

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
UNIVERSITE AKLI MOAND OULHADJ-BOUIRA



Faculté des Sciences et Sciences Appliquées  
Département Génie Électrique

**Mémoire de fin d'étude**

Présenté par

**GACEM HADJIRA**

**KERNANE MESSAOUDA**

**En vue de l'obtention du diplôme de Master en :**

Filière : Génie Électrique

Option : Électronique Des Systèmes Embarqués

***Thème :***

**Implémentation De La Commande Intelligente Dans Un Robot à Deux  
Degrés De Liberté**

**Devant le jury composé de :**

ISSAOUNI Salim	Grade	UAMOB	Président
YAHIOU Abd El Ghani	Grade	UAMOB	Encadreur
ZARIFI Adel	Grade	UAMOB	Co Encadreur
BEN SAFIA Yassine	Grade	UAMOB	Examineur

Année Universitaire 2019/2020

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Table des figures</b>	<b>iv</b>
<b>Liste des tableaux</b>	<b>vii</b>
<b>Introduction Général</b>	<b>1</b>
<b>1 généralités sur les robots</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Historique . . . . .	3
1.3 Définition . . . . .	4
1.3.1 Définition de Robot . . . . .	4
1.3.2 Définition de Robotique . . . . .	4
1.4 Type de robot . . . . .	4
1.4.1 Robot manipulateur . . . . .	4
1.4.2 Robot mobile . . . . .	5
1.5 Robot industriel . . . . .	5
1.6 Architecture des robots . . . . .	6
1.7 Degré de liberté (DDL) . . . . .	8
1.8 Structures des robots . . . . .	8
1.8.1 Structure à chaîne cinématique ouverte (structure sérielle) . . . . .	8
1.8.2 Structure à chaîne cinématique fermé (structure parallèle ) . . . . .	9
1.8.3 Structure hybride . . . . .	9
1.9 Classification des robots industriels . . . . .	9
1.9.1 Classification fonctionnel . . . . .	10
1.9.2 Classification géométrique . . . . .	12
1.10 Domaine d'application des Robots . . . . .	14
1.11 Avantages et inconvénients du robot . . . . .	15

1.11.1	Avantages . . . . .	15
1.11.2	Inconvénients . . . . .	15
1.12	Conclusion . . . . .	16
<b>2</b>	<b>Modélisation d'un robot anthropomorphe à deux degré de liberté</b>	<b>17</b>
2.1	Introduction . . . . .	17
2.2	Les notations et les outils mathématiques mis en œuvre pour la modélisation des robots . . . . .	18
2.2.1	Représentation d'un point . . . . .	18
2.2.2	Représentation d'un vecteur . . . . .	18
2.2.3	Représentation d'un plan . . . . .	18
2.2.4	Transformation homogène . . . . .	19
2.3	Modélisation d'un robot manipulateur anthropomorphe . . . . .	22
2.3.1	Présentation du robot manipulateur utilisé . . . . .	22
2.3.2	Modèle géométrique . . . . .	22
2.3.3	Modèle cinématique . . . . .	24
2.3.4	Le modèle dynamique . . . . .	26
2.4	Conclusion . . . . .	28
<b>3</b>	<b>Étude théorique de la commande intelligente ( la logique floue)</b>	<b>29</b>
3.1	Introduction . . . . .	29
3.2	Le but d'utiliser la logique floue . . . . .	29
3.3	La logique classique et la logique floue . . . . .	30
3.4	La commande floue . . . . .	30
3.5	Historique . . . . .	31
3.6	Les ensembles flous . . . . .	31
3.6.1	variable linguistique . . . . .	33
3.6.2	Les Différentes formes et ces fonctions d'appartenances flous . . . . .	33
3.6.3	Les opérateurs de la logique floue . . . . .	34
3.7	Structure d'un contrôleur flou . . . . .	36
3.7.1	Fuzzification . . . . .	36
3.7.2	l'inférence floue . . . . .	37
3.7.3	Base de règles . . . . .	38
3.7.4	défuzzification . . . . .	38
3.8	Domaines d'application . . . . .	40
3.9	Les avantages et les inconvénients . . . . .	40

---

3.10 conclusion . . . . .	41
<b>4 simulation d'un robot anthropomorphe à deux degrés de liberté</b>	<b>42</b>
4.1 introduction . . . . .	42
4.2 la géométrie inverse . . . . .	43
4.3 Présentation de la solution floue . . . . .	43
4.3.1 Définition d'ANFIS . . . . .	44
4.3.2 Data génération . . . . .	44
4.3.3 Construire des réseaux ANFIS . . . . .	46
4.3.4 Validation des réseaux ANFIS . . . . .	47
4.3.5 Construire une solution autour des réseaux ANFIS formés . . . . .	48
4.4 conclusion . . . . .	49
<b>5 implémentation de la logique floue</b>	<b>50</b>
5.1 Introduction . . . . .	50
5.2 Partie matériels( Partie Hard) . . . . .	50
5.2.1 présentation générale de l'Arduino . . . . .	50
5.2.2 servomoteur . . . . .	53
5.2.3 cabale USB . . . . .	54
5.3 partie logiciel (partie soft ) . . . . .	54
5.3.1 Matlab support package for Arduino Hardware . . . . .	54
5.3.2 l'installation de MATLAB Support Package for Arduino Hardware . . . . .	55
5.4 Transformation du programme vers la carte Arduino . . . . .	56
5.5 L'implémentation de la logique floue dans notre robot . . . . .	56
5.5.1 concept d'un contrôleur flou . . . . .	56
5.5.2 les étapes de commande floue . . . . .	58
5.5.3 L'inférence Mamdani . . . . .	61
5.6 conclusion . . . . .	66
<b>Conclusion Général</b>	<b>67</b>
<b>Bibliographie</b>	<b>69</b>

# Table des figures

1.1	Robot manipulateur	5
1.2	Robot Mobile	5
1.3	architecture d'un robot	6
1.4	représentation d'une articulation rotoïde.	7
1.5	représentation d'une articulation prismatique.	7
1.6	Constituent d'un robot	8
1.7	Structure série d'un robot.	8
1.8	Structure parallèle d'un robot.	9
1.9	Structure hybride d'un robot.	9
1.10	Manipulateur à commande manuelle.	10
1.11	Manipulateur automatique à cycle pré-réglé.	11
1.12	Robot chimiste intelligent.	11
1.13	Robot programmable.	12
1.14	Structure d'un robot cartésien.	12
1.15	Structure d'un robot cylindrique.	13
1.16	Structure d'un robot sphérique.	13
1.17	Structure d'un robot SCARA.	13
1.18	Structure d'un robot Anthropomorphe	14
1.19	Les domaines d'application de robot.	14
1.20	stocks opérationnel des robots industriels par principales industries dans le monde en fin d'année 2012-2013.(14)	15
2.1	transformation d'un repère	19
2.2	transformation d'une translation pure	20
2.3	la transformation de rotation suivant les axe x,y et z	21
2.4	Robot anthropomorphe à 2 degrés de liberté.	22
2.5	modèle géométrique direct et inverse.	22
2.6	Résolution des angles d'articulation d'un robot anthropomorphe à 2 ddl.	24

3.1	classification de l'âge selon les ensembles flous et classique . . . . .	32
3.2	Les Différentes formes pour les fonctions d'appartenances . . . . .	33
3.3	Opérateur « ET ». . . . .	35
3.4	Opérateur « ou » . . . . .	35
3.5	complémentation . . . . .	36
3.6	structure d'un contrôleur flou . . . . .	36
3.7	représentation de centre de gravité . . . . .	39
4.1	illustration montrant le bras robotique à deux articulations avec les deux angles, $\theta_1$ et $\theta_2$ . . . . .	43
4.2	illustration montrant toutes les valeurs possibles de $\theta_1$ et $\theta_2$ . . . . .	45
4.3	Coordonnées X-Y générées pour toutes les combinaisons $\theta_1$ et $\theta_2$ à l'aide de formules géométriques directes . . . . .	45
4.4	les plages d'erreurs des angles. . . . .	48
4.5	: GUI pour la modélisation géométrique inverse. . . . .	49
5.1	carte Arduino UNO . . . . .	51
5.2	les composants du carte Arduino UNO . . . . .	51
5.3	servomoteur SG90 . . . . .	53
5.4	première étape d'installation du Matlab support package . . . . .	55
5.5	deuxième étape d'installation du Matlab support package . . . . .	56
5.6	Robot Anthropomorphe à 2DDL . . . . .	57
5.7	Définition de la structure globale (entrée, sortie, type d'opérateurs). . . . .	57
5.8	Les fonctions d'appartenances de la variable d'entrée eX . . . . .	59
5.9	Les fonctions d'appartenances de la variable d'entrée eY . . . . .	59
5.10	Les fonctions d'appartenances de la variable de sortie Ang11 . . . . .	60
5.11	Les fonctions d'appartenances de la variable de sortie Ang22 . . . . .	60
5.12	Les règles possibles . . . . .	61
5.13	fuzzification de la variable d'entrée "Erreur X" . . . . .	62
5.14	fuzzification de variable d'entrée "Erreur Y" . . . . .	62
5.15	les résultats de règle « 1 » aux sorties . . . . .	63
5.16	les résultats de règle « 2 » aux sorties . . . . .	63
5.17	agrégation des règles de sortie Ang11 . . . . .	64
5.18	agrégation des règles de sortie Ang22 . . . . .	64
5.19	les résultats des règles 1 et 2 aux sorties sous matlab . . . . .	64
5.20	défuzzification des résultats sur Matlab . . . . .	65
5.21	l'Agrégation et défuzzification de Ex et Ey selon les 25 règles . . . . .	65

5.22 Observation de la surface1 . . . . . 66

5.23 Observation de la surface2 . . . . . 66

# Liste des tableaux

- 1    Tableau des abréviations . . . . . viii
- 2    Tableau de Notation . . . . . ix
  
- 3.1 Comparaison entre les deux méthodes . . . . . 39
  
- 5.1 les caractéristiques d'Arduino UNO(28) . . . . . 52
- 5.2 les caractéristiques de servomoteur . . . . . 54
- 5.3 Les noms des sous ensembles . . . . . 58



# Abréviations

<b>Acronyme</b>	<b>Signification</b>
RUR	Rassum's Universel Robot
AFNOR	Association Française de normalisation
AFRI	Association Française de Robotique Industrielle
IFR	Fédération Interactionnel de la Robotique
SMA	Système Mécanique Articulé
DDL	Degré De Liberté
SCARA	Sélective Compliance Assembly Robot Arm
PPP	3 Articulations Prismatiques
RPP	Articulation Rotoïde avec 2 Articulations Prismatiques
RRR	3 Articulations Rotoïdes
LF	logique floue
RLF	Régulateur par Logique Floue
Trans	Translation
Rot	Rotation
MGD	Modèle Géométrique Direct
MGI	Modèle Géométrique Inverse
MCD	Modèle Cinématique Direct
MCI	Modèle Cinématique Inverse
ANFIS	Adaptative-Network-based Fuzzy Inference System
FIS	Système Inférence Floue

TABLE 1 – Tableau des abréviations

# Notation

$R_i$	le repère $i$
$W$	le facteur d'échelle
$u$	vecteur
$\mu(x)$	degré d'appartenance
$cg$	limite gauche
$cd$	limite droite
$wg$	largeur gauche
$wd$	largeur droite
$c$	le centre de gaussienne
$\sigma$	la largeur de gaussienne
$\cap$	l'intersection
$\cup$	l'union
$y^*$	centre de gravité

TABLE 2 – Tableau de Notation

# Introduction Général

A notre époque les chercheurs travaillent pour le bien d'humanité et la facilité du vie quotidienne. Pour ces dernières on a besoin à des outils ,des moyens sophistiqués touchés par la technologie tel que la Robotique. ce qu'est généralement utilisé pour remplacer les humaines dans des situations ou ces dernier sont incapables d'effectuer le travail des situations dangereuses. car les robot permet d'effectuer des taches et de manipuler des objets. Parmi les plus connu sont les robot manipulateurs.

Les robots manipulateurs sont les plus utilisés dans les systèmes industriels, car ils sont flexibles permettent d'augmenter la productivité, réduire les coûts et améliorer la qualité de la production. Nous retrouvons également ce type des robots dans les hôpitaux ou ils assistent l'être humain dans des procédures chirurgicales difficiles ainsi que dans des milieux qui lui sont hostiles, tel que dans des pièces à températures très élevée ou radioactive. Ils ont également fait leur preuves dans des domaines tel que l'agriculture, la construction, l'exploitation des fond marins ou de l'espace.

Selon les développements scientifiques spécifiquement de l'électronique et de l'informatique, aussi que : automatique, mécanique et matériaux. Les dernières recherches du robotique sont concentrées presque entièrement sur la commande intelligente des robots manipulateurs pour données une certaine autonomie et permettre de se diffuser dans des nouveaux domaines (les robots de services ,les robots chirurgicaux, les robots d'exploitations etc.. ). Pour commander ou simuler le comportement d'un système mécanique articulé (robot) on doit disposer d'un technique de commande.

les techniques de la commande intelligente sont multiples et considérées comme une solution très intéressante pour les systèmes non linéaires ou il est difficile d'établir un modèle mathématique. Ces outils sont utilisant dans la conception, la modélisation et la commande de systèmes complexe tel que les robots. parmi de ces techniques : les réseaux de neurones ,la logique floue ,les algorithmes génétiques...etc.

L'objectif de notre travail consiste à commander un bras manipulateur rigide à deux axes, en

position en garantissant la stabilité et la robustesse. La loi de commande que nous proposons est la logique floue.

## Organisation de mémoire

Cette mémoire est organisée comme suit :

Au premier chapitre, nous avons donné une généralité sur les robots ( la définition et les types des robots, l'architecture et ces composantes, leurs caractéristiques et les structures ).

Dans le deuxième chapitre, nous avons présenté les notations et les outils mathématiques pour la modélisation des robots, ainsi que nous avons précisé les différents modèles d'un robot manipulateur anthropomorphe à deux degrés de liberté.

Dans le troisième chapitre, nous avons étudié théoriquement les principes de la logique floue et sa structure de fonctionnement générale : fuzzification, inférence floue, base de règles et défuzzification.

Au quatrième chapitre, nous avons donné une simulation à notre robot à base de l'application de l'ANFIS avec utilisation de *Matlab*.

En dernier chapitre, nous avons fait la réalisation et l'implémentation de la commande intelligente dans notre robot avec la carte Arduino UNO par l'utilisation de *Matlab*.

Finalement, une conclusion générale résume notre travail.

# Chapitre 1

## généralités sur les robots

### 1.1 Introduction

Les inventions scientifiques sont devenues un grand service à l'humanité, dont le plus important est les inventions électroniques qui répondent actuellement à de nombreux domaines, notamment dans le domaine de l'industrie et de la commande, certaines applications qui nécessitent l'intervention de l'homme se font de plus en plus remplacer par des machines électroniques parmi ces machines : les robots.

De nos jours, les robots sont devenus des outils nécessaires dans l'environnement de fabrication, permettant d'effectuer des tâches complexes, dangereuses ou répétitives.

Dans ce chapitre, on va donner une idée générale sur les robots, l'historique de robots, leur définition, leurs types, leurs structures, leur classification ainsi que leurs avantages, inconvénients et leur domaine d'application.

### 1.2 Historique

Le mot "robot" vient d'un mot tchèque "robota" qui signifie travail forcé, il a été utilisé pour la première fois en 1920 par Karel Capek dans une pièce intitulée "R.U.R" (Rassum's Universal Robots) (1).

Le mot robotique a été utilisé pour la première fois en 1941 par le célèbre écrivain de science-fiction, Isaac Asimov(2).

Voici quelques dates importantes qui ont marqué l'évolution de la robotique :

- 1947 : première manipulateur électrique télé opéré .

- 1954 : première robot programmable.
- 1961 : utilisation d'un robot industriel, commercialisé par la société UNIMATION (USA), sur une chaîne de montage de Général Motors.
- 1961 : premier robot avec contrôle en effort.
- 1963 : utilisation de la vision pour commander un robot (3).
- 1971 : création de l'association japonaise des robots.
- 1978 : PUMA (Programmable Universel Machine for Assembly) développée par UNIMATION/ VICARM, USA, avec le soutien de Général Motors (1).
- 1985 : premier robot avec un bras de manipulation formé de 3 parallélogrammes : Delta.
- 2015 : l'arrivée des robots collaboratifs (4).

## 1.3 Définition

### 1.3.1 Définition de Robot

Un robot est un système mécanique poly-articulé mû par des actionneurs et commandée par un ordinateur, qui est destiné à effectuer une grande variété de tâches.

### 1.3.2 Définition de Robotique

La robotique est la science et la technologie derrière, la conception, la fabrication et l'application des robots, On peut dire aussi que tout dispositif comporte une partie opérationnelle qui réalise la tâche et une partie décisionnelle qui contrôle la partie opérationnelle. (5).

## 1.4 Type de robot

Il existe deux types des robots : robot manipulateur et robot mobile :

### 1.4.1 Robot manipulateur

La structure mécanique d'un robot manipulateur se compose d'une séquence de corps rigides (liens) inter-connectées au moyen d'articulations ; le manipulateur se caractérise par un bras qui assure la mobilité, un poignet qui confère de la dextérité et un effecteur qui effectue la tâche demandée au robot.(6).

Le robot manipulateur est un robot ancrés physiquement sur leur lieu de travail et généralement mis en place pour une tâche précise répétitive (7).



**Fig. 1.1** – Robot manipulateur

### 1.4.2 Robot mobile

La principale caractéristique des robots mobiles est la présence d’une base mobile qui permet au robot de se déplacer librement dans l’environnement .

Les robots mobiles peuvent se déplacer de manière autonome (dans une installation industrielle, un laboratoire, une surface planétaire, etc.) c’est –à-dire sans l’aide d’opérateurs humains externes (7).



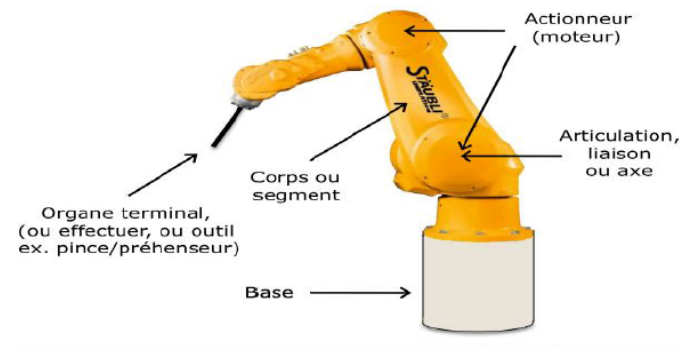
**Fig. 1.2** – Robot Mobile

## 1.5 Robot industriel

Selon AFNOR( Association française de normalisation) Le robot industriel c’est un manipulateur commandé en position ,reprogrammable,polyvalent, à plusieurs degrés de liberté capable de manipuler des matériaux , des pièces , des outils ou des dispositifs spécialisés au cours des mou-

vements variables ,et programmé pour l'exécution d'une variété de tâches .ils souvent l'apparence d'un ou plusieurs bras se terminant par un poignet .son unité de commande utilise notamment un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances (8).

