouteille existent déjà et selon leurs prix, chaque système a sa particularité. Pour la mise en caisse et l'emballage, la tendance est aujourd'hui orientée vers l'utilisation des bras manipulateurs. Cependant, les bras manipulateurs étant une technologie de pointe, tout le monde ne peut se l'offrir en raison de son prix et surtout que, pour une meilleure efficacité, ces bras devront être conçus spécialement pour un processus particulier [5].

D'une manière particulière, dans notre pays, il y a quelques décennies les boissons embouteillées étaient exclusivement produites par les grandes marques internationales. Cependant, depuis quelques années, les entrepreneurs locaux se sont lancés dans la production des boissons embouteillées après transformation des boissons qui jadis étaient produites de manière artisanales, améliorant ainsi la qualité et le gain financier. Aujourd'hui, cette entreprise a pris de l'ampleur et on trouve presque partout une augmentation accrue du nombre d'entreprises qui œuvrent dans l'industrie des boissons et qui produisent des catégories très variées allant de l'eau minérale aux boissons alcoolisées et en passant par les jus. Malgré la multiplication du nombre de ces entreprises, l'automatisation dans ces entreprises n'est pas encore forte du fait du coût d'investissement très important. Cela s'observe surtout pour la mise en caisse et l'emballage de bouteilles. Le problème devient encore plus complexe pour les entreprises produisant plusieurs marques de boissons. Cela en ce sens que la mise en caisse ou

soit l'emballage doit être fait en prenant en compte le type de boisson qui des fois se distinguent par la couleur, la forme ou les dimensions de la bouteille. Pour une main d'œuvre humaine cela peut conduire à des erreurs de tris qui peuvent être dues à la fatigue, la rapidité du tri n'est pas aussi garantie car la main d'œuvre humaine peut se retrouver rapidement débordé [6]. Ce problème logistique qui est d'une grande importance pour assurer la qualité du produit et l'accroissement des gains financiers de l'entreprise mérite une attention particulière.

0.3. Questions de recherche

Cette problématique nous mène aux interrogations suivantes :

1. Quelles sont les meilleures stratégies de conception pour intégrer un contrôle à la fois manuel et par application mobile dans un bras robotisé, en assurant une expérience utilisateur fluide et intuitive ?
2. Quelle technologie de communication et de commande optimale pourrait assurer une réactivité en temps réel entre le bras robotisé, la commande manuelle et l'application mobile ?
3. Comment optimiser la conception électromécanique, électronique et cinématique du bras robotisé pour garantir des mouvements précis et fluides en réponse aux commandes manuelle et de l'application mobile ?

0.4. Formulation des hypothèses

Pour rendre automatique les tâches répétitives comme le tri d'objets, il existe plusieurs solutions qui pourraient être exploitées. La plupart de ces solutions font recours à la robotique ou soit plus généralement à des automates programmables industriels. La réalisation de ce

dispositif combinant une commande manuelle et par application mobile nécessite une conception qui assurera le fonctionnement indépendant de ces deux modes de commande. L'activation d'un mode de commande devra aussi empêcher l'activation de l'autre mode. Ainsi donc l'algorithme de commande devra être conçu pour permettre ce fonctionnement indépendant. Cet algorithme devra permettre le basculement d'un mode à l'autre.

Les solutions au problème de tri dépendent généralement du problème à résoudre. Pour de problème de logistique comme le tri d'objets, un bras robotique pourrait être adaptée car il permettrait à obtenir plus d'efficacité et de rapidité. Ces solutions nécessiteraient un choix objectif de technologie à employer pour pouvoir s'assurer d'une transmission en temps réel de la commande. La commande d'un bras manipulateur nécessitant une vision de l'environnement de travail par l'opérateur, la technologie Bluetooth serait adaptée en ce sens que c'est une technologie de proximité. En dotant le bras robotique d'un capteur avec une sensibilité par rapport à un élément clé permettrait de ranger les objets d'une manière autonome selon un critère bien défini. Ainsi avec un capteur sensible aux couleurs nous pourrions être capables de distinguer les objets en fonction des couleurs [5].

La modélisation du dispositif devra permettre d'établir un modèle capable de combiner les bons paramètres géométrique, cinématique et électrique pour s'assurer une exécution précise de commandes. En ce qui concerne la commande, un régulateur basé sur la logique floue pourrait permettre une meilleure direction du bras manipulateur en évitant les obstacles dans le processus de rangement [6] [7]. Un système de tri doté d'un algorithme permettant de mettre en marche les actionneurs en fonctions de grandeurs mesurées par les capteurs nous permettrait de réaliser un système automatique.

0.5. Justification du choix du sujet et motivations

L'automatisation est l'un de défi majeur du moment. Ce défi une fois relevé peut permettre l'essor économique et industriel de l'Afrique. Les travaux de manutention sont répétitifs. Ces travaux sont fatigants, pénibles et généralement moins valorisés car moins rémunérés. L'utilisation de systèmes automatiques de tri, comme un bras robotique, permettra de gagner

en efficacité, en temps et de réduire la main d'œuvre humaine ou soit l'affecter à d'autres tâches. L'utilisation de systèmes automatiques de tri permettra aussi de lutter contre certaines maladies du travail dues aux tâches répétitives dans la logistique.

0.6. Énoncé des objectifs de recherche

0.6.1. L'objectif général

Ce travail a pour objectif général de concevoir, de modéliser et de commander un bras robotique pour permettre le tri d'objets et son automatisation dans les entreprises locales de fabrication de boissons embouteillées. Il s'agit d'un élément qui permettra d'améliorer leur système de production pour une meilleure efficacité.

0.6.2. Les objectifs spécifiques

Pour atteindre cet objectif, nous nous sommes fixés les objectifs spécifiques suivants :

- Etudier et comprendre le fonctionnement de bras manipulateurs.
- Faire la modélisation du bras manipulateur
- Concevoir un système de commande pour le bras manipulateur qui consistera en un régulateur flou. Ce régulateur sera conçu grâce au logiciel Matlab.
- Concevoir un algorithme qui va permettre l'automatisation du bras manipulateur grâce aux données reçus de capteurs de positions, de couleurs...
- Concevoir une application mobile permettant de commander le bras robotique.

0.7. Méthodologie

Pour réaliser cette recherche, nous utilisons les méthodes et techniques suivantes :

- La méthode analytique pour analyser systématiquement les différents éléments de notre système. La méthode déductive devra permettre, grâce aux lois mathématiques, de cerner la mécanique qui régit les mouvements.

- La technique documentaire nous a permis grâce à une fouille dans les ouvrages, articles et cours sur le domaine d'approfondir nos connaissances d'une manière générale sur les bras robotiques et leurs commandes.
- La technique expérimentale nous a permis de concevoir notre régulateur grâce au logiciel MATLAB avant de faire la réalisation du bras robotique au laboratoire.

0.8. Structure du mémoire

Structurellement ce travail débute par une introduction générale, ensuite vont suivre 4 chapitres.

- Le premier chapitre est consacré aux généralités sur les bras robotiques; il donne un aperçu général sur les robots en présentant les types, les caractéristiques et la classification des robots en général et de bras robotisé en particulier ainsi que leurs modes de commande.
- Le second est consacré à la modélisation du bras robotique; il présente et montre les méthodologies pour faire la modélisation géométrique, cinématique et dynamique du bras robotisé.
- Le troisième concerne la conception du bras robotisé; le régulateur flou y sera conçu ainsi que l'ensemble du dispositif. Nous finirons ce chapitre par la conception de l'application mobile devant servir pour la commande du bras robotique.
- Le quatrième concerne la réalisation du bras robotisé où la conception sera matérialisée et les résultats commentés.
- Enfin une conclusion générale clôture le travail.

Chapitre 1 : GÉNÉRALITÉS LES BRAS ROBOTISÉS

1.1 INTRODUCTION

Dans ce chapitre nous abordons des notions assez larges sur les robots et les bras robotisés. Nous commençons par présenter les généralités sur les robots, ensuite nous présentons les différents types de robot avant de faire la classification de robots. Le chapitre se termine par l'étude de manière succincte des bras robotisés.

1.2 LA ROBOTIQUE

1.2.1 GENERALITES SUR LES ROBOTS

1.2.1.1 DEFINITION

Un robot est un système qui exécute un certain nombre de fonctions délicates et ponctuelles d'une manière précise et sensible qu'un être humain ne peut facilement exécuter en raison de la difficulté d'accès du lieu, des risques encourus, des coûts de la main d'œuvre humaine ou de la grande précision requise par ces tâches [8]. D'une manière technique, un robot est un système mécanique polyarticulé mû par des actionneurs et commandé par un ordinateur grâce aux données fournies par un certain nombre de capteurs et qui est destiné à effectuer une grande variété de tâches selon un programme fixe ou modifiable [9].

Pour être qualifié de robot, une machine doit être capable d'obtenir des informations sur son environnement, et d'effectuer différentes tâches comme la locomotion ou la manipulation d'objets. Il doit être aussi programmable et capable de fonctionner d'une manière autonome ou soit en interaction avec les humains [10]. Ci-dessous, nous retrouvons quelques images d'illustrations des robots humanoïdes.

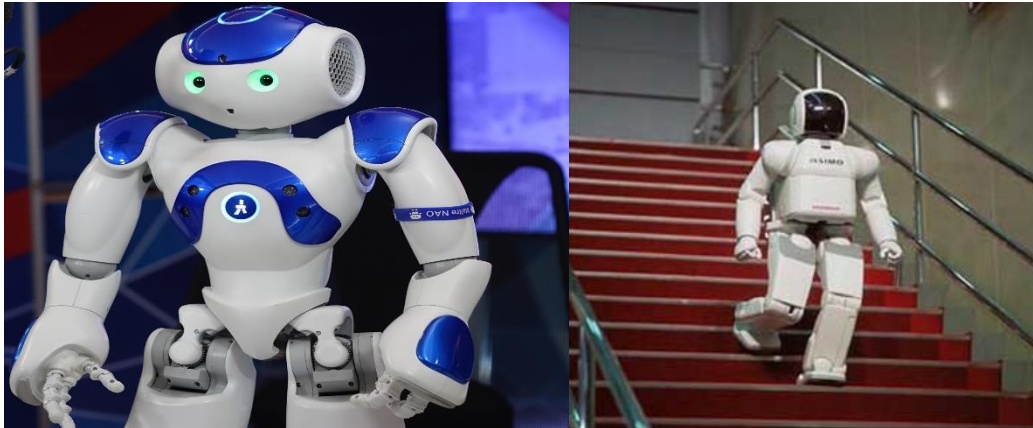


Figure 1 : images d'illustration des robots humanoïdes [11]

1.2.1.2 CARACTERISTIQUES DES ROBOTS

En fonction de l'application qu'on lui réserve, voici quelques paramètres à prendre dans le choix d'un robot [12]:

- La charge maximale transportable par le robot : elle doit être déterminée dans les conditions les plus défavorables (c'est-à-dire, en élongation maximale).
- L'architecture du Système Mécanique Articulée. : en fonction de cette architecture, nous avons des robots à structure rigides, des robots à articulations flexibles. Le choix est guidé par la tâche à réaliser.
- Le volume ou espace de travail : il est défini comme l'ensemble des points atteignables par l'organe terminal du robot.
- L'exactitude de positionnement du robot, définie par une position et une orientation dans l'espace cartésien. C'est l'erreur entre le point qu'on souhaite attendre et le point finalement atteint par le robot calculé via le modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de position, à la flexibilité du système mécanique.
- La répétabilité : Ce paramètre caractérise la capacité du robot à retourner vers un point donné (position et orientation). La répétabilité correspond à l'erreur maximum de positionnement sur un point prédéfini dans le cas de trajectoires répétitives.
- La vitesse de déplacement (vitesse maximale en élongation maximale), l'accélération

- La masse du robot
- Le coût du robot
- La maintenance . [10] [12].

1.2.2 TYPES DE ROBOTS

La classification dépend de critères choisis. Selon leurs fonctionnalités, les robots peuvent être classés en trois grandes catégories [10] :

1.2.2.1 LES ROBOTS MANIPULATEURS INDUSTRIELS

Un robot manipulateur industriel constitue l'association d'un manipulateur et d'organes apte à acquérir, traiter et gérer des informations issues de l'environnement. Ils sont dotés de sens artificiels et de ressources leur permettant de résoudre des problèmes limités. Ces robots sont essentiellement orientés vers les applications nécessitent une adaptabilité a la tâche : assemblage, soudage à l'arc, vérification [10], ...

1.2.2.2 LES ROBOTS DIDACTIQUES

Les robots didactiques sont de version ou format réduite des robots. Ils ont un rôle de formation et d'enseignement, ils peuvent aussi être utilisés pour effectuer des tests de faisabilités d'un poste robotisé [14].

1.2.2.3 LES ROBOTS MOBILES AUTONOMES

Les possibilités sont plus vastes, du fait de leur mobilité. Notamment, ils peuvent être utilisés en zone dangereuse (nucléaire, incendie, sécurité civil, déminage. . . etc.), inaccessible (océanographie, spatial). De tels robots font appel à des capteurs et à des logiciels sophistiqués. On peut distinguer deux types de locomotion : les robots marcheurs qui imitent la démarche humaine, et les robots mobiles qui ressemblent plus à des véhicules [13].

Ce travail porte sur les robots manipulateurs industriels.

1.2.3 CLASSIFICATION DE ROBOTS MANIPULATEURS INDUSTRIELS

La classification des bras robotiques est une tâche difficile, car il existe plusieurs critères pour les différencier. Le point de vue choisi pour cette classification qui met en évidence la structure de robots est les suivant [14] [15]. Il s'agit du point de vue géométrique.

1.2.3.1 CLASSIFICATION GÉOMÉTRIQUE

Du point de vu géométrique, il existe différentes architectures d'un robot porteur :

- La structure cartésienne (PPP),
- La structure cylindrique (RPP ou PRP),
- La structure sphérique ou polaire (RRP),
- La structure dite SCARA (RRP) et enfin
- La structure anthropomorphe (RRR).

1.2.3.1.1 Structure cartésienne (PPP)

Les caractéristiques d'une structure cartésienne (PPP) sont :

1. Trois axes, deux à deux en série avec trois degrés de liberté,
2. Une très bonne précision et une grande lenteur.
3. Le volume de travail est un parallélépipède dont les dimensions sont les translations permises par les trois liaisons prismatiques [14].

Ci-dessous, nous retrouvons quelques images d'illustrations de la structure cartésienne.

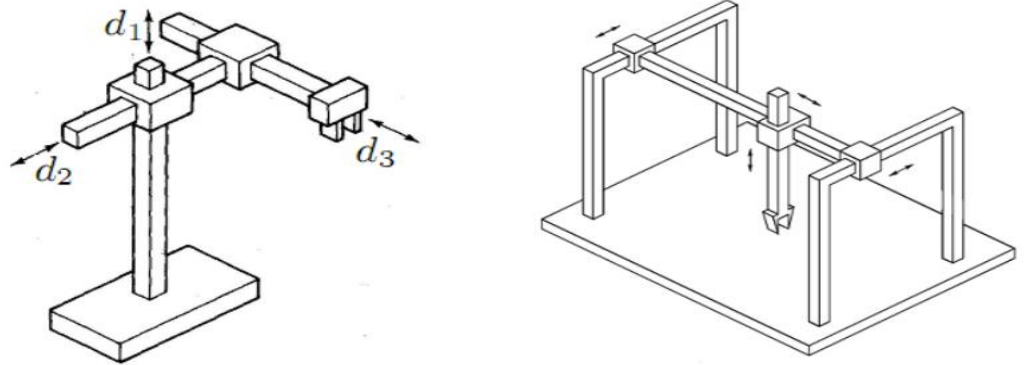


Figure 2 : illustration de la structure cartésienne [16].

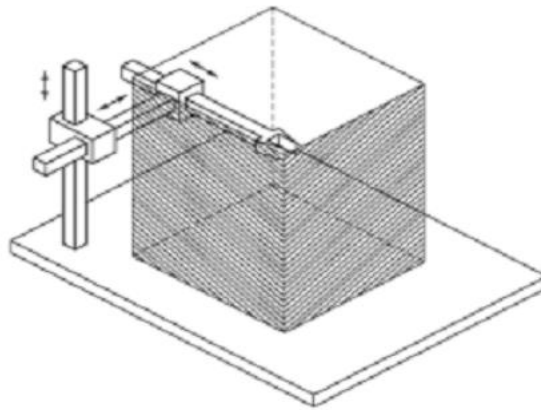


Figure 3 : volume de travail pour une structure cartésienne [16]

1.2.3.1.2 La structure cylindrique (RPP) ou (PRP)

Les caractéristiques d'une structure cylindrique (RPP ou PRP) sont :

1. Trois axes non perpendiculaires en série avec trois degrés de liberté et une grande rapidité.
2. Le volume de travail est un cylindre plein ou creux, autrement dit, le volume de travail est un tore à section rectangulaire, dont la hauteur L est la translation permise par une liaison prismatique [15].

Ci-dessous, nous retrouvons quelques images d'illustrations de la structure cylindrique.



Figure 4 :illustration de la structure cylindrique [16]

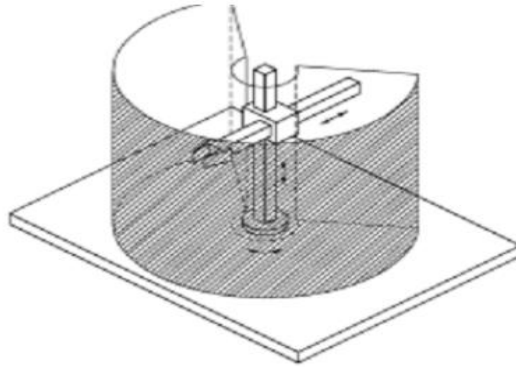


Figure 5 :volume de travail de la structure cylindrique [16]

1.2.3.1.3 La structure sphérique ou polaire à axe de rotation orthogonale (RRP)

Pour cette structure, nous avons un volume de travail en forme de sphère creuse, dont les rayons intérieur et extérieur sont fixés soit par la disposition de la liaison prismatique et la translation qu'elle permet, soit par les longueurs des deux parties du bras du robot [14].

Ci-dessous, nous retrouvons quelques images d'illustrations de la structure sphérique.

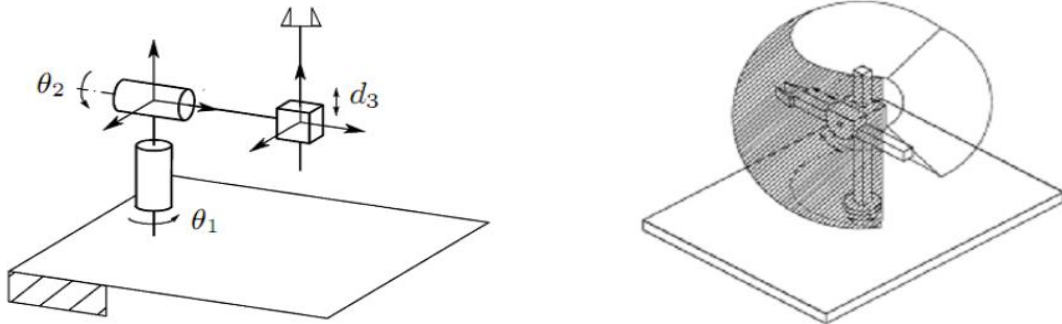


Figure 6 :images d'illustration de la structure polaire et de son espace de travail [16]

1.2.3.1.4 La structure dite SCARA

Les caractéristiques d'une structure SCARA (Selective Compliance Articulated Robot for Assembly) sont : 3 axes en série cylindrique (RRP) ayant trois degrés de liberté.

Les particularités de cette structure sont qu'elle est précise et très rapide [15].

Ci-dessous, nous retrouvons quelques images d'illustrations de la structure SCARA.

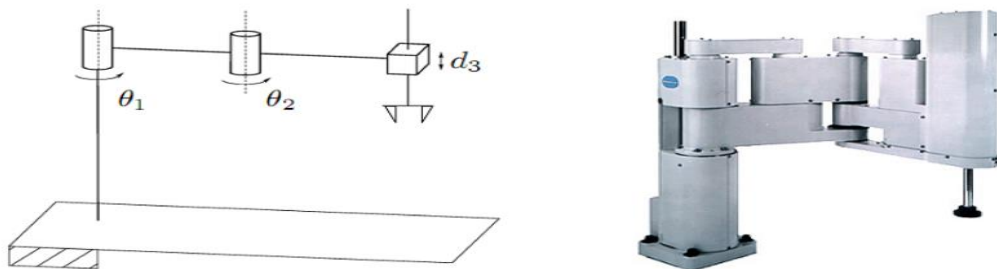


Figure 7 :images d'illustration de la structure SCARA [16]

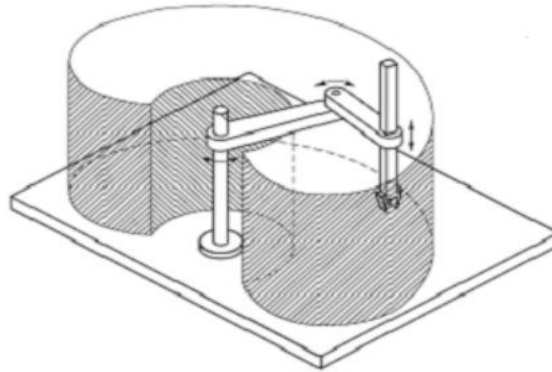


Figure 8 : illustration du volume de travail de la structure SCARA [16]

1.2.3.1.5 La structure anthropomorphe (RRR)

L'architecture de cette structure est la plus généraliste qui reproduit le bras humain. Son enveloppe de travail a une cinématique et une dynamique complexe, mais sa configuration est plus flexible [14]. Ci-dessous, nous retrouvons quelques images d'illustrations de la structure anthropomorphe.

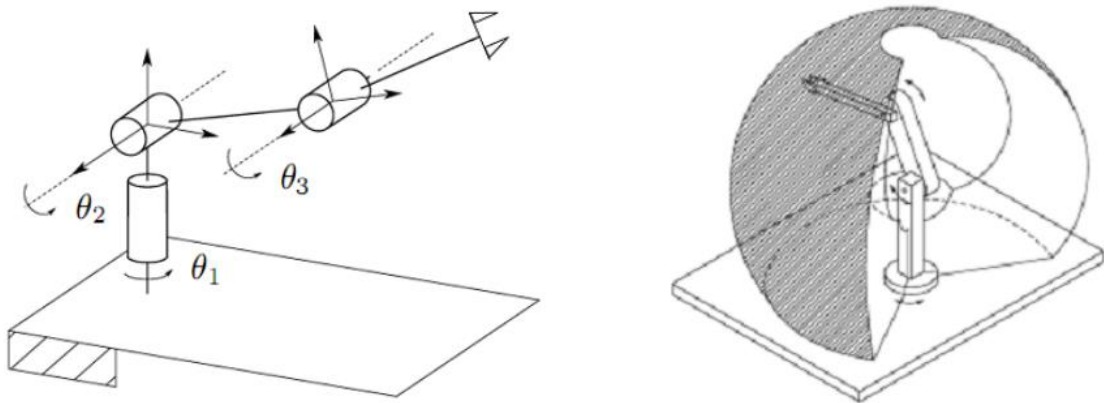


Figure 9 : images d'illustration de la structure anthropomorphe et de son espace de travail [16]

1.3 LES BRAS ROBOTIQUES

1.3.1 LES ROBOTS MANIPULATEURS

Les bras manipulateurs ou bras robotiques sont des bras artificiels constitués généralement de plusieurs liens rigides connectés par des articulations. La grande majorité des robots industriels sont des robots manipulateurs qui ont comme tâche de positionner des outils pour des tâches spécifiques comme la soudure, la peinture, etc [14]. Ci-dessous, nous retrouvons une image d'illustration des robots manipulateurs.

