

République Algérienne Démocratique et Populaire  
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et d'Informatique  
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes  
de MASTER ACADEMIQUE**  
Spécialité : **commande des systèmes**

*Présenté par*  
**MALLEK Farida**  
**YAHY Zahia**

Mémoire dirigé par **M.SAIDI Khayreddine**

Thème

**Application des algorithmes génétiques  
pour l'optimisation des contrôleurs d'un  
système non linéaire**

*Mémoire soutenu publiquement le 18 septembre 2014 devant le jury composé de :*

**M Ahmed MAIDI**

Grade, Lieu d'exercice, Président

**M Khayreddine SAIDI**

Grade, Lieu d'exercice, Rapporteur

**M Rezki HADOUCHE**

Grade, Lieu d'exercice, Examineur

**M Mouhand Achour TOUAT**

Grade, Lieu d'exercice, Examineur

# *Remerciements*

Nous tenons tout d'abord à exprimer nos sincères remerciements à notre promoteur Monsieur **SAIDI Khayreddine**, qui a accepté de nous encadrer et guider notre travail jusqu'à la réalisation de ce mémoire. Qu'il trouve ici, le témoignage de notre profonde gratitude et notre reconnaissance pour les conseils qu'il n'a cessé de nous prodiguer et pour la confiance qu'il nous a accordé en acceptant de diriger ce mémoire.

Ensuite, les enseignants de la faculté de Génie Electrique pour la qualité de la formation qu'ils nous ont assurée, ainsi que tous les enseignants du département automatique.

Enfin, nos amis (es) : **MOKRANI. Noual**, **KHLIFA AKLI. Amar**, et tous ceux et celles qui ont contribué de près ou de loin pour réaliser ce modeste travail.

# *Dédicaces*

*Au nom du DIEU clément et miséricordieux et que le salut de DIEU soit sur son*

*Prophète MOHAMED.*

*Je dédie ce modeste travail à mon père, ma mère et mes sœurs et mon frère*

**A** ma chère amie et binôme Farida

*Et tous mes amis(es)*

*Souhila en particulier*

**Zahia**



# *Dédicaces*

Au nom du DIEU clément et miséricordieux et que le salut de DIEU soit sur son prophète MOHAMED.

Je dédie ce modeste travail aux personnes qui me sont les plus chères :

La lumière de ma vie et l'espoir de mon existence, source d'affection, de courage et d'inspiration qui a autant sacrifié pour me voir atteindre ce jour...ma chère mère.

Celui qui m'a rendu telle que je suis et telle qu'il souhaite, source du respect, en témoignage de ma profonde reconnaissance pour tous les efforts et le soutien qui m'a toujours apporté...mon cher père.

Mes chers frères et sœurs et toute ma famille.

A ma chère amie et binôme zahia et tous mes amis (es).

***FARIDA MALLEK***

## **Sommaire**

**Introduction Générale.....**

### **Chapitre I**

#### **Etude Et Modélisation Des Bras Manipulateurs**

I.1 Introduction.....	1
I.2 Etude des robots.....	1
I.3 Description géométrique des robots .....	8
I.4 Description cinématique .....	15
I.5 Description dynamique des robots .....	19
I.5.1 Modèle dynamique des robots.....	19
I.5.2 Forme générale des équations dynamiques .....	20
I.6 Conclusion .....	22

### **Chapitre II : Logique Floue Et Algorithmes Génétiques**

II.1 Introduction .....	23
II.2 La logique floue et la commande floue .....	24
II.2.1 Principe de la logique floue .....	24
II.2.2 Les domaines d'application de la logique floue .....	24
II.2.3 Quelques concepts sur la logique floue .....	25
II.3 Réglage par logique floue.....	28
II.4 Conception de contrôleur flou.....	35
II.5 synthèse des contrôleurs flous .....	35
II.5.1 Les systèmes experts.....	35
II.5.2 Régulateur de type Mamdani .....	35
II.5.3 Régulateur flou de type Sugeno .....	36
II.6 Les avantages .....	36
II.7 Détermination des paramètres d'adaptation d'un contrôleur flou en fonction des paramètres classiques .....	37
II.8 Avantages et désavantages du réglage par la logique floue.....	38
II.9 Exemple d'application .....	38
II.9.1 Présentation du système.....	38
II.9.2 Commande du pendule par un proportionnel dérivé flou de type Mamdani.....	40
II.10 Optimisation par les algorithmes génétiques .....	43