## 1.6 Architecture des robots



**Fig. 1.3** – architecture d'un robot

- **La Base** :le support attaché a l'origine du premier élément de la structure mécanique articulée constituant le bras.
- **Le Bras** :chaîne d'éléments constituée de connecteurs linéaires ou rotatifs, qui sont entraînées les unes par rapport aux autres et portant un effecteur terminal.
- **Le Poignet** : l'ensemble mécanique à l'extrémité du bras de robot, qui est constitué d'articulations et permet d'orienter l'effecteur terminal.
- **Le porteur** :représente l'essentiel du système mécanique articulé SMR (segment, articulation, effecteur terminal, actionneur) ; il a pour rôle d'amener l'organe terminal dans une situation (position ou orientation) donnée imposé par la tâche. Il constituer de :
  - **corps segment** : Corps solides rigides susceptibles d'être en mouvement par rapport à la base du porteur, et les uns par rapport aux autres (9).
  - **Articulation ou axe** :L'articulation lié deux corps successifs en limitant le nombre de degré de liberté (d.d.l) de l'un par rapport à l'autre.

$0 < m \leq 6$  ou **m** est le degré de liberté (ou mobilité).  $m = 1$  (le cas le plus fréquent en robotique) l'articulation est : soit rotoïde,soit prismatique

  - a. **L'articulation rotoïde (R)** : qui produit un mouvement rotatif relatif entre deux liens.
  - b.**L'articulation prismatique (P)** :crée un mouvement de translation relatif entre deux liens (6).



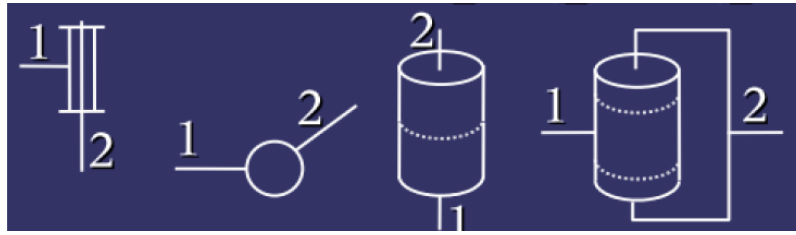


Fig. 1.4 – représentation d'une articulation rotoïde.

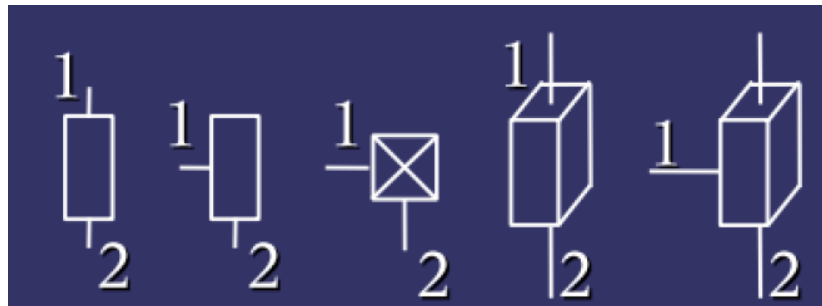


Fig. 1.5 – représentation d'une articulation prismatique.

- **Actionneur** : Les actionneurs sont les dispositifs de conversion d'énergie utilisés à l'intérieur d'un robot, la fonction principale des actionneurs est de convertir l'énergie en un phénomène physique utilisable comme les mouvements.
  - **Effecteur terminal** : l'appareil fixé au poignet par le connecteur qui effectue le travail pour le robot (par exemple : pince, outil, capteur, mesure instrument). Pour saisir, soulever et manipuler des pièces (9).
- **Capteur** : les capteurs sont des systèmes sensoriels qui peuvent acquérir des données sur l'état interne du système mécanique (capteurs proprioceptifs, tels que les transducteurs de position) ainsi que sur l'état externe de l'environnement (capteurs extéroceptifs, tels que les capteurs de force et les caméras) (6).
- **unité de contrôle** : assume simultanément trois rôles différents :
- a**-un rôle d'information : qui consiste à collecter et traiter les informations fournies par les capteurs.
  - b**-un rôle de décision : qui consiste à planifier le mouvement géométrique de la structure de manipulateur à partir de la définition de la tâche fournie par l'opérateur humain et de l'état du système et de son environnement transmis par le capteur.
  - c**-un rôle de la communication : qui consiste à organiser le flux d'informations entre l'unité de contrôle, le manipulateur et son environnement (10).

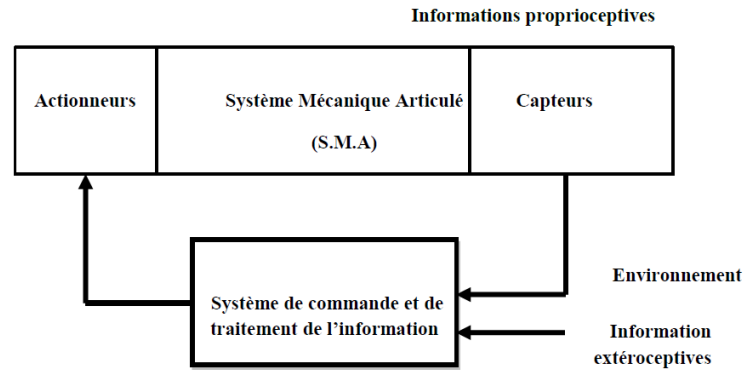


Fig. 1.6 – Constituent d'un robot .

## 1.7 Degré de liberté (DDL)

C'est le nombre des mouvements indépendants qu'un objet peut effectuer, chaque articulation dans un système robotisé donne au robot un degré de liberté.

## 1.8 Structures des robots

### 1.8.1 Structure à chaîne cinématique ouverte (structure sérielle)

Caractérisé par le fait qu'une seule chaîne cinématique relie la base fixe à l'organe terminal mobile.

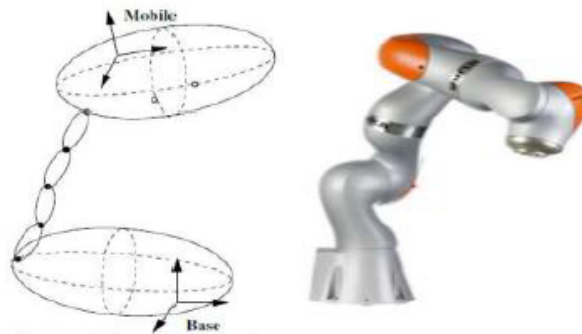


Fig. 1.7 – Structure série d'un robot.

### 1.8.2 Structure à chaîne cinématique fermé (structure parallèle )

Caractérisés par le fait que plus d'une chaîne cinématique assure la liaison entre la base fixe et l'organe terminal mobile.

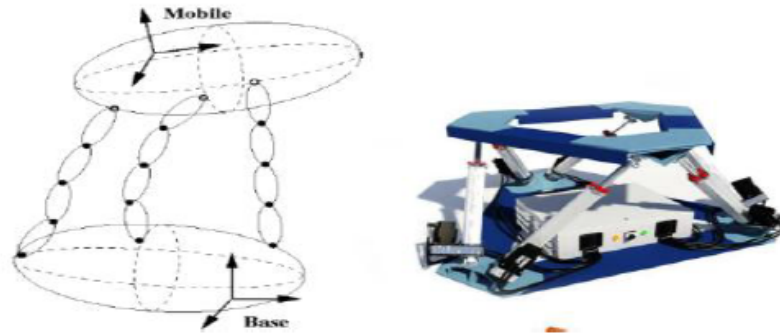


Fig. 1.8 – Structure parallèle d'un robot.

### 1.8.3 Structure hybride

Combinaison de chaîne cinématique sérielle et parallèle(11).

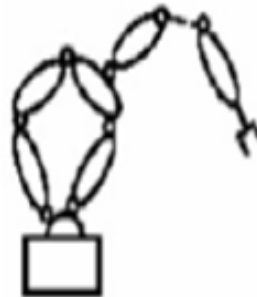


Fig. 1.9 – Structure hybride d'un robot.

## 1.9 Classification des robots industriels

Elle est divisé en deux classes : fonctionnel et géométrique

## 1.9.1 Classification fonctionnel

Selon AFRI (l'association française de robotique industrielle) on distingue quatre classes de robots : manipulateur manuel, manipulateur automatique, robot intelligent et robot programmable (8) :

### 1.9.1.1 Manipulateur Manuel

Est un manipulateur à commande manuelle ou à télécommande avec au moins quatre degrés de liberté.

#### a-Manipulateur sous commande manuelle

la commande s'exerce par action directe sur la charge ou à proximité de celle-ci par exemple bras pour une charge en mouvement.

#### b-Télémanipulateur

la commande est exercée à distance à l'aide d'un boîtier de commande, ou d'un bras principale par exemple : manipulation de matériaux dans un environnement hostile : sous l'eau, dans les centrales nucléaires.etc. . .



Fig. 1.10 – Manipulateur à commande manuelle.

### 1.9.1.2 Manipulateur Automatique

C'est un engin manipulateur avec un minimum de deux degrés de liberté, il suit un cycle automatique, selon une séquence fixe ou variable.

#### a.Manipulateur automatique à séquence fixe

Les manipulateurs à séquence fixe qui admettent un contrôle automatique, mais difficile à reprogrammer.

### b. Manipulateur à séquence variable

Les manipulateurs à séquence variable qui ont un contrôle automatique est qui sont reprogrammé mécaniquement, exemple robot tout ou rien (12).



**Fig. 1.11** – Manipulateur automatique à cycle prééglé.

#### 1.9.1.3 Robot intelligent

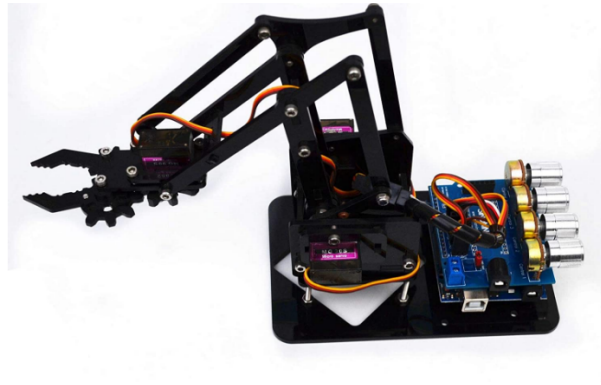
Un robot ayant les moyens de comprendre son environnement et la capacité de mener à bien une tâche malgré les changements des conditions environnantes dans les quelles elle doit être effectuée (12).



**Fig. 1.12** – Robot chimiste intelligent.

#### 1.9.1.4 Robot programmable

Les robots programmables fonctionnent en suivant les instructions données par un ordinateur. Mais pour qu'ils obéissent, un humain a dû informer l'ordinateur des instructions à donner et quand les exécuter.



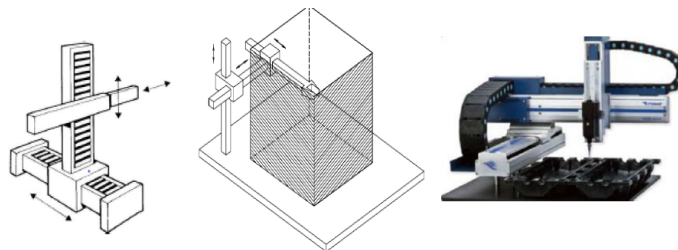
**Fig. 1.13** – Robot programmable.

## 1.9.2 Classification géométrique

Selon la structure géométrique des robots manipulateurs ou bien l'architecture de porteur on classe ces robots en cinq classes qui sont plus réponde dans l'industrie :

### 1.9.2.1 Robot a Structure Cartésien PPP

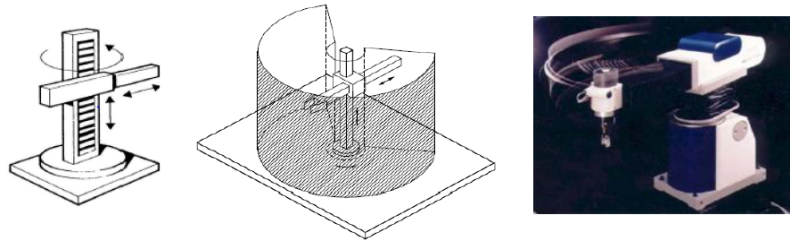
La géométrie cartésienne est réalisée par trois articulations prismatiques dont les axes sont typiquement mutuellement orthogonaux (6).



**Fig. 1.14** – Structure d'un robot cartésien.

### 1.9.2.2 Robot a Structure Cylindrique (RPP) ou (PRP)

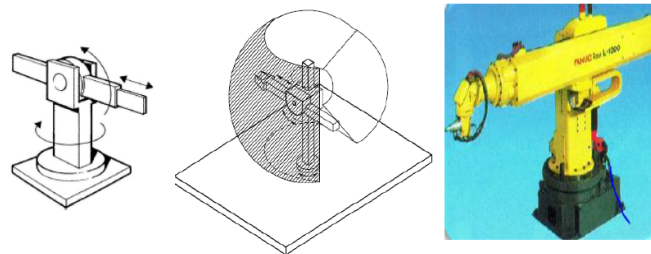
Un bras de robot est appelé robot cylindrique si la première ou la deuxième articulation d'un robot cartésien est remplacée par une articulation rotative .la position centrale du poignet d'un robot cylindrique peut être décrite par un ensemble de coordonnées cylindriques associer aux trois variables articulaires.



**Fig. 1.15** – Structure d'un robot cylindrique.

### 1.9.2.3 Robot a Structure Sphérique RRP

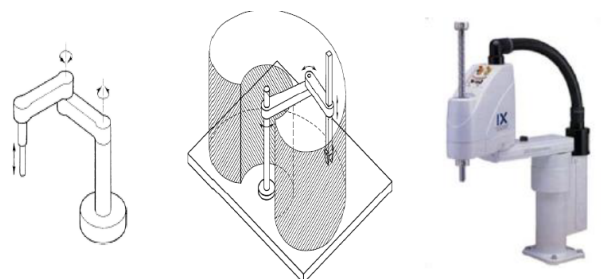
Robot a structure sphérique est un robot avec deux articulations rotatifs et un articulation prismatique (13).



**Fig. 1.16** – Structure d'un robot sphérique.

### 1.9.2.4 Robot a Structure SCARA

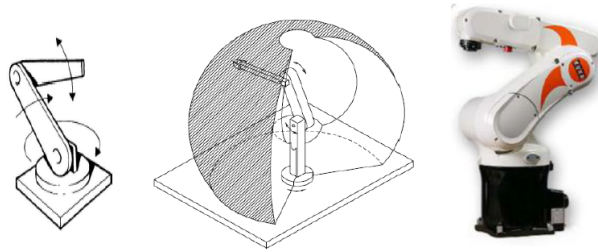
La géométrie SCARA qui peut être réaliser en disposant deux articulations rotatives et une articulation prismatique de manière à ce que tous les axes de mouvement soient parallèles pour assurer une rigidité élevée aux charges verticales et une conformité aux charges horizontales.



**Fig. 1.17** – Structure d'un robot SCARA.

### 1.9.2.5 Robot a Structure Anthropomorphe

Est réalisée par trois articulations révolutionnaires (rotatifs), l'axe révolutionnaire de la première articulation est orthogonal aux axes des deux autres qui sont parallèle. La structure anthropomorphe est la plus à droite, car toutes les articulations sont révolutives.

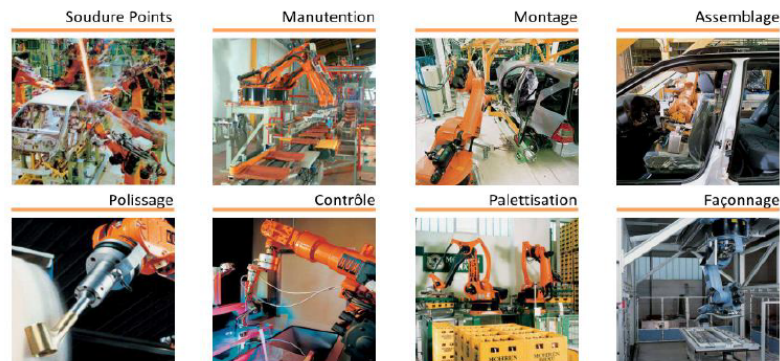


**Fig. 1.18** – Structure d'un robot Anthropomorphe .

●Selon IFR (Fédération Interactionnel de la Robotique) jusqu'un 2005 : 59% des robots manipulateurs installés dans le monde ont une géométrie anthropomorphe ,20% ont une géométrie cartésienne, 12% ont une géométrie cylindrique et 8% ont une géométrie SCARA (6).

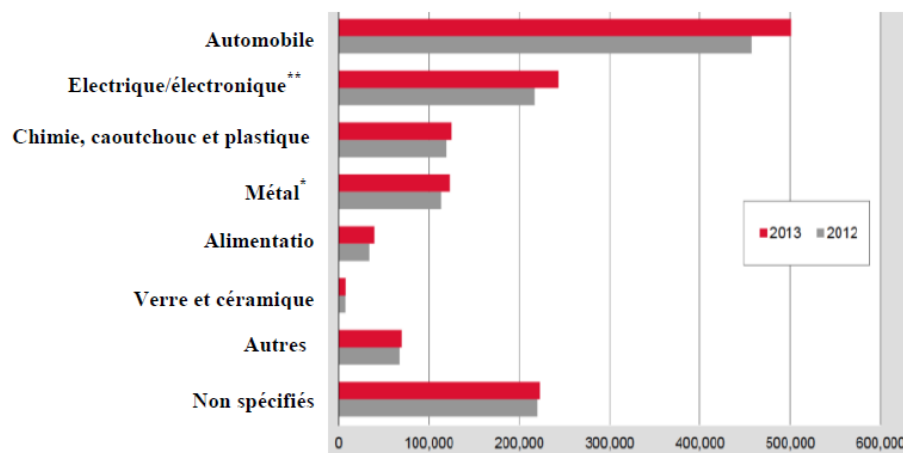
## 1.10 Domaine d'application des Robots

Les robots sont maintenant utilisés dans des nombreuses applications, notamment : soudage, scellement et peinture; manutention, assemblage et inspection des matériaux, et dans les industries non automobiles telles que l'électronique, les produits de consommation, les produits pharmaceutique et services.



**Fig. 1.19** – Les domaines d'application de robot.





† y compris les produits métalliques, les métaux de base et l'industrie des machines.

\*\* y compris communication, précision informatique et médicale.

**Fig. 1.20** – stocks opérationnel des robots industriels par principales industries dans le monde en fin d'année 2012-2013. (14)

## 1.11 Avantages et inconvénients du robot

### 1.11.1 Avantages

- Les robots augmentent la productivité, la sécurité, l'efficacité, qualité et cohérence des produits.
- Les robots peuvent traiter plusieurs tâches simultanément.
- Les robots peuvent être beaucoup plus précis qu'humain.
- Les robots peuvent fonctionner dans des environnements dangereux sans besoin.
- amélioration de la qualité de produit

### 1.11.2 Inconvénients

- Les robots n'ont pas la capacité de répondre en cas d'urgence.
- Les robots sont coûteux, en raison du coût initial de l'équipement, coûts d'installation, besoin de périphériques, de formation et de programmation.

- Remplacement du travail humain (plus de chômage).
- Coût de recyclage important pour les chômeurs et les utilisateurs de nouvelle technologie.

## 1.12 Conclusion

On s'attend à ce que la part de l'utilisation des robots augmente dans les domaines de la vie quotidienne et dans l'industrie avec le développement de la technologie et des matériaux utilisés dans sa fabrication.