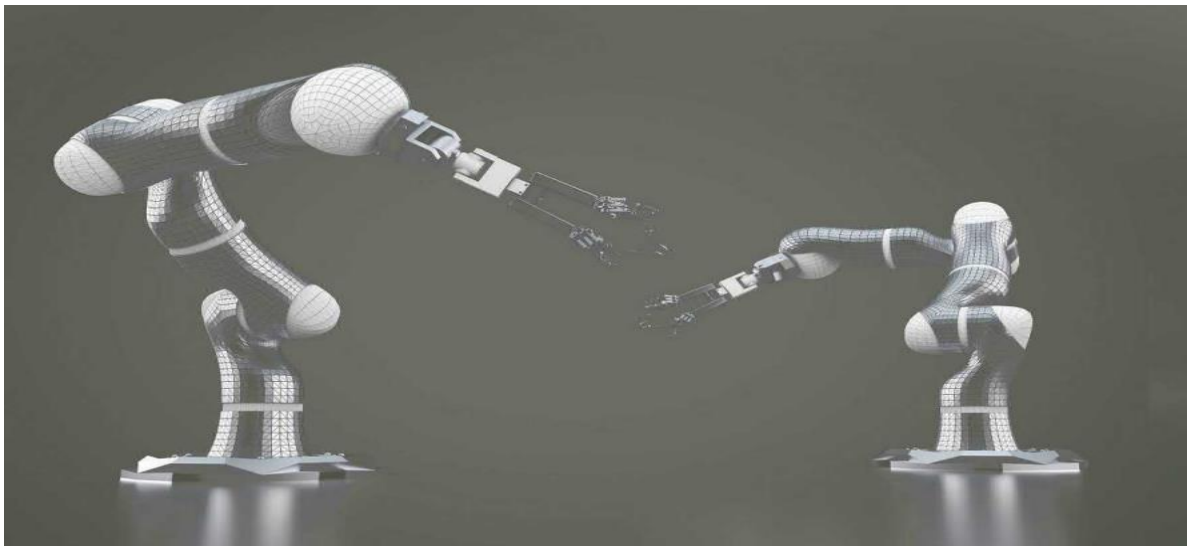


Figure 10 :images d'illustration du bras robotique [11]

1.3.2 DEGRÉS DE LIBERTÉ

Le positionnement complet d'une pièce dans l'espace nécessite six paramètres indépendants, appelés degré de liberté. Ces paramètres sont choisis parmi les trois coordonnées cartésiennes pour positionner un point de la pièce et les trois angles pour orienter cette pièce [13].

Pour un système robotisé, le degré de liberté est défini par le nombre de paramètres indépendants qui fixent la situation de l'effecteur ou organe terminal (position et orientation). Il représente le nombre de mouvements élémentaires (rotation et translation) par rapport à un repère de base [17].

1.3.3 CONSTITUANTS D'UN BRAS ROBOTISÉS

Nous distinguons 5 parties dans un robot manipulateur : les actionneurs, le système mécanique articulé et organe terminal, les capteurs, la commande. Ci-dessous, nous retrouvons quelques images d'illustrations des bras robotiques avec leurs constituants.

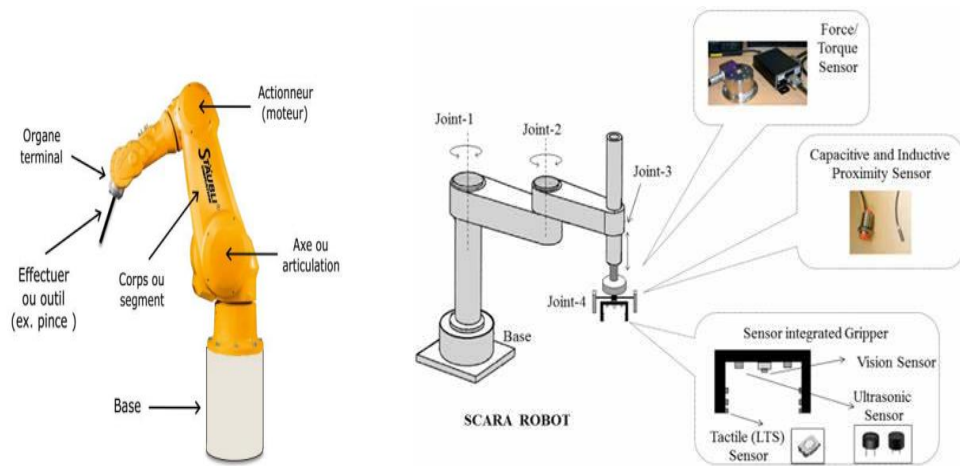


Figure 11 : images d'illustration d'un bras robotique avec ses constituants [17]

1.3.3.1 Les actionneurs

Les actionneurs sont les muscles des bras robotiques. Ce sont des composants permettant essentiellement de mettre en mouvement les organes des machines. Ils convertissent l'énergie électrique, hydraulique ou pneumatique en de l'énergie mécanique. Ces actionneurs sont presque toujours contrôlés par des signaux de commande électriques. Ce sont essentiellement des moteurs et des vérins [15].

1.3.3.2 Système mécanique articulé

Le système mécanique articulé (SMA) est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Son architecture est composée de plusieurs chaînes de corps rigides assemblés par des liaisons appelées articulations, généralement les uns à la suite des autres où chaque solide est mobile par rapport au précédent. Cette mobilité s'exprime en termes

de degrés de liberté. Le rôle du SMA est d'amener l'organe terminal dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données. Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés [10].

1.3.3.3 Joints

Les joints sont les articulations d'un robot. Une articulation lie deux corps successifs en limitant le nombre de degrés de liberté de l'un par rapport à l'autre. Soit m le nombre de degré de liberté, encore appelé mobilité de l'articulation. La mobilité d'une articulation est telle que : $0 \leq m \leq 6$.

Lorsque $m = 1$; ce qui est fréquemment le cas en robotique, l'articulation est dite simple : soit rotoïde, soit prismatique.

Les joints en robotique se ramène à une combinaison de joints prismatiques et de joints rotoïdes [9].

- **Joint Rotoïde**

Il s'agit d'une articulation de type pivot, réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. Il est noté R. Ci-dessous, nous retrouvons quelques images d'illustrations des joints rotoïdes.

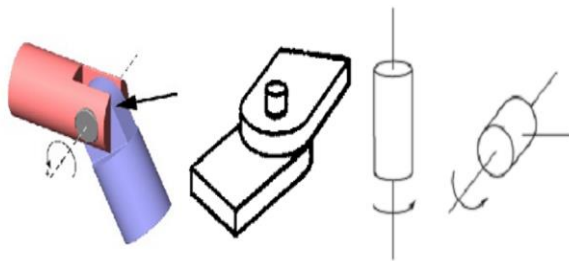


Figure 12 : image d'illustration de joints Rotoïde [16]

- **Joint prismatique**

Il s'agit d'une articulation de type glissière (translation), notée P, réduisant le mouvement entre deux corps à une translation le long d'un axe commun. Il est noté P.

Ci-dessous, nous retrouvons quelques images d'illustrations des joints prismatiques.

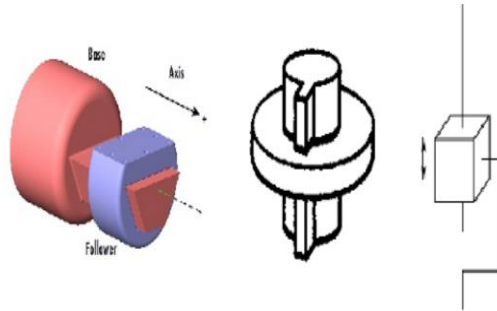


Figure 13 :image d'illustration de joints prismatiques [16]

1.3.3.4 Organe terminal

En parlant d'organe terminale, il s'agit de l'interface permettant au robot d'interagir avec son environnement. Il est constitué de dispositifs destinés à manipuler des objets (dispositifs de serrage, dispositifs magnétiques...), ou à les transformer (outils, torche de soudage, pistolet de peinture...). Un organe terminal peut être multifonctionnel, au sens où 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 [18]. Ci-dessous, nous retrouvons une images d'illustration de l'organe terminale d'un bras manipulateur.



Figure 14 :image d'illustration d'un effecteur [11]

1.3.3.5 Capteurs

Les capteurs sont des dispositifs qui permettent la collecte de l'information sur l'état interne d'un robot ou de son environnement. Les capteurs sont dits proprioceptifs lorsqu'ils mesurent l'état interne du robot (positions et vitesses des articulations) et extéroceptifs lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle) [10] [15].

1.3.4 COMMANDES DES BRAS ROBOTISÉS

Deux types de mouvements apparaissent quand on parle de commande du bras manipulateur. Un premier type considère que les mouvements nécessaires pour la réalisation de la tâche sont exécutés dans l'espace libre. Le deuxième type considère des mouvements spécifiques avec des forces de contact pour l'organe terminal qui se déplace dans un espace contraint. Toute tâche de robotique de service est réalisée par une combinaison de ces deux types de mouvement [19].

1.3.4.1 Commande dans l'espace libre

Nous nous limiterons à une description simple des différentes techniques de commande dans l'espace libre. Nous présentons tout d'abord le schéma de commande par articulation pour progresser jusqu'à des techniques plus performantes comme la commande optimale.

- **Commande par articulation**

2. Cette technique est utilisée par des robots manipulateurs qui utilisent des servomoteurs avec de forts rapports de réduction. Lorsque le système présente un comportement linéaire, l'asservissement du mouvement peut être réalisé par des techniques classiques de commande. Nous parlons alors d'une commande décentralisée de type PID [20].

- **Commande jacobienne**

3. Cette technique utilise la matrice jacobienne inverse du bras manipulateur pour calculer les vitesses de consigne aux articulations. Elle est aussi connue sous le nom

de commande à mouvement résolu. Les approches les plus courantes sont de type : mouvement à vitesse résolu, mouvement à accélération résolu et mouvement à force résolu. Son principal inconvénient est l'utilisation de l'inverse de la matrice jacobienne, qui peut devenir singulière dans certaines conditions [21].

- **Commande par découplage non linéaire**

4. Cette technique est aussi connue sous les noms de commande dynamique ou commande par couple calculé. Lorsque l'application exige des évolutions rapides avec des contraintes dynamiques, la commande doit prendre en compte les forces d'interaction. Ce type de technique considère l'ensemble des articulations et, pour les découpler, utilise la théorie du découplage non linéaire. Cette théorie utilise le modèle dynamique du robot pour le calcul de la loi de commande, ce qui conduit à des lois de commande centralisées non linéaires. Cette méthode dépend fortement du modèle du système, elle est très sensible aux imprécisions du modèle qui entraînent un découplage imparfait. Ceci constitue son principal inconvénient [22].

- **Commande fondée sur une fonction de Lyapunov**

5. Des méthodes basées sur une fonction de Lyapunov ont été utilisées pour la commande des bras manipulateurs de façon satisfaisante pour des tâches de suivi. Particulièrement lorsqu'on cherche à garantir la convergence asymptotique et non à linéariser le système ou à obtenir le découplage [23].

- **Commande passive**

6. Cette technique considère le robot comme un système passif, c'est à dire un système qui dissipe de l'énergie. De telles lois de commande permettent de modifier l'énergie naturelle du robot pour qu'il réalise la tâche. En utilisant le formalisme de Hamilton, la commande cherche à minimiser l'énergie du système en utilisant un bloc non linéaire passif dans la boucle de retour [24].

- **Commande adaptative**

7. Ce type de techniques vise à corriger les déficiences de la commande par découplage non linéaire, comme la connaissance approximative des paramètres du modèle du robot ou pour s'adapter aux différentes conditions opératoires. Ce type de schémas cherche à estimer ou ajuster en ligne les valeurs des paramètres utilisées dans le calcul de la loi de commande. Les avantages de ce type de techniques sont évidents, malheureusement la puissance de calcul demandée au système constitue un inconvénient important [25].

- **Commande prédictive**

8. Ce type de schémas, en utilisant le modèle du système et les consignes, est capable de prédire son évolution, de telle manière qu'il est capable d'agir en fonction de l'erreur de prédiction. Un grand avantage de ce type d'approche est lié au fait que l'erreur de prédiction n'est pas contaminée par les bruits de mesure mais la dépendance au modèle reste forte [26].

- **Commande robuste**

9. Si les paramètres du modèle ne sont pas connus de façon précise et si l'incertitude sur les paramètres admet des bornes connues, alors les techniques de commande robuste peuvent être utilisées. Cette technique utilise une surface de glissement où la stabilité du système est garantie [27].

- **Commande optimale**

10. Pour réaliser une tâche, il peut exister un grand nombre de solutions. Dans ce cas, il peut être souhaitable de choisir une solution qui satisfasse un certain critère [28].

1.3.4.2 Commande dans l'espace contraint

Lorsque l'organe terminal entre en contact avec une surface, la chaîne cinématique du robot est fermée sur l'environnement, et du fait de la raideur de l'ensemble, de faibles variations de position du point de contact peuvent induire des efforts importants. Lorsque le robot est

contraint par l'environnement suivant toutes les directions, aucun déplacement de l'organe terminal n'est possible et il ne peut qu'exercer des efforts sur l'environnement ; en revanche, lorsque le robot est dans l'espace libre, l'absence de tout contact empêche la génération d'efforts. Ainsi, les notions de commande en position et de commande en effort s'excluent mutuellement : on ne peut contrôler simultanément une force et un déplacement suivant une même direction [19].

Dans la pratique, on associe un repère de compliance à l'organe terminal, de façon à ce que pour chacun des 6 degrés de liberté on associe soit une « position » soit un « effort ».

Il existe cinq types de commande dans l'espace contraint, les trois premiers n'utilisent pas de consigne d'effort : compliance passive, commande par raideur active et commande en impédance. Les deux suivantes utilisent la consigne d'effort : la commande hybride et la commande hybride externe [29].

1.4 CONCLUSION PARTIELLE

Les bras robotiques jouent un rôle essentiel dans l'automatisation des processus industriels et logistiques, permettant leur déploiement dans des domaines où la main humaine était traditionnellement l'outil de travail, ainsi que dans des environnements inaccessibles à l'homme, tout en offrant une précision et une rapidité exceptionnelles.

Dans ce chapitre, nous avons exposé les concepts fondamentaux relatifs aux robots, en détaillant leurs caractéristiques, leurs divers types et leurs classifications. Parmi la gamme des robots, nous nous sommes focalisés sur un type particulier qui fera l'objet de notre étude dans ce travail : les bras manipulateurs. Parmi les bras manipulateurs ou bras robotisé, nous avons choisi la structure anthropomorphe avec 3 degrés de liberté pour notre étude car elle est la plus proche du bras humain. Nous avons également présenté les composants essentiels des bras robotiques et examiné différentes approches pour leur commande.

Chapitre 2

MODÉLISATION DU BRAS MANIPULATEUR

2.1 INTRODUCTION

Dans ce chapitre, nous explorons la modélisation des bras robotiques en établissant les fondements mathématiques nécessaires pour réaliser diverses représentations. Notre démarche commence par la présentation de modèles géométriques, cinématiques et dynamiques pour les bras robotiques de manière générale. Ensuite, nous nous penchons sur les modèles géométriques et cinématiques spécifiques de notre bras robotique anthropomorphe, composé de 3 articulations rotoïdes. Le chapitre se clôture par une conclusion partielle.

2.2 TYPES DE MODELES DE TRANSFORMATION

La conception et la commande des robots nécessitent la détermination de certains modèles mathématiques, tels que :

- Les modèles de transformation entre l'espace opérationnel dans laquelle est définie la situation de l'organe terminal et l'espace articulaire dans laquelle est définie la configuration du robot. Ces modèles de transformation sont très importants puisque les robots sont commandés dans l'espace articulaire, tandis que les tâches sont définies dans l'espace opérationnel [16].

Parmi ces modèles, on distingue :

- ◆ Les modèles géométriques direct et inverse qui expriment la position de l'organe terminal en fonction des variables articulaires du mécanisme et inversement,
- ◆ Les modèles cinématiques directes et inverses qui expriment la vitesse de l'organe terminal en fonction de la vitesse des articulations et inversement.

- Les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercées par les actionneurs et les positions, vitesses et accélérations des articulations [13].

2.3 CONSTITUTION GÉOMÉTRIQUE DES ROBOTS MANIPULATEURS

Un robot manipulateur est une cinématique constituée de $n+1$ corps, possédant deux corps particuliers et des actionneurs. Le premier corps particulier (C_0), appelé base, est le socle du robot, le seconde corps particulier (C_n), appelé terminal, est le support d'un préhenseur ou d'outil. Les n corps sont liés entre eux par des liaisons (articulations), suivant une structure de chaîne. Ces liaisons sont de nature prismatique P ou Rotoïde R. Les corps et les liaisons du robot manipulateur sont numérotés de 0 à n dans un ordre croissant en partant du socle [13]. Ci-dessous, nous retrouvons une image d'illustration de la constitution géométrique des robots manipulateurs.

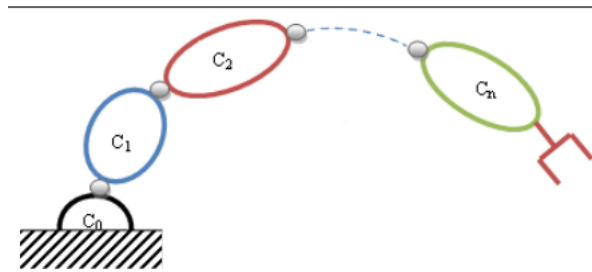


Figure 15 :structure d'un bras manipulateur série [3]

Soit $({}^iP_x, {}^iP_y, {}^iP_z)$ les coordonnées cartésiennes d'un point P arbitraire, mesuré dans le repère $R_i(O_i, x_i, y_i, z_i)$, Les coordonnées homogènes du point P sont $(w {}^iP_x, w {}^iP_y, w {}^iP_z, w)$ où w est le facteur d'échelle, dans la robotique $w = 1$. Les coordonnées homogènes du point P sont représentées par le vecteur colonne [9] [18] [13]:

$${}^iP = \begin{bmatrix} {}^iP_x \\ {}^iP_y \\ {}^iP_z \\ 1 \end{bmatrix} \quad \text{Équation 2-1}$$

Les coordonnées homogènes d'un plan Q d'équation :

$${}^i\alpha x + {}^i\beta y + {}^i\gamma z + {}^i\delta = 0 \quad \text{Équation 2-2}$$

exprimées selon le repère R_i , sont données par le vecteur ligne :

$${}^iQ = [{}^i\alpha \quad {}^i\beta \quad {}^i\gamma \quad {}^i\delta] \quad \text{Équation 2-3}$$

Si un point P appartient à un plan Q, le produit matriciel ${}^iQ \cdot {}^iP$ est nul :

$$[{}^i\alpha \quad {}^i\beta \quad {}^i\gamma \quad {}^i\delta] * \begin{bmatrix} {}^iP_x \\ {}^iP_y \\ {}^iP_z \\ 1 \end{bmatrix} = 0 \quad \text{Équation 2-4}$$

La transformation (translation et/ou rotation) d'un repère R_i au repère R_j est représentée par la matrice de transformation homogène iT_j de dimension (4×4) où :

$${}^iT_j = [{}^iS_j \quad {}^in_j \quad {}^ia_j \quad {}^iP_j] = \begin{bmatrix} S_x & n_x & a_x & P_x \\ S_y & n_y & a_y & P_y \\ S_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Équation 2-5}$$

où, iS_j , in_j , ia_j contiennent les composants des vecteurs unitaires suivant x_j, y_j, z_j respectivement du repère R_j exprimées dans le repère R_i et iP_j représente les coordonnées de l'origine O_j de repère R_j exprimées dans R_i .

On peut dire également que la matrice iT_j définit le repère R_j dans le repère R_i .

La matrice de transformation s'écrit aussi sous la forme :

$${}^i T_j = \begin{bmatrix} {}^i A_j & {}^i P_j \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i s_j & {}^i n_j & {}^i a_j & {}^i P_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Équation 2-6}$$

Finalement, la matrice de transformation ${}^i T_j$:

— Est considéré comme la représentation du repère R_j dans le repère R_i .

— Permet le passage du repère R_i au repère R_j .

Soit $\text{Trans}(a, b, c)$ la transformation d'une translation pure où a , b , et c sont les translations le long des axes x , y et z respectivement. Quand l'orientation est conservée, la matrice de transformation de cette translation a la forme suivante :

$${}^i T_j = \text{Trans}(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Équation 2-7}$$

La notation $\text{Trans}(u, d)$ indique la translation d'une valeur d le long de l'axe u . Ainsi la matrice $\text{Trans}(a, b, c)$ peut être décomposée en produit de trois matrice :

$\text{Trans}(x, a) \times \text{Trans}(y, b) \times \text{Trans}(z, c)$, par n'importe quel ordre de multiplication.

Ci-dessous, nous retrouvons une image d'illustration de repères.

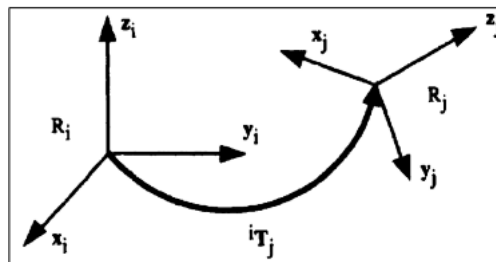


Figure 2 : illustration de repères pour lesquels on fait la transformation [13]

Soit $\text{Rot}(x, \theta)$ la transformation correspondante à une rotation pure d'angle θ autour de l'axe x , la matrice de transformation de cette rotation s'écrit :

$${}^i T_j = \text{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{rot}(x, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Équation 2-8}$$

où, $\text{rot}(x, \theta)$ désignant la matrice d'orientation de dimension (3×3) . De la même façon on définit les matrices de transformation des rotations autour des axes y et z [10].

$${}^i T_j = \text{Rot}(y, \theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{rot}(y, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Équation 2-9}$$

$${}^i T_j = \text{Rot}(z, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{rot}(z, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Équation 2-10}$$

2.4 MODELE GEOMETRIQUE DIRECT

Afin de calculer l'équation géométrique directe d'un bras manipulateur à chaîne ouverte simple, une méthode systématique et générale doit être appliquée pour définir la position et l'orientation relative de deux corps consécutifs. Les méthodes de Denavit-Hartenberg, de Khalil et Kleinfinger sont les plus répandus pour faire ce calcul. Le problème est comment déterminer deux repères attachés aux deux corps et comment calculer les transformations entre eux. Nous partirons de quelques hypothèses pour la définition des repères du corps [13].

La structure du bras étant composée de n corps C_i , $i = 1$ à n , en plus de la base (C_0), reliés entre eux par des liaisons Rotoïde ou prismatique, l'articulation i relie le corps C_i au corps C_{i-1} . Afin de définir la relation entre les positions des corps on associe à chaque corps C_i le repère $R_i(O_i, x_i, y_i, z_i)$ tels que :

— L'axe z_i est porté par l'axe de l'articulation i .