## *Sommaire*

---

II. 10.1 Introduction .....	43
II.10.2 Terminologie.....	43
II. 10.3 Principe.....	43
II.10 .4 But.....	44
II.10.5 Applications .....	44
II.10.6 les opérateurs .....	44
II.10.6.1 Codage binaire et réel des variables .....	44
II.10.6.2 Sélection .....	45
II.10.6.3 Croisement .....	46
II.10 6.4 Mutation .....	48
II.10.7 Explications des éléments d'un algorithme génétique .....	50
II. 11 Conclusion.....	52

### **Chapitre III : Simulation Et Résultats**

III.1 Introduction.....	52
III.2 Utilisation de la logique floue: contrôleur en Simulink.....	52
III.3 Exemples d'application de la commande floue.....	53
III.3.1 Commande d'un pendule inversé .....	53
III.3.2 Commande d'un bras manipulateur à 2ddl .....	59
III.4 application de la commande classique sur un bras à 2ddl.....	61
III.5 l'optimisation par les algorithmes génétiques.....	69
III.6 Conclusion .....	74
Conclusion Générale .....	
Références bibliographique.....	

# Introduction Générale

---

Les progrès réalisés dans les différentes disciplines permettent de construire des robots et autres systèmes « mécatroniques » performants et fiables.

Les situations économiques actuelles dans les pays industrialisés notamment la crise qui a suivi le « choc pétrolier » et l'émergence de nouveaux pays industrialisés, conduisent à une compétition industrielle très vive, ne laissant survivre que les industries capables de produire à faible coût ou de fournir rapidement des produits nouveaux et de qualité supérieure et garantie. Ces exigences ont conduit au développement d'une automatisation flexible impliquant l'implantation des robots dans de nombreuses branches d'activités. C'est l'apparition des robots et leur association à d'autres machines automatiques d'inspiration « classique » qui permettent d'automatiser des opérations considérées non automatisables.

L'utilisation de robots est très répandue dans le milieu industriel. Dans plusieurs applications, les robots sont utilisés afin de répéter une tâche prédéterminée. On peut penser par exemple à des robots qui effectuent une soudure ou qui déplacent précisément une pièce d'un endroit donné à un autre le long d'une chaîne de montage. Une fois mis en place et programmés, ils sont alors pratiquement autonomes et fonctionnent sans l'aide d'un opérateur. Par contre, pour d'autres applications plus complexes ou afin d'être plus efficace, il peut être avantageux de vouloir combiner la robustesse et la force d'un robot avec les capacités d'analyse et d'adaptation d'un être humain. On a alors recours, à des robots coopératifs, qui servent à assister un opérateur dans les tâches qu'il doit accomplir.

Dans le domaine scientifique et celui de l'enseignement, l'automatique a souvent recours à des cas d'études particuliers, qui sont représentatifs de grandes classes d'applications. De plus, avec l'expérience, la connaissance de ces cas s'est affinée et ils fournissent aujourd'hui une base idéale pour comparer de façon valable les avantages et les inconvénients d'approches différentes. Le pendule inversé et le manipulateur robot sont les uns de ces cas typiques qui occupent une place importante dans l'industrie.