Dans ce chapitre, nous avons donné une idée générale sur les robots qui permet de servir la construction de notre robot manipulateur qui va respecter les classifications et les définitions donner dans ce chapitre , puis dans le chapitre suivant on s'intéressera à la modélisation d'un robot anthropomorphe à deux degrés de liberté.

# Chapitre 2

## Modélisation d'un robot anthropomorphe à deux degré de liberté

### 2.1 Introduction

La conception et la commande des robots manipulateurs est actuellement les uns des principales préoccupations de la recherche en robotique. Afin de concevoir un système de simulation ou de contrôle performant, certaines exigences et connaissances du robot sont nécessaires. La modélisation est une étape importante dans la commande des robots elle nous permet d'avoir une vision globale de l'état de comportement du robot sous forme d'équation mathématique, tels que : Des modèles de transformation entre l'espace opérationnel (dans lequel la position de l'effecteur terminal est définie) et l'espace articulaire (dans lequel la configuration du robot est définie) on distingue ce qui suit :

- Modèles géométrique direct et inverse donnant la localisation de l'effecteur terminal en fonction des coordonnées d'articulation du mécanisme et inversement.
- Modèles cinématique direct et inverse donnant la vitesse de l'effecteur terminal en terme de vitesse de joint et inversement.
- Modèles dynamiques donnant les relations entre les couples et les forces des actionneurs, et les positions, vitesses et accélérations des articulations.

Dans ce chapitre on va présenter quelques méthodes pour établir ces modèles, on va commencer par les outils mathématique pour modéliser les mouvements des corps solides dans l'espace puis

on va donner une modélisation d'un robot anthropomorphe à deux degrés de liberté.

## 2.2 Les notations et les outils mathématiques mis en œuvre pour la modélisation des robots

### 2.2.1 Représentation d'un point

Soit  $P$  un point arbitraire en coordonnées cartésiennes  $(P_x^i, P_y^i, P_z^i)$ , mesuré dans le repère  $R_i (O_i, X_i, Y_i, Z_i)$ , les coordonnées homogènes du point  $P$  sont  $(W.p_x^i, W.p_y^i, W.p_z^i, W)$  ou  $W$  est le facteur d'échelle, dans la robotique  $W = 1$ .

$$P^i = \begin{bmatrix} P_x^i \\ P_y^i \\ P_z^i \\ 1 \end{bmatrix} \quad (2.1)$$

### 2.2.2 Représentation d'un vecteur

Pour représenter un vecteur, on se sert de quatre composantes. Trois pour les coordonnées cartésiennes et la quatrième égale à zéro (0).

$$u = \begin{bmatrix} u_x \\ u_y \\ u_z \\ 0 \end{bmatrix} \quad (2.2)$$

### 2.2.3 Représentation d'un plan

les coordonnées homogènes d'un plan  $Q$  de l'équation :  $\alpha x + \beta y + \gamma z + \delta = 0$   
 Exprimé selon le repère  $R_i$ , sont données par le vecteur ligne  $Q^i$  :

$$Q^i = [\alpha^i \beta^i \gamma^i \delta^i] \quad (2.3)$$

si un point  $P$  appartient à un plan  $Q$ , le produit matriciel  $Q^i.P^i$  est nul (15) :

$$Q^i P^i = [\alpha^i \beta^i \gamma^i \delta^i] \begin{bmatrix} P_x^i \\ P_y^i \\ P_z^i \\ 1 \end{bmatrix} = 0 \quad (2.4)$$

## 2.2.4 Transformation homogène

### a. Transformation des repères

Le développement mathématique est basé sur une matrice de transformation homogène ( $4 \times 4$ ), la transformation homogène  $T_j^i$  représente la transformation de repère  $R_i$  pour encadrer  $R_j$  est définie comme :

$$T_j^i = [s_j^i, n_j^i, a_j^i, p_j^i] = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

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

$$T_j^i = \begin{bmatrix} R_j^i & P_j^i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_j^i & n_j^i & a_j^i & P_j^i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

- Ou  $s_j^i$ ,  $n_j^i$  et  $a_j^i$  représente la matrice d'orientation .
- $R_j^i$  : indique le vecteur unitaire suivant des axes  $x_j$  ,  $y_j$  et  $z_j$  du repère  $R_j$  exprimés dans le repère  $R_i$  .
- $P_j^i$  est le vecteur exprimant l'origine de repère  $R_j$  dans le repère  $R_i$  citeWisama.

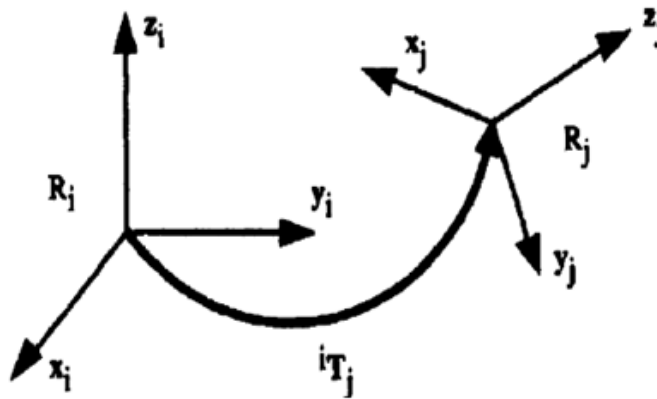


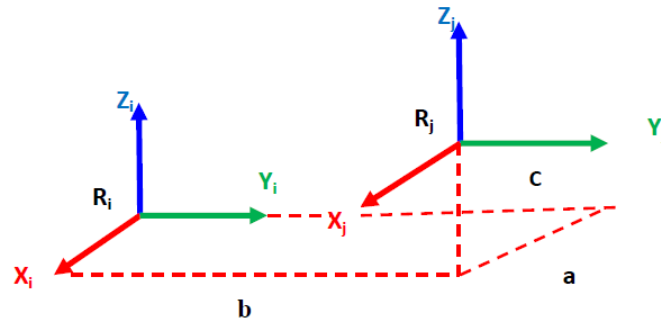
Fig. 2.1 – transformation d'un repère

### b. transformation de translation

soit  $\text{Trans}(a,b,c)$  la transformation d'une translation pure au  $a, b$  et  $c$  sont les translations suivant des axes  $x, y$  et  $z$  respectivement. Quand l'orientation est conservée, la matrice de transformation

de cette translation a la forme suivant (15) :

$$T_j^i = Trans(a,b,c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$



**Fig. 2.2** – transformation d'une translation pure

la matrice  $Trans(a,b,c)$  peut être décomposée en produit de trois matrice :  $Trans(x,a) Trans(y,b) Trans(z,c)$ .

$$Trans(a,b,c) = Trans(x,a)Trans(y,b)Trans(z,c).$$

$$\text{translation suivant l'axe x : } Trans(a,b,c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{translation suivant l'axe y : } Trans(a,b,c) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{translation suivant l'axe z : } Trans(a,b,c) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

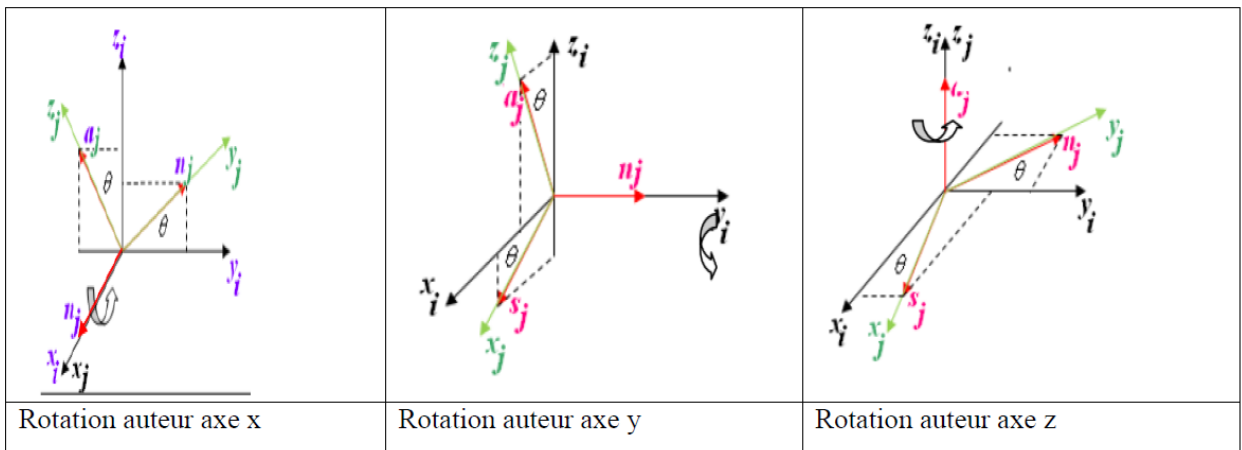
### c. Transformation de rotation :

transformation correspondant aux rotations autour des axes x,y ou z d'un angle  $\theta$ (17) :

$$Rot(x,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

$$Rot(y,\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 0 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

$$Rot(z,\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

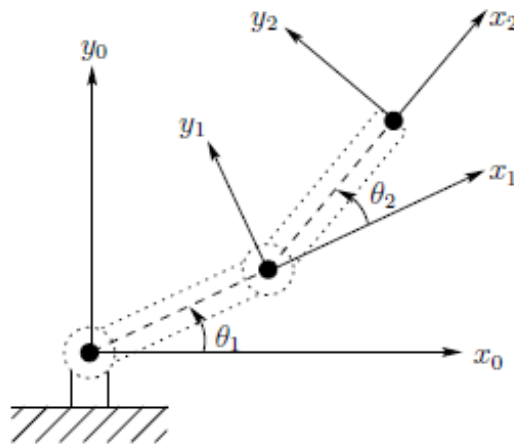


**Fig. 2.3** – la transformation de rotation suivant les axe x,y et z

## 2.3 Modélisation d'un robot manipulateur anthropomorphe

### 2.3.1 Présentation du robot manipulateur utilisé

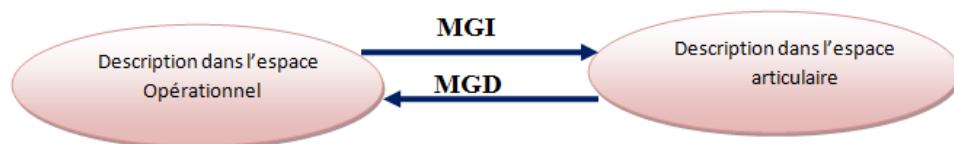
On utilise dans notre étude un robot anthropomorphe à deux (02) degrés de liberté (Fig.2.4). Il est constitué de deux segments de masses et longueurs  $\alpha_1 m_1$  et  $\alpha_2 m_2$  respectives. La position de l'organe terminal est complètement déterminée en utilisant deux coordonnées  $(x,y)$  qui sont dites coordonnées opérationnelles. La configuration du robot est déterminée en utilisant les angles  $(\theta_1$  et  $\theta_2)$  qui sont dites coordonnées articulaires.



**Fig. 2.4** – Robot anthropomorphe à 2 degrés de liberté.

### 2.3.2 Modèle géométrique

Il permet d'établir la relation entre la position de l'organe terminal (coordonnées opérationnelles) et la configuration du robot (coordonnées articulaires).



**Fig. 2.5** – modèle géométrique direct et inverse.



### Modèle géométrique direct (MGD)

Représente les relations calculant les coordonnées opérationnelles, donnant la localisation de l'effecteur terminal, en fonction des coordonnées articulaires. Dans le cas d'une simple chaîne ouverte, il peut être représenté par la matrice de transformation  $T_n^0$  :

$$T_n^0 = T_1^0(\theta_1)T_2^1(\theta_2)\dots T_n^{n-1}(\theta_n). \quad (2.11)$$

Le modèle géométrique direct du robot peut également être représenté par la relation :

$$X = f(\theta) \quad (2.12)$$

$\theta$  : étant le vecteur des coordonnées d'articulation, telles que :

$$\theta = [\theta_1 \theta_2 \dots \theta_n]^T \quad (2.13)$$

$X$  : les coordonnées opérationnelles telles que(16) :

$$X = [X_1 X_2 \dots X_m]^T. \quad (2.14)$$

Dans le cas du robot anthropomorphe, et d'après la figure 2.4, le MGD est donné par le système d'équations suivant :

$$\begin{aligned} x = x_2 &= \alpha_1 \cos(\theta_1) + \alpha_2 \cos(\theta_1 + \theta_2) \\ y = y_2 &= \alpha_1 \sin(\theta_1) + \alpha_2 \sin(\theta_1 + \theta_2). \end{aligned} \quad (2.15)$$

### Modèle géométrique Inverse (MGI)

Est le problème inverse Consiste à calculer les coordonnées articulaires a un emplacement donnée de l'effecteur terminal, quand il existe, ce qui peut représente par la relation(16) :

$$\theta = f^{-1}(X) \quad (2.16)$$

Dans le cas d'un robot anthropomorphe, la détermination du MGI se fait en 2 étapes.

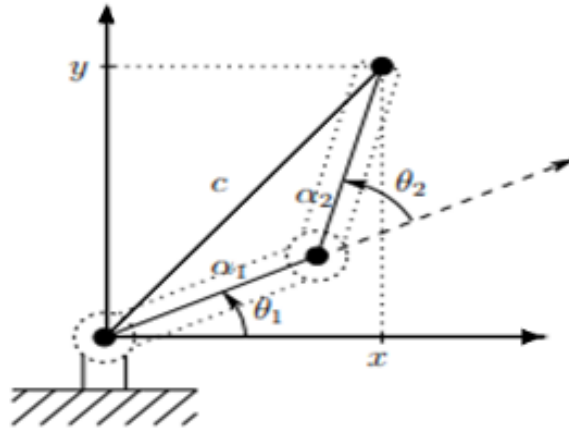
**Étape 1** : Calcul de l'angle  $\theta_2$

Considérons le diagramme de Figure.2.6 , on utilisant la loi des cosinus pour trouver l'angle  $\theta_2$  :

$$\cos(\theta_2) = \frac{x^2 + y^2 - \alpha_1^2 - \alpha_2^2}{2\alpha_1\alpha_2} = D \quad (2.17)$$

D'où

$$\theta_2 = \cos^{-1}(D) = \arccos\left(\frac{x^2 + y^2 - \alpha_1^2 - \alpha_2^2}{2\alpha_1\alpha_2}\right) \quad (2.18)$$



**Fig. 2.6** – Résolution des angles d'articulation d'un robot anthropomorphe à 2 ddl.

cependant, une meilleure façon de trouver  $\theta_2$  et de trouver à partir de l'équation (2.17) et de l'équation suivant alors :

$$\sin^2 \theta_2 + \cos^2 \theta_2 = 1 \quad (2.19)$$

D'où

$$\sin \theta_2 = \pm \sqrt{1 - D^2} \quad (2.20)$$

Donc on aura :

$$\begin{cases} \cos \theta_2 = D \\ \sin \theta_2 = \pm \sqrt{1 - D^2} \\ \Rightarrow \tan \theta_2 = \frac{\sin \theta_2}{\cos \theta_2} = \frac{\pm \sqrt{1 - D^2}}{D} \Rightarrow \theta_2 = \pm \arctan\left(\frac{\sqrt{1 - D^2}}{D}\right) \end{cases} \quad (2.21)$$

Ceci traduit le fait que pour une position donnée de l'organe terminal le robot peut avoir deux configurations possibles.

**Etape 2** Calcul de l'angle  $\theta_1$  :

$$\theta_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{\alpha_2 \sin(\theta_2)}{\alpha_1 + \alpha_2 \cos(\theta_2)}\right) \quad (2.22)$$

On Obtient finalement le MGI Suivant :

$$\begin{cases} \theta_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{\alpha_2 \sin(\theta_2)}{\alpha_1 + \alpha_2 \cos(\theta_2)}\right) \\ \theta_2 = \pm \arctan\left(\frac{\sqrt{1 - D^2}}{D}\right) \text{ avec } D = \frac{x^2 + y^2 - \alpha_1^2 - \alpha_2^2}{2\alpha_1\alpha_2} \end{cases} \quad (2.23)$$

### 2.3.3 Modèle cinématique

Il permet d'exprimer la vitesse de l'organe terminal (vitesses opérationnelles) en fonction des vitesses articulaires et inversement.

### Modèle cinématique direct (MCD)

Exprime les vitesses des coordonnées opérationnelles  $\dot{X}$  en fonction des vitesses articulaires  $\dot{\theta}$  :

$$\dot{X} = J(\theta).\dot{\theta} \quad (2.24)$$

Ou :  $J(\theta)$  : indique la matrice jacobienne du mécanisme telle que :

$$J(\theta) = \frac{\partial X}{\partial \theta} \quad (2.25)$$

Cette Matrice jacobienne apparait dans le calcul du modèle différentiel direct qui donne les variations différentielles  $dx$  des coordonnées opérationnelles en terme de variation différentielles des coordonnées articulationnelles  $d\theta$  telle que :

$dx = J(\theta) d\theta$ . (16) Dans notre cas

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (2.26)$$

Ce qui donne le MCD suivant :

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\alpha_1 \sin(\theta_1) - \alpha_2 \sin(\theta_1 + \theta_2) & -\alpha_2 \sin(\theta_1 + \theta_2) \\ \alpha_1 \cos(\theta_1) + \alpha_2 \cos(\theta_1 + \theta_2) & \alpha_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (2.27)$$

on pose  $X = \begin{bmatrix} x \\ y \end{bmatrix}$  et  $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$  donc on obtient :  $\dot{X} = j\dot{\theta}$

### Modèle cinématique inverse (MCI)

L'objectif du MCI est d'exprimer les vitesses articulaires en fonction de vitesse opérationnelle. il est obtenu en inversant la matrice de jacobienne (16).

$$\dot{\theta} = J^{-1}(\theta)\dot{X} \quad (2.28)$$

Pour le robot planaire utilisé, nous obtenons le MCI :

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \frac{1}{\alpha_1 \alpha_2 \sin \theta_2} \begin{bmatrix} \alpha_2 \cos(\theta_1 + \theta_2) & \alpha_2 \sin(\theta_1 + \theta_2) \\ -\alpha_1 \cos(\theta_1) - \alpha_2 \cos(\theta_1 + \theta_2) & -\alpha_1 \sin(\theta_1) - \alpha_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (2.29)$$

### 2.3.4 Le modèle dynamique

Les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples et/ou forces applique aux actionneurs et les positions, vitesses et accélérations des articulations ,On représente le modèle dynamique par une relation de la forme :

$$\Gamma = f(\theta, \dot{\theta}, \ddot{\theta}, F) \quad (2.30)$$

Où :

$\Gamma$  : Vecteur des couples et forces des actionneurs.

$\theta$  : Vecteur des positions articulaires.

$\dot{\theta}$  : Vecteur des vitesses articulaires.

$\ddot{\theta}$  : Vecteur des accélérations articulaires.

$F$  : Effort extérieur ( force et couple) , à exerce par l'organe terminal [17].

Le formalisme d'Euler-Lagrange sera utilisé dans ce chapitre pour établir le modèle dynamique du robot à 2 degrés de liberté.