— L'axe x_i est porté par la perpendiculaire commune aux axes z_i et z_{i+1} .

— L'axe y_i est formé par la règle droite pour compléter les coordonnées du système (x_i, y_i, z_i) .

Ci-dessous, nous retrouvons une image d'illustration du bras robotisés RRR avec ses paramètres.

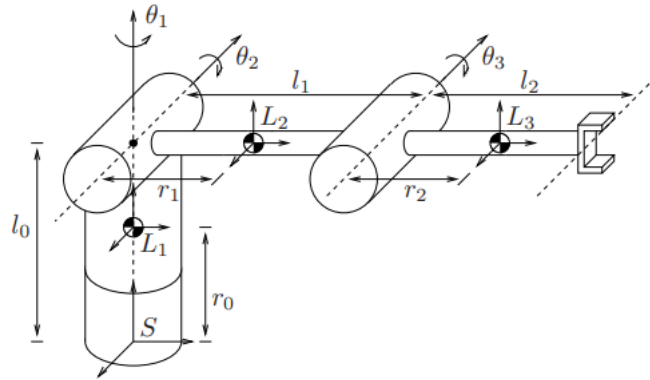


Figure 3: illustration du bras robotisés RRR avec ses paramètres [14]

La matrice de transformation du repère R_{i-1} au repère R_i s'exprime en fonction des paramètres de Denavit-Hartenberg suivants :

α_i : L'angle de rotation entre l'axe z_{i-1} et l'axe z_i autour de x_{i-1} .

d_i : La distance entre l'axe z_{i-1} et l'axe z_i le long de l'axe x_{i-1} .

θ_i : L'angle de rotation entre les axes x_{i-1} et x_i autour de z_i .

r_i : la distance entre les axes x_{i-1} et x_i le long de l'axe z_{i-1} .

La variable q_i de l'articulation i , définissant l'orientation ou la position relative entre les articulations $i-1$ et i , est soit θ_i , soit r_i , selon le type d'articulation est Rotoïde ou prismatique respectivement. Ceci est défini par la relation [13]:

$$q_i = \bar{\sigma}_i \theta_i + \sigma_i r_i$$

Équation 2-11

Avec :

- $\sigma_i = 0$ si l'articulation i est Rotoïde.
- $\sigma_i = 1$ si l'articulation i est prismatique.
- $\bar{\sigma}_i = 1 - \sigma_i$.

De la même façon on définit la variable q_i :

$$\bar{q}_i = \sigma_i \theta_i + \bar{\sigma}_i r_i \quad \text{Équation 2-12}$$

La matrice de transformation définissant R_i dans R_{i-1} est donnée par :

$${}^i T_i = \text{Trans}(x, d_i) \text{Trans}(z, r_i) \text{Rot}(z, \theta_i) \text{Rot}(z, \theta_i) \text{Rot}(x, \alpha_i) \quad \text{Équation 2-13}$$

$${}^i T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\alpha_i \sin\theta_i & d_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\cos\theta_i \sin\alpha_i & -d_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & r_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Équation 2-14}$$

Ci-dessous, nous retrouvons une image d'illustrations des éléments de la matrice de transformation.

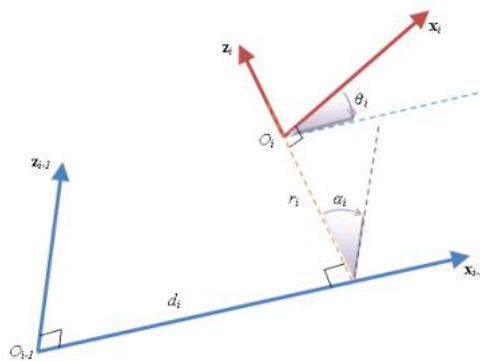


Figure 4: illustrations des éléments de la matrice de transformation sur leurs repères [13]

Le modèle géométrique direct (MGD) est l'ensemble des relations permettant de définir la position de l'organe terminal du robot en fonction de ses coordonnées articulaires. Pour un robot à chaîne ouverte simple le MGD est défini par la matrice de transformation 0T_n [13]:

$${}^0T_n = {}^0T_1(q_1) \times {}^1T_2(q_2) \times {}^2T_3(q_3) \dots {}^{n-1}T_n(q_n) \quad \text{Équation 2-15}$$

Le MGD peut aussi être représenté par la relation :

$$X = f(q) \quad \text{Équation 2-16}$$

où q est le vecteur des variables articulaires tels que :

$$q = [q_1 \ q_2 \ q_3 \ \dots \ q_n]^T$$

2.5 MODELE GEOMETRIQUE INVERSE

Le modèle géométrique direct fournit l'emplacement de l'organe terminal en fonction des coordonnées articulaires. Le modèle géométrique inverse (MGI) consiste à déterminer les variables articulaires correspondant à une situation spécifique de l'organe terminal. Lorsqu'elles existent. Trois méthodes de calcul de MGI sont répandues [13]:

- La méthode de Paul, qui 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 avec trois articulations Rotoïde ou trois articulations prismatiques.
- La méthode de *Raghavan et Roth*, donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égale 16. Ci-dessous, nous retrouvons une image d'illustration du passage du modèle direct vers le modèle inverse.

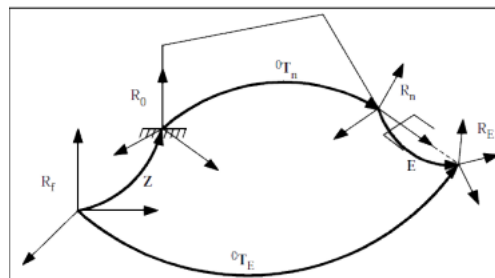


Figure 5: illustration du passage du modèle direct vers le modèle inverse [13]

Quand le modèle géométrique inverse ne peut pas être obtenu, des techniques numériques peuvent être employées. Ces techniques emploient la méthode de Newton-Raphson ou des méthodes fondées sur la transposé de la matrice Jacobienne [29].

Soit ${}^{i-1}T_E^d$ la matrice de transformation homogène représente la position désirée du repère outil RE par rapport au repère atelier R_f .

En général on peut exprimer ${}^{i-1}T_E^d$ sous la forme :

$${}^fT_E^d = Z {}^0T_n(q)E \quad \text{Équation 2-17}$$

Tels que :

- Z est la matrice de transformation définissant le repère de base du robot dans le repère atelier R_f .
- 0T_n est la matrice de transformation du repère terminal dans le repère R_0 .
- E est la matrice de transformation du repère outil RE dans le repère terminal R_n .

Mettant tous les termes connus dans le membre gauche, on obtient :

$$U_0 = {}^0T_n(q) \quad \text{Équation 2-18}$$

Avec, $U_0 = Z^{-1} {}^fT_E^d E^{-1}$

Le MGD donnant $X = f(q)$, $q = [q_1 \ q_2 \ q_3 \ \dots \ q_n]^T$, $X = [x_1 \ x_2 \ x_3 \ \dots \ x_m]^T$ ou n est le nombre de coordonnées opérationnelles et m le nombre de coordonnées articulaires, le problème est de résoudre un système de m équations à n inconnues. Le nombre de solutions dépend de l'architecture du robot manipulateur et de l'amplitude des articulations. Pour un système moins complexe, grâce aux méthodes de résolution des équations trigonométriques, on peut déterminer le modèle géométrique inverse.

2.5.1 LA MÉTHODE DE PAUL

Soit la matrice de transformation homogène d'un robot manipulateur :

$${}^0T_n = {}^0T_1(q_1) \times {}^1T_2(q_2) \times {}^2T_3(q_3) \dots {}^{n-1}T_n(q_n) \quad \text{Équation 2-19}$$

Soit U_0 la situation désirée telle que :

$$U_0 = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Équation 2-20}$$

On cherche à résoudre le système d'équations suivant :

$$U_0 = {}^0T_1(q_1) \times {}^1T_2(q_2) \times {}^2T_3(q_3) \dots {}^{n-1}T_n(q_n)$$

Pour résoudre l'équation ci-haut, Paul a proposé une méthode : la Méthode de Paul. Elle consiste à pré-multiplier successivement les deux membres de l'équation par les matrices ${}^i T_{i-1}$ pour i allant de 1 à $n-1$. Ces opérations permettent d'isoler les variables d'articulations l'une après l'autre [9].

$$U_{j+1} = {}^{j+1}T_j U_j = {}^{j+1}T_n \text{ pour } j = 0, \dots, n.$$

Pour un robot à six degrés de liberté on procède comme suit :

$$U_0 = {}^0T_1 \times {}^1T_2 \times {}^2T_3 \times {}^3T_4 \times {}^4T_5 \times {}^5T_6$$

$${}^1T_0 U_0 = {}^1T_2 \times {}^2T_3 \times {}^3T_4 \times {}^4T_5 \times {}^5T_6$$

$${}^2T_1 U_1 = {}^2T_3 \times {}^3T_4 \times {}^4T_5 \times {}^5T_6$$

$${}^3T_2 U_2 = {}^3T_4 \times {}^4T_5 \times {}^5T_6$$

$${}^4T_3 U_3 = {}^4T_5 \times {}^5T_6$$

$${}^5T_4 U_4 = {}^5T_6$$

2.6 MODELE CINEMATIQUE DIRECT

Le modèle cinématique direct (MCD) d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires [11]. Il s'écrit:

$$\dot{X} = J(q)\dot{q} \quad \text{Équation 2-21}$$

où, $J(q)$ désigne la matrice Jacobienne de dimension $(m \times n)$ du mécanisme, égale à $\frac{\partial X}{\partial q}$.

Le calcul de la matrice Jacobienne peut se faire en dérivant le MGD, $X = f(q)$, à partir de la relation suivante :

$$J_{ij} = \frac{\partial f_i}{\partial q_j} \quad i=1,\dots,m ; j=1,\dots, n \quad \text{Équation 2-22}$$

Où J_{ij} est l'élément (i, j) de la matrice jacobienne J .

Cette méthode est pratique pour des robots simples avec un nombre réduit de degrés de liberté. Le calcul de la matrice Jacobienne de base, connu sous le nom de matrice Jacobienne cinématique, est plus pratique pour un robot général de degré de liberté égale n . La matrice Jacobienne obtenue (sans calcul de la dérivée du MGD) relie les vecteurs des vitesses de translation et de rotation V_n et W_n et les vitesses articulaires [29] :

$$\begin{bmatrix} V_n \\ W_n \end{bmatrix} = J_n \dot{q} \quad \text{Équation 2-23}$$

où V_n et W_n sont les vitesses linéaire et angulaire du repère R_n respectivement.

La vitesse \dot{q}_k de l'articulation k produit une vitesse linéaire ($V_{k,n}$) et une vitesse angulaire ($W_{k,n}$) sur le repère terminal R_n . Deux cas se présentent :

1. Si l'articulation k est prismatique ($\sigma_k = 1$)

$$\begin{cases} V_{k,n} = a_k \dot{q}_k \\ W_{k,n} = 0 \end{cases} \quad \text{Équation 2-24}$$

où, a_k est le vecteur unitaire porté par l'axe z_k de l'articulation k . Ci-dessous, nous retrouvons une image d'illustration de l'accélération angulaire prismatique.

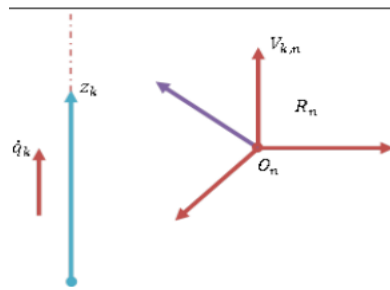


Figure 6 : illustration de l'orientation de l'accélération angulaire [13]

2. Si l'articulation k est rotoïde ($\sigma_k = 0$)

$$\begin{cases} V_{k,n} = a_k \dot{q}_k \times L_{k,n} = (a_k \times L_{k,n}) \dot{q}_k \\ W_{k,n} = a_k \dot{q}_k \end{cases} \quad \text{Équation 2-25}$$

$L_{k,n}$ désigne le vecteur $\overrightarrow{O_k O_n}$

De façon générale, les vecteurs $V_{k,n}$ et $W_{k,n}$ s'écrivent sous la forme :

$$\begin{cases} V_{k,n} = a_k \dot{q}_k \times L_{k,n} = [\sigma_k a_k + \overline{\sigma}_k (a_k \times L_{k,n})] \dot{q}_k \\ W_{k,n} = \overline{\sigma}_k a_k \dot{q}_k \end{cases} \quad \text{Équation 2-26}$$

Ci-dessous, nous retrouvons une image d'illustration de l'accélération angulaire rotoïde.

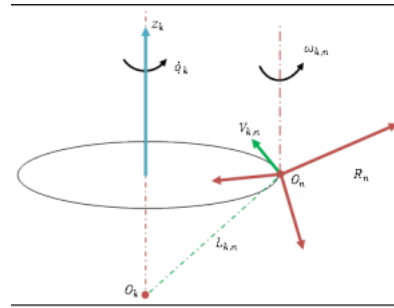


Figure 7: illustration de l'orientation de l'accélération angulaire [13]

Les vitesses linéaires et angulaires de l'organe terminal peuvent être écrites comme :

$$\begin{cases} V_n = \sum_{k=1}^n V_{k,n} = \sum_{k=1}^n [\sigma_k a_k + \overline{\sigma}_k (a_k \times L_{k,n})] \dot{q}_k \\ W_n = \sum_{k=1}^n W_{k,n} = \sum_{k=1}^n \overline{\sigma}_k a_k \dot{q}_k \end{cases} \quad \text{Équation 2-27}$$

Nous avons aussi :

$$J_n = \begin{bmatrix} \sigma_1 a_1 + \overline{\sigma}_1 (a_1 \times L_{1,n}) & \dots & \sigma_k a_k + \overline{\sigma}_k (a_k \times L_{k,n}) \\ \overline{\sigma}_1 a_1 & \dots & \overline{\sigma}_k a_k \end{bmatrix} \quad \text{Équation 2-28}$$

Se référant les vecteurs de J_n par rapport au repère R_i , on obtient la matrice jacobienne ${}^i J_n$, telle que :

$${}^i \begin{bmatrix} V_n \\ W_n \end{bmatrix}_n = {}^i J_n \dot{q} \quad \text{Équation 2-29}$$

En général, on calcul V_n et W_n dans R_n et R_0 , la matrice jacobienne correspondante est ${}^n J_n$ ou ${}^0 J_n$ respectivement. Ces matrices peuvent être aussi calculées en utilisant une matrice ${}^i J_n$, $j=0, \dots, n$, grâce à l'expression suivante :

$${}^s J_n = \begin{bmatrix} {}^s A_i & 0_3 \\ 0_3 & {}^s A_i \end{bmatrix} {}^i J_n \quad \text{Équation 2-30}$$

Où ${}^s A_i$ est la matrice d'orientation du repère R_i dans R_s .

La k ième colonne de la matrice ${}^i J_n$ notée par ${}^i J_{n:k}$ est donnée par:

$${}^i J_{n:k} = \begin{bmatrix} \sigma_k \times {}^i a_k + \overline{\sigma}_k \times {}^i A_k \times {}^k \widehat{a}_k \times {}^k L_{n,k} \\ \overline{\sigma}_k \times {}^i a_k \end{bmatrix} \quad \text{Équation 2-31}$$

2.7 MODELE CINEMATIQUE INVERSE

Le modèle cinématique inverse (MCI) donne les vitesses articulaires \dot{q} correspondants à une vitesse désirée \dot{X} de l'organe terminal. Le modèle cinématique inverse s'obtient par la solution d'un système d'équations linéaires soit analytiquement, soit numériquement. Les solutions analytiques diminuent le nombre d'opérations de façon remarquable par rapport au solutions numériques, mais il faut traiter les cas singuliers séparément. Les solutions numériques sont plus générales et traitent tous les cas de la même façon [5].

le MCI s'écrit sous la forme : $\dot{X} = J\dot{q}$. La méthode la plus générale consiste à calculer J^{-1} , la matrice inverse de J , qui permet de déterminer les vitesses articulaires \dot{q} grâce à la relation :

$$\dot{q} = J^{-1}\dot{X} \quad \text{Équation 2-32}$$

Lorsque la matrice J a la forme : $J = \begin{bmatrix} A & 0 \\ B & C \end{bmatrix}$ Équation 2-33

. Les matrices A et C étant carrées inversibles, il est facile de montrer que l'inverse de cette

matrice s'écrit : $J^{-1} = \begin{bmatrix} A^{-1} & 0 \\ -C^{-1}BA^{-1} & C^{-1} \end{bmatrix}$ Équation 2-34

La résolution du problème se ramène donc à l'inversion de la matrice J [24].

2.8 MODELE DYNAMIQUE

Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires [13] [26] [11]. Il est représenté par la relation :

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad \text{Équation 2-35}$$

Avec :

- Γ : vecteur des couples des actionneurs, selon que l'articulation est Rotoïde ou prismatique.
- q : vecteur des positions articulaires.
- \dot{q} : vecteurs des vitesses articulaires.
- \ddot{q} : vecteurs des accélérations articulaires.
- f_e : vecteurs représentant l'effort extérieur qu'exerce le robot sur l'environnement.

L'Équation 2-36 est appelée modèle dynamique inverse (ou modèle dynamique) parce qu'elle définit le système d'entrée en fonction des variables sorties. Le modèle dynamique direct décrit les accélérations articulaires en fonction des positions, vitesses et couples. Il est représenté par la relation suivante [24] :

$$\ddot{q}=g(q, \dot{q}, \Gamma, f_e)$$

Équation 2-37

Le modèle dynamique joue un rôle important dans la conception et le fonctionnement des robots. Pour la conception, le modèle dynamique inverse peut être utilisé pour choisir les actionneurs, alors que le modèle dynamique direct est utilisé pour effectuer des simulations, afin de tester les performances du robot. En ce qui concerne les fonctionnalités du robot, le modèle dynamique inverse est utilisée pour calculer les couples actionneurs, qui sont nécessaires pour réaliser un mouvement souhaité. Il est également utilisé pour identifier les paramètres dynamiques qui sont nécessaires à la fois pour le contrôle et la simulation [29]. Plusieurs approches sont proposées pour obtenir le modèle dynamique des robots. Les plus souvent utilisé dans la robotique sont le formalisme de Lagrange, et le formalisme de Newton-Euler. L'approche de Newton-Euler est basées sur les forces et les moments agissent entre les liens. La formulation de Newton-Euler peut être considéré comme une approche basée sur l'équilibre des forces, la formulation de Lagrange est une approche basée sur l'énergie. Bien sûr, pour le même manipulateur, les deux donnent les mêmes équations du mouvement [27].

2.8.1 FORMALISME DE LAGRANGE

On considère un système idéal sans frottement ou élasticité, exerçant ni des forces ni des moments sur l'environnement. La formulation de Lagrange décrit le comportement d'un système dynamique en termes d'énergie. Les équations de Lagrange sont généralement écrites sous la forme [5]:

$$\Gamma = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q}$$

Équation 2-38

où L est le Lagrangien du système, se définit comme la différence entre l'énergie cinétique K et l'énergie potentiel P :

$$L = K - P$$

Équation 2-39

Pour obtenir l'équation dynamique générale du robot, on détermine les énergies cinétiques et potentielles, et le lagrangien, puis les remplacés dans l'équation de Lagrange. Dans modélisation, q sera le vecteur des variables articulaires, se composant de l'angle d'articulation

θ_i et le déplacement de l'articulation d_i . Alors que Γ est un vecteur qui a comme composantes les couples correspondants aux angles θ_i , et les forces f_i correspondantes aux déplacements d_i [26].

2.8.2 L'ÉNERGIE CINÉTIQUE

L'énergie cinétique k_i de la i ième articulation est donnée par l'expression [13] :

$$k_i = \frac{1}{2} m_i v_{c_i}^T v_{c_i} + \frac{1}{2} \times {}^i W_i^T \times {}^{c_i} I_i \times {}^i W_i \quad \text{Équation 2-40}$$

où le premier terme est l'énergie cinétique due à la vitesse linéaire et le deuxième terme est l'énergie cinétique due à la vitesse angulaire. L'énergie cinétique totale du manipulateur est la somme de l'énergie cinétique des différentes articulations c'est-à-dire :

$$K = \sum_{k=1}^n k_i \quad \text{Équation 2-41}$$

Les vitesses $v_{c_i}^T$ et ${}^i W_i^T$ sont des fonctions de q et \dot{q} respectivement, donc on voit que l'énergie cinétique d'un manipulateur peut être décrite par une formule en fonction de la position et de la vitesse, $k(q, \dot{q})$. En fait, l'énergie cinétique d'un manipulateur est donnée par :

$$K(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad \text{Équation 2-42}$$

où $M(q)$ est une matrice $n \times n$ de l'énergie cinétique appelée matrice d'inertie. Une expression de la forme ci-dessus est appelé la forme quadratique. L'énergie cinétique doit toujours être positive, pour cela la matrice d'inertie du manipulateur doit être une matrice définie positive [3].

2.8.3 L'ÉNERGIE POTENTIEL

L'énergie potentiel du i ième articulation, a l'expression suivantes :

$$p_i = m_i \times {}^0 g^T \times {}^0 T_{c_i} + p_{refs} \quad \text{Équation 2-43}$$

où ${}^0 g$ est le vecteur de gravité (3×1), ${}^0 T_{c_i}$ une transformation homogène localisant le centre de la i ième articulation, et p_{refs} est une constante choisie de sorte que la valeur minimum de

p_i soit 0. L'énergie potentielle totale est la somme de l'énergie potentielle dans les différentes articulations, c'est-à-dire [13]:

$$P = \sum_{k=1}^n p_i \quad \text{Équation 2-44}$$

Puisque ${}^0T_{c_i}$ sont en fonction de $q_1, q_2, q_3, \dots, q_n$, l'énergie potentielle est en fonction de la position de l'articulation, $P(q)$. Donc le Lagrangien sera :

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) \quad \text{Équation 2-45}$$

2.8.4 EQUATION DU MOUVEMENT D'UN ROBOT MANIPULATEUR

Le Lagrangien a la forme :

$$L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - P(q) \quad \text{Équation 2-46}$$

Où $M(q)$ est la matrice d'inertie du manipulateur et $P(q)$ est l'énergie potentielle due au gravité. Il est convenable d'écrire l'énergie cinétique sous forme d'une somme :

$$L(q, \dot{q}) = \frac{1}{2} \sum_{i,j=1}^n M_{ij}(q) \dot{q}_i \dot{q}_j - P(q) \quad \text{Équation 2-47}$$

en substituant dans l'équation de Lagrange, les équations de mouvements sont données par :

$$\Gamma_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad \text{Équation 2-48}$$

où Γ_i sont les forces (couples) agissant sur l'articulation i , en utilisant l'équation (1.52) on a :

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} = \frac{d}{dt} \left(\sum_{k=1}^n M_{ik}(q) \dot{q}_k \right) = \sum_{j=1}^n \left(M_{ij}(q) \ddot{q}_j + \dot{M}_{ij}(q) \dot{q}_j \right) \quad \text{Équation 2-49}$$

$$\frac{\partial L}{\partial q_i} = \frac{1}{2} \sum_{j,k=1}^n \frac{\partial M_{kj}(q)}{\partial q_i} \dot{q}_k \dot{q}_j - \frac{\partial P(q)}{\partial q_i} \quad \text{Équation 2-50}$$

Le terme $\dot{M}(q)$ peut-être augmenté en termes de dérivées partiels.

$$\Gamma_i = \sum_{j=1}^n M_{ij}(q) \ddot{q}_j + \sum_{j,k=1}^n \frac{\partial M_{kj}(q)}{\partial q_k} \dot{q}_j \dot{q}_k - \frac{1}{2} \frac{\partial M_{kj}(q)}{\partial q_i} \dot{q}_k \dot{q}_j + \frac{\partial P(q)}{\partial q_i} \quad \text{Équation 2-51}$$

Afin de mettre les équations du mouvement sous forme d'un vecteur, on définit la matrice $C(q, \dot{q})$, telle que :

$$C_{ij}(q, \dot{q}) = \frac{1}{2} \sum_{k=1}^n \left(\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{kj}}{\partial q_i} \right) \dot{q}_k \quad \text{Équation 2-52}$$

Maintenant l'équation (1.58) peut être écrite comme :

$$\Gamma = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) \quad \text{Équation 2-53}$$

où :

- Γ : vecteur des couples appliqués aux articulations.
- $M(q)$: matrice d'inertie du robot.
- $C(q, \dot{q}) \dot{q}$: vecteur des forces de Coriolis et centrifuges.
- $G(q)$: vecteur des forces de gravité

Pour un système avec frottement ou élasticité, le modèle dynamique est :

$$\Gamma = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F_e \quad \text{Équation 2-54}$$

Il s'agit d'une équation différentielle du second ordre pour le mouvement du manipulateur en fonction des couples articulaires appliqués [13]. Les matrices M et C , qui résument les propriétés d'inertie du manipulateur, ont les propriétés :

- La matrice d'inertie $M(q)$ est symétrique et définie positive.
- La matrice $M(q)$ est donnée comme suit :
 - $M_m I \leq M(q) \leq M_M I$
 - avec M_m et M_M des scalaires positives, et I la matrice d'inertie.