Afin de commander les systèmes industriels, plusieurs méthodes ont été mises en œuvre. Les performances d'un régulateur classique restent, malgré l'existence de cette diversité, bien souvent limitées en raison de la complexité réelle des systèmes à commander (non linéarité, variation des paramètres, ...). [14]

L'apparition de la commande par logique floue a permis de contrôler les systèmes mal définis, mal modélisés, voire n'ayant aucun modèle mathématique. Cette nouvelle approche utilise les connaissances et l'expérience acquise de l'expert, formulée sous forme de règles. Elle se montre très proche du raisonnement humain lors de la génération de la commande.

# Introduction Générale

---

Depuis la première application du formalisme de la logique floue à la commande des systèmes proposée par Mamdani, plusieurs travaux ont montré que le contrôle à logique floue est une méthode adéquate pour la commande des procédés mal définis ou complètement inconnus et qui ne peuvent pas être modélisés mathématiquement d'une manière aisée. par des experts formulées sous forme de règles. Cependant il se peut que les experts ne puissent pas transcrire leurs connaissances et expériences sous forme d'un contrôleur à logique floue. C'est à partir de ces constatations qu'ont été développées plusieurs méthodes optimales pour la conception des contrôleurs flous. Parmi ces méthodes, nous avons retenu les algorithmes génétiques.

Les algorithmes génétiques(AG) sont des algorithmes d'optimisation, appartenant à la famille des algorithmes évolutionnaire, fondés sur les mécanismes de la sélection naturelle et de la génétique. Leur fonctionnement est extrêmement simple. On part d'une population de solutions potentielle (chromosomes) initiales arbitrairement choisies, on évalue leur performances relative (fitness) qui permet de quantifier sa qualité .Sur la base de ces performances on crée une nouvelle population de solution potentielles en utilisant des opérateurs évolutionnaires simples : la sélection, le croisement, et la mutation. On recommence ce cycle jusqu'à l'obtention d'une solution satisfaisante (optimale).

Ce mémoire est organisé comme suit:

La première partie de chapitre I est consacré à des généralités et étude sur les robots : leurs constituants et domaines d'application.

Dans la suite du ce chapitre, nous avons étudié la modélisation d'un bras manipulateur où nous avons calculé le modèle dynamique de ce bras en nous basant sur le formalisme d'Euler-Lagrange.

Dans le deuxième chapitre, nous étudierons les différentes techniques de l'intelligence artificielle à savoir: les bases théoriques de la logique floue, les algorithmes génétiques qui nous serviront lors de la phase de stabilisation du pendule et du bras, le concept de variable linguistique, les notions de signification et de description floues.

A la fin de la synthèse de chaque type de commande, nous présenterons les résultats de simulation du système commandé mis à l'épreuve de perturbations et de variations paramétriques, choses qui nous permettront d'évaluer les performances et la robustesse de ces commandes.

Enfin, nous terminons notre travail par une conclusion générale.

# Introduction Générale

---

**I.1 Introduction**

Pour commander un robot ou pour simuler son comportement on doit disposer de modèles. Plusieurs niveaux sont possibles selon les objectifs, les contraintes de tâche et de performances recherchées : modèles géométriques, cinématiques et dynamiques [7].

L'obtention de ces différents modèles n'est pas aisée, la difficulté variant selon la complexité de la cinématique de la chaîne articulée. Entrent en ligne le nombre de degrés de liberté, le type des articulations mais aussi le fait que la chaîne peut être ouverte simple, arborescente ou fermée[7].

Ces modèles, qu'ils soient utilisés en simulation ou en commande, doivent représenter le comportement géométrique, cinématique ou dynamique du robot de façon réaliste. Il est donc nécessaire de mettre en œuvre des procédures efficaces d'identification de leurs paramètres constitutifs [7].

Pour qu'une commande puisse effectivement être implantée sur un contrôleur de robot, les modèles doivent être calculés et donc, le nombre d'opérations à effectuer doit être minimum. Les techniques proposées ont été développées dans ce souci et conduisent aux modèles les plus performants [7].