#### • Formalisme d'Euler-Lagrange

Le formalisme ou la méthode d'Euler-Lagrange nous permet de décrire les équations de mouvements de plusieurs corps articulés les uns par rapport aux autres. Son usage est plus simple, pour les non spécialistes, que d'autres formalismes, même si elle exige souvent des calculs plus longs. La forme générale de l'équation d'Euler-Lagrange représentant le mouvement d'un robot constitue de  $n$  degrés de liberté et de  $n$  articulation, s'écrit comme suit :

$$\Gamma_i = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} + \frac{\partial F_D}{\partial \dot{\theta}_i}, i = 1 \dots n \quad (2.31)$$

Avec

$L$  : Lagrangien du système, défini par :  $L = E_c - E_p$ , où  $E_c$  est l'énergie cinétique du système et  $E_p$  son énergie potentielle.

$F_D$  : Fonction de dissipation ayant la forme générale  $F_D = \frac{1}{2} \sum_{i=1}^n f_i \dot{\theta}_i^2$ , où  $f_i$  sont les coefficients de frottement visqueux.

Ces équations dynamiques seront dérivées en 5 étapes :

- Calcul de l'énergie cinétique
- Calcul de l'énergie potentielle
- Calcul de la fonction de dissipation
- Calcul du Lagrangien
- Faire la différentiation

Certaines hypothèses simplificatrices sont à prendre en considération afin de permettre l'élaboration du modèle dynamique :

- Les frottements sont de nature linéaire et visqueuse
- Les segments sont supposés rigides
- Les actionneurs sont idéaux, c'est-à-dire leur sortie est directement proportionnelle au signal de commande
- Les masses des actionneurs sont supposées incluses avec les masses des segments

**Etape 1** : Calcul de l'énergie cinétique

$$E_c = \frac{1}{2} \sum_{i=1}^n I_i \dot{\theta}_i^2 \quad (2.32)$$

Où  $I_i$  est le moment d'inertie du segment  $i$  par rapport à un axe de rotation passant par son extrémité,  $I_i = \frac{1}{3} m_i \alpha_i^2$ . L'énergie cinétique aura l'expression suivante :

$$E_c = \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} m_2 v_{G_2}^2 \quad (2.33)$$

Où  $V_{G_2}$  est la vitesse de translation du centre de gravité de la 2ème tige .

On obtient :

$$E_c = a \dot{\theta}_1^2 + b \dot{\theta}_2^2 + c \dot{\theta}_1 \dot{\theta}_2 \quad (2.34)$$

avec

$$a = \frac{1}{6} m_1 \alpha_1^2 + \frac{1}{2} m_2 \alpha_1^2 + \frac{5}{12} m_2 \alpha_2^2 + \frac{1}{2} m_2 \alpha_1 \alpha_2$$

$$b = \frac{5}{12} m_2 \alpha_2^2$$

$$c = \frac{5}{6} m_2 \alpha_2^2 + \frac{1}{2} m_2 \alpha_1 \alpha_2$$

**Etape 2** : Calcul de L'énergie potentielle

$$E_p = d. \sin(\theta_1) + e. \sin(\theta_1 + \theta_2) \quad (2.35)$$

Avec

$$d = \left(\frac{m_1}{2} + m_2\right) \alpha_1 g, \quad e = \frac{1}{2} m_2 \alpha_2 g.$$

**Etape 3** : Détermination de La fonction de dissipation

$$F_D = \frac{1}{2} (f_1 \dot{\theta}_1^2 + f_2 \dot{\theta}_2^2) \quad (2.36)$$

**Etape 4 :**Lagrangien du système

$$L = E_c - E_p = a\dot{\theta}_1^2 + b\dot{\theta}_2^2 + c\dot{\theta}_1\dot{\theta}_2 - d.\sin(\theta_1) - e.\sin(\theta_1 + \theta_2) \quad (2.37)$$

**Etape 5 :**Différentiation

$$\begin{cases} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = 2a\ddot{\theta}_1 + c\ddot{\theta}_2 \\ \frac{\partial L}{\partial \theta_1} = -d.\cos(\theta_1) - e.\cos(\theta_1 + \theta_2) \\ \frac{\partial F_D}{\partial \dot{\theta}_1} = f_1\dot{\theta}_1 \end{cases} \quad (2.38)$$

$$\begin{cases} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) = 2a\ddot{\theta}_2 + c\ddot{\theta}_1 \\ \frac{\partial L}{\partial \theta_2} = -e.\cos(\theta_1 + \theta_2) \\ \frac{\partial F_D}{\partial \dot{\theta}_2} = f_2\dot{\theta}_2 \end{cases} \quad (2.39)$$

nous avons :

$$\Gamma_1 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1} + \frac{\partial F_D}{\partial \dot{\theta}_1}$$

$$\Gamma_2 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \frac{\partial L}{\partial \theta_2} + \frac{\partial F_D}{\partial \dot{\theta}_2}$$

Donc a partir des équations (2.38) et (2.39), on obtient le modèle dynamique suivant :

$$\begin{cases} \Gamma_1 = 2a\ddot{\theta}_1 + c\ddot{\theta}_2 + d.\cos(\theta_1) + e.\cos(\theta_1 + \theta_2) + f_1\dot{\theta}_1 \\ \Gamma_2 = 2b\ddot{\theta}_2 + c\ddot{\theta}_1 + e.\cos(\theta_1 + \theta_2) + f_2\dot{\theta}_2 \end{cases} \quad (2.40)$$

## 2.4 Conclusion

Dans ce chapitre nous avons présenté les notations et les outils mathématiques mis en œuvre pour la modélisation des robots, puis on a donné une présentation de notre robot, on a aussi exposé les différents modèles (géométrique, cinématique et dynamique) utilisés qui permettent de commander, de programmer et exécuter les tâches requises. Nous avons vu que :

Les modèles géométriques directes et inverses expriment les positions de l'organe terminal en fonction des variables articulaires et inversement, les modèles cinématiques directs et inverses expriment les vitesses de l'organe terminal en fonction des variables articulaires 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, les positions, vitesses et accélérations des articulations.

Dans le chapitre suivant on s'intéressera à l'étude théorique de la logique floue dans un robot a 2ddl.

# Chapitre 3

## Étude théorique de la commande intelligente ( la logique floue)

### 3.1 Introduction

Aujourd'hui l'utilisation de la commande intelligente sera plus intéressante pour les ingénieurs et les industriels automatisés. Elle est représentée par un ensemble de méthodes et de techniques basées sur les outils de l'intelligence artificielle ( les réseaux neuronaux, la logique floue et les algorithmes génétiques).

La logique floue c'est une méthode moderne de la commande intelligente a été développée à l'origine au début des années 1960 par le professeur Lotfi Zadeh, qui a donné un nouveau type de paradigme informatique capable de modéliser la propre incertitude du raisonnement humain. Cet outil est utilisé car de sa amélioration des performances et la simplicité des implémentations.

Dans ce chapitre, on va introduire les principes de base de la LF : la définition de la commande floue et les ensembles flous, sa structure de fonctionnement ,ses opérations...etc.

### 3.2 Le but d'utiliser la logique floue

Pour des structures simples comme le bras robotisé à deux articulations, il est possible de déduire mathématiquement les angles au niveau des articulations en fonction de l'emplacement souhaité de la pointe du bras. Cependant, avec des structures plus complexes (par exemple : des bras robotiques à n articulations fonctionnant dans un espace d'entrée tridimensionnel), la

déduction d'une solution mathématique pour la géométrie inverse peut s'avérer difficile.

En utilisant la logique floue, nous pouvons construire un système d'inférence floue qui déduit la géométrie inverse si la géométrie directe du problème est connue, évitant ainsi le besoin de développer une solution analytique. En outre, la solution floue est facilement compréhensible et ne nécessite pas de connaissances de base particulières pour la comprendre et l'évaluer.

Dans la section suivante, un aperçu général pour développer une telle solution est décrit, et plus tard, les étapes détaillées sont élaborées.

### 3.3 La logique classique et la logique floue

La logique classique ne peut traiter que les deux états : vrai ou faux. Elle se trouve incapable de représenter convenablement ces formules linguistiques, c'est là où intervient la logique floue.

La logique floue est une extension de la logique classique qui permet la modélisation des imperfections des données et se rapproche dans une certaine mesure de la flexibilité du raisonnement humain.

Tous les résultats obtenus par la logique classique doivent être retrouvés par la logique floue, car la logique classique est un sous-ensemble de la logique floue.

### 3.4 La commande floue

La commande floue est introduite par Mamdani et Assilian ; mais aussi largement développée au Japon au début des années 80(21).

Le principe fonctionnel de la commande floue est consisté sur le traitement des problèmes de commande de processus le plus souvent à partir de connaissance des experts ou d'opérateurs qualifiés travaillant sur le processus(22)

On peut utiliser la commande floue dans : Machines-outils, Groupes d'ascenseurs, D'appareils électroménagers, Caméras, Voitures ou hélicoptères sans pilote. . .



## 3.5 Historique

développement de la logique floue : (18)

- 1961 : Lotfi Zadeh affirme dans son article pour un nouveau type de mathématiques « floues ».
- 1965 : Lotfi Zadeh introduit le concept de décors floue.
- 1969 : Marinos (Duke université) mène la première recherche visant à la mise en œuvre matérielle de la logique floue.
- 1972 : M.sugeno présent l'idée pour mesure floue.
- 1974 : E.Mamdani présente une application floue pour contrôler une machine à vapeur dans un cadre académique.
- 1980 : Yamakawa construire le premier circuit flou avec des composants bipolaires discret.
- 1983 : la première application industrielle dans un four à ciment au Danemark.
- 1984 : Togai et Watanabe présentent la première implémentation VLSI des contrôleurs flous.
- 1986 : Hitachi met rn service un système de métro contrôlé flou.
- 1987 : Yamakawa présente le premier contrôleur analogique flou.
- 1988 : Togai implémente les premiers processeurs numériques flous.
- 1989 : LIFE : le laboratoire de recherche internationale sur le flou en ingénierie commence au Japon.
- 1990 : Yamakawa crée l'Institut des Systèmes de Logique Floue (FLSI) au Japon.
- 1992 : la première conférence internationale de l'IEEE sur le système flou.
- 1993 : le premier numéro des transactions IEEE sur les systèmes flous.

## 3.6 Les ensembles floues

La fonction caractéristique de l'ensemble ne prend pas que les deux valeurs discrètes (0 ou 1).

- Valeur 0 ... (l'élément n'appartient pas ...).
- Valeur 1 ... (l'élément appartient à l'ensemble...)(19).

Exemple : l'ensemble A des âges compris entre 20 et 40 ans pour la fonction caractéristique :

$$A = \begin{cases} 1 & \text{si } 20 \leq x \leq 40. \\ 0 & \text{sinon.} \end{cases} \quad (3.1)$$

X : c'est la variable de l'âge.

A : l'ensemble des âges.

Un ensemble flou est défini par une fonction d'appartenance qui peut prendre toutes les valeurs réelles comprises entre 0 et 1 (19).

$\mu : x \rightarrow [0; 1]$ .

B : ensemble flou.

X3

X1

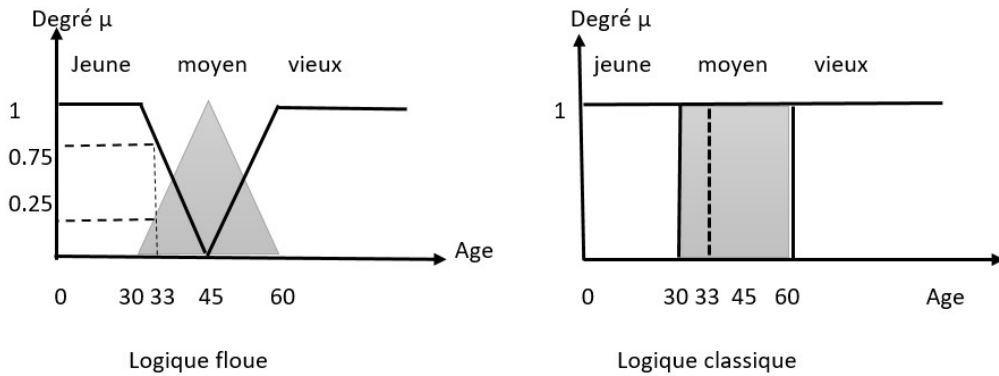
X2

$\mu$  : ensemble (élément) = degré d'appartenance compris entre 0 et 1 (20).

**Degré d'appartenance** : à L'univers de discours est l'ensemble des valeurs réelles que peut prendre la variable floue  $x$  .et  $\mu_B(x)$  c'est le degré d'appartenance de l'élément  $x$  à l'ensemble flou B.

**-Un exemple simple** : une personne de 33 ans appartient à l'ensemble « Jeune » avec une valeur  $\mu=0.75$  de la fonction d'appertenance et à l'ensemble « Moyens » avec une valeur  $\mu=0.25$ .

- Variable linguistique :  $\hat{\text{Age}}$
- Valeur d'une variable linguistique : Jeune, Moyen, Vieux, ...
- Ensembles flous : "jeune", "moyen", "Vieux", ...
- Plage de valeurs : (0, 30, 45,60,...)
- Fonction d'appartenance :  $0 < \mu < 1$



**Fig. 3.1** – classification de l'âge selon les ensembles flous et classique

### 3.6.1 variable linguistique

Une variable linguistique est une variable dont les valeurs sont des mots ou des phrases. Un langage naturel ou artificiel plutôt que numérique. Selon une variable linguistique est définie par :

-Son nom : "x".

-Son ensemble de termes : "TS (x)", qui est l'ensemble des valeurs linguistiques ou des étiquettes de "x".

-La variable de base "u", qui prend en charge les valeurs linguistiques de "x". En d'autres termes, les fonctions d'appartenances aux valeurs linguistiques de «x» sont définies dans le domaine de «u».

-L'univers du discours «U» associé à la variable de base «u».(18)

### 3.6.2 Les Différentes formes et ces fonctions d'appartenances floues

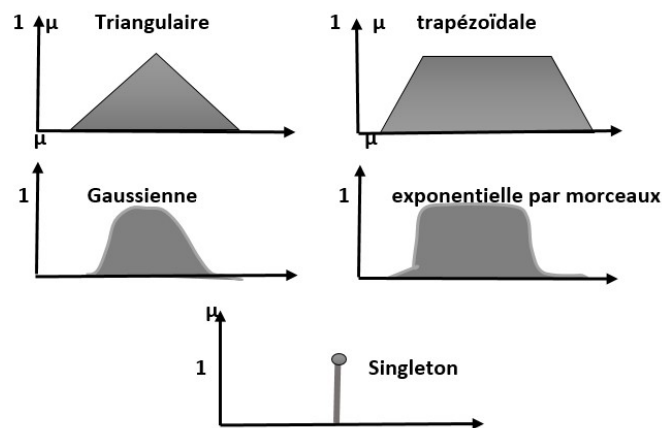


Fig. 3.2 – Les Différentes formes pour les fonctions d'appartenances

#### 3.6.2.1 Les fonctions d'appartenances floues

- a- Fonction d'appartenance trapézoïdale.

$$\mu(x; a; b; c; d) = \max\left(0; \min\left(\frac{x-a}{b-a}; 1; \frac{d-x}{d-c}\right)\right) \quad (3.2)$$

Où a, b, c et d sont les coordonnées des apex du trapèze. Si b=c, nous obtenons une fonction d'appartenance triangulaire.

- b- Fonction d'appartenance exponentielle par morceaux.

$$\mu(x; cg; cd; wg; wd) = \begin{cases} \exp\left(-\left(x - \frac{cg}{2wg}\right)^2\right) & \text{si } x < cg \\ \exp\left(-\left(x - \frac{cd}{2wd}\right)^2\right) & \text{si } x > cd \\ 1 & \text{autrement} \end{cases} \quad (3.3)$$

Où  $cg$  et  $cd$  sont les limites gauche et droite, respectivement et  $wg$  et  $wd$  sont les largeurs gauche et droite, respectivement. Pour  $cg=cd$  et  $wg=wd$ .

- c- La fonction d'appartenance gaussienne est obtenue.

$$\mu(x; c; \sigma) = \exp\left(-\left(x - \frac{c}{2\sigma}\right)^2\right) \quad (3.4)$$

Où  $c$  c'est le centre de la gaussienne et  $\sigma$  sa largeur.

$$\mu(x) \begin{cases} 1 & \text{si } x = x_0 \\ 0 & \text{autrement} \end{cases} \quad (3.5)$$

on utilise généralement trois formes géométriques pour les fonctions d'appartenances : trapézoïdale, triangulaire et de cloche. Les deux premières formes sont les plus employées car à sa simplicité (25).

### 3.6.3 Les opérateurs de la logique floue

On étudie les opérations de base sur les ensembles flous. Dans la suite, on suppose que  $A$  et  $B$  sont des ensembles flous définis dans le même univers de discours  $U$  ayant respectivement  $\mu_A$  et  $\mu_B$  comme fonction d'appartenance (21).

Il existe quatre opérateurs de base pour gérer les variables floues : l'intersection, l'union, complément et implication. Mais les deux opérateurs les plus souvent dans la commande par la logique floue : opérateur(ET) et opérateur (OU)

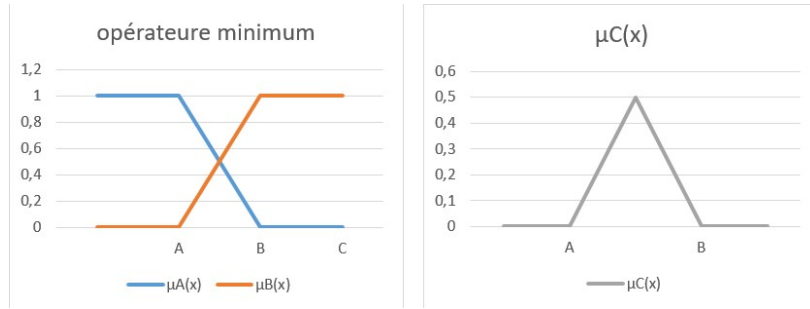
#### 3.6.3.1 Opérateur(ET)

L'opérateur « ET » correspond l'intersection on a donc :

$$C = A \cap B = A \text{ ET } B \quad (3.6)$$

Dans la logique floue, l'intersection est réalisée dans la plupart des cas par la formulation de minimum, appliqué aux fonctions d'appartenance  $\mu_A(x)$  et  $\mu_B(x)$  des ensembles A et B, à savoir :

$$\mu_c(x) = \min[\mu_A(x), \mu_B(x)] \tag{3.7}$$



**Fig. 3.3** – Opérateur « ET ».

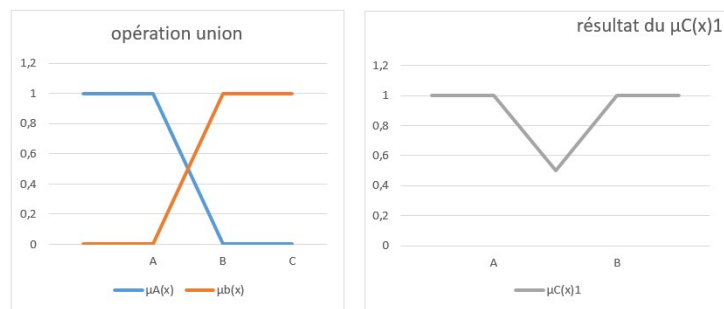
### 3.6.3.2 Opérateur (OU)

L'opérateur « OU » correspond à l'union on a donc :

$$C = A \cup B = A \text{ ou } B. \tag{3.8}$$

La réalisation de l'union se fait en générale par la formulation maximum :

$$\mu_c(x) = \max[\mu_A(x), \mu_B(x)] \tag{3.9}$$



**Fig. 3.4** – Opérateur « ou »

### 3.6.3.3 Opérateur (NON)

L'opérateur « non » correspond à la complémentation d'un ensemble A. on a donc :

$$C = \bar{A} = Non(A) \quad (3.10)$$

Est défini par les éléments de x qui n'appartiennent pas à l'ensemble A. Dans le cas de la logique floue, cette définition peut être exprimée par les fonctions d'appartenance de la manière suivante :

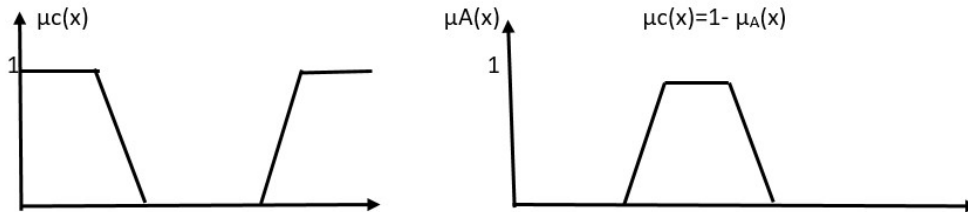


Fig. 3.5 – complémentation

## 3.7 Structure d'un contrôleur flou

La structure de base d'un contrôleur flou se compose de quatre parties principales un bloc de fuzzification, une base des règles, un mécanisme d'inférence et un bloc de défuzzification :

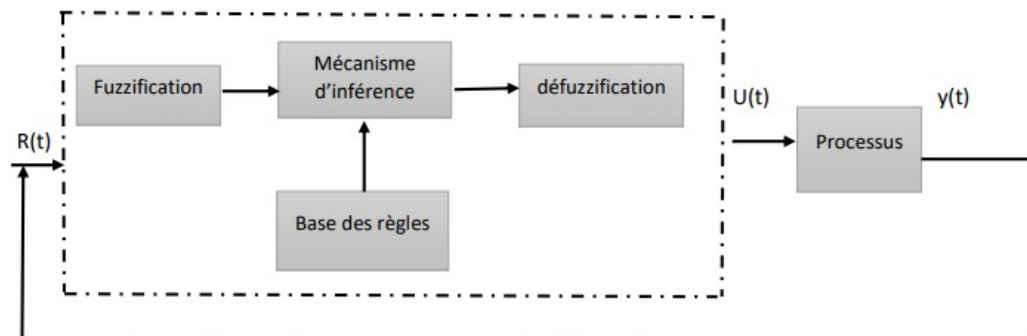


Fig. 3.6 – structure d'un contrôleur flou

### 3.7.1 Fuzzification

Est l'étape de conversion des entrées numériques en groupes obscurs dans le système obscur (en degré d'appartenance flou par évaluation d'une fonction d'appartenance. (19))(24).