- L'inverse de la matrice $M(q)$ est borné :

$$\square \frac{1}{M_M} \leq M^{-1}(q) \leq \frac{1}{M_m}$$

- La matrice $N(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q})$ est antisymétrique.
- La matrice $C(q, \dot{q})$ vérifie la relation suivante :

$$\square C(q, x)y = C(q, y)x \quad \text{Équation 2-55}$$

avec x, y les vecteurs de vitesse de dimension $(n \times 1)$.

- la matrice $C(q, \dot{q})\dot{q}$ peut-être écrite sous la forme :

$$\square C(q, \dot{q})\dot{q} = B(q)[q\dot{q}] + C(q)[\dot{q}^2] \quad \text{Équation 2-56}$$

- $B(q)$: une matrice de dimension $n \times n(n-1)$ des coefficients de Coriolis ses éléments sont égaux à :

$$b_{ijk} = \frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \quad \text{Équation 2-57}$$

- $C(q)$: est une matrice de dimension $(n \times n)$ des coefficients centrifuges et $[\dot{q}^2]$ est un vecteur de dimension $(n \times n)$, les éléments de $C(q)$ sont donnés par :

$$C(q) = \frac{1}{2} \left(2 \frac{\partial m_{ki}}{\partial q_k} - \frac{\partial m_{jj}}{\partial q_k} \right) \quad \text{Équation 2-58}$$

2.9 MODÉLISATION DU ROBOT ÉTUDIÉ

Ci-dessous, nous avons la représentation dans l'espace d'un bras robotisé à structure anthropomorphe avec ces paramètres de Denavit-Hartenberg.

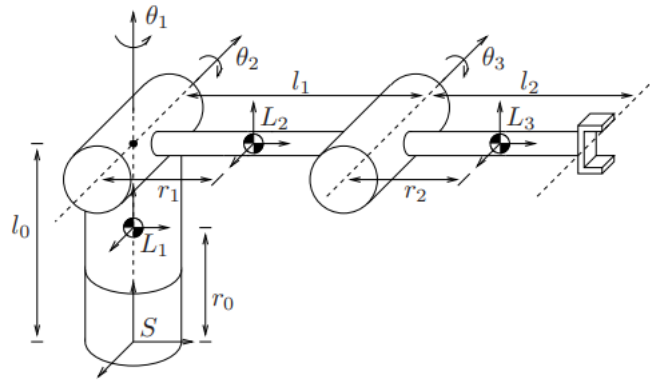


Figure 8: illustration du bras robotisés RRR avec ses paramètres

2.9.1 MODELE GEOMETRIQUE DIRECTE

Voici les paramètres de Denavit-Hartenberg déterminés pour notre bras robotisé :

Tableau 1 : tableau donnant les paramètres de Denavit-Hartenberg du robot étudié

	σ_i	α_i	d_i	r_i	θ_i
1	0	$\frac{\pi}{2}$	0	l_1	q_1
2	0	0	l_2	0	q_2
3	0	0	l_3	0	q_3

Les matrices de transformations pour chaque articulation sont ainsi données par :

$${}^0T_1 = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & 0 \\ \sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2 \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & -l_2 \sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & l_3 \cos q_3 \\ \sin q_3 & \cos q_3 & 0 & -l_3 \sin q_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Le modèle géométrique directe est exprimé comme suit :

$${}^0T_3 = {}^0T_1 \times {}^1T_2 \times {}^2T_3$$

La situation de l'organe terminal est défini par la matrice :

$${}^0T_3 = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Après calcul de 0T_3 , on obtient par identification :

$$P_x = l_2 \cdot \cos(q_1) \cdot \cos(q_2) + l_3 \cdot \sin(q_2) \cdot \sin(q_3) \cdot \cos(q_1) + l_3 \cdot \cos(q_1) \cdot \cos(q_2) \cdot \cos(q_3)$$

$$P_y = l_2 \cdot \sin(q_1) \cdot \cos(q_2) + l_3 \cdot \sin(q_1) \cdot \sin(q_2) \cdot \sin(q_3) + l_3 \cdot \sin(q_1) \cdot \cos(q_2) \cdot \cos(q_3)$$

$$P_z = l_1 - l_2 \cdot \sin(q_2) + l_3 \cdot \sin(q_2) \cdot \cos(q_3) - l_3 \cdot \sin(q_3) \cdot \cos(q_2)$$

Après simplification le modèle géométrique directe devient:

$$P_x = l_2 \cdot \cos(q_1) \cdot \cos(q_2) + l_3 \cdot \cos(q_1) \cdot \cos(q_3 - q_2)$$

$$P_y = l_2 \cdot \sin(q_1) \cdot \cos(q_2) + l_3 \cdot \sin(q_1) \cdot \cos(q_3 - q_2)$$

$$P_z = l_1 - l_2 \cdot \sin(q_2) + l_3 \cdot \sin(q_3 - q_2)$$

Test du modèle

Pour $q_1 = q_2 = q_3 = 0$

$$P_x = l_2 + l_3$$

$$P_y = 0$$

$$P_z = l_1$$

Ci-dessous, nous retrouvons une image d'illustration de la position prise par le bras quand on lui applique les angles ci-hauts.

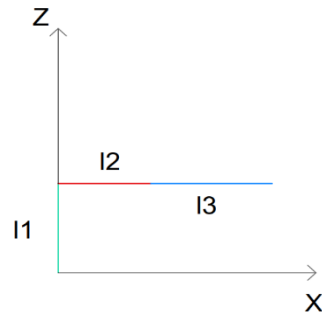


Figure 9: illustration de la position de du bras robotique après application des vitesses angulaires de test

Pour $q_1 = q_2 = 0, q_3 = 90$ degré

$$P_x = l_2$$

$$P_y = 0$$

$$P_z = l_1 + l_3$$

Ci-dessous, nous retrouvons une image d'illustration de la position prise par le bras quand on lui applique les angles ci-hauts

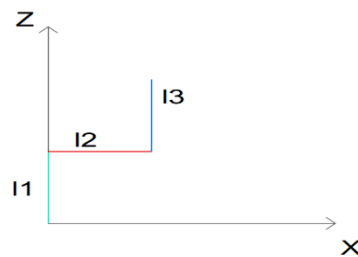


Figure 10: illustration de la position de du bras robotique après application de deuxième valeurs de vitesses angulaires de test

2.9.2 MODELE GEOMETRIQUE INVERSE

A partir des équations du modèle géométrique directe, nous pouvons obtenir grâce aux méthodes de résolution des équations trigonométriques, le modèle géométrique inverse suivant :

$$q_3 = \cos^{-1} \left(\frac{P_x^2 + P_y^2 + (P_z - l_1)^2 - l_3^2 - l_2^2}{2l_1 l_3} \right)$$

$$q_2 = \cos^{-1} \left(\frac{(l_2 + l_3 \cos q_3)(P_x^2 + P_y^2)^{\frac{1}{2}} + l_3 \sin q_3 (P_z - l_1)^2}{(l_2 + l_3 \cos q_3)^2 + (l_3 \sin q_3)^2} \right)$$

$$q_2 = \sin^{-1} \left(\frac{l_3 \sin q_3 (P_x^2 + P_y^2)^{1/2} + (l_2 + l_3 \cos q_3)(P_z - l_1)^2}{(l_2 + l_3 \cos q_3)^2 + (l_3 \sin q_3)^2} \right)$$

$$q_1 = \cos^{-1} \left(\frac{P_x}{l_3 \cos(q_3 + q_2) + l_2 \sin q_2} \right) = \sin^{-1} \left(\frac{P_y}{l_3 \cos(q_3 + q_2) + l_2 \sin q_2} \right)$$

Test du modèle

Pour :

$$P_x = l_2 + l_3$$

$$P_y = 0$$

$$P_z = l_1$$

On parvient à obtenir grâce au modèle géométrique inverse : $q_1 = q_2 = q_3 = 0$.

Ci-dessous, nous retrouvons une image d'illustration de la position prise par le bras quand on lui applique les valeurs de P_x , P_y , P_z .

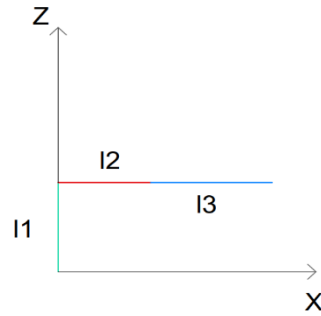


Figure 11: illustration de la position de du bras robotique après application des vitesses angulaires de test

Pour :

$$P_x = l_2$$

$$P_y = 0$$

$$P_z = l_1 + l_3$$

On parvient à obtenir grâce au modèle géométrique inverse : $q_1 = q_2 = 0, q_3 = 90$ degré.

Ci-dessous, nous retrouvons une image d'illustration de la position prise par le bras quand on lui applique les valeurs de P_x , P_y , P_z .

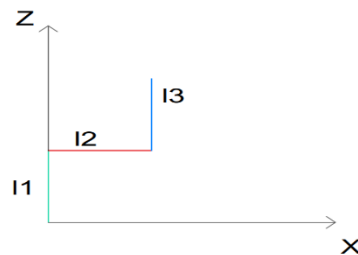


Figure 12: illustration de la position de du bras robotique après application de deuxièmes valeurs de vitesses angulaires de test.

2.9.3 MODELE CINEMATIQUE DIRECT

Le modèle cinématique directe est exprimé par :

$$\dot{X} = J(q)\dot{q} \quad \text{Équation 2-59}$$

La matrice jacobienne peut être déterminée grâce au modèle géométrique direct par :

$$J_{ij} = \frac{\partial f_i}{\partial q_j} \quad i=1,\dots,m ; j=1,\dots, n \quad \text{Équation 2-60}$$

Le modèle géométrique de notre bras robotisé est exprimé par les équations suivantes :

$$f_x = P_x = l_2 * \cos(q_1) * \cos(q_2) + l_3 * \cos(q_1) * \cos(q_3 - q_2)$$

$$f_y = P_y = l_2 * \sin(q_1) * \cos(q_2) + l_3 * \sin(q_1) * \cos(q_3 - q_2)$$

$$f_z = P_z = l_1 - l_2 * \sin(q_2) + l_3 * \sin(q_3 - q_2)$$

Le modèle cinématique direct est donc exprimé par :

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \frac{\partial f_x}{\partial q_2} & \frac{\partial f_x}{\partial q_3} \\ \frac{\partial f_y}{\partial q_1} & \frac{\partial f_y}{\partial q_2} & \frac{\partial f_y}{\partial q_3} \\ \frac{\partial f_z}{\partial q_1} & \frac{\partial f_z}{\partial q_2} & \frac{\partial f_z}{\partial q_3} \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

Après calcul des dérivées nous obtenons comme éléments de la matrice jacobienne les éléments suivants :

$$\frac{\partial f_x}{\partial q_1} = -l_2 * \sin(q_1) * \cos(q_2) - l_3 * \sin(q_1) * \cos(q_3 - q_2)$$

$$\frac{\partial f_x}{\partial q_2} = -l_2 * \cos(q_1) * \sin(q_2) + l_3 * \cos(q_1) * \sin(q_3 - q_2)$$

$$\frac{\partial f_x}{\partial q_3} = -l_3 * \cos(q_1) * \sin(q_3 - q_2)$$

$$\frac{\partial f_y}{\partial q_1} = l_2 * \cos(q_1) * \cos(q_2) + l_3 * \cos(q_1) * \cos(q_3 - q_2)$$

$$\frac{\partial f_y}{\partial q_2} = -l_2 * \sin(q_1) * \sin(q_2) + l_3 * \sin(q_1) * \sin(q_3 - q_2)$$

$$\frac{\partial f_y}{\partial q_3} = -l_3 * \sin(q_1) * \sin(q_3 - q_2)$$

$$\frac{\partial f_z}{\partial q_1} = 0$$

$$\frac{\partial f_z}{\partial q_2} = -l_2 * \cos(q_2) - l_3 * \cos(q_3 - q_2)$$

$$\frac{\partial f_z}{\partial q_3} = l_3 * \cos(q_3 - q_2)$$

Le modèle cinématique directe devient donc :

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = \begin{bmatrix} -l_2 * \sin(q_1) * \cos(q_2) - l_3 * \sin(q_1) * \cos(q_3 - q_2) & -l_2 * \cos(q_1) * \sin(q_2) + l_3 * \cos(q_1) * \sin(q_3 - q_2) & -l_3 * \cos(q_1) * \sin(q_3 - q_2) \\ l_2 * \cos(q_1) * \cos(q_2) + l_3 * \cos(q_1) * \cos(q_3 - q_2) & -l_2 * \sin(q_1) * \sin(q_2) + l_3 * \sin(q_1) * \sin(q_3 - q_2) & -l_3 * \sin(q_1) * \sin(q_3 - q_2) \\ 0 & -l_2 * \cos(q_2) - l_3 * \cos(q_3 - q_2) & l_3 * \cos(q_3 - q_2) \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

2.9.4 MODELE CINEMATIQUE INVERSE

Le modèle cinématique inverse est exprimée par :

$$\dot{q} = J^{-1} \dot{X}$$

Équation 2-61

Le modèle cinématique inverse est donc exprimé par :

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \frac{\partial f_x}{\partial q_2} & \frac{\partial f_x}{\partial q_3} \\ \frac{\partial f_y}{\partial q_1} & \frac{\partial f_y}{\partial q_2} & \frac{\partial f_y}{\partial q_3} \\ \frac{\partial f_z}{\partial q_1} & \frac{\partial f_z}{\partial q_2} & \frac{\partial f_z}{\partial q_3} \end{bmatrix}^{-1} * \begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix}$$

Ici la tâche principale va consister à inverser la matrice jacobienne trouver en déterminant le modèle cinématique direct.

Après inversion de la matrice jacobienne, nous allons obtenir un modèle sous la forme :

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} -l_2 * \sin(q_1) * \cos(q_2) - l_3 * \sin(q_1) * \cos(q_3 - q_2) & -l_2 * \cos(q_1) * \sin(q_2) + l_3 * \cos(q_1) * \sin(q_3 - q_2) & -l_3 * \cos(q_1) * \sin(q_3 - q_2) \\ l_2 * \cos(q_1) * \cos(q_2) + l_3 * \cos(q_1) * \cos(q_3 - q_2) & -l_2 * \sin(q_1) * \sin(q_2) + l_3 * \sin(q_1) * \sin(q_3 - q_2) & -l_3 * \sin(q_1) * \sin(q_3 - q_2) \\ 0 & -l_2 * \cos(q_2) - l_3 * \cos(q_3 - q_2) & l_3 * \cos(q_3 - q_2) \end{bmatrix}^{-1} * \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix}$$

2.9.5 MODELE DYNAMIQUE

Le modèle dynamique d'un robot manipulateur humanoïde a 3 articulations est sous la forme :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \quad [30]$$

Avec:

- M : la matrice d'inertie.
- C : la matrice de termes Coriolis et centrifuge
- g : le vecteur des termes de force de gravité

Les équations de vecteurs des couples appliquées aux articulations sont les suivants :

$$\begin{cases} \tau_1 = M_{11}\ddot{q}_1 + M_{12}\ddot{q}_2 + M_{13}\ddot{q}_3 + h_{11}\dot{q}_1 + h_{12}\dot{q}_2 + h_{13}\dot{q}_3 + g_1 \\ \tau_2 = M_{21}\ddot{q}_1 + M_{22}\ddot{q}_2 + M_{23}\ddot{q}_3 + h_{21}\dot{q}_1 + h_{22}\dot{q}_2 + h_{23}\dot{q}_3 + g_2 \\ \tau_3 = M_{31}\ddot{q}_1 + M_{32}\ddot{q}_2 + M_{33}\ddot{q}_3 + h_{31}\dot{q}_1 + h_{32}\dot{q}_2 + h_{33}\dot{q}_3 + g_3 \end{cases} \quad [30]$$

Étant donné que nous avons choisi de commander notre bras robotique avec son modèle géométrique inverse, nous nous limitons donc à donner la forme générale du modèle dynamique de notre bras robotique à 3 articulations rotoïdes vu que son modèle dynamique ne sera pas utilisé.

2.10 CONCLUSION PARTIELLE

La modélisation de bras robotiques représente une avancée dans la robotique en ce sens qu'elle permet de créer de représentations précises et prédictives du comportement de bras robotiques.

Ces représentations sont obtenues grâce aux modèles mathématiques et physiques. Ces modèles rendent possibles la planification et le contrôle des mouvements du bras robotique.

Dans ce chapitre nous avons présenté les différentes manières de modéliser les bras robotiques. Nous avons présenté les bases mathématiques derrière chaque modélisation et abordé les différentes manières permettant d'implémenter ces modélisations. Nous avons ensuite donné le modèle géométrique direct et inverse de notre bras robotique anthropomorphe à 3 articulations. Nous avons aussi présenté son modèle cinématique. Ensuite nous avons montré la forme générale d'un modèle dynamique d'un bras robotique anthropomorphe à 3 articulations.

Chapitre 3 : CONCEPTION DU BRAS ROBOTISÉ

3.1 INTRODUCTION

Dans ce chapitre nous abordons la conception du bras robotisé. Nous présentons ainsi la constitution du bras robotisé en relevant les différents éléments électriques permettant son bon fonctionnement. Nous donnons aussi les différents modes de commandes prévus pour le bras robotisé. Dans le cadre de ces différents modes de commande nous allons devoir concevoir un régulateur basé sur la logique floue et une application Android.

3.2 CONSTITUTION DU BRAS ROBOTISÉ

Comme nous l'avons vu au premier chapitre, nous distinguons 5 parties dans un robot manipulateur : les actionneurs, le système mécanique articulé, l'organe terminal, les capteurs et la commande, avec un microcontrôleur comme élément central.

3.2.1 LA CARTE ARDUINO

Le montage électrique se fait avec comme élément centrale, une carte Arduino. Arduino est une plate-forme électronique open source basée sur du matériel et des logiciels faciles à utiliser. La carte Arduino est généralement construite autour d'un microcontrôleur Atmel AVR (ATmega328, ATmega32u4 ou ATmega2560 pour les versions récentes, ATmega168, ATmega1280 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). La carte Arduino que nous allons utiliser pour notre projet est de type Arduino MEGA 2560 [31].

Arduino MEGA 2560 dispose :

- de 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée) ;

- de 16 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques);
- de 4 UART (port série matériel) ;
- d'un quartz 16Mhz ;
- d'une connexion USB ;
- d'un connecteur d'alimentation jack ;
- d'un connecteur ICSP (programmation "in-circuit") ;
- et d'un bouton de réinitialisation (reset) ;

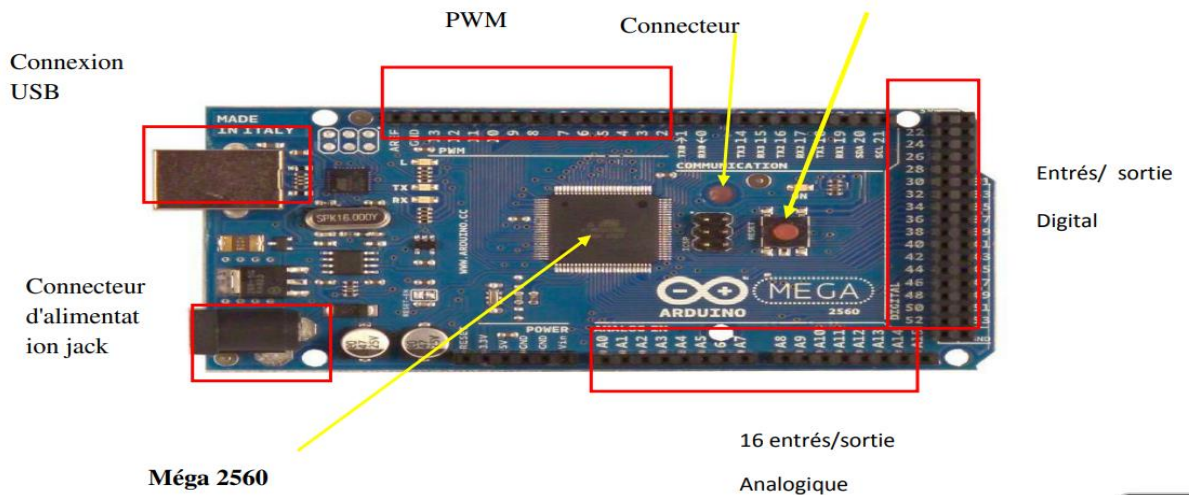


Figure 13 : la carte arduin- mega 2560 [31].

La carte Arduino MEGA contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur ; pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB). Ci-dessous, un tableau reprenant les caractéristiques de l'Arduino MEGA 2560:

Tableau 2 : caractéristiques de l'arduino-mega [31]

Microcontrôleur	ATMEGA 2560
Tension de fonctionnement	5V
Tension d'alimentation recommandée	7-12V
Tension d'alimentation limite	6-20V
Broches E/S numériques	54 dont 14 disposant d'une sortie PWM
Broches d'entrées analogiques	16
Vitesse d'horloge	16MHz

Le choix de cette carte est essentiellement motivé par le nombre de broches élevé qu'il fournit.