Dans ce chapitre une étude détaillée sur les robots manipulateurs a été présentée, où nous avons parlé de la géométrie, de la constitution, de la cinématique, et de la dynamique de ce type de robots.

Ce chapitre est une base pour le développement ultérieur du modèle dynamique du robot que nous allons utiliser pour la commande.

**I.2 Etude des robots :****I.2.1 Définition d'un robot :**

Un robot est défini comme étant un appareil automatique capable de manipuler des tâches ou d'exécuter des opérations (numériques ou analogiques) selon un programme fixe ou modifiable. Autrement dit, le robot est une machine programmable qui imite des actions d'une créature intelligente, pouvant manipuler des objets en réalisant des mouvements variés. [7]

La définition du robot se différencie selon les pays :

Au Japon, on y qualifie de robot « tout mécanisme permettant d'effectuer, en tout ou en partie, une tâche normalement réalisée par l'homme.

Mais la définition américaine du robot est beaucoup plus spécifique « un robot est un manipulateur reprogrammable à fonction multiples. Il est conçu pour déplacer des matériaux,

des pièces, des outils ou des instruments spécialisés suivant des trajectoires variables programmées, en vue d'accomplir des tâches très diverse.

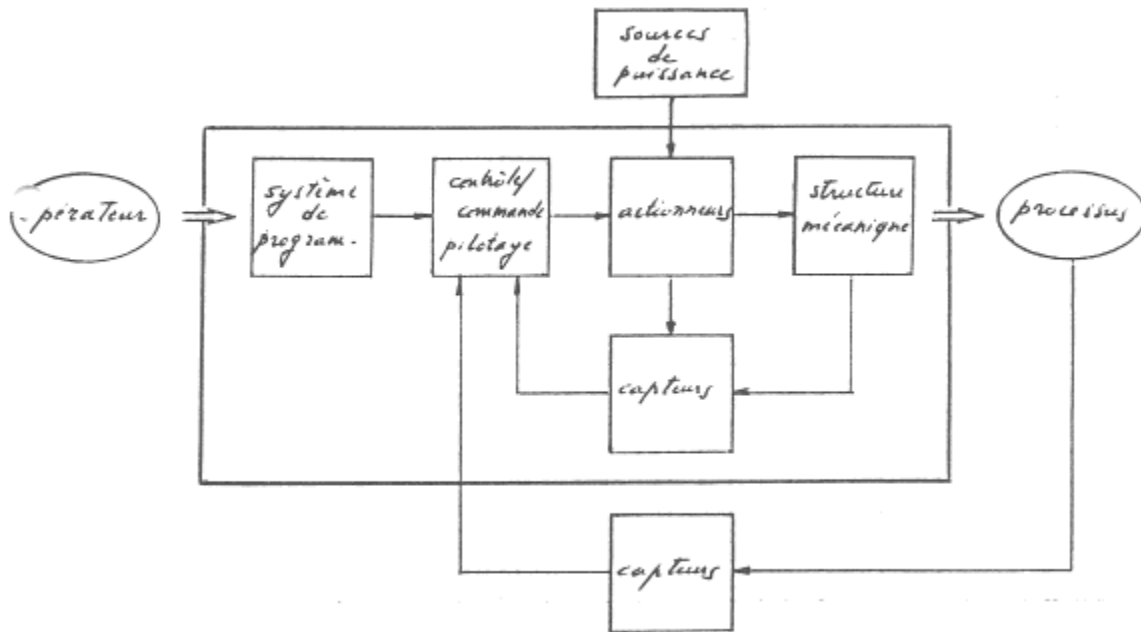
D'après l'Association Française de Normalisation (AFNOR), un robot est un « manipulateur commandé en position, reprogrammable, polyvalent, plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété tâches. Il a 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. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel » [7].

### **I.2.2 Constituants d'un robot :**

Un robot est essentiellement une structure mécanique poly articulée, équipée d'actionneurs, de capteurs, d'une unité de traitement de l'information et de sources d'énergie ; l'analogie avec l'être humain et plus généralement avec tout individu vertébré est immédiate : squelette, muscles, sens et cerveau.