On propose maintenant trois fuzzifiers : Fuzzifier singleton, Gaussian flou et traingulaire flou.

### **Fuzzification d'une variable numérique**

Pour fuzzifier une variable numérique il faut donner :

- L'univers de discours, c'est-à-dire la plage de variation possibles des entrées et des sorties.
- Une répartition en classe floue de cet univers.
- Les fonctions d'appartenance de chacune de ces classes.

### **3.7.2 l'inférence floue**

C'est un mécanisme de décision. Il permet à partir d'un fait observé de la base des règles floues une décision en exploitant le raisonnement approximatif. Dans les inférences de régulateur par logique floue interviennent les opérateurs ET et OU. L'opérateur ET s'applique aux variables à l'intérieur d'une règle tandis que l'opérateur OU lie les différentes règles. A cause de l'empieusement des fonctions d'appartenances, en général deux ou plusieurs règles sont activées en même temps. Ce fait doit être pris en considération lors de la réalisation de l'opérateur OU.

Il existe plusieurs possibilités pour réaliser ces opérateurs qui s'appliquent aux fonctions d'appartenances. On introduit alors la notion de méthode d'inférence. Elle détermine la réalisation des différents opérateurs dans une inférence, permettant ainsi un traitement numérique de cette dernière.

Pour le réglage par logique floue, on utilise en général une des méthodes suivantes :

#### **3.7.2.1 Méthode de Mamdani**

Mamdani est la méthode d'inférence floue est la méthodologie floue la plus courante. Cette méthode a été parmi les premiers systèmes de contrôle construits en utilisant la théorie des ensembles flous.il a été proposé par Mamdani (1975) comme une tentative de contrôler une combinaison de machine à vapeur et de chaudière en synthétisant un ensemble de règles de contrôle linguistique obtenues auprès d'opérateurs humains expérimentés.

### 3.7.2.2 Méthode Sugeno

Les systèmes flous TS emploient des fonctions linéaires de variables d'entrée comme implémentées en conséquence dans le système neuronal-flou.

### 3.7.2.3 Méthode Tuskamoto

Dans cette méthode, le conséquent de chaque règle floue est représenté par un ensemble flou avec une fonction d'appartenance monotone.

## 3.7.3 Base de règles

La base de règles floues, ou base de connaissances, contient des règles floues décrivant le comportement du système, elle est le cœur du système entier dans le sens où tous les autres composants sont utilisés pour interpréter et combiner ces règles afin de former le système final elle est composée :

1. D'une base de données fournissant les informations nécessaires pour les fonctions de Normalisation.

2. La base de règle constitue un ensemble d'expressions linguistiques structurées autour d'une connaissance d'expert, et représentée sous forme de règles :

Si «condition » Alors conséquences

## 3.7.4 défuzzification

Si le processus d'occultation est la porte d'entrée du monde logique mystérieux, alors le processus de décompression en est la porte de sortie. Grâce à ce processus, les valeurs linguistiques ambiguës sont converties en valeurs numériques qui sont faciles à gérer pour les ordinateurs et les machines en général.(24)

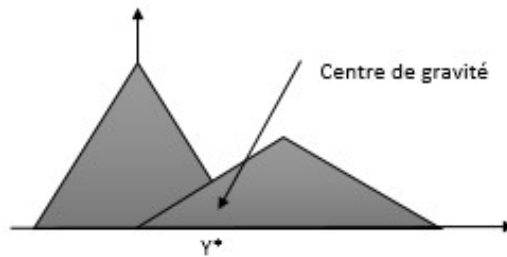
Les Stratégies de défuzzification les plus couramment employées :

### 3.7.4.1 La méthode du centre de gravité

Le défuzzifier du centre de gravité spécifie  $y^*$  comme centre de la zone couverte par la fonction d'appartenance de B'. Peut être déterminé à l'aide de la relation suivante :

$$y^* = \frac{\int y\mu_{B'}(y)dy}{\int \mu_{B'}(y)dy} \quad (3.11)$$





**Fig. 3.7** – représentation de centre de gravité

On utilise cette méthode généralement presque toujours en commande floue à cause de sa simplicité de calculs et sa sortie unique(26).

#### 3.7.4.2 Méthode du maximum

La méthode du maximum simple, rapide et facile mais elle introduit des ambiguïtés et une discontinuité de la sortie .la commande en sortie est égale à la commande ayant la fonction d'appartenance maximale (26).

#### 3.7.4.3 Méthode de la moyenne des maxima

Elle considère, comme valeur de sortie, la moyenne de toutes les valeurs pour lesquelles la fonction d'appartenance issue de l'inférence est maximale. est plutôt employée lorsqu'il s'agit de discriminer une valeur de sortie(26)

Comparaison entre les deux méthodes maximum et centre de gravité.

Méthode	Centre de gravité	Maximum
plausibilité	oui	oui
simplicité de calcul	Non	oui
continuité	oui	Non

TABLE 3.1 – Comparaison entre les deux méthodes

## 3.8 Domaines d'application

Les domaines d'application de la logique floue sont multiples mais on résumer quelques :

Automatisme :

- Automatisme dans les appareils grand public ;
- Automatisme industriels ;
- Automatisme d'entreprise de transport ;

Informatique :

- Programmations et développement ;
- Intelligence artificielle et systèmes experts ;

Mathématiques appliquées :

- Recherche opérationnelle ;
- Statistique ;

## 3.9 Les avantages et les inconvénients

**Les avantages**

- Le non nécessité d'une modélisation mathématique rigoureuse du processus ;
- La maîtrise du procédé avec un comportement complexe ;
- La possibilité d'implanter des connaissances (linguistiques) de l'opérateur de processus ;
- La simplicité de définition et de conception.
- Robustesse vis-à-vis les incertitudes de modélisation.

**Les inconvénients**

- Aucunes méthodes formelles pour l'ajustement ;
- L'approche artisanale et non systématique ;
- La difficulté de montrer la stabilité dans tous les cas ;
- Il n'existe pas de théorie générale qui caractérise rigoureusement la stabilité, la robustesse, . . . etc

### **3.10 conclusion**

Dans ce chapitre nous avons présenté les principes théoriques des techniques de la commande intelligente basés sur la logique floue. tout d'abord on a donné quelques définitions bref, après on a présenté la structure générale de commande floue et ces étapes(avec les méthodes les plus connus) .et on a terminé par un ensembles des domaines d'application.  
la simulation de notre robot sera montrer dans le chapitre suivant.

# Chapitre 4

## simulation d'un robot anthropomorphe à deux degrés de liberté

### 4.1 introduction

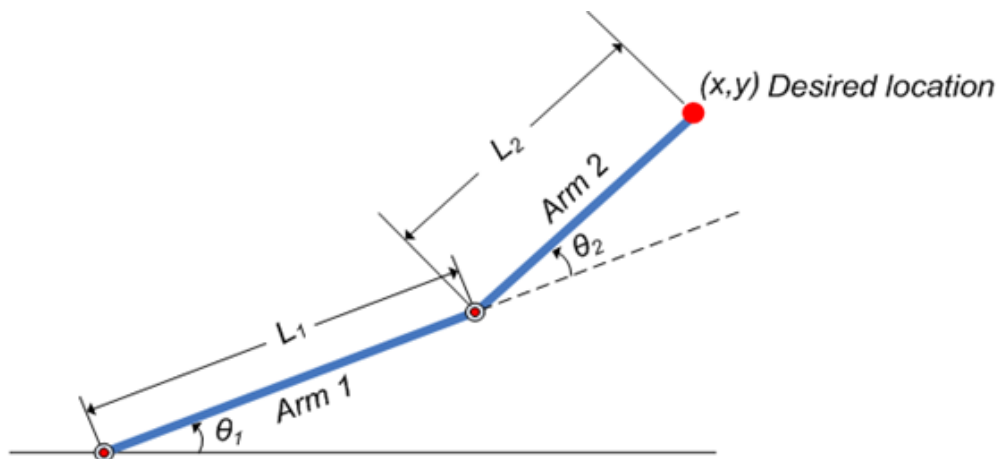
*MATLAB* (*MATrixLABoratory*) est un outil très flexible et outil de programmation simple. Mais il peut également être utilisé comme langage de programmation de haut niveau. *MATLAB* est notre choix car il offre des avantages importants en comparaison avec d'autres langages de programmation. Ce produit MathWorks™ contient un noyau général et des boîtes à outils pour les applications spécialisées. un débutant devrait commencer par le noyau. Comme déjà mentionné, la langue est facile à apprendre et à utiliser, mais il offre des instructions de contrôle flou, des fonctions, des données structures, instructions d'entrée / sortie et autres fonctionnalités. La bibliothèque de fonctions mathématiques fournit un grand ensemble de fonctions pour une large gamme de algorithmes. L'interface utilisateur graphique *MATLAB GUI* est également très bon et les fonctions correspondantes sont faciles à utiliser. Il est également possible de écrire des programmes C qui interagissent avec le code *MATLAB*.

Dans ce chapitre on va donner la simulation de notre robot dans *Matlab* à l'aide de la géométrie inverse et la logique floue ,on va définir d'abord la géométrie inverse , l'intérêt de la logique floue , en suit on va donner une présentation de la solution floue qui est représenté en réseaux ANFIS pour contrôler notre bras de robot.

## 4.2 la géométrie inverse

Dans un bras robotisé à deux articulations, étant donné les angles des articulations, les équations géométriques donnent l'emplacement de la pointe du bras. la géométrie inverse fait référence au processus inverse. Étant donné un emplacement souhaité pour la pointe du bras robotique, quels devraient être les angles des articulations afin de localiser la pointe du bras à l'emplacement souhaité. Il existe généralement plus d'une solution et peut parfois être un problème difficile à résoudre.

Il s'agit d'un problème typique en robotique qui doit être résolu pour contrôler un bras robotique afin d'exécuter les tâches pour lesquelles il est désigné. Dans un espace d'entrée bidimensionnel, avec un bras robotisé à deux articulations et étant donné la coordonnée souhaitée, le problème se réduit à trouver les deux angles impliqués. Le premier angle est entre le premier bras et le sol (ou tout ce à quoi il est attaché). Le deuxième angle est entre le premier bras et le deuxième bras.



**Fig. 4.1** – illustration montrant le bras robotique à deux articulations avec les deux angles,  $\theta_1$  et  $\theta_2$

## 4.3 Présentation de la solution floue

Les formules géométriques du bras robotique à deux articulations étant connues avant, les coordonnées  $x$  et  $y$  de la pointe du bras sont déduites pour toute la plage d'angles de rotation des deux articulations. Les coordonnées et les angles sont sauvegardés pour être utilisés comme données d'apprentissage pour former un réseau ANFIS (Adaptive Neuro-Fuzzy Inference System).

Lors de la formation, le réseau ANFIS apprend à mapper les coordonnées  $(x, y)$  aux angles  $(\theta_1, \theta_2)$ . Le réseau ANFIS formé est ensuite utilisé dans le cadre d'un système de contrôle plus large pour contrôler le bras robotique. Connaissant l'emplacement souhaité du bras robotique, le système de commande utilise le réseau ANFIS formé pour déduire les positions angulaires des articulations et applique une force sur les articulations du bras robotique en conséquence pour le déplacer vers l'emplacement souhaité.

### 4.3.1 Définition d'ANFIS

ANFIS est synonyme de système d'inférence neuro-floue adaptatif. Il s'agit d'une technique hybride neuro-floue qui apporte les capacités d'apprentissage des réseaux de neurones aux systèmes d'inférence floue. L'algorithme d'apprentissage ajuste les fonctions d'appartenance d'un système d'inférence floue de type Sugeno en utilisant les données d'entrée / sortie d'apprentissage.

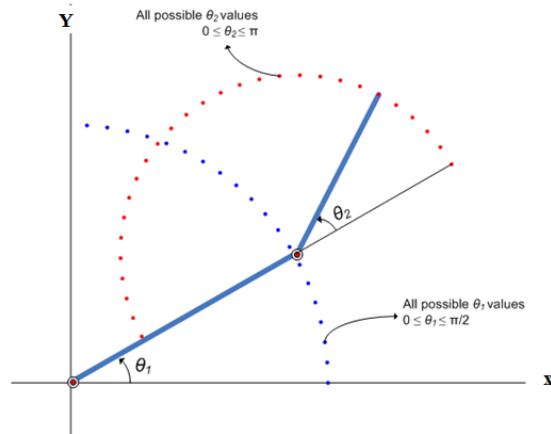
Dans ce cas, les données d'entrée / sortie se réfèrent au jeu des données «coordonnées / angles». Les coordonnées agissent comme entrée pour l'ANFIS et les angles agissent comme sortie. L'algorithme d'apprentissage apprend à l'ANFIS à mapper les coordonnées aux angles grâce à un processus appelé apprentissage. À la fin de la formation, le réseau ANFIS formé aurait appris la carte entrées-sorties et serait prêt à être déployé dans la solution de système de contrôle plus large.

### 4.3.2 Data génération

Soit  $\theta_1$  l'angle entre le premier bras et le sol. Soit  $\theta_2$  l'angle entre le deuxième bras et le premier bras (reportez-vous à la figure 4.2 pour l'illustration). Soit la longueur du premier bras  $l_1$  et celle du second bras  $l_2$ .

Supposons que la première articulation a une liberté de rotation limitée et qu'elle peut pivoter entre 0 et 90 degrés. De même, supposons que la deuxième articulation a une liberté de rotation limitée et peut tourner entre 0 et 180 degrés. (Cette hypothèse supprime la nécessité de traiter certains cas particuliers qui confondront le discours).

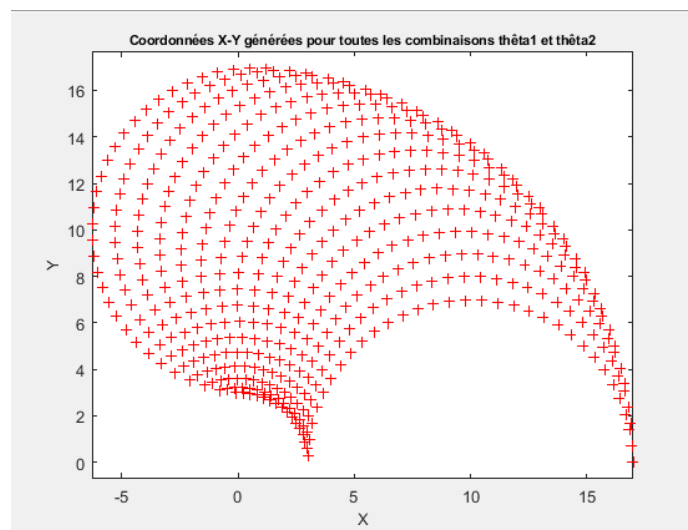
Par conséquent,  $0 \leq \theta_1 \leq \pi/2$  et  $0 \leq \theta_2 \leq \pi$ .



**Fig. 4.2** – illustration montrant toutes les valeurs possibles de  $\theta_1$  et  $\theta_2$

Maintenant, pour chaque combinaison de valeurs  $\theta_1$  et  $\theta_2$ , les coordonnées  $x$  et  $y$  sont déduites à l'aide de formules de la géométrie directe.

Le graphique suivant montre tous les points de données X-Y générés en parcourant différentes combinaisons de  $\theta_1$  et  $\theta_2$  et en déduisant les coordonnées  $x$  et  $y$  pour chacun. Le graphique est illustré plus loin pour une meilleure compréhension.



**Fig. 4.3** – Coordonnées X-Y générées pour toutes les combinaisons  $\theta_1$  et  $\theta_2$  à l'aide de formules géométriques directes

### 4.3.3 Construire des réseaux ANFIS

Une approche pour construire une solution ANFIS pour ce problème, est de construire deux réseaux ANFIS, l'un pour prédire  $\theta_1$  et l'autre pour prédire  $\theta_2$ .

Pour que les réseaux ANFIS puissent prédire les angles, ils doivent être entraînés avec des échantillons de données d'entrée-sortie. Le premier réseau ANFIS sera formé avec les coordonnées X et Y en entrée et les valeurs  $\theta_1$  correspondantes en sortie. La matrice `data1` contient l'ensemble de données x-y- $\theta_1$  requis pour former le premier réseau ANFIS. Par conséquent, `data1` sera utilisé comme ensemble de données pour former le premier réseau ANFIS.

De même, le deuxième réseau ANFIS sera formé avec les coordonnées X et Y en entrée et les valeurs  $\theta_2$  correspondantes en sortie. La matrice `data2` contient l'ensemble de données x-y- $\theta_2$  nécessaire pour former le deuxième réseau ANFIS. Par conséquent, `data2` sera utilisé comme ensemble de données pour former le deuxième réseau ANFIS.