3.2.2 LES ACTIONNEURS

Les actionnaires utilisés pour notre bras robotisé sont des servomoteurs. Un servomoteur est tout simplement un moteur à courant continu qui est asservi en position à l'aide d'un capteur de position (typiquement un potentiomètre) et un circuit électronique interne au moteur. Ils sont pilotés par un fil de commande et alimentés par deux autres fils ; le premier est relié à l'alimentation positive +5 ou +6 V selon le servomoteur, le deuxième est relié à la masse (GND). Le signal de commande est de type modulation de largeur d'impulsion (PWM). En modifiant le rapport cyclique de ce signal, on indique au moteur quelle est la position désirée dans une plage de positions possibles, généralement entre 0 et 180°. La fréquence du signal à modulation de largeur d'impulsion est habituellement de l'ordre de 50 Hz (30 Hz pour certains modèles) avec des impulsions durant de 1 à 2 millisecondes. La figure 27 illustre les impulsions et les angles qui y correspondent [32].

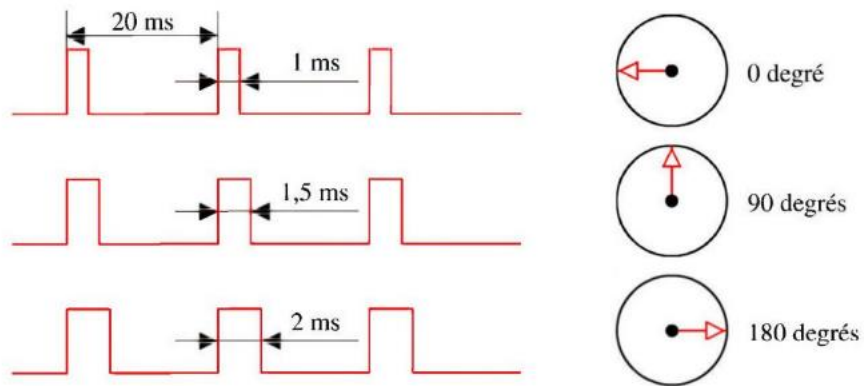


Figure 14 :illustrations des impulsions et des angles correspondants pour un servomoteur [32]

Nous avons utilisé deux types de servomoteurs illustrés par la figure ci-dessous.



Figure 15 :(a) est le SG90 et (b) est le MG996R [32]

Les caractéristiques de ces servomoteurs sont reprises dans le tableau ci-dessous :

Tableau 3 : tableau des caractéristiques des servomoteurs SG90 et MG996R [32]

	SG90	MG996R
Poids	9g	55g
Couple d'arrêt	1.8kgf.cm	9.4kgf.cm(4.8V) 11kgf.cm(6V)

Tension de fonctionnement	4.8V -5V	4.8V – 7.2V
vitesse	0.1s/60°	0.17s/60° (4.8V) 0.14s/60°(6V)
Rotation d'angle	180°	180°

Idéalement, nous aurions dû sélectionner les servomoteurs en fonction de leur couple d'arrêt, mais en raison de contraintes budgétaires, nous avons dû nous contenter des servomoteurs disponibles sur le marché local, ce qui a significativement limité la capacité de charge de notre bras robotisé.

3.2.3 LE SYSTEME MECANIQUE ARTICULÉ

Notre bras manipulateur étant un bras anthropomorphe, avec 3 degrés de liberté, nous avons subdivisé la structure de son système mécanique articulé en 3 parties qui sont : la base, l'avant-bras et le bras. Nous avons conçu ces parties grâce au logiciel de dessin 123Design et les avons imprimées en matériels plastiques grâce à une imprimante 3D.

Les figures 31, 32 et 33 illustrent les images correspondantes respectivement à la base, à l'avant-bras et au bras telles que conçu avec 123Design :

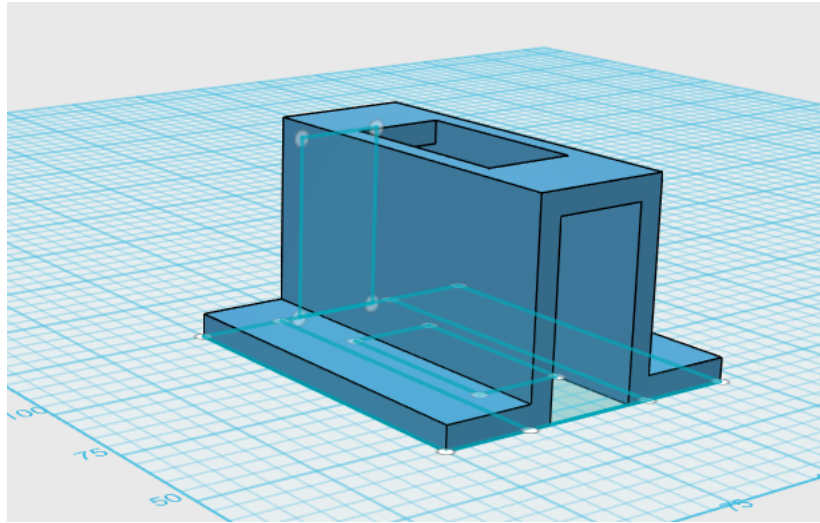


Figure 16 : illustration de la base du SMA

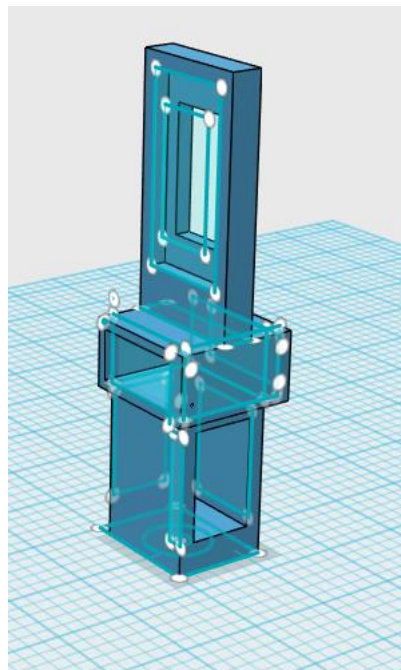


Figure 17 : illustration de l'avant-bras du SMA

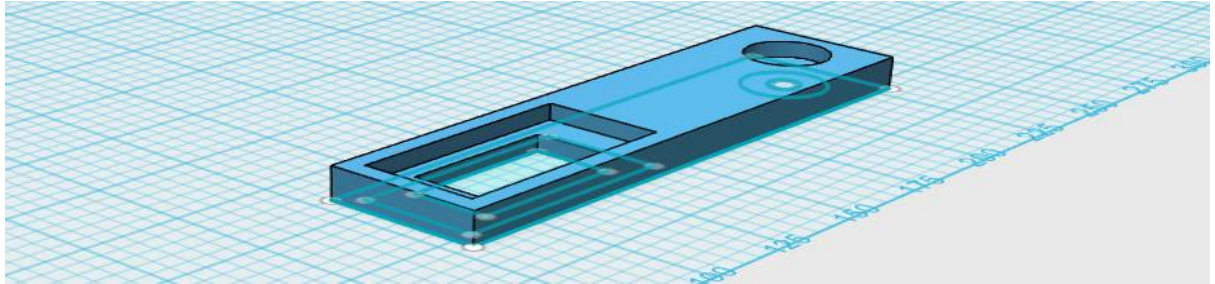


Figure 18 :illustration du bras du SMA

3.2.4 ORGANE TERMINALE

L'organe terminale du bras robotique dépend de la mission qu'on assigne au bras. Pour notre cas, la mission est la prise de bouteilles. L'organe terminale est dotée ainsi donc de deux doigts pour faciliter la prise, ces doigts sont montés sur une structure jouant le rôle de la main.

Ci-dessous les images d'illustrations de ces parties telles que conçu avec 123Design :

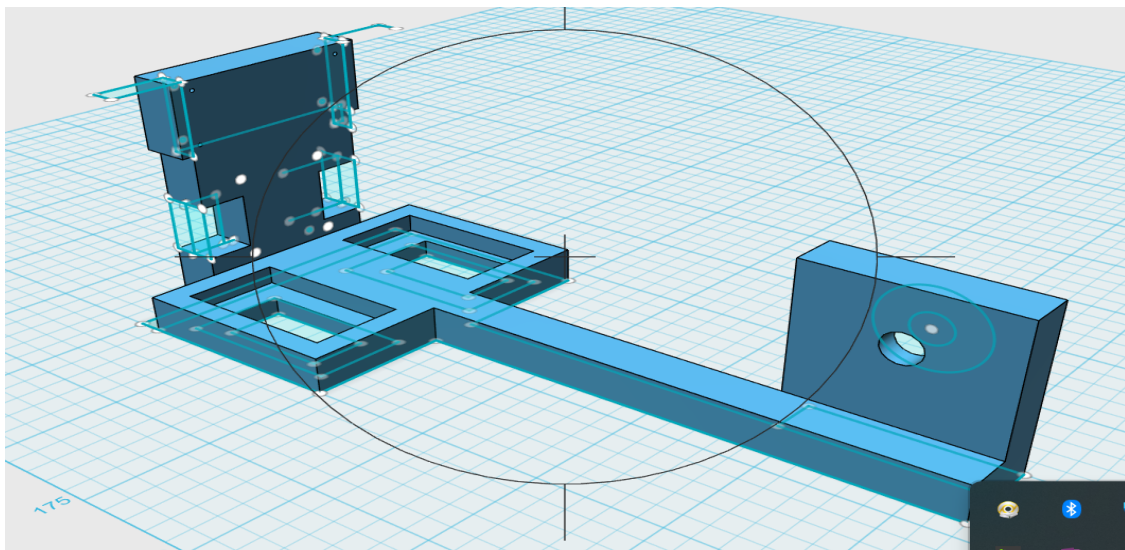


Figure 19 :illustration de la main de l'organe terminale qui va porter les doigts

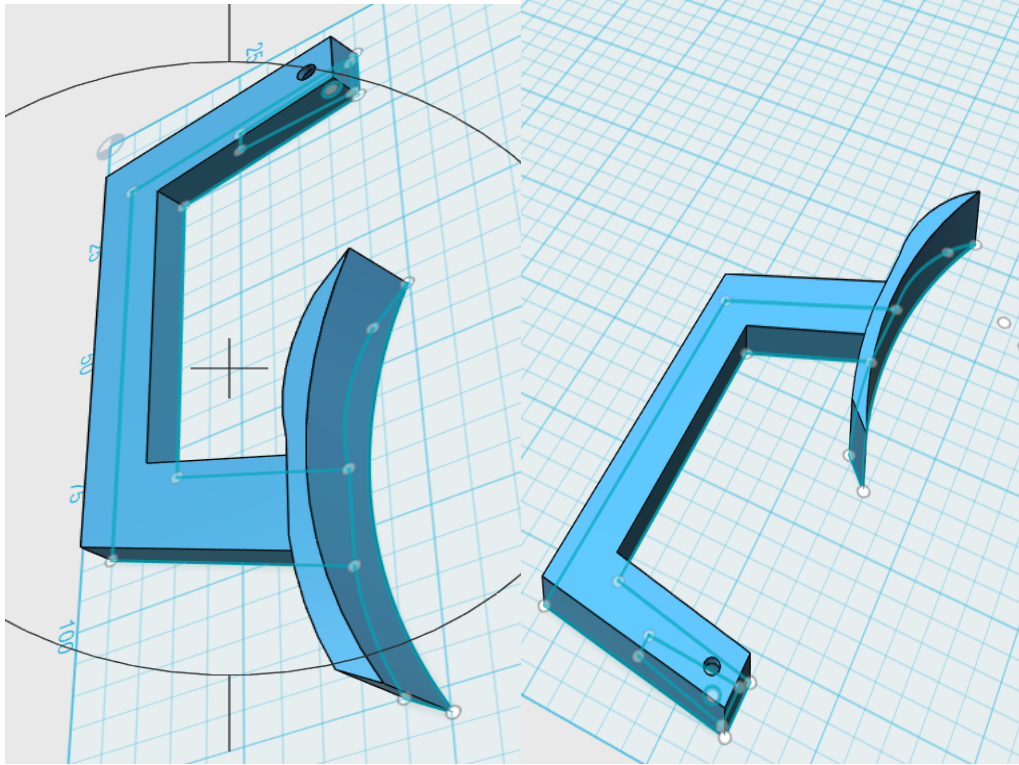


Figure 20 : illustrations de deux doigts de l'organe terminale du bras

3.2.5 LES CAPTEURS

Pour ce projet, deux types de capteurs seront utilisés. Il s'agit d'un capteur de couleur et de 3 capteurs de distances.

3.2.5.1 Le Capteur de distance à ultrasons HC-SR04

C'est celui que nous utilisons dans ce travail. Le capteur à ultrasons (également souvent appelé sonar ou télémètre à ultrasons) détermine la distance d'un objet de la même manière que les chauves-souris ou les dauphins. Le capteur de distance HC-SR04 génère un signal à bande étroite à une fréquence de 40 kHz et capte le signal réfléchi (écho). Sur la base du temps de propagation du son vers et depuis l'objet, la distance en centimètres de l'obstacle peut être déterminée assez précisément. Ci-dessous une image d'illustration du HC SR 04 [31].



Figure 21 : image d'illustration du HC SR 04 [31]

Caractéristiques:

- Dimensions: 45 mm x 20 mm x 15 mm
- Plage de mesure: 2 cm à 400 cm
- Resolution de la mesure: 0.3 cm
- Angle de mesure efficace: 15 °
- Largeur d'impulsion sur l'entrée de déclenchement: 10 μ s

Les caractéristiques de HC SR 04 sont reprises dans le tableau ci-dessous :

Tableau 4 :tableau regroupant les caractéristiques du capteurs HC SR 04 [31]

Paramètres électriques	HC SR 04
tension de fonctionnement	5V
Courant de fonctionnement	15 mA
fréquence de fonctionnement	40kHz
Distance minimale	2cm
Distance maximale	400cm
Couverture angulaire	15 degré

Ce capteur a été choisi à cause de facilité d'utilisation et de sa disponibilité sur le marché.

3.2.5.2 Le capteur de couleur

Le capteur de couleur est un détecteur de couleurs TCS3200. Il est composé de 4 LEDS blanches servant à éclairer l'objet exposé si nécessaire. Le capteur peut détecter et mesurer une gamme quasi illimitée de couleurs visibles. Ci-dessous une image d'illustration du TCS 3200 [33].

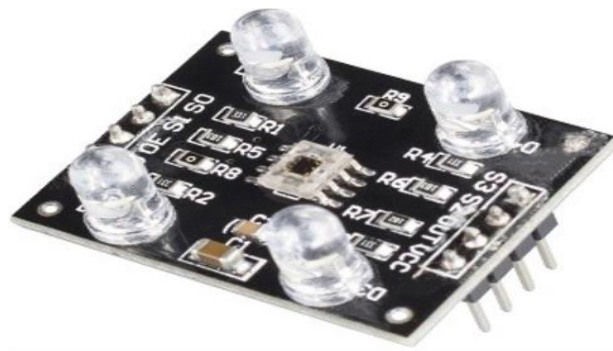


Figure 22 : illustration du TCS 3200 [33]

Il est caractérisé par:

- Tension d'entrée: (2,7 V à 5,5 V)
- Interface: TTL numérique.
- Conversion haute résolution de l'intensité lumineuse en fréquence
- Couleur programmable et fréquence de sortie pleine échelle
- Fonction de mise hors tension.
- Température de travail: -40 degré Celsius à 85 degré Celsius
- Taille: 28,4 x 28,4 mm (1,12 x 1,12 ").

Il dispose d'une Matrice de photo détectrice de 8x8. Cette Matrice de photodétectrice est composée de filtres rouges, verts et bleus. En effet lorsqu'on choisit un filtre de couleur, il ne peut laisser passer qu'une couleur particulière et empêche les autres de passer. Dans le capteur, se trouve un oscillateur qui produit un signal dont la fréquence est proportionnelle à l'intensité de la couleur choisie. Ce signal est ensuite transmis à un convertisseur qui le convertit en fréquence. Ci-dessous, l'illustration de la structure interne du capteur [33].

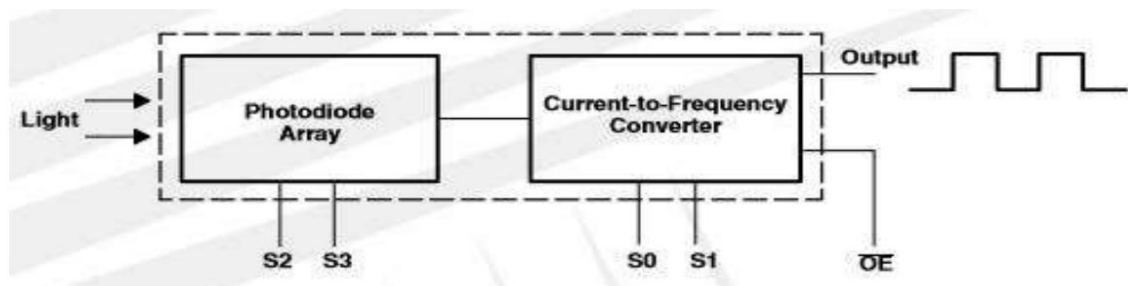


Figure 23 : l'illustration de la structure interne du capteur [33].

Le capteur TCS3200 est constitué de 8 pins:

- S0-S1: sélection de mise à l'échelle des fréquences de sortie
- S2-S3: sélection du type des photos détectrices (Filtres)

- OUT: fréquence de sortie (signal carré)
- OE: broche d'activation de sortie (OE)
- Vcc-GND broches d'alimentation.

Ce capteur a été choisi à cause de sa disponibilité sur le marché malgré les défis liés à son utilisation comme la nécessité d'étalonnage du capteur à chaque changement de luminosité de l'environnement de travail.

3.3 TYPE DE COMMANDE DU BRAS ROBOTISÉ

Nous avons opté pour divers modes de commande du bras robotisé, étant donné que les articulations du bras robotisé sont équipées de servomoteurs. Le système de commande du bras robotisé offre trois options : commande manuelle, commande via une application mobile et mode de fonctionnement automatique.

3.3.1 COMMANDE MANUELLE

La commande manuelle consiste à commander le bras robotisé manuellement grâce aux potentiomètres. Un opérateur humain donne des angles aux servomoteurs grâce aux potentiomètres pour atteindre finalement une cible.

- **Le potentiomètre**

Un potentiomètre est un type de résistance variable à trois bornes, dont une est reliée à un curseur se déplaçant sur une piste résistante limitée par les deux autres bornes. Ce système permet de recueillir, entre la borne reliée au curseur et une des deux autres bornes, une tension qui dépend de la position du curseur et de la tension à laquelle est soumise la résistance. Ci-dessous, une image d'illustration d'un potentiomètre [31].



Figure 24 : image d'illustration d'un potentiomètre [31]

Le potentiomètre possède 3 boches : une alimentation, une sortie analogique et une masse

En tournant l'axe du potentiomètre, nous modifions la résistance vers l'interface de sortie. L'entrée analogique de l'Arduino est codée sur 10 bits ($2^{10} = 1024$ valeurs). Quand nous envoyons la tension en sortie du potentiomètre vers l'entrée de l'Arduino, celle-ci va être convertie en un nombre numérique [31].

3.3.2 COMMANDE PAR APPLICATION MOBILE

La commande par application mobile, comme la commande manuelle consiste à commander le bras robotisé manuellement, non plus grâce aux potentiomètres mais grâce aux curseurs d'une application mobile. Un operateur humain donne des angles aux servomoteurs grâce aux curseurs d'une application mobile pour atteindre finalement une cible. L'application mobile communique avec le bras robotisé grâce à la technologie Bluetooth.

- **Module BluetoothHC-06 :**

La communication en Bluetooth est une norme permettant l'échange bidirectionnel de données à très courte distance en utilisant des ondes radio UHF sur une bande de fréquence de 2,4 GHz. Son objectif est de simplifier les connexions entre les appareils électroniques en supprimant des liaisons filaires. Pour ce qui est du module Bluetooth HC-06 est un module Bluetooth PPS (protocole de port série) facile à utiliser, conçu pour la configuration d'une connexion série. Ce

qui facilite l'interfaçage avec les microcontrôleurs via deux broches Rx et Tx, telle que la broche Rx de ce dernier doit être raccordée à la broche Tx du module Bluetooth HC-06 et la broche Tx doit être raccordée à la broche Rx du module HC-06. Ci-dessous une image d'illustration du module Bluetooth HC-06 [31].

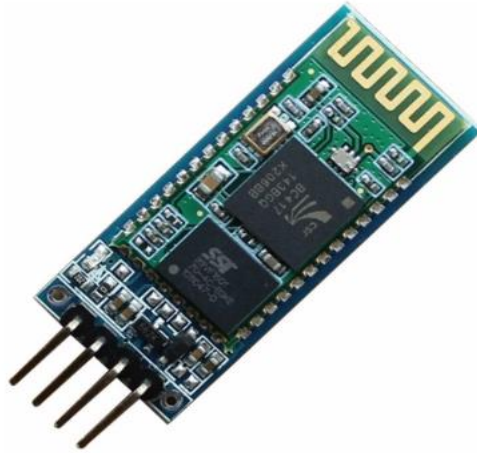


Figure 25 : d'illustration du module Bluetooth HC-06

Le HC-06 contient 4 broches :

- VCC broche d'alimentation connectée à la broche 5V de l'Arduino.
- GND masse connectée à la broche GND de l'Arduino
- RX broche de réception connecté à la broche de transmission (TX) de l'Arduino
- TX broche de transmission connecté à la broche de réception (RX) de l'Arduino [31]

Ce capteur a été choisi à cause de facilité d'utilisation et de sa disponibilité sur le marché.