On voit aussi ici les concours des différentes disciplines : mécanique, électronique, automatique et informatique.

Le robot est en effet conçu pour saisir et déplacer des objets. Ceci suppose donc une structure mécanique. La maîtrise d'un objet dans l'espace implique sa localisation et son orientation. Pour localiser un point de l'objet dans l'espace, il faut disposer de 3 degrés de Liberté (ddl), qui peut être des translations ou des rotations. Un point de l'objet étant fixe, si l'on désire orienter l'objet de manière quelconque, il faut disposer en plus de 3 autres ddl qui sont nécessairement des rotations.



*Figure I.1: les constitutions d'un robot*

Un robot manipulateur est constitué aussi de deux sous-ensembles distincts, un (ou plusieurs) organe terminal et une structure mécanique articulée :

- Sous le terme organe terminal, on regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétique, à dépression...) ou à transformer (outils, torche de soudage, pistolet de peinture...). Il s'agit donc d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, c'est-à-dire qu'il est équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel mais interchangeable. Un robot enfin, peut être multi bras chacun des bras portant un organe terminal différent. Par la suite, nous utilisons indifféremment le terme « organe terminal, préhenseur, outil ou effecteur » pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique [7].
- La structure mécanique articulée est un mécanisme ayant une structure plus au moins proche de celle du bras humain. Elle permet de remplacer, ou de prolonger son action. Son rôle est d'amener l'organe terminal dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données. Son architecture est une chaîne cinématique de corps généralement rigides, ou supposés

comme tels, assemblés par des liaisons appelées **articulations**. Les chaînes peuvent être soit **ouvertes simples**, soit **arborescentes**, soit **fermées**. [7]

### I.2.3 Notion de degré de liberté :

Précisons la notion d'**articulation** : Une articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre.

Soit **m** le nombre de degré de liberté résultant, encore appelé **mobilité de l'articulation**. [7]

La mobilité d'une articulation est telle que :

$$0 \leq m \leq 6.$$

Lorsque **m = 1** ; ce qui est fréquemment le cas en robotique, l'articulation est dite simples : [7]

**Rotoïde** ou **prismatique**.

### I.2.4 Articulation :

#### Articulation rotoïde :

Il s'agit d'une articulation de type pivot réduisant le mouvement entre deux corps d'une rotation autour d'un axe qui leur commun. La situation relative entre deux corps est donnée par l'angle autour de cet axe. L'articulation rotoïde est représentée par **R** (voir figure)[7]



*Figure I. 2 : symbole de l'articulation rotoïde*

❖ Exemple d'une simple rotation :

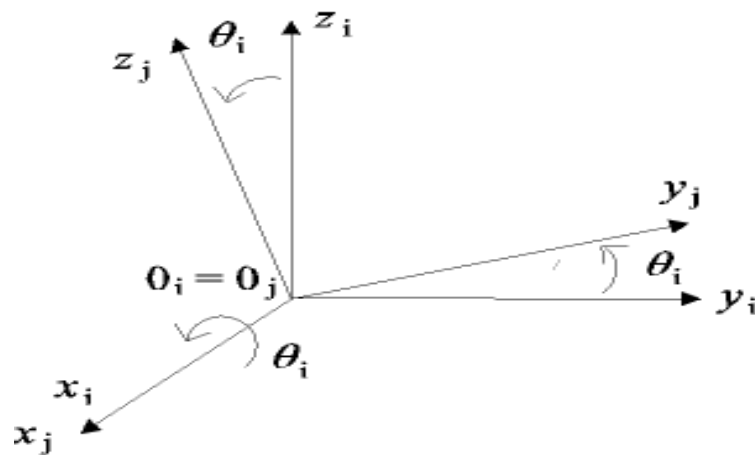


Figure I.3 : une simple rotation