Pour former un réseau ANFIS, spécifiez d'abord les options de formation à l'aide de la commande `anfisOptions`. Pour ce travail, spécifiez un objet FIS avec 7 fonctions d'appartenances pour chaque variable d'entrée. Entraînez le système pour 150 époques et supprimez l'affichage de la fenêtre de commande des informations d'entraînement.

```
opt = anfisOptions;
opt.InitialFIS = 7;
opt.EpochNumber = 150;
opt.DisplayANFISInformation = 0;
opt.DisplayErrorValues = 0;
opt.DisplayStepSize = 0;
opt.DisplayFinalResults = 0;
```

Une fois l'apprentissage terminé, les deux réseaux ANFIS ont appris à approcher les angles ( $\theta_1$ ,  $\theta_2$ ) en fonction des coordonnées (x, y). Un avantage de l'utilisation de l'approche floue est que le réseau ANFIS peut maintenant approximer les angles pour des coordonnées qui sont similaires mais pas exactement les mêmes que celles avec lesquelles il a été formé. Par exemple, les réseaux ANFIS formés sont désormais capables d'approximer les angles des coordonnées situées entre deux points inclus dans l'ensemble de données d'apprentissage. Cela permettra au contrôleur final de déplacer le bras en douceur dans l'espace d'entrée.



Nous avons maintenant deux réseaux ANFIS formés qui sont prêts à être déployés dans le système plus grand qui utilisera ces réseaux pour contrôler le bras robotique.

#### 4.3.4 Validation des réseaux ANFIS

Après avoir formé les réseaux, une étape de suivi importante consiste à valider les réseaux pour déterminer dans quelle mesure les réseaux ANFIS fonctionneraient à l'intérieur du système de contrôle plus large. Puisque ce problème concerne un bras robotique à deux articulations dont les formules de géométrie inverse peuvent être dérivées, il est possible de tester les réponses que les réseaux ANFIS produisent avec les réponses des formules dérivées. Supposons qu'il soit important pour les réseaux ANFIS d'avoir de faibles erreurs dans la plage de fonctionnement  $0 < x < 2$  et  $8 < y < 10$ . Les valeurs  $\theta_1$  et  $\theta_2$  sont déduites mathématiquement des coordonnées  $x$  et  $y$  à l'aide de formules de géométrie inverse.

```
[X,Y] = meshgrid(x,y);
c2 = (X.^2 + Y.^2 - l1^2 - l2^2)/(2*l1*l2);
s2 = sqrt(1 - c2.^2);
THETA2D = atan2(s2,c2); % theta2 est déduit

k1 = l1 + l2.*c2;
k2 = l2*s2;
THETA1D = atan2(Y,X) - atan2(k2,k1); % theta1 est déduit

XY = [X(:) Y(:)];
THETA1P = evalfis(anfis1,XY); % theta1 prédit par anfis1
THETA2P = evalfis(anfis2,XY); % theta2 prédit par anfis2

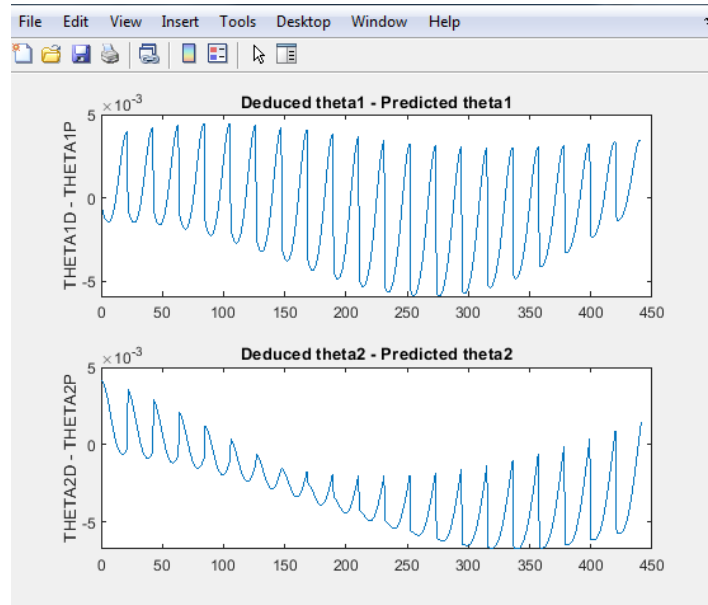
theta1diff = THETA1D(:) - THETA1P;
theta2diff = THETA2D(:) - THETA2P;
figure
subplot(2,1,1);
plot(theta1diff);
ylabel('THETA1D - THETA1P','fontsize',10)
title('Deduced theta1 - Predicted theta1','fontsize',10)

subplot(2,1,2);
plot(theta2diff);
ylabel('THETA2D - THETA2P','fontsize',10)
title('Deduced theta2 - Predicted theta2','fontsize',10)
```

THETA1D et THETA2D sont les variables qui contiennent les valeurs de  $\theta_1$  et  $\theta_2$  déduites à l'aide des formules de géométrie inverse. Les valeurs  $\theta_1$  et  $\theta_2$  prédites par les réseaux anfis formés sont obtenues en utilisant la commande evalfis qui évalue un FIS pour les entrées données.

Ici, evalfis est utilisé pour trouver les sorties FIS pour les mêmes valeurs x-y utilisées précédemment dans les formules de géométrie inverse.

Maintenant, nous pouvons voir à quel point les sorties FIS sont proches des valeurs déduites.



**Fig. 4.4** – les plages d’erreurs des angles.

Les erreurs sont dans la plage  $1e^{-3}$ , ce qui est un nombre assez bon pour l’application dans laquelle il est utilisé. Cependant, cela peut ne pas être acceptable pour une autre application, auquel cas les paramètres de la fonction anfis peuvent être modifiés jusqu’à ce qu’une solution acceptable est arrivé à. En outre, d’autres techniques telles que la sélection des entrées et d’autres moyens de modéliser le problème peuvent être explorés.

### 4.3.5 Construire une solution autour des réseaux ANFIS formés

Maintenant donné une tâche spécifique, telle que des robots ramassant un objet dans une chaîne de montage, le système de contrôle plus grand utilisera les réseaux ANFIS formés comme référence, un peu comme une table de consultation, pour déterminer quels doivent être les angles des bras, étant donné un emplacement souhaité pour la pointe du bras. Connaissant les angles souhaités et les angles courants des articulations, le système appliquera une force appropriée sur les articulations des bras pour les déplacer vers l’emplacement souhaité.

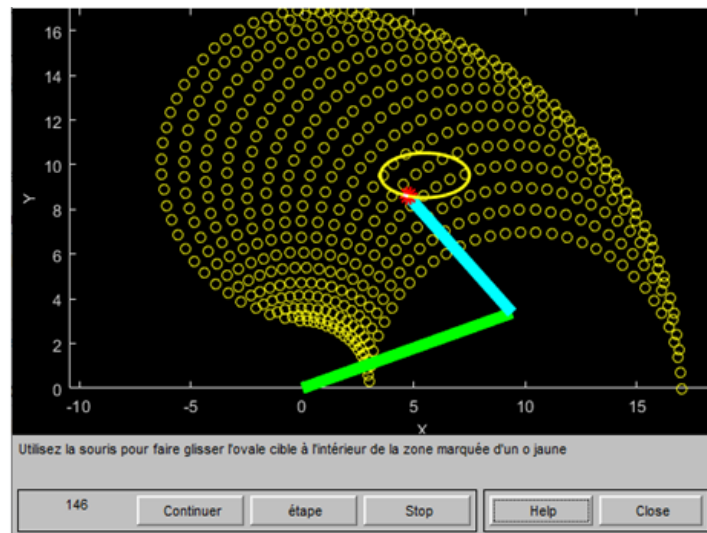


Fig. 4.5 – : GUI pour la modélisation géométrique inverse.

Les deux réseaux ANFIS utilisés dans l'exemple ont été pré-entraînés et sont déployés dans un système plus grand qui contrôle la pointe du bras du robot à deux articulations pour tracer une ellipse dans l'espace d'entrée. L'ellipse à tracer peut être déplacée. Déplacez l'ellipse vers un emplacement légèrement différent et observez comment le système répond en déplaçant la pointe du bras robotique de son emplacement actuel vers le point le plus proche du nouvel emplacement de l'ellipse. Observez également que le système répond en douceur tant que l'ellipse à tracer se trouve dans les points marqués «o» qui représentent la grille de données utilisée pour former les réseaux. Une fois que l'ellipse est déplacée hors de la plage de données avec laquelle elle a été entraînée, les réseaux ANFIS réagissent de manière imprévisible. Cela souligne l'importance de disposer de données pertinentes et représentatives pour la formation. Les données doivent être générées en fonction de la plage de fonctionnement prévue pour éviter de tels problèmes d'imprévisibilité et d'instabilité.

## 4.4 conclusion

Dans ce chapitre nous avons donné la simulation de notre robots à l'aide de la géométrie inverse et les réseaux ANFIS ,nous avons données l'intérêt de la logique floue ,la définition de l'anfis, la construction et la validation de deux réseaux anfis et cela a pour but de déduire les positions angulaires des articulation et applique une force sur les articulations du bras robotique en conséquence pour le déplacer vers l'emplacement souhaité.

# Chapitre 5

## implémentation de la logique floue

### 5.1 Introduction

Les systèmes embarqués sont composés d'une partie logiciel « Software » qui contient les logiciels de programmation et d'une partie Matérielle« Hardware » qui contient les composants nécessaires pour la réalisation des tâches .

Dans ce chapitre,tout d'abord nous allons présenter la partie hard qui contient les deux composants électroniques nécessaires pour la réalisation et la conception de notre projet (robot manipulateur à 2dd) sont : l'arduino Uno qui est utilisé pour commander le robot,et le servomoteur SG90 qui permet de faire les rotations des bras ,ensuite on va expliquer la partie soft (l'installation de Matlab support package pour Arduino hardware ) et en fin on va terminer ce chapitre par l'implémentation de la logique floue sur ce robot à l'aide de Matlab.

### 5.2 Partie matériels( Partie Hard)

Pour construire notre projet on a besoin d'un ensemble des composants sont :les pièces des tâches de robot, une carte *Arduino UNO*, 2 servomoteurs de type SG90, câble *USB* et un ensemble des fils de connections.

#### 5.2.1 présentation générale de l'Arduino

Arduino est un projet créé par des passionnés en Italie (2005) pour des étudiants.Il existe plusieurs de types,toutes sont différentes les une au autres parmi les plus utilisées :Les cartes Mega ,Uno ,Duemilanove,Leonardo, NANO et la carte Mega Adk. Pour notre projet nous avons choisis

d'utiliser la carte portant le nom de « UNO »(31).

### 5.2.1.1 définition de l'Arduino UNO

C'est un circuit électronique programmable construit d'un microcontrôleur et des composants minimum pour réaliser des fonctions plus ou moins évoluées à bas coût. Elle possède une interface USB pour la programmer. Cette carte est une plat forme open source peut être programmée directement avec « l'Arduino software ».(27)



Fig. 5.1 – carte Arduino UNO

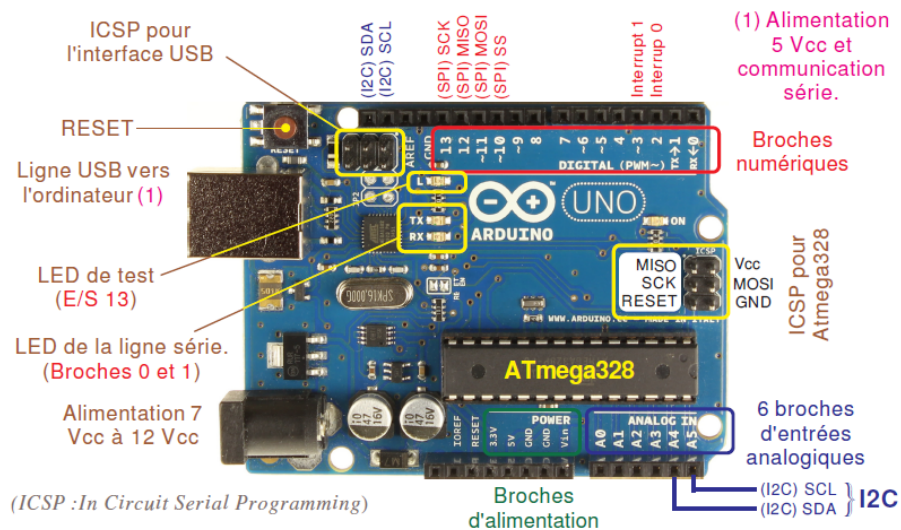


Fig. 5.2 – les composants du carte Arduino UNO

On utilise la carte Arduino pour développer des applications matérielles industrielles légères ou des objets interactifs.

### 5.2.1.2 Caractéristiques d'Arduino UNO

Microcontrôleur	Atmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM)
Intensité maxi disponible par broche E/S	(5V) 40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée – 500 mA max si port USB utilisé seul
Mémoire Programme Flash	32 KB (Atmega328) dont 0.5 KB sont utilisés par le bootloader
Mémoire SRAM (mémoire volatile)	2 KB (Atmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (Atmega328)
Vitesse d'horloge	16 MHz

TABLE 5.1 – les caractéristiques d'Arduino UNO(28)

### 5.2.1.3 Alimentation d'UNO

On peut alimenter La carte Arduino UNO par l'USB avec l'ordinateur ou par une alimentation externe de 7 à 12 volts (peut venir soit un adaptateur AC-DC ou un pile). La source est sélectionnée automatiquement.

### 5.2.1.4 Fonctionnement de l'Arduino UNO

le principe de fonctionnement est simple :

1. On réalise le programme sur un ordinateur.
2. On connecte l'ordinateur à l'Arduino via une prise USB.
3. On envoie le programme sur l'Arduino.
4. L'Arduino exécute enfin le programme de manière autonome.

## 5.2.2 servomoteur

Un servomoteur (signifie « esclave ») est un ensemble complet de mécanique et d'électronique. Permet de déplacer un bras, sur lequel est fixé un objet, jusqu'à une certaine position (ou angle de rotation) et On peut piloter des rotations avec l'Arduino.

C'est un ensemble mécanique et électronique comprenant :

- d'un moteur à courant continu
- un réducteur en sortie de ce moteur diminuant la vitesse mais augmentant le couple
- d'un axe de rotation
- un capteur de position de l'angle d'orientation de l'axe (très souvent un potentiomètre)
- une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique à trois fils qui permet d'alimenter le moteur et de lui transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion plus communément appelé PWM(29).

il y a plusieurs types de servomoteurs ,mais dans notre projet on a choisi le servomoteur SG90.

### 5.2.2.1 servomoteur SG90

Minuscule et léger avec une puissance de sortie élevée. Le servomoteur peut pivoter d'environ 180 degrés (90 dans chaque direction), et fonctionne comme les types standard avec un format plut petit. Vous pouvez utiliser n'importe quel code dédié aux servomoteurs,matériel ou bibliothèque pour contrôler ce type de moteur(30).



Fig. 5.3 – servomoteur SG90

Le SG90 dispose de trois fils :

- Marron ( Masse ou GND )
- Rouge ( Alimentation 5V)
- Orange ( le signal de commande PWM )

### 5.2.2.2 Caractéristique de servomoteur SG90

Dimensions	22 x 11.5 x 27 mm
Poids	9 gr
Tension d'alimentation	4.8v à 6v
Vitesse	0.12 s / 60° sous 4.8v
Couple	1.2 Kg / cm sous 4.8v
Amplitude	de 0 'a 180°

TABLE 5.2 – les caractéristiques de servomoteur

### 5.2.3 cabale USB

câble USB est utilisé pour transférer des données entre les périphériques et les ordinateurs. Elle est la norme universellement acceptée de l'intégration de l'ordinateur avec de nombreux appareils électroniques grand public.

## 5.3 partie logiciel (partie soft )

Il y a plusieurs logiciels pour programmer l'Arduino ; l'Arduino IDE est le plus utilisé, ainsi, il est disponible sur le site officiel pour télécharger gratuitement ([www.arduino.cc](http://www.arduino.cc)), Il existe d'autres programmes, parmi lesquels LabView, Atmel Studio et Matlab, pour notre projet on utilise Matlab pour programmer l'Arduino.

### 5.3.1 Matlab support package for Arduino Hardware

Le Support Package for Arduino Hardware permet à l'utilisateur d'utiliser MATLAB pour communiquer avec la carte Arduino via un câble USB. Ce package est basé sur un programme serveur exécuté sur la carte, qui écoute les commandes arrivant via le port série, exécute les commandes et, si nécessaire, renvoie un résultat. Ce package de support prend en charge le matériel

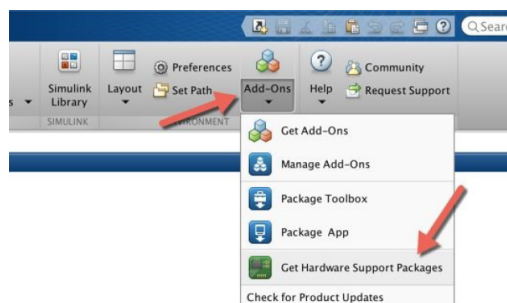


Arduino Uno, Arduino Mega 2560 et Arduino Due. De plus, cette approche aide l'utilisateur :

- Commencez immédiatement la programmation sans boîtes à outils supplémentaires.
  - Travailler dans MATLAB pour le développement et le débogage interactifs.
  - Développer de manière interactive des programmes pour acquérir des données analogiques et numériques et pour contrôler les moteurs à courant continu, servomoteurs et moteur pas à pas.
- Les principaux avantages de l'utilisation de MATLAB pour la programmation Arduino sont :
- Lire et écrire les données du capteur de manière interactive sans attendre la compilation du code.
  - Analysez les données du capteur à l'aide de milliers de fonctions prédéfinies pour le traitement du signal, l'apprentissage automatique, la modélisation mathématique, etc.
  - Visualisez rapidement les données à l'aide de la vaste gamme de types de tracés de MATLAB.

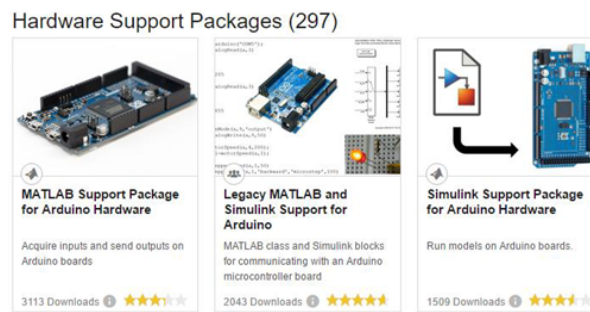
### 5.3.2 l'installation de MATLAB Support Package for Arduino Hardware

1. Créez un compte dans MATLAB fonctionne « mathworks » <http://www.mathworks.com/>.
2. Installez Matlab support package for arduino :
  - a) Dans MATLAB, sélectionnez HOME > Add-Ons > Get Hardware Support Packages (Fig5.4).



**Fig. 5.4** – première étape d'installation du Matlab support package

- b) La fenêtre Add-On Explorer apparaîtra, cliquez sur Matlab Support Packages for Arduino Hardware (Fig5.5).



**Fig. 5.5** – deuxième étape d’installation du Matlab support package

- c) Cliquez sur le bouton d’installation et sélectionnez l’option d’installation .
- d) Connectez-vous avec le compte précédemment créé.
- e) Acceptez le contrat de licence, puis cliquez sur le bouton Suivant dans la fenêtre le premier logiciel. Attendez que tous les packages soient téléchargés et installés. Désactivez la case Ouvrir des exemples et cliquez sur Terminer.