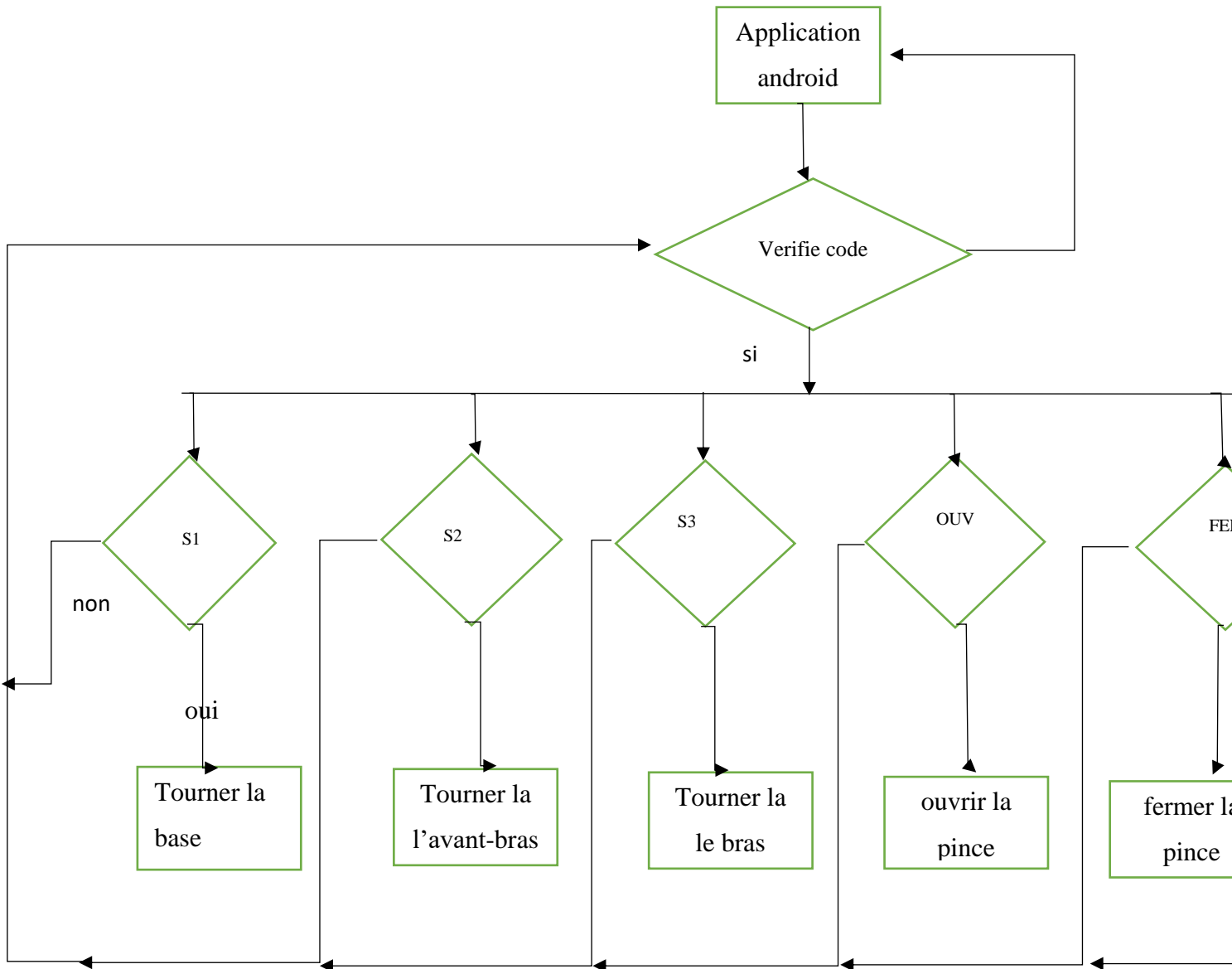
3.3.2.1 App Inventor pour Android

C'est un logiciel de développement d'applications créé par Google, pour pallier la difficulté d'utilisation de Android Studio par des débutants. Il est actuellement entretenu par le Massachusetts Institute of Technology (MIT).

Il simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont pas familiers avec les langages de programmation. Il est basé sur une interface graphique similaire à Scratch et à celle de StarLogo TNG. Grâce à son interface entièrement graphique et à l'absence totale de ligne de code, il est particulièrement adapté à l'initiation à la programmation [34].

La commande par application mobile de notre bras robotisé sera possible grâce aux données que nous allons envoyer par Bluetooth au microcontrôleur. Pour chaque articulation, une consigne sera envoyée et permettra la commande du servomoteur de l'articulation comme nous pouvons le voir sur le logigramme suivant.

Logigramme de l'application de commande du bras robotisé



3.3.3 COMMANDE AUTOMATIQUE

La commande automatique permet au bras robotisé, grâce aux capteurs de distance HC-SR 04 et au capteur de couleur TCS 3200, de pouvoir prendre et déplacer un objet dans le sens voulu en fonction de sa couleur. Nous allons donc concevoir un algorithme qui pourra commander chaque articulation en fonction des données reçues de ces capteurs. Le système comporte 3 capteurs de distances et un capteur de couleur.

Un capteur de distance est placé sur l'avant-bras du bras robotisé, dont le rôle est de détecter la présence d'un objet. En cas de détection de l'objet, une commande est faite pour que le bras et la main du bras robotisé se dirigent vers l'objet. Cette commande est faite lorsque l'objet se trouve à une distance comprise entre 20 et 23cm du capteur de distance se trouvant sur l'avant-bras du bras robotisé et le servomoteur de la base se positionne initialement à 90 degrés. Les doigts du bras robotisé, initialement ouverts, se ferment sur l'objet, et ensuite la détection de couleur est lancée. Si la couleur de l'objet est bleue, l'objet est déplacé à 0 degré par rapport au servomoteur de l'articulation de la base, si elle est rouge, l'objet est déplacé vers 180 degrés. Deux capteurs de distances sont placés sur deux surfaces opposées du bras manipulateur. Ces deux capteurs servent pour la détection et l'évitement d'obstacles. L'évitement d'obstacle est basé sur la logique floue.

3.4 REGULATEUR A LOGIQUE FLOU

La logique floue est une extension de la logique booléenne créée par Lotfi Zadeh en 1965 en se basant sur sa théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la vérification d'une condition, permettant ainsi à une condition d'être dans un autre état que vrai ou faux, la logique floue confère une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions et des incertitudes. Un des intérêts de la logique floue pour formaliser le raisonnement humain est que les règles sont énoncées en langage naturel. Voici par exemple quelques règles de conduite qu'un conducteur suit, en supposant qu'il tienne à son permis : Si le feu est rouge... si ma vitesse est élevée...alors je freine fort. Si

le feu est rouge... si ma vitesse est faible..... alors je maintiens ma vitesse [35]. Ce régulateur flou est choisi ici essentiellement pour permettre de gérer au mieux l'incertitude lié à la position(distance) que peut prendre un obstacle par rapport au bras robotisé.

3.4.1 PRINCIPE D'UN REGULATEUR FLOU

Le régulateur flou, comme tout régulateur, a pour tâche de produire une loi de commande $u(t)$ pour chaque combinaison de ses entrées de sorte que l'erreur tend vers zéro le plus vite possible. [2] La structure interne de ce régulateur flou est représentée dans le diagramme fonctionnel suivant :

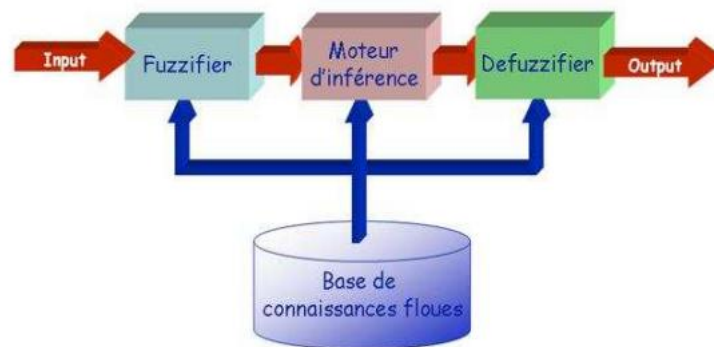


Figure 26 : le diagramme fonctionnel d'un régulateur flou. [35]

Nous avons trois blocs appelés fuzzification, inférence, et défuzzification qui ont comme fonctionnalité :

Fuzzification : Le processus de la fuzzification consiste à simplement attribuer aux différentes variables linguistiques d'entrée des fonctions d'appartenance convenables. Cette opération est nécessaire vu que la mesure de la sortie et la donnée de la référence ou consigne sont toujours exprimées en des variables qui sont nettes et précises et que le régulateur flou ne peut traiter que des variables floues [7].

Inférence floue : L'inférence est le processus par lequel des actions ou opérations floues sont prises au fur et à mesure que des informations sur les entrées sont recueillies et ce selon les règles définissant le système. Ils existent plusieurs méthodes d'inférence dont :

- La méthode d'inférences max-min ou méthode de Mamdani: réalise le « OU » reliant les règles par l'opérateur max et le « ALORS » par l'opérateur min.
- La méthode d'inférence max-produit ou méthode de Larsen : réalise le « OU » reliant les règles par l'opérateur max et le « ALORS » par l'opérateur produit.
- La méthode d'inférence somme-produit ou méthode de Sugeno: réalise le « OU » reliant les règles par l'opérateur somme et le « ALORS » par l'opérateur produit.

Défuzzification : L'étape précédente produit une loi de commande exprimée en termes d'une fonction d'appartenance donc une loi de commande floue. Comme le système à asservir ne peut accepter qu'une commande définie et nette, cette fonction d'appartenance doit être traduite en une commande non floue, nette et précise. Ce processus est connu sous le nom de défuzzification [7].

3.4.2 CONCEPTION DU REGULATEUR FLOU

Le régulateur flou que nous concevons prend en entrée les valeurs de distances mesurées par les deux capteurs de distance HC-SR 04 et donne en sortie un angle pour le servomoteur commandant l'avant-bras du bras robotisé. Nous avons choisi un contrôleur de type Mamdani, dont la conclusion est un ensemble flou. Ce modèle présente l'avantage d'être facilement interprétable et bien adapté aux problèmes d'évitement d'obstacles [36]. La fonction d'appartenance utilisée est gaussienne.

Ci-dessous une illustration de la conception du régulateur sous Matlab.

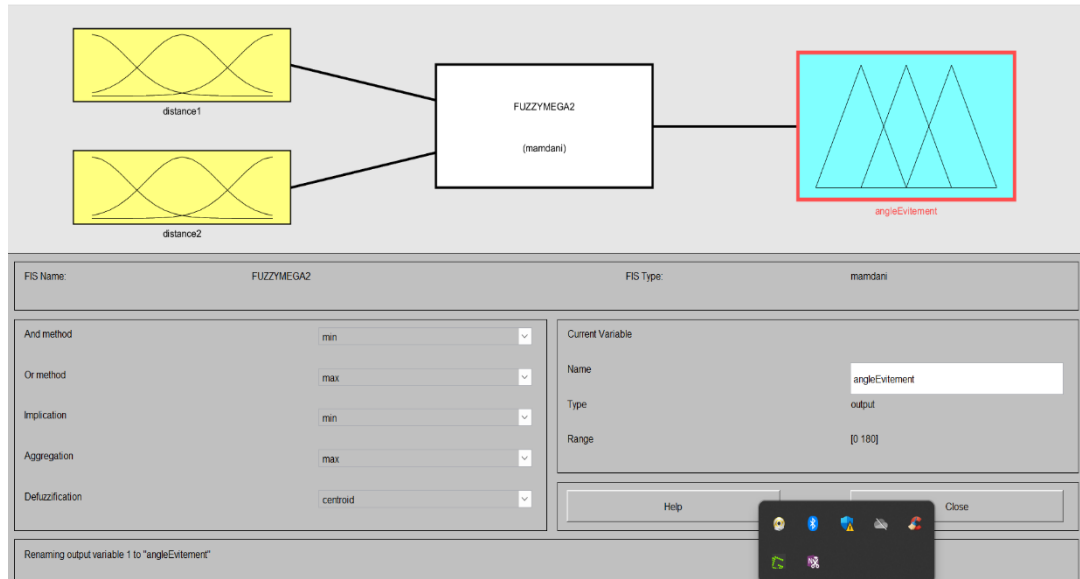


Figure 27 : le régulateur type Mamdani conçu avec Matlab.

Nous avons en entrée, distance1 et distance2, en sortie nous avons angleEvitement.

3.4.2.1 La fuzzification de la distance d'obstacle

On suppose que le bras robotisé peut percevoir un obstacle dans une distance comprise entre 0 et 30 cm. La fonction d'appartenance est représentée en 3 sous-ensembles flous, comme les figures ci-dessous l'indiquent :

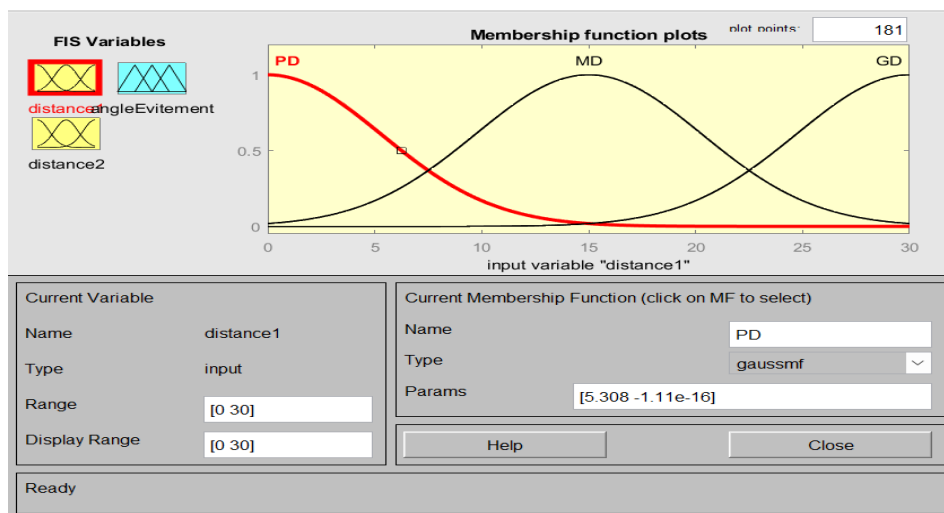


Figure 28 : illustrations des sous-ensembles flous de la variable distance1.

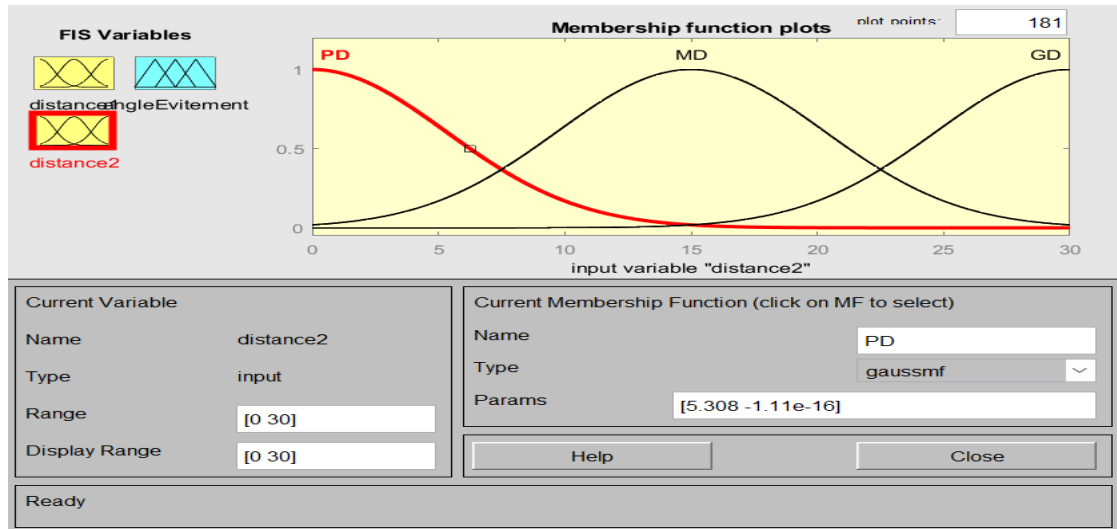


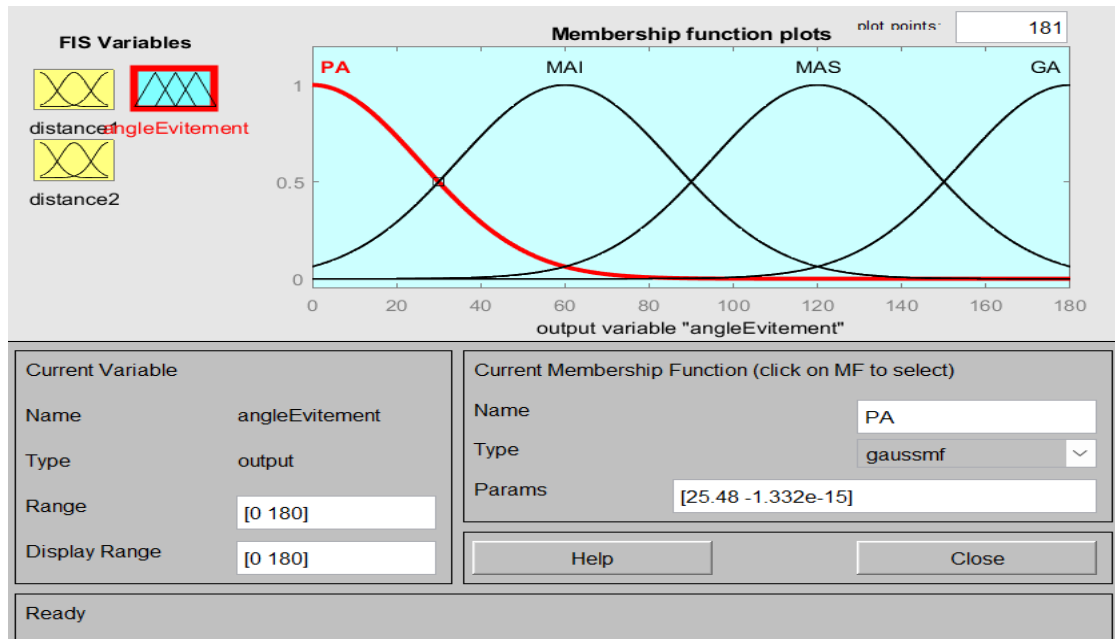
Figure 29 : illustrations des sous-ensembles flous de la variable distance1.

Avec :

- PD : petite distance
- MD : moyenne distance
- GD : grande distance.

3.4.2.2 La fuzzification de l'angle d'évitement

On suppose que le bras robotisé peut éviter un obstacle avec un angle compris entre 0 et 180 degrés. La fonction d'appartenance est représentée en 4 sous ensemble flous, comme les figures ci-dessous l'indiquent :



Avec :

- PA : petit angle
- MAI : moyen angle inferieur
- MAS : moyen angle superieur.
- GA : grand angle

3.4.2.3 La base de règles floues du régulateur.

Un régulateur flou prend généralement la forme d'une série de règles «Si-alors». L'inférence est basée sur des opérations min et max afin d'effectuer l'inférence des règles et l'opérateur max pour l'agrégation des règles.

Ces règles sont les suivantes :

- Si (distance1 est PD) alors (angleEvitement est MAS)
- Si (distance1 est MD) alors (angleEvitement est MAI)
- Si (distance1 est GD) alors (angleEvitement est PA)
- Si (distance2 est PD) alors (angleEvitement est MAS)
- Si (distance2 est MD) alors (angleEvitement est MAI)
- Si (distance2 est GD) alors (angleEvitement est PA)

La surface de décision floue de notre régulateur est donnée par la figure suivante :

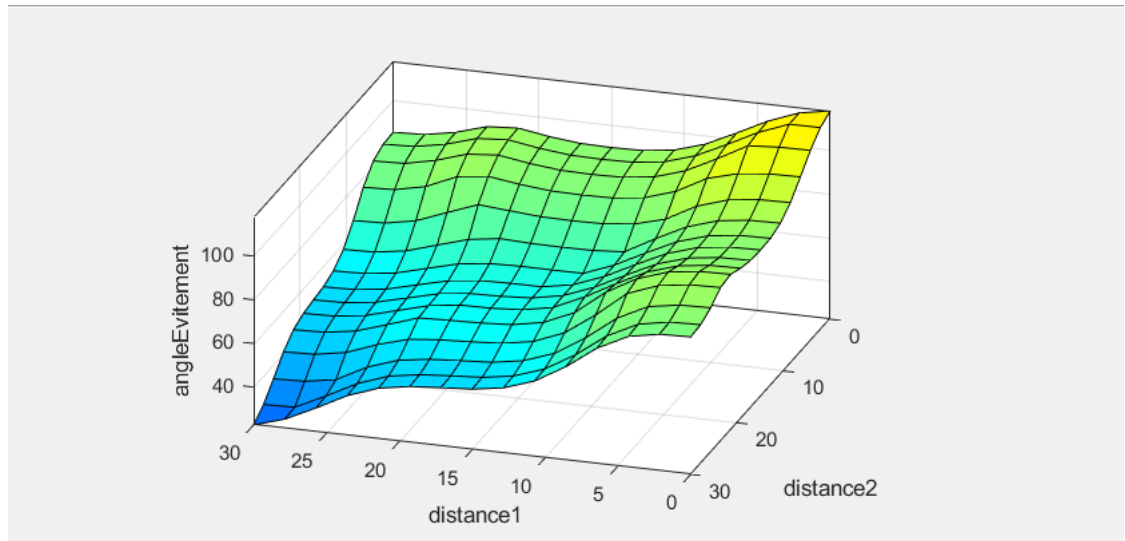


Figure 30 : surface de décision floue.

Un programme Arduino sera généré, au chapitre suivant, grâce à ce modèle conçu avec Matlab et sera implémenté dans notre algorithme pour la commande autonome du bras manipulateur.

3.5 CONCLUSION PARTIELLE

Ce chapitre était consacré à la conception du bras robotisé. Pour cela, les modèles conçus avec le logiciel 123Design ont été présentés pour chaque partie de la structure du bras robotisé. Nous avons aussi relevé les différents éléments électriques permettant un bon fonctionnement du bras robotisé en présentant leurs caractéristiques et modes de fonctionnement. Nous avons ensuite abordé les différents modes de commandes prévues pour le bras robotisé. Ces modes ont été présentés et leurs fonctionnements expliqués. Dans le cadre de ces différents modes de commande, nous avons conçu un régulateur flou et avons présenté le logigramme de l'application Android devant servir pour la commande par application mobile du bras robotisé.

Chapitre 4 : REALISATION DU BRAS ROBOTISÉ

4.1 INTRODUCTION

Dans ce chapitre nous abordons la réalisation du bras robotisé. Nous présentons ainsi la structure mécanique du bras robotisé et sa constitution électrique. Nous réalisons également l'application conçu. Nous réalisons aussi l'implémentation du régulateur flou ainsi que les modes de commande du bras robotisé.

4.2 STRUCTURE MECANIQUE DU BRAS ROBOTISÉ

La structure mécanique de notre bras robotisé est faite, comme nous l'avons vu, de la base, de l'avant-bras, du bras, de la main du bras avec ses doigts au bout. Ces éléments après avoir été modélisés ont été imprimé en plastique grâce aux imprimantes 3D Creativity Ender-3. Les images qui suivent illustrent ces parties avec les servomoteurs qui vont assurer la mobilité de ces articulations.



Figure 31 : la base du bras robotisé.



Figure 32 : *l'avant-bras du bras robotisé.*

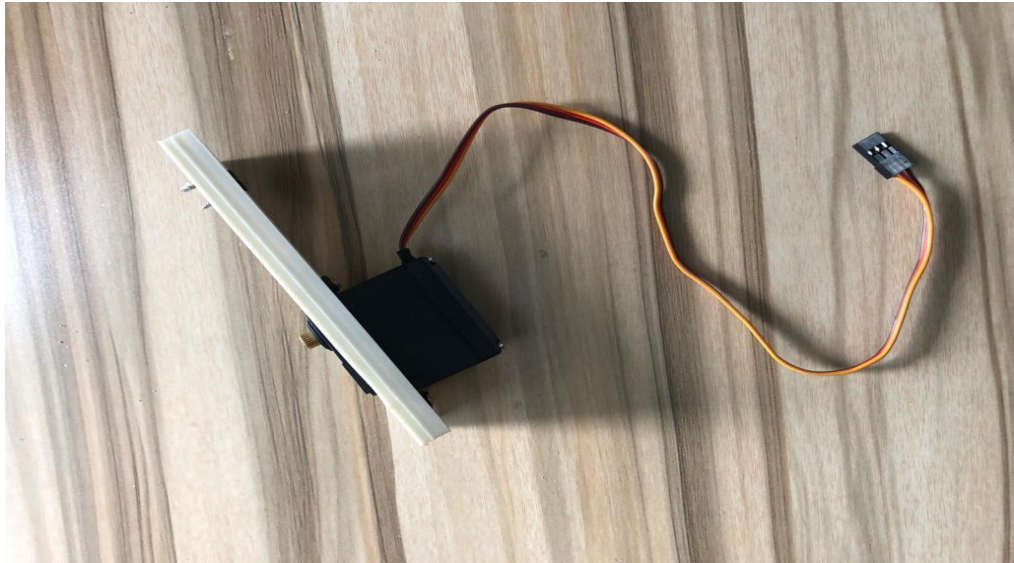


Figure 33 : la bras du bras robotisé.



Figure 34 : la main du bras robotisé.

La structure complètement assemblée du bras robotisé est illustrée sur la figure suivante :

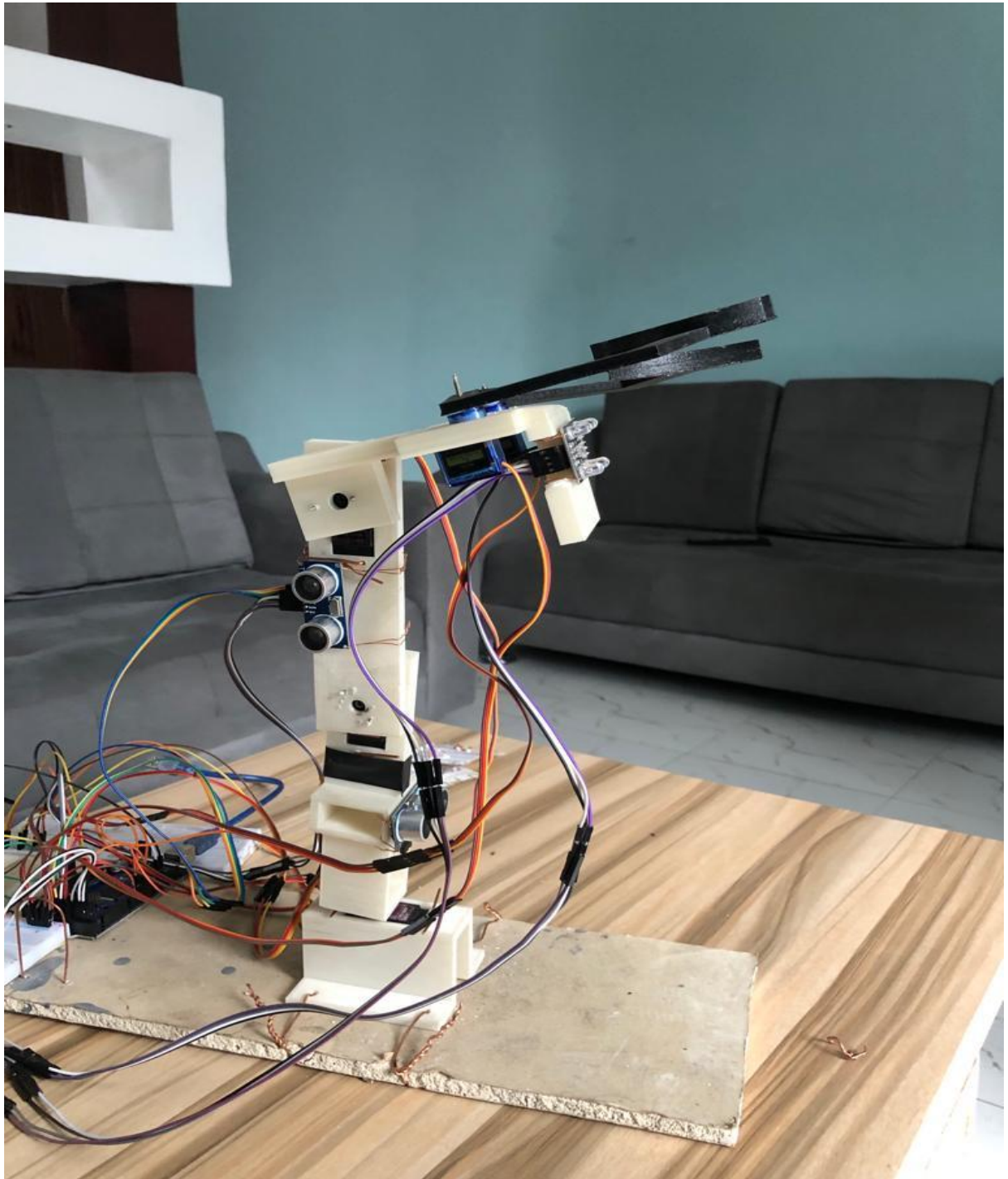


Figure 35 :la structure du bras robotisé complètement assemblée.

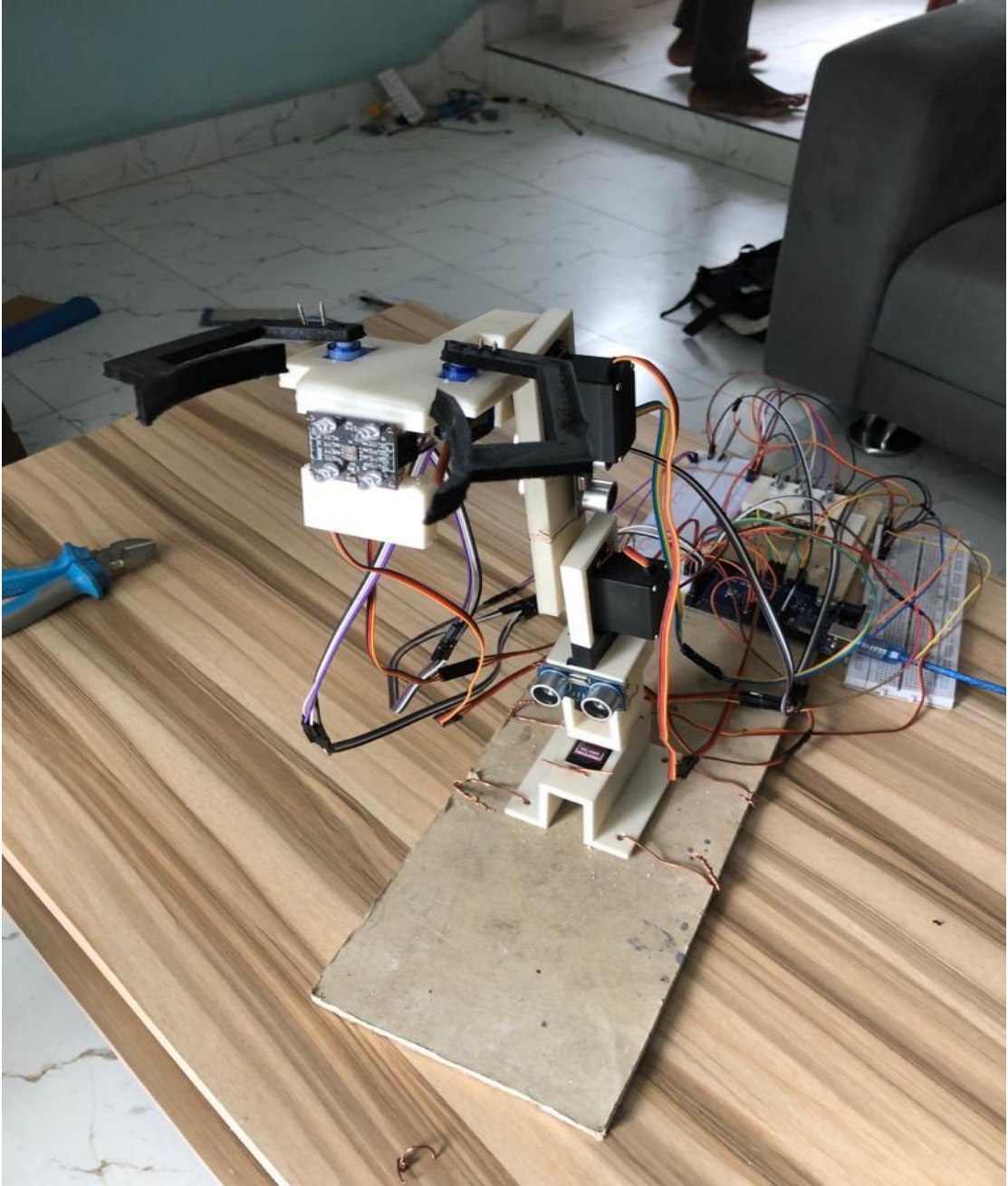


Figure 36 : la structure du bras robotisé complètement assemblée

4.3 MONTAGE ELECTRIQUE DU BRAS ROBOTISÉ

Le montage de tout le système du bras robotisé a été illustré sur l'image ci-dessous dont le lien du fichier source sera fournie en annexe.

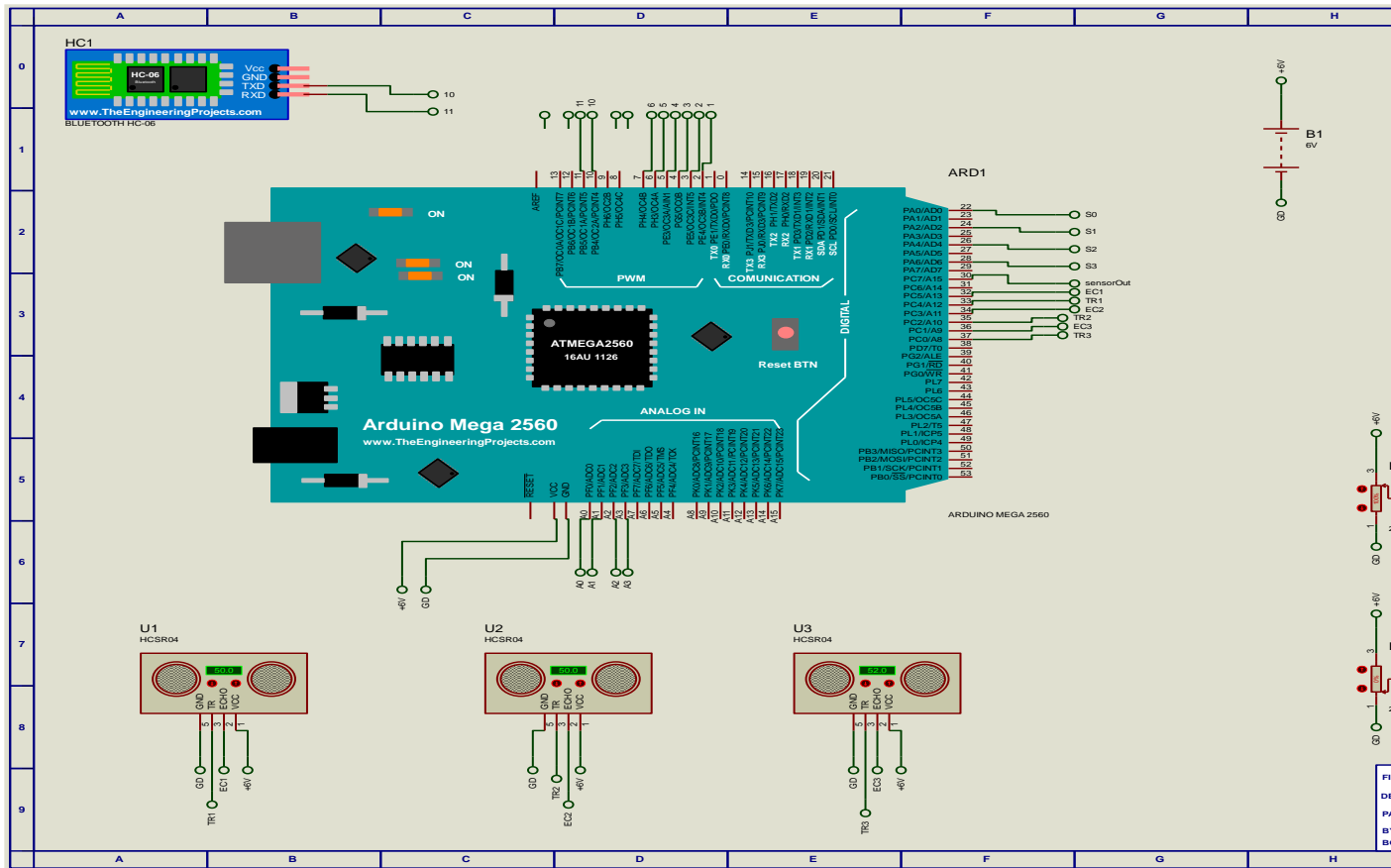
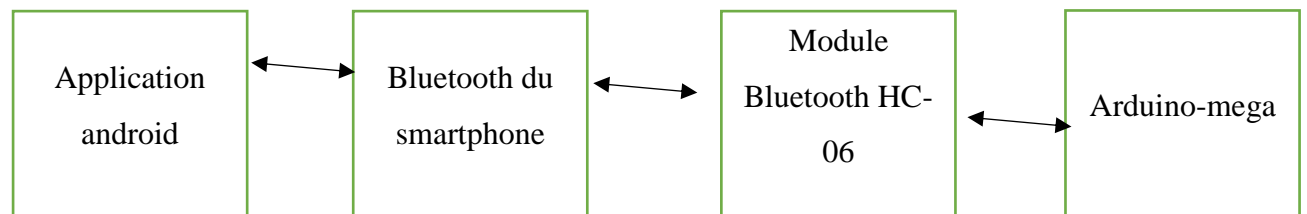


Figure 37 : image du logiciel Proteus du montage de tout le système.

4.4 APPLICATION MOBILE

En ayant recours à MIT App Inventor, nous avons développé une application personnalisée qui nous permet de commander notre robot. Cela implique que nous avons la capacité d'échanger des données entre une application Android et le robot. Notre application établit une connexion avec la carte Arduino du robot via la fonction Bluetooth de notre smartphone, en utilisant le module Bluetooth HC-06, comme décrit dans le schéma-bloc ci-dessous :



L'interface utilisateur de l'application mobile comme illustrée par la figure 53 suivante offre 3 options : la commande à distance (par application mobile), la commande automatique et la déconnexion du Bluetooth.

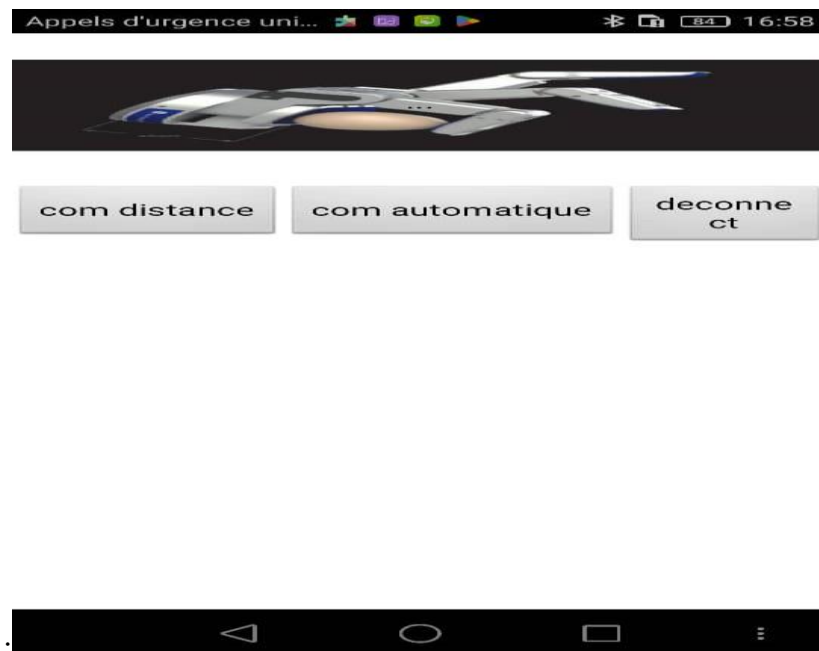


Figure 38 : image de l'interface de l'application

Comme illustrée par la figure 53 suivante, une fois que le choix de la commande à distance a été fait, nous avons les curseurs pour commander les articulations ainsi que les boutons ouverture et fermeture de l'effecteur :



Figure 39 : image de l'interface de l'application

Ci-dessous les images du programme de notre application dans MIT appinventor :

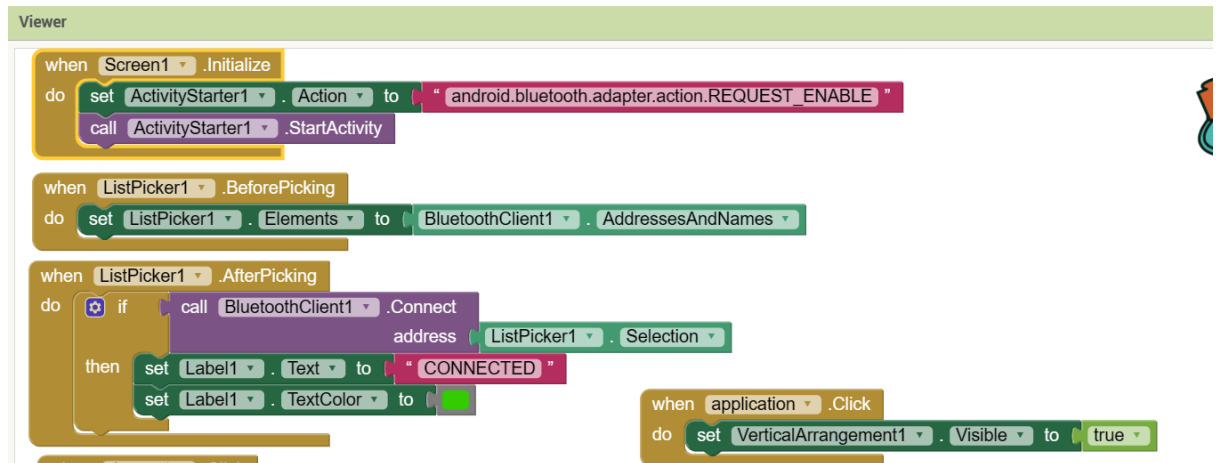


Figure 40 : le programme dans MIT appinventor

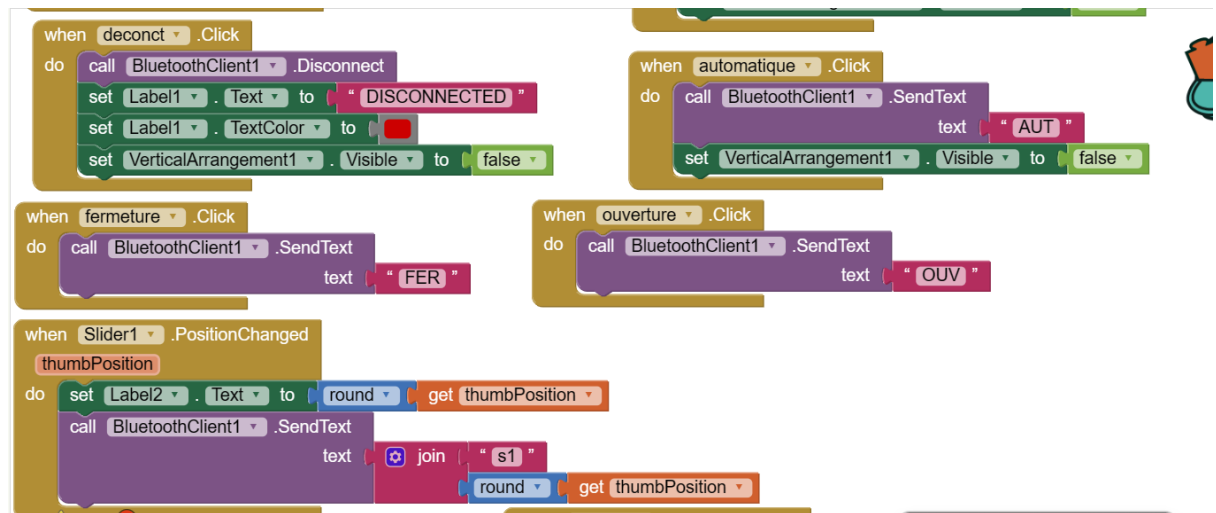
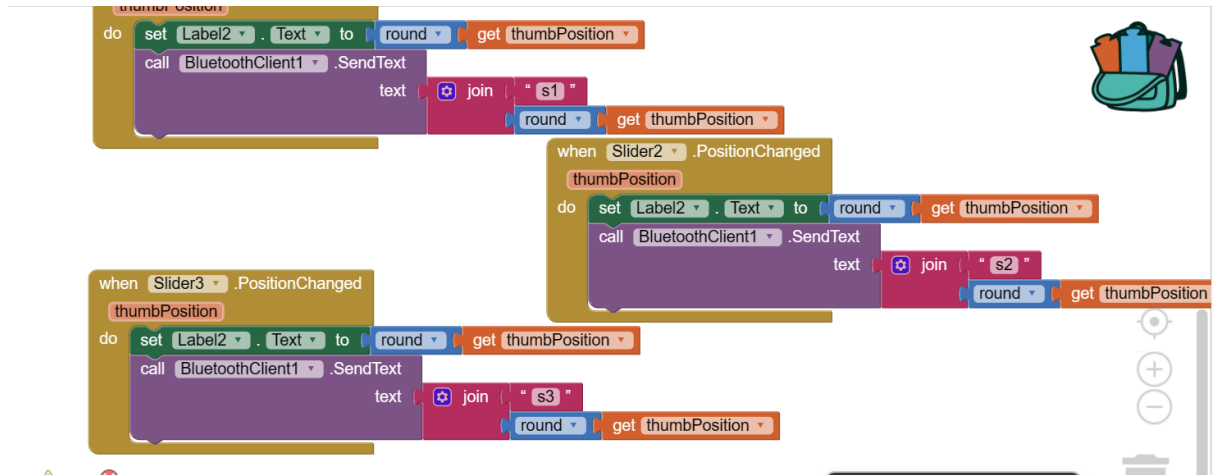


Figure 41 : le programme de l'application dans MIT appinventor



Ces figures montrent le programme qui permet à notre application d'accomplir le travail qu'on lui assigne.

Figure 42 : le programme de l'application dans MIT appinventor

4.5 COMMANDE DU BRAS ROBOTISÉ

Nous avons déjà vu que la commande manuelle est faite avec 4 potentiomètres, en raison d'un potentiomètre pour chaque articulation et un potentiomètre pour actionner les doigts du bras robotisé comme illustré sur la figure 57 suivante.

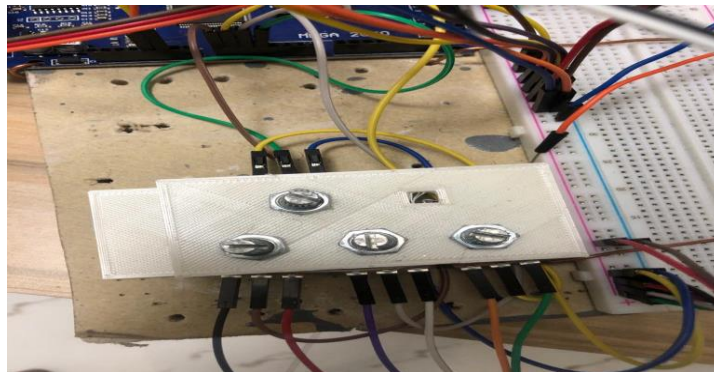


Figure 43 : les potentiomètres pour la commande manuelle

La commande automatique est basée sur un algorithme qui consiste à donner aux servomoteurs des angles précis en fonctions des données reçus de 3 capteurs de distances. Le régulateur flou

conçu au chapitre précédent a été intégré au programme grâce aux codes Arduino générés à partir des fichiers du logiciel Matlab de ce régulateur. Ce code a été adapté pour l'intégrer au programme principal.