## 5.4 Transformation du programme vers la carte Arduino

Sur la plupart des cartes, vous devez voir les LEDs des lignes RX et TX clignoter rapidement, témoignant que le programme est bien transféré. Durant le transfert, le bouton devient jaune et le logiciel Matlab affiche un message indiquant que le transfert est en cours.

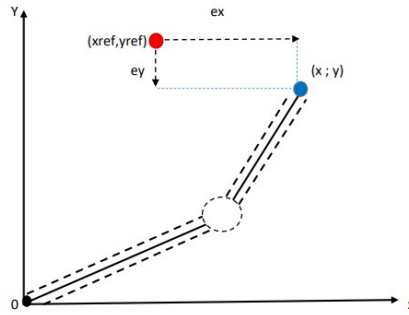
## 5.5 L’implémentation de la logique floue dans notre robot

Parmi les nombreuses applications de la logique floue, la commande floue s’avère être le champ d’application le plus actif à travers le monde.

### 5.5.1 concept d’un contrôleur flou

Pour atteindre l’objectif voulu c’est-à-dire la commande du robot manipulateur à 2ddl on va utiliser la logique floue (fuzzy).

On prend l’exemple de notre robot ayant 2 entrées l’erreur X ( $eX$ ) et l’erreur Y ( $eY$ ) et 2 sorties Ang11 et Ang22.



**Fig. 5.6** – Robot Anthropomorphe à 2DDL

### Les données de notre graphe :

$X_{ref}$  : le coordonnée de l'objet sur l'axe « x ».

$Y_{ref}$  : le coordonnée de l'objet sur l'axe « y ».

$X$  : le coordonnée de l'angle  $\theta$  sur l'axe « x ».

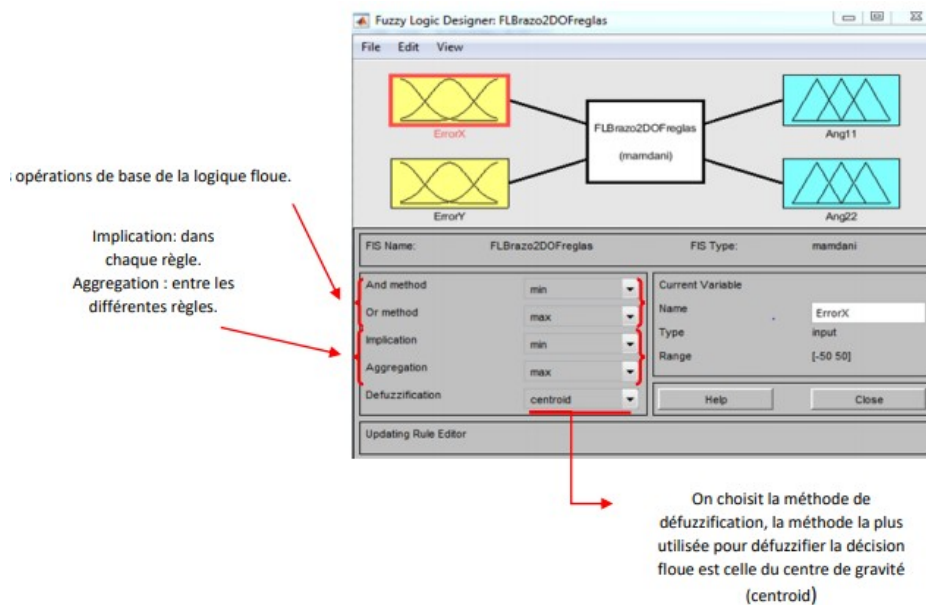
$Y$  : le coordonnée de l'angle  $\theta$  sur l'axe « y »

$Ex$  : l'erreur x.

$Ey$  : l'erreur y.

Telle que :

$Ex = X_{ref} - X$  et  $Ey = Y_{ref} - Y$ .



**Fig. 5.7** – Définition de la structure globale (entrée, sortie, type d'opérateurs).

- Le mot Aggregation (en français Agrégation) veut dire : rassemblement, collecte.
- Le mot Inference (en français Inférence) vient du verbe : inférer qui veut dire : tirer une conséquence de quelque chose, conclure, déduire.

### 5.5.2 les étapes de commande floue

Etape1 : Fuzzification des variables d'entrée

- Définition des fonctions d'appartenance de tout les variables d'entrée .
- passage : transformation les grandeurs physiques à des grandeurs flous ( variables linguistiques).

#### Les sous ensembles floue des différents variables linguistiques

- La variable d'entrée erreur X « eX » entre -50 et 50 à des valeurs linguistiques telque :  
T-ex = eXmN, eXN, eXnull, eXP, eXmP

- La variable d'entrée erreur Y « eY » entre -50 et 50 à des valeurs linguistiques tel que :  
T-ey = eYmN, eYN, eYnull, eYP, eYmP

**Les noms de sous ensembles sont**

<b>L'erreur X</b>	<b>Le Nom</b>
eXN	erreur X Négatif
eXnull	erreur X null
eXmP	erreur X max Positif
eXP	erreur X Positif
eXmN	erreur X max Négatif
<b>L'erreur Y</b>	<b>Le Nom</b>
eYmN	erreur Y max Négatif
eYN	erreur Y Négatif
eYnull	erreur Y null
eYmP	erreur Y max Positif
eYP	erreur Y Positif

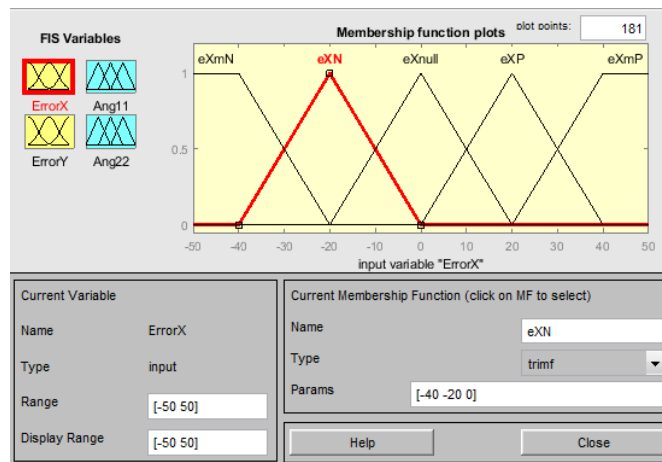
TABLE 5.3 – Les noms des sous ensembles

- La variable de sortie Ang11 entre -20 et 20 à des valeurs linguistiques tels que :  
Ang11=n17,n15,n14,n10,null,p5,p7,p10,p14,p15,p17

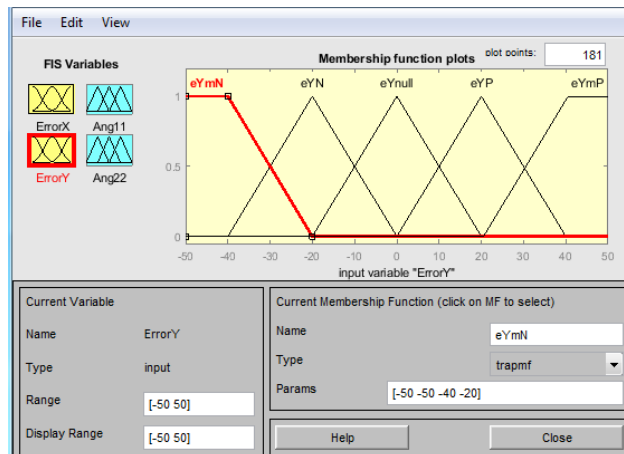
- La variable de sortie Ang22 entre -20 et 20 à des valeurs linguistiques tels que :  
 $Ang22 = n17, n15, n14, n10, null, p5, p7, p10, p14, p15, p17$

### Le choix de fonction d'appartenance

Pour les variables d'entrées les fonctions d'appartenances associées aux sous ensembles flous sont triangulaires et trapézoïdales comme montré la Fig.5.8 et la Fig.5.9.



**Fig. 5.8** – Les fonctions d'appartenances de la variable d'entré eX



**Fig. 5.9** – Les fonctions d'appartenances de la variable d'entré eY

Pour les variables de sorties les fonctions d'appartenances associées aux sous ensembles flous sont triangulaires comme montré la Fig.5.10 et la Fig.5.11.

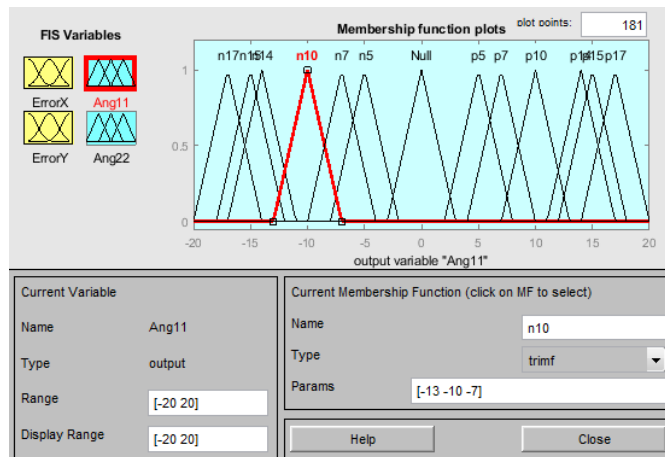


Fig. 5.10 – Les fonctions d'appartenances de la variable de sortie Ang11

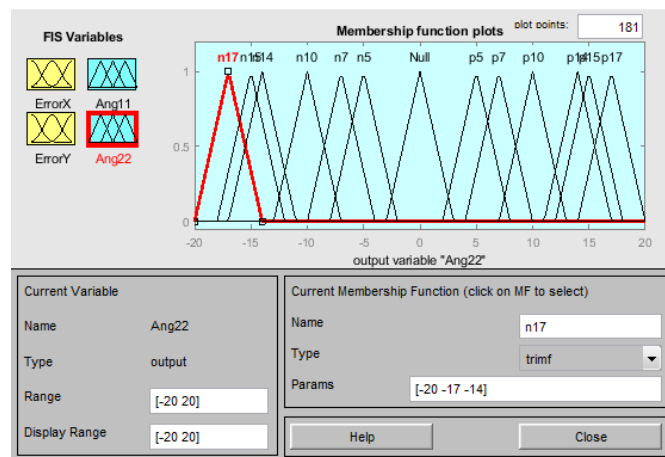


Fig. 5.11 – Les fonctions d'appartenances de la variable de sortie Ang22

Etape2 : inférence(avec la base des règles) :

- prend les décision des règles.

Supposons que l'erreur eX est 30 et l'erreur eY est -20, on va effectuons l'inférence Mamdani pour calculer les angles Ang11 et Ang22. Les règles possibles sont :

1. Si (ErrorX est eXmN) et (ErrorY est eYmP) alors (Ang11 est p15)(Ang22 est p15) (1)
2. Si (ErrorX est eXN) et (ErrorY est eYmP) alors (Ang11 est p14)(Ang22 est p10) (1)
3. Si (ErrorX est eXnull) et (ErrorY est eYmP) alors (Ang11 est p17)(Ang22 est Null) (1)
4. Si (ErrorX est eXP) et (ErrorY est eYmP) alors (Ang11 est p14)(Ang22 est n10) (1)
5. Si (ErrorX est eXmP) et (ErrorY est eYmP) alors (Ang11 est p15)(Ang22 est n15) (1)
6. Si (ErrorX est eXmN) et (ErrorY est eYP) alors (Ang11 est p10)(Ang22 est p14) (1)
7. Si (ErrorX est eXN) et (ErrorY est eYP) alors (Ang11 est p5)(Ang22 est p5) (1)
8. Si (ErrorX est eXnull) et (ErrorY est eYP) alors (Ang11 est p7)(Ang22 est Null) (1)
9. Si (ErrorX est eXP) et (ErrorY est eYP) alors (Ang11 est p5)(Ang22 est n5) (1)
10. Si (ErrorX est eXmP) et (ErrorY est eYP) alors (Ang11 est p10)(Ang22 est n14) (1)
11. Si (ErrorX est eXmN) et (ErrorY est eYnull) alors (Ang11 est Null)(Ang22 est p17) (1)
12. Si (ErrorX est eXN) et (ErrorY est eYnull) alors (Ang11 est Null)(Ang22 est p7) (1)
13. Si (ErrorX est eXnull) et (ErrorY est eYnull) alors (Ang11 est Null)(Ang22 est Null) (1)
14. Si (ErrorX est eXP) et (ErrorY est eYnull) alors (Ang11 est Null)(Ang22 est n7) (1)
15. Si (ErrorX est eXmP) et (ErrorY est eYnull) alors (Ang11 est Null)(Ang22 est n17) (1)
16. Si (ErrorX est eXmN) et (ErrorY est eYN) alors (Ang11 est n10)(Ang22 est p14) (1)
17. Si (ErrorX est eXN) et (ErrorY est eYN) alors (Ang11 est n5)(Ang22 est p5) (1)
18. Si (ErrorX est eXnull) et (ErrorY est eYN) alors (Ang11 est n7)(Ang22 est Null) (1)
19. Si (ErrorX est eXP) et (ErrorY est eYN) alors (Ang11 est n5)(Ang22 est n5) (1)
20. Si (ErrorX est eXmP) et (ErrorY est eYN) alors (Ang11 est n10)(Ang22 est n14) (1)
21. Si (ErrorX est eXmN) et (ErrorY est eYmN) alors (Ang11 est n15)(Ang22 est p15) (1)
22. Si (ErrorX est eXN) et (ErrorY est eYmN) alors (Ang11 est n14)(Ang22 est p10) (1)
23. Si (ErrorX est eXnull) et (ErrorY est eYmN) alors (Ang11 est n17)(Ang22 est Null) (1)
24. Si (ErrorX est eXP) et (ErrorY est eYmN) alors (Ang11 est n14)(Ang22 est n10) (1)
25. Si (ErrorX est eXmP) et (ErrorY est eYmN) alors (Ang11 est n15)(Ang22 est n15) (1)

**Fig. 5.12** – Les règles possibles

Etape3 : Agrégation des sorties des règles :

l'opérateur utilisé pour agréger les règles est de type conjonctif ou disjonctif. Ainsi, en commande, l'implication étant généralement de type conjonctif, cela revient à considérer que les règles sont liées par un opérateur OU. En pratique, on utilise l'opérateur max.

Etape4 : Défuzzification

- transforme les grandeurs flous en valeurs déterminées.

### 5.5.3 L'inférence Mamdani

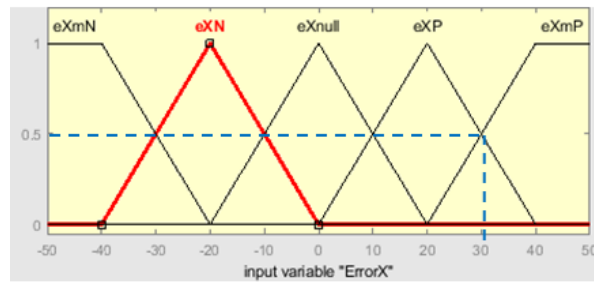
L'inférence Mamdani procède selon quatre étapes :

- **Etape1 :Fuzzification des variables d'entrée :**

pour eX=30

$$\mu_{eXP}(30) = 0.5$$

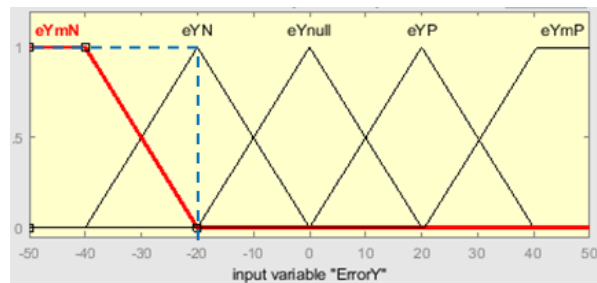
$$\mu_{eXmP}(30) = 0.5$$



**Fig. 5.13** – fuzzification de la variable d'entrée "Erreur X"

pour  $eY = -20$

$$\mu_{eYN}(-20) = 1$$



**Fig. 5.14** – fuzzification de variable d'entrée "Erreur Y"

$$\mu_{eYnull}(-20) = 0$$

$$\mu_{eYmN}(-20) = 0$$

### • Etape2 :inférence des règles

Concerne l'élaboration de base de règle du contrôleur, Cette dernier est obtenu à partir de l'expérience et des connaissances acquises sur le système à commander.

Dans cette étape, il s'agit d'établir les relations qui lient les variables de sorties aux variables d'entrées. Ces relations appelées les règles de type Si-ALORS.

**la règle 1** :Si (Erreur X est  $eXP = 0.5$ ) et (Erreur Y est  $eYN=1$ ) Alors (Ang11 est  $n5$ )(Ang22 est  $n5$ )



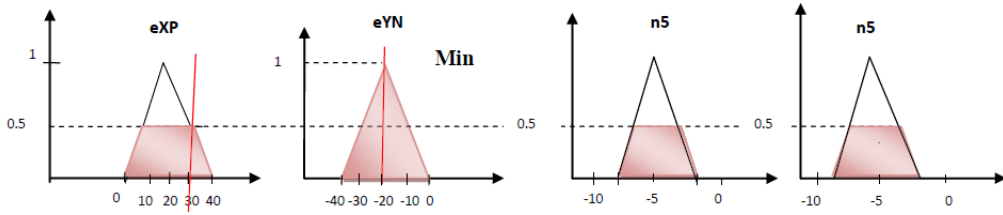


Fig. 5.15 – les résultats de règle « 1 » aux sorties

**la règle 2** :Si (Erreur X est  $eXmP = 0.5$ ) et (Erreur Y est  $eYN = 1$ ) alors (Ang11 est  $n10$ )(Ang22 est  $n10$ )

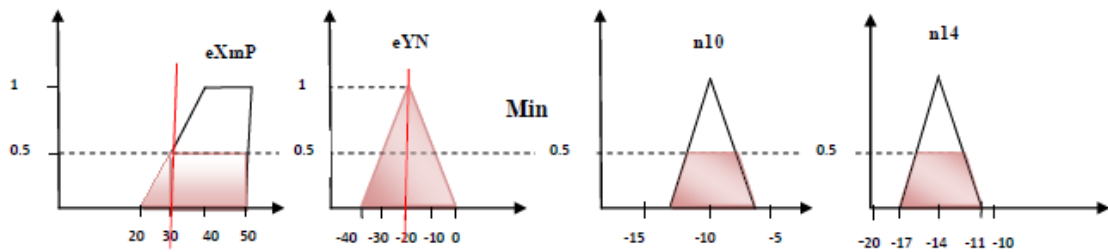


Fig. 5.16 – les résultats de règle « 2 » aux sorties

### Etape3 :l'agrégation

L'agrégation est le processus d'unification des sorties de toutes les règles floues, en d'autres termes, on assemble les fonctions d'appartenances des résultats des 2 règles précédentes ( $n5+n10$ ) et ( $n5+n14$ ) multiplié par leur degrés d'appartenances c'est-à-dire [ $(n5+n10)*0.5$  pour Ang11 et  $(n5+n14)*0.5$  pour Ang22], on obtient alors un seul sous ensemble à la sortie 1 et un seul sous ensemble à la sortie 2.

les résultats sont illustré dans la Fig.5.17 et Fig.5.18 ci-dessus :

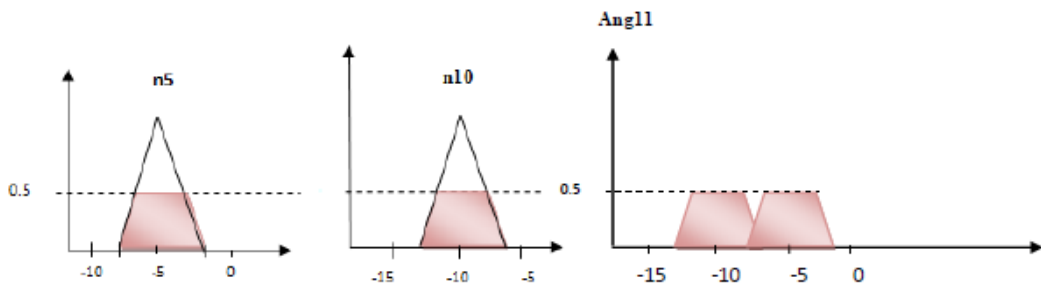


Fig. 5.17 – agrégation des règles de sortie Ang11

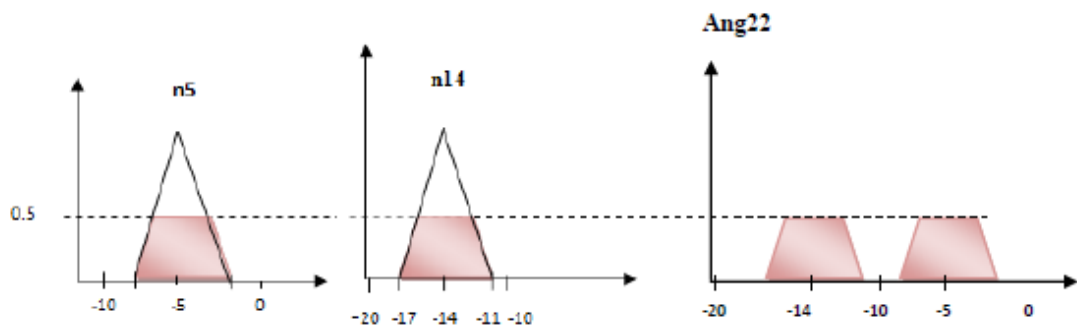


Fig. 5.18 – agrégation des règles de sortie Ang22

selon les règles 1 et 2 le «et » logique dans la méthode de Mamdani se traduit la valeur minimum « Min » entre ces deux degrés d'appartenances (0.5 et 1), les résultats est alors 0.5 pour Ang11 et 0.5 pour Ang22 dans les deux règles.