4.6 CONCLUSION PARTIELLE

Dans ce chapitre, nous avons exposé la fabrication du bras robotisé en détaillant les composants que nous avons imprimés et en présentant l'assemblage complet de l'appareil. Ensuite, nous avons détaillé l'installation électrique du système, qui repose sur l'utilisation de l'Arduino Mega. De plus, nous avons décrit l'application que nous avons développée avec MIT App Inventor, en expliquant les méthodes de commande via des applications mobiles et en fournissant les codes correspondants. Enfin, nous avons exploré les modes de contrôle manuel et automatique.

CONCLUSION GENERALE

Ce travail portait sur la conception et la réalisation d'un bras robotisé a commande manuelle et par application mobile. Dans la réalisation de ce travail, nous nous sommes proposés de répondre aux questions suivantes :

1. Quelles sont les meilleures stratégies de conception pour intégrer un contrôle à la fois manuel et par application mobile dans un bras robotisé, en assurant une expérience utilisateur fluide et intuitive ?
2. Quelle technologie de communication et de commande optimale pourrait assurer une réactivité en temps réel entre le bras robotisé, la commande manuelle et l'application mobile ?
3. Comment optimiser la conception électromécanique, électronique et cinématique du bras robotisé pour garantir des mouvements précis et fluides en réponse aux commandes manuelle et de l'application mobile ?

Au premier chapitre, nous avons exposé les concepts fondamentaux relatifs aux robots, en détaillant leurs caractéristiques, leurs divers types et leurs classifications. Parmi la gamme des robots, nous nous sommes focalisés sur un type particulier qui a fait l'objet de notre étude dans ce travail : les bras manipulateurs. Nous avons également présenté les composants essentiels des bras robotiques et examiné différentes approches pour leur commande.

Au second chapitre, nous avons présenté les différentes manières de modéliser les bras robotiques. Nous avons présenté les bases mathématiques derrière chaque modélisation et abordé les différentes manières permettant d'implémenter ces modélisations. Nous avons ensuite donné le modèle géométrique direct et inverse de notre bras robotique anthropomorphe à 3 articulations. Nous avons aussi présenté son modèle cinématique. Ensuite nous avons montré la forme générale d'un modèle dynamique d'un bras robotique anthropographe à 3 articulations. Cette modélisation a été faite pour garantir que nous aurons des mouvements précis en réponse aux commandes manuelle et de l'application mobile et cela en vue de répondre à la problématique posée en sa troisième question.

Au troisième chapitre, nous avons abordé la constitution du bras robotisé en présentant les modèles conçus avec le logiciel 123Design pour chaque partie de la structure du bras robotisé. Nous avons aussi relevé les différents éléments électriques permettant un bon fonctionnement du bras robotisé en présentant leurs caractéristiques et modes de fonctionnement. Nous avons aussi abordé les différents modes de commandes prévues pour le bras robotisé. Ces modes ont été présentés et leurs fonctionnements expliqués. Dans le cadre de ces différents modes de commande, nous avons conçu un régulateur flou et avons présenté le logigramme de l'application Android devant servir pour la commande par application mobile du bras robotisé.

Au quatrième chapitre, nous avons abordé la réalisation du bras robotisé en présentant les parties du bras robotique que nous avons imprimée et en montrant l'ensemble du dispositif déjà monté. Ensuite nous avons présenté le montage électrique du système qui est basé sur l'arduino-mega. Nous avons ensuite présenté l'application conçue avec MIT appinventor et les modalités de commande par applications mobile ainsi que les codes.

La réalisation du dispositif du bras robotisé combinant une commande manuelle et par application mobile est fait d'une manière à assurer un fonctionnement indépendant de ces deux modes de commande en ce sens que la commande manuelle n'est possible que dans le cas où le connexion Bluetooth n'est pas établie. Et la commande par application mobile se fait grâce à la connexion Bluetooth, cela répondant à la réponse à la problématique posée en sa première question.

Concernant le choix de technologie à employer pour pouvoir s'assurer d'une transmission en temps réel de la commande, comme posée dans la problématique en sa deuxième question, nous avons opté pour la technologie Bluetooth à cause de la modestie de son implémentation. Le bras manipulateur nécessitant une vision de l'environnement de travail par l'opérateur, la technologie Bluetooth est adaptée en tant que technologie de proximité.

Dans la réalisation de ce travail, nous avons eu à être confronter principalement aux défis de disponibilité de composants électriques. C'est dans ce sens qu'il a été difficile pour nous de trouver de servomoteurs capables de supporter le poids de charges que normalement notre bras devrait soulever. Ainsi le dispositif que nous présentons ne pourra pas être testé en charge car déjà avec le poids de sa structures certains servomoteurs sont déjà à la limite de la surcharge.

Comme défis, nous avons également le capteur de couleur qui avant toute utilisation doit être étalonné pour l'adapter aux conditions de luminosité de la pièce dans laquelle elle est placée. Cela est un défi car l'algorithme doit être mis à jour dès que les conditions de luminosité de la pièce changent.

Comme perspective, nous recommandons premièrement l'utilisation de servomoteurs avec un grand couple pour pouvoir soulever de poids plus importants. Nous recommandons également l'utilisation du modèle dynamique du bras robotisé pour une commande plus robuste, comme la commande PID, car tenant compte des paramètres dynamiques du bras.

Bibliographie

- [1] P. Ughetto, «Automatisation et robotisation : Quels effets sur le travail et les compétences dans la logistique ?,» HAL, Marne-la-Vallée Cedex, 2018.
- [2] CIM GmbH, «Automatisation de la logistique de stock ,Facteurs de succès, potentiels et avantages des systèmes de stockage automatique,» CIM GmbH, Livry-Gargan-Straße, 2018.
- [3] KOBRINSKI, Bras manipulateurs de Robots, Architecture et theories, Moscou: Mir, 1989.
- [4] V. Govaere, «La preparation de commande en logistique, Mutations technologiques et risques professionnels,» *INRS departement homme au travail*, p. 12, 2009.
- [5] A. Lazinica, Mobile Robots Towards New Applications, Mammendorf : pro literatur Verlag, 2006.
- [6] B. MUSHAGE, Cours de regulation et systemes asservis, Goma: ULPGL, 2022.
- [7] A. NKIEDIEL, Cours d'intelligence artificielle, GOMA: ULPGL, 2023.
- [8] M. LAROUSSE, Larousse illustree, PARIS: MAISON LAROUSSE, 2009.
- [9] u. d. Bouira, Cours de robotique, BOUIRA: Universite de Bouira.
- [10] J.-L. Boimond, La Robotique, Angers: ISTIA, Université Angers.
- [11] S. H. V. MARK W. SPONG, robot modeling and control second edition, John Wiley & Sons, Ltd, 2020.
- [12] F. MORBIDI, Initiation a la robotique, Université de Picardie Jules Verne, 2021.
- [13] A. Ghoul, Modelisation des robots manipulateurs, universite Tahri Mohammed Bechar, 19 fevrier 2016.
- [14] F. BOUAKRIF, Cours de Robotique, Jijel: Université Mohamed Seddik Benyahia-Jijel, 2020.

- [15] H. FRIKHA, commande PID et par logique floue pour l'amélioration de la finesse et de la précision des mouvements d'un bras robotisé à plusieurs axes et degrés de liberté, UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES, 2021.
- [16] B. Abderrahim, Cours de Robotique, UNIVERSITE KASDI MERBAH OUARGLA, 2015.
- [17] M. M. Ayoub, Développement d'un contrôleur multi-agents pour les robots manipulateurs mobiles : Cas du robot RobuTER/ULM, Blida: UNIVERSITÉ DE BLIDA -1-, 2014.
- [18] M. E. H. DAACHI, Cours de Robotique, Université Mohamed El Bachir El Ibrahimy Bordj Bou Arreridj, 2014.
- [19] I. Herrera-Aguilar, Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service, HAL, 2007.
- [20] B. G. e. M. Renaud, Modèles des Robots Manipulateurs application à leur commande., Cepadues Editions., 1984.
- [21] D. E. Whitney, Resolved motion rate control of manipulators and human prostheses, IEEE Trans. Man-Machine, 1969.
- [22] M. J. Er, Recent developments and futuristic trends in robot manipulator control., Asia-Pacific Workshop on Advances in Motion Control Proceedings., 1993.
- [23] B. S. e. G. B.] C. Canudas, Theory of Robot Control., Springer, 1997.
- [24] L. S. e. B. Siciliano, Modelling and Control of Robot Manipulators, Springer, 2000.
- [25] J. J. E. S. e. W. Li, Adaptive strategies in constrained manipulation, . IEEE Int. Conf. on Robotics and Automation, volume Raleigh, N.C, 1987.
- [26] R. T. P. B. e. D. D. R. Hedjar, Feedback nonlinear predictive control of rigid link robot manipulators, Dans Proceedings of the American Control Conference, 2002.
- [27] J. J. E. Slotine, Robust control of robot manipulators., The International Journal of Robotics Research, 1985.

- [28] K. S. e. S. Tadakuma, Minimum jerk control of power assisting robot based on human arm behavior characteristic, International Conference on Systems, Man and Cybernetics, 2004.
- [29] O. B. e. O. Khatib, Real-time path modification for mobile manipulation, hayama: The eight International Symposium of Robotics Research, 1997.
- [30] B. Mohammed, Conception et analyse d'un robot manipulateur 3R, MEDEA: UNIVERSITE YAHIA FARES DE MEDEA, 2020.
- [31] E. BARTMANN, Le Grand livre d'Arduino, EYROLLES, 2015.
- [32] C. FAYROUZ, Le systeme servomoteur, 2020.
- [33] a. france, «manuel d'utilisation du capteur de couleur TCS 3200».
- [34] S. SIG, Debuter avec MIT appinventor, 2014.
- [35] B. H. E. C. Rezqi, Commande par logique floue d'un moteur a courant continu, Université de M'sila.
- [36] M. M. & O. K. HATEM GHODBANE, «commande d'un bras manipulateur a 6 d.d.l avec evitement d'obstacles par la logique floue,» *COURIER DU SAVOIR*, n° %110, pp. 15-22, 20210.
- [37] JUnit.org. [En ligne]. Available: <http://www.junit.org>. [Accès le 2 Janvier 2019].
- [38] B. M. Said, Robotique Industrielle, Bejaia: Université Abderrahmane Mira - Bejaia, 2019.
- [39] C. MARTINEZ, L'homme et le robot: une collaboration est elle possible?, geneve: Hesso, 2018.
- [40] S. S. S. L. Richard M. Murray, A Mathematical Introduction to Robotic Manipulation, , CRC Press, 1994.
- [41] J. J. Craig, Introduction to Robotics Mechanics and Control, Pearson Education International, 2005.
- [42] R. n. i. p. l. r. Enjeux, ROBIN RIVATON, FONDAPOL, 2012.

ANNEXE1: Code arduino du programme

```
#include "fis_header.h"
#include <Ultrasonic.h>
#define Echo1 32
#define Trig1 33
#define Echo2 34
#define Trig2 35
#define Echo3 36
#define Trig3 37
Ultrasonic ultrasonic1(Trig1, Echo1);
Ultrasonic ultrasonic2(Trig2, Echo2);
Ultrasonic ultrasonic3(Trig3, Echo3);
#include <SoftwareSerial.h>
#include <Servo.h>
Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;
Servo myservo5;
int bluetoothTx = 10;
int bluetoothRx = 11;
String dataIn = "";
String data = "MAN";
int pos;
int potpin1 =0;
int potpin2 =1;
int potpin3 =2;
int potpin4 =3;
int val1;
int val2;
int val3;
int val4;
int man=200;
```

```

int distanceMax= 25;
int distanceMin= 2;

long dist1;
long dist2;
long dist3;
// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 1;
// Number of rules to the fuzzy inference system
const int fis_gcR = 6;

FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];

#define S0 22
#define S1 24
#define S2 26
#define S3 28
#define sensorOut 30

// Stores frequency read by the photodiodes
int red = 0;
int green = 0;
int blue = 0;
String couleur=" none";
void Color() {
    // Setting RED (R) filtered photodiodes to be read
    digitalWrite(S2,LOW);
    digitalWrite(S3,LOW);
    // Reading the output frequency
    red = pulseIn(sensorOut, LOW);
    delay(100);

```

```
// Setting GREEN (G) filtered photodiodes to be read
digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
// Reading the output frequency
green = pulseIn(sensorOut, LOW);
delay(100);

// Setting BLUE (B) filtered photodiodes
//to be read
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
// Reading the output frequency
blue = pulseIn(sensorOut, LOW);
delay(100);
pinMode(sensorOut, INPUT);

}

SoftwareSerial bluetooth(blueetoothTx,bluetoothRx);
void setup() {
  // put your setup code here, to run once:
myservo1.attach(2);
myservo2.attach(3);
myservo3.attach(4);
myservo4.attach(5);
myservo5.attach(6);
pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);
pinMode(Trig1, OUTPUT);
pinMode(Trig2, OUTPUT);
pinMode(Trig3, OUTPUT);
```

```
pinMode(Echo1, INPUT);
pinMode(Echo2, INPUT);
pinMode(Echo3, INPUT);
Serial.begin(9600);
bluetooth.begin(9600);
bluetooth.setTimeout(100);
  delay(20);
  myservo1.write(90);
  myservo2.write(60);
  myservo3.write(60);
}

void loop() {
// data = "MAN";
  if(data=="MAN")
  {
    //Serial.println(dataIn);
    int pos;
    int potpin1 =0;
    int potpin2 =1;
    int potpin3 =2;
    int potpin4 =3;
    int val1;
    int val2;
    int val3;
    int val4;
    val1=analogRead(potpin1);
    val1=map(val1,0,1023,0,180);
    myservo1.write(val1);
    delay(15);
    val2=analogRead(potpin2);
    val2=map(val2,0,1023,0,180);
    myservo2.write(val2);
    delay(15);
```

```
val3=analogRead(potpin3);
val3=map(val3,0,1023,0,180);
myservo3.write(val3);
delay(15);
```

```
val4=analogRead(potpin4);
val4=map(val4,0,1023,0,180);
if(val4>90)
{
myservo4.attach(5);
myservo5.attach(6);
myservo4.write(85);
myservo5.write(5);
delay(50);
myservo4.detach();
myservo5.detach();
}
else
{
myservo4.attach(5);
myservo5.attach(6);
myservo4.write(0);
myservo5.write(90);
delay(50);
myservo4.detach();
myservo5.detach();
}
}
```

```
// put your main code here, to run repeatedly:
if (bluetooth.available() > 0) {
  data = "OP";
  //myservo1.write(60);
  //myservo2.write(60);
  //myservo3.write(60);
```

```

dataIn = bluetooth.readString(); // Read the data as string

if(dataIn.startsWith("AUT"))
{
int dist1 =ultrasonic1.Ranging(CM);
int dist2 =ultrasonic2.Ranging(CM);
int dist3 =ultrasonic3.Ranging(CM);
Serial.println(dist2);

if (dist1<=distanceMax && dist1>distanceMin)
{

myservo3.write(40);
delay(1000);
myservo2.write(90);
// delay(1000);
//servo2.detach();
// Read Input: input1

Serial.println("Color detected: ");
Color();

if(red<blue && red<green && red<120&& red>52){
couleur="rouge";
Serial.println(couleur);
delay(1000);
}
else if(blue < blue < red && blue < green && blue<190 && blue>90) {
couleur="bleu";
Serial.println(couleur);
delay(1000);
}
}

```

```
}

else {
couleur="none",
  Serial.println(couleur);

  delay(5000);
}
if(couleur.startsWith("bleu"))
{
  if(dist2<40)
  {
    g_fisInput[0] = dist2;
    // Read Input: input2
    // g_fisInput[1] = distance1;

    g_fisOutput[0] = 0;

//servo2.attach(2);
    fis_evaluate();
    Serial.println(g_fisOutput[0]);
myservo2.write(g_fisOutput[0]);
  }
  delay(1000);
myservo1.write(180);
  delay(1000);
}
if(couleur.startsWith("rou"))
{
  if(dist2<40)
  {
    g_fisInput[1] = dist3;
    // Read Input: input2
    // g_fisInput[1] = distance1;
```

```

    g_fisOutput[0] = 0;

    //servo2.attach(2);
    fis_evaluate();
    //Serial.println(g_fisOutput[0]);
    myservo2.write(g_fisOutput[0]);
    }
    delay(1000);
    myservo1.write(0);
    delay(1000);
    }

}
else
{
    myservo2.write(180);
    delay(1000);
    myservo3.write(40);
    }

}

// If "Waist" slider has changed value - Move Servo 1 to position
if (dataIn.startsWith("s1")) {
    String dataInS = dataIn.substring(2, dataIn.length()); // Extract only the number. E.g. from "s1120" to "120"
    pos = dataInS.toInt();
    Serial.println(pos);
    myservo1.write(pos);
    delay(20);
    }

if (dataIn.startsWith("s2")) {
    String dataInS = dataIn.substring(2, dataIn.length()); // Extract only the number. E.g. from "s1120" to "120"
    pos = dataInS.toInt();

```

```
    Serial.println(pos);
myservo2.write(pos);
delay(20);
}

if (dataIn.startsWith("s3")) {
    String dataInS = dataIn.substring(2, dataIn.length()); // Extract only the number. E.g. from "s1120" to "120"
    pos = dataInS.toInt();
    Serial.println(pos);
    myservo3.write(pos);
    delay(20);
}

if(dataIn.startsWith("FER"))
{
myservo4.attach(5);
myservo5.attach(6);
myservo4.write(85);
myservo5.write(5);
delay(50);
myservo4.detach();
myservo5.detach();
}
if(dataIn.startsWith("OUV"))
{
myservo4.attach(5);
myservo5.attach(6);
myservo4.write(0);
myservo5.write(90);
delay(50);
myservo4.detach();
myservo5.detach();
}
}
```

```

}

//*****
// Support functions for Fuzzy Inference System
//*****

// Gaussian Member Function
FIS_TYPE fis_gaussmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE s = p[0], c = p[1];
    FIS_TYPE t = (x - c) / s;
    return exp(-(t * t) / 2);
}

FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return min(a, b);
}

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size, _FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;

    if (size == 0) return ret;
    if (size == 1) return array[0];

    ret = array[0];
    for (i = 1; i < size; i++)
    {
        ret = (*pfnOp)(ret, array[i]);
    }
}

```

```

    }

    return ret;
}

/*****
// Data for Fuzzy Inference System
/*****
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =
{
    fis_gaussmf
};

// Count of member function for each Input
int fis_gIMFCount[] = { 3, 3 };

// Count of member function for each Output
int fis_gOMFCount[] = { 4 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { 5.308, -1.11e-16 };
FIS_TYPE fis_gMFI0Coeff2[] = { 5.308, 15 };
FIS_TYPE fis_gMFI0Coeff3[] = { 5.308, 30 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2, fis_gMFI0Coeff3 };
FIS_TYPE fis_gMFI1Coeff1[] = { 5.308, -1.11e-16 };
FIS_TYPE fis_gMFI1Coeff2[] = { 5.308, 15 };
FIS_TYPE fis_gMFI1Coeff3[] = { 5.308, 30 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2, fis_gMFI1Coeff3 };
FIS_TYPE** fis_gMFI0Coeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff };

// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { 25.48, -1.332e-15 };
FIS_TYPE fis_gMFO0Coeff2[] = { 25.48, 60 };

```

```

FIS_TYPE fis_gMFO0Coeff3[] = { 25.48, 120 };
FIS_TYPE fis_gMFO0Coeff4[] = { 25.48, 180 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2, fis_gMFO0Coeff3,
fis_gMFO0Coeff4 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff };

// Input membership function set
int fis_gMFI0[] = { 0, 0, 0 };
int fis_gMFI1[] = { 0, 0, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1 };

// Output membership function set
int fis_gMFO0[] = { 0, 0, 0, 0 };
int* fis_gMFO[] = { fis_gMFO0 };

// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1 };

// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1 };

// Rule Inputs
int fis_gRI0[] = { 1, 0 };
int fis_gRI1[] = { 2, 0 };
int fis_gRI2[] = { 3, 0 };
int fis_gRI3[] = { 0, 1 };
int fis_gRI4[] = { 0, 2 };
int fis_gRI5[] = { 0, 3 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4, fis_gRI5 };

// Rule Outputs
int fis_gRO0[] = { 3 };
int fis_gRO1[] = { 2 };
int fis_gRO2[] = { 1 };
int fis_gRO3[] = { 3 };

```

```

int fis_gRO4[] = { 2 };
int fis_gRO5[] = { 1 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5 };

// Input range Min
FIS_TYPE fis_gIMin[] = { 0, 0 };

// Input range Max
FIS_TYPE fis_gIMax[] = { 30, 30 };

// Output range Min
FIS_TYPE fis_gOMin[] = { 0 };

// Output range Max
FIS_TYPE fis_gOMax[] = { 180 };

//*****
// Data dependent support functions for Fuzzy Inference System
//*****
FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x, int o)
{
    FIS_TYPE mfOut;
    int r;

    for (r = 0; r < fis_gcR; ++r)
    {
        int index = fis_gRO[r][o];
        if (index > 0)
        {
            index = index - 1;
            mfOut = (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
        else if (index < 0)
        {
            index = -index - 1;

```

```

        mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
    }
    else
    {
        mfOut = 0;
    }

    fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);
}
return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);
}

FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet, int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) / (FIS_RESOLUTION - 1);
    FIS_TYPE area = 0;
    FIS_TYPE momentum = 0;
    FIS_TYPE dist, slice;
    int i;

    // calculate the area under the curve formed by the MF outputs
    for (i = 0; i < FIS_RESOLUTION; ++i){
        dist = fis_gOMin[o] + (step * i);
        slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
        area += slice;
        momentum += slice*dist;
    }

    return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) : (momentum / area));
}

//*****
// Fuzzy Inference System
//*****

void fis_evaluate()

```

```

{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, };
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;

    // Transforming input to fuzzy Input
    int i, j, r, o;
    for (i = 0; i < fis_gcI; ++i)
    {
        for (j = 0; j < fis_gIMFCount[i]; ++j)
        {
            fuzzyInput[i][j] =
                (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i], fis_gMFICoeff[i][j]);
        }
    }

    int index = 0;
    for (r = 0; r < fis_gcR; ++r)
    {
        if (fis_gRType[r] == 1)
        {
            fuzzyFires[r] = FIS_MAX;
            for (i = 0; i < fis_gcI; ++i)
            {
                index = fis_gRI[r][i];
                if (index > 0)
                    fuzzyFires[r] = fis_min(fuzzyFires[r], fuzzyInput[i][index - 1]);
                else if (index < 0)
                    fuzzyFires[r] = fis_min(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
            }
        }
    }
}

```

```

        else
            fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
    else
    {
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
        }
    }

    fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
    sW += fuzzyFires[r];
}

if (sW == 0)
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
    }
}
else
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet, o);
    }
}

```

}
 }
 }

ANNEXE2: COUT DES MATERIAUX.

NOM	QUANTITE	PRIX UNITAIRE	PRIX TOTAL
ARDUINO MEGA	1	30\$	30\$
TCS3200	1	25\$	25\$
MG996R	3	15\$	45\$
SG90	2	9\$	18\$
HC 06	1	8\$	8\$
HC SR04	3	5\$	15\$
BATTERIE 9V	1	1\$	1\$
POTENTIOMETRES	4	1\$	4\$
JUMPERS	100PCS	0.1\$	10\$
TOTAL			156\$