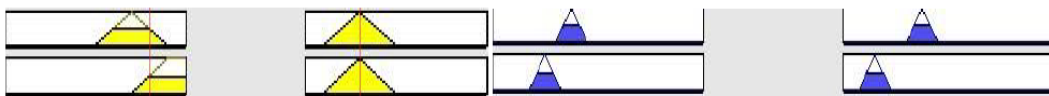


Fig. 5.19 – les résultats des règles 1 et 2 aux sorties sous matlab

#### Etape4 :la Défuzzification

La défuzzification de Mamdani se fait par le calcul du centre de gravité (la méthode du centroïde) de la surface obtenu après l'agrégation. les résultats sont illustré dans la figure ci-dessus :

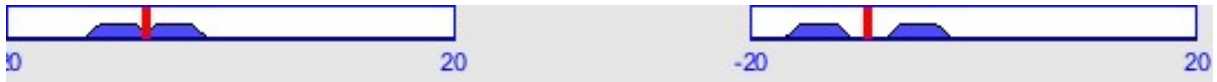


Fig. 5.20 – défuzzification des résultats sur Matlab

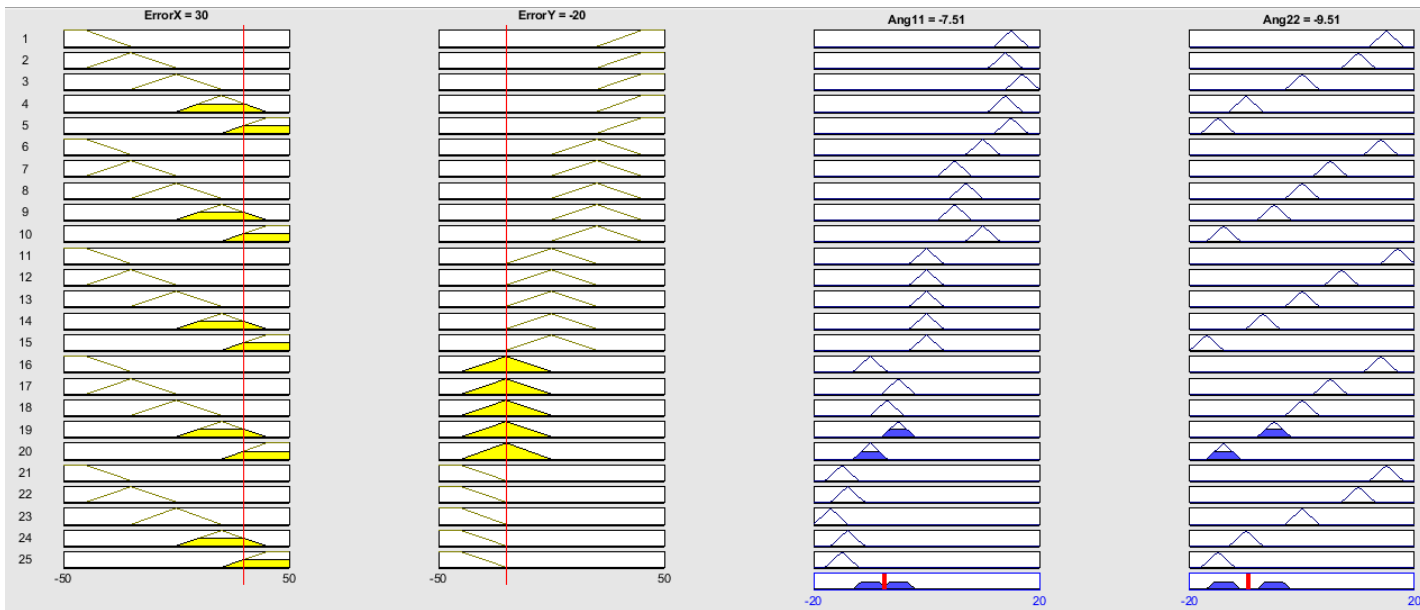
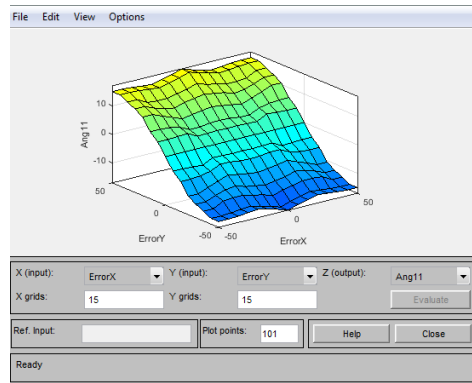


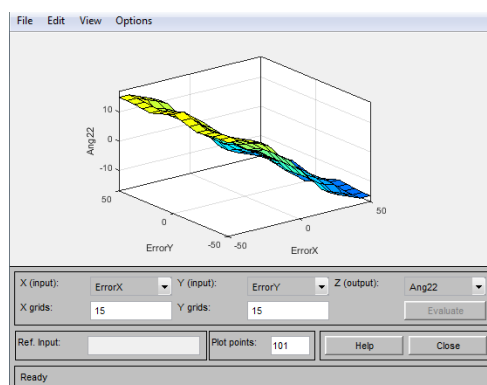
Fig. 5.21 – l'Agrégation et défuzzification de Ex et Ey selon les 25 règles

### La représentation graphique

La surface de décision (contrôle) du contrôleur conçu est donnée sur les figures 5.22 et 5.23. Elles représentent la non linéarité et la souplesse de la commande floue.



**Fig. 5.22** – Observation de la surface1



**Fig. 5.23** – Observation de la surface2

## 5.6 conclusion

Dans ce chapitre nous avons donné une présentation de la partie hard qui contient les deux composants nécessaires pour la réalisation de notre projets qui sont le servomoteur sg90 et l'arduino uno et la partie soft qui contient le logiciel de programmation Matlab, on a donné la définition de l'arduino uno , le servomoteur sg90 et leur caractéristiques ensuite on a donné les étapes d'installation de Matlab Support package for Arduino Hardware .

En fin on a fait une implémentation de la logique floue où on appliqué l'inférence de Mamdani afin de contrôler notre robot.

# conclusion général

L'objectif principal assigné à ce travail est de concevoir et réalisation d'un robot manipulateur à deux degrés de liberté qui est commandé pour le suivi fidèle de trajectoires en position .cette commande basé sur la logique floue.

nous avons privilégié,dans notre mémoire,l'étude des différentes approches de modélisation mathématique du robot manipulateur,trois classes des modèles mathématiques permettent de couvrir l'ensemble des tâches exécuter par les robots industriels,le modèle géométrique(direct et inverse),cinématique(direct et inverse) ,et le modèle dynamique. l'intérêt principal d'utiliser ces modèles réside dans son utilité dans deux domaines : la commande, la conception et la simulation des robots,ou il était question de décrire la modélisation géométrique, dans le but de repérer ,avec précision le robot par rapport a son environnement.

dans la partie de simulation on a utilisé le modèle géométrique inverse et l'ANFIS pour déduire les position angulaires des articulations possibles de l'emplacement du robot et cela pour vérifier le fonctionnement de la commande floue .

A la fin de cette mémoire nous somme terminé notre travail par l'aspect pratique le plus intérêt pour ce projet qui est représenté la réalisation et l'implémentation de robot manipulateur.Le système actionneur utilisé pour notre robot est composé : d'un ensemble de 2 servomoteurs (SG90), qui sont répartis sur le robot, commandé avec la carte d'Arduino UNO, pour faire animer le robot manipulateur et ainsi voir son fonctionnement lorsqu'il effectue sa tache.nous avons appliqué la logique floue (l'inférence de Mamdani) pour assurer la commande de notre robot qui nos a donné une meilleurs résultats.

Ce travail nous a permis de mettre en valeur nos connaissances théoriques et pratiques, que nous avons cumulées pendant toute la durée de nos études, il nous a permis aussi d'apprendre une nouvelle expérience dans la programmation des cartes Arduino avec la commande intelligente. Il est évident que notre travail est loin d'être parfait, mais il peut constituer une base très intéres-

sante pour les promotions à venir qui voudront travailler sur ce type de sujet.

# Bibliographie

- [1] Theveryidea , *history of industrial robots from the first installation until today*, compiled by the international federation of robotics-IFR,Germany, 2012.
- [2] Silvestri Josef, *VEX Robotics : unit 2-introduction to robotics*,2015.
- [3] Jean-LouisBoimond,Cour *ROBOTIQUE*,UniversitéAngers,2008.  
<https://www.ensta-bretagne.fr/jaulin/masrterds-cours-robot-biomond.pdf>
- [4] M.CHERIF, *Robotique industriel de arts et métiers science et technologie*,2019.
- [5] Cour *Généralités sur la robotique- introduction aux structures rigides et flexibles*,2016.  
[thesis.univ-biskra.dz/2421/2/chapitre1.pdf](https://thesis.univ-biskra.dz/2421/2/chapitre1.pdf)
- [6] Bruno Siciliano ,Lorenzo Sciavicco , Luigi Villani and Giuseppe Oriolo , *Robotics Modeling , Planning and Control* , Springer-Verlage, London,2009.
- [7] Laëtitia Matignon , *Introduction à la robotique ,Licence 1ère année*université de Caen ,France,2011/2012.
- [8] Hamdi Hossine , *Introduction à la Robotique : nouvelle édition revue augmentée et corrigée*, Université Constantine,2002-2003
- [9] Michel Parent and Claude Laugeau ,*Robot Technology,LOGIC AND PROGRAMMING*, London,1983.
- [10] Pierre Duysinx and Michel Geradin , *An Introduction To Robotics : Mechanical Aspects* ,university of Liège, Novembre 2004.
- [11] M. Bouri, R. Clavel , Cour *Généralités Robots sériels et parallèles* ,2017.
- [12] Philippe Coiffet and Michel Chirouze , *An Introduction To Robot Technology*, France,1982.
- [13] Shakhathreh Fareed, *The Basics of Robotics*, Lahti university of Applied Sciences Machine-and production technology, 2011.

- 
- [14] Ahmed Joubair, Chargé de cours, ÉTS ,cour *Robots industriels*.
- [15] Abdallah Ghoul , *Modélisation des robots manipulateurs*,19 février 2016 .
- [16] Etienne Dombre and Wisama Khalil ,*Modeling, Performance Analysis and Control of Robot Manipulators* ,France,2007.
- [17] Richard P.Paul , *Robot Manipulators : Mathematics, Programming, and Control* ,ISBN 0-262-16082-X , England,1981.
- [18] C. Daulibe /M .Verleysen /Paul G.A, *Design of Analog Fuzzy Logic Controllers in CMOS Technologies*,2003.
- [19] F. CHEVRIE F. GUÉLY, *Cahier technique n° 191. La logique floue* , mars 1998.
- [20] Équipe de Recherche en Ingénierie des Connaiss AncEs (ERICAE). *La logique floue Benoît Potvin*,décembre 2002.
- [21] Yann MORERE, *Les Réseaux Neuro-Flous*, le 17 mai 2001.
- [22] Saint Germain. *universitaires de France*,1<sup>er</sup> édition 1993,3<sup>ème</sup> édition 1999.
- [23] Chung-Chun Kung, Chia-Chang Liao, *Fuzzy-Sliding Mode Controller Design for Tracking Control of Non-Linear System*,IEEE Proceeding Of the American Control Conference Baltimore, Maryland June 1994 .
- [24] Awad Allah Tayfor Ali, *Fuzzy control systems*.
- [25] MINH TA CAO,*Commande numérique de machines asynchrones par la logique floue*, Thèse de philosophie doctorat faculté des études supérieur de l'université laval 1997.
- [26] K.J.schmusker, *Fuzzy sets,Natural langage computations,and risk analysis* Rockville,1984 .
- [27] Chelly Nizar,Ghared Amine,*commande d'un système thermique à l'aide de la carte Arduino Uno*,Hammamet Mai 2014.
- [28] younsi A.Robot Suiveur de ligne,cour "la carte arduino Uno :EIE LP CONDORCET".
- [29] F.Ferrero,cour *Moteurs et Transistors MOS*.
- [30] DATA SHEET,*SG90 9 g Micro Servo*.
- [31] Z.Med , N Riadh , *conception et réalisation d'un système de commande d'une habitation*,Mémoire de Mester, université de Tizi-Ouzou,2017/2018.



# Dedicace

*Je dédie ce travail ;*

*-Aux les plus chers à mon cœur, ma mère et mon père, ceux qui m'ont aidée du mieux qu'ils pouvaient pour réussir.*

*-A mes adorables sœurs et mes frères qui m'ont toujours encouragés et soutenu moralement.*

*-A toutes ma famille GACEM, et mes amies.*

*-Au promo master 2 en Electrotechnique des systèmes embarqués : cession 2019-2020.*

# Dedicace

*Au nom d'Allah, le Clément et le Miséricordieux.*

*Tout d'abord je tiens à remercier le Dieu Tout Puissant de m'avoir donné le courage et la patience pour arriver à ce stade afin de réaliser ce travail que je dédie :*

*A mes très chers parents, dont le sacrifice, la patience, le soutien, l'aide et les encouragements sont le secret de ma réussite. Sans eux je ne serai pas ce que je suis aujourd'hui.*

*A ma très chère grande mère les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour elle.*

*A mes chères soeurs et mes chers frères pour leurs encouragements permanents, et leur soutien moral.*

*A toutes ma famille "KERNANE", mes ami(es) et mes ami(es) de la promotion Master 2 ESE session 2019/2020 et tous ceux qui ont contribué à ma réussite.*

# Remerciements

Ce document présente les travaux effectués dans le cadre de notre projet de fin d'étude de Master au Département de Génie Électrique de la Faculté de Sciences et de Sciences Appliquées de l'Université Akli Mouhand Oulhadj de Bouira .

Nos remerciements vont premièrement à Dieu tout puissant pour la volonté, la santé et la patience, qu'il nous a donnée durant toutes ces longues années.

Ainsi, nous tenons à exprimer notre gratitude à notre encadreur Monsieur YAHIOU Abd El Ghani et notre co-encadreur Monsieur ZARIFI Adel pour avoir d'abord proposé ce thème, pour suivi continuel tout le long de la réalisation de ce mémoire et qui n'ont pas cessé de nous donner leurs conseils et remarques.

Nous tenons également à adresser nos remerciements à messieurs les membres du jury pour l'honneur qu'ils nous font en participant au jugement de ce travail.

En fin merci à tous ceux qui, de près ou de loin, nous ont aidé et donc ont contribué au succès de ce travail.

# Résumé

Les systèmes non linéaires représentent une grande partie des systèmes étudiés dans l'industrie parmi les quels on trouve les robots manipulateurs, c'est pour cette raison que le développement des lois de commande non linéaire est devenu très intéressant pour l'amélioration des performances.

Notre projet consiste à faire une implémentation de la commande intelligente dans un robot manipulateur à deux degrés de liberté basé sur les concepts de la logique floue , cette technique montre l'amélioration des performances du robot distant du point de vu robustesse, précision de suivi et de stabilité et d'assurer une certaine autonomie au robot.

nous commençons notre mémoire par la présentation d'un aperçu général sur les robots, et nous avons fait une étude détaillée sur les modèles utilisés pour le contrôle et la commande du robot à savoir le modèle géométrique inverse,ensuite on a résumé toutes les notions de base nécessaires à la compréhension du domaine de la logique floue,puis on a consacré à la simulation de notre robot sous Matlab à base de Modèle géométrique inverse et l'ANFIS, après dans la partie de la Réalisation, on a présenté les parties hard et soft et nous avons fait l'implémentation de la logique floue pour commander le robot.

Nous terminons notre mémoire par une conclusion générale de cette étude et nous proposons des perceptives à notre travail.

## Mots-clés

Robot manipulateur 2ddl,Modèle géométrique inverse,La logique floue,Arduino UNO,Matlab.

# Abstract

Nonlinear systems represent a large part of the systems studied in industry among which we find manipulator robots, it is for this reason that the development of nonlinear control laws has become very interesting for the improvement of performance.

Our project consists in making an implementation of the intelligent control in a manipulator robot with two degrees of freedom based on the concepts of fuzzy logic, this technique shows the improvement of the performance of the remote robot from the point of view of robustness, tracking precision and stability and ensure a certain autonomy for the robot.

we start our thesis with the presentation of a general overview on robots, and we made a detailed study on the models used for the control and the command of the robot namely the inverse geometric model, then we have summarized all the basic notions necessary for understanding the field of fuzzy logic, then we devoted to the simulation of our robot under Matlab based on an inverse geometric model and ANFIS, after in the Realization part, we presented the hard and soft parts and we implemented fuzzy logic to control the robot.

We end our work with a general conclusion of this study and we offer perceptive to our work.

## Keywords

Robot manipulator 2ddl, Inverse geometric model, Fuzzy logic, Arduino UNO, Matlab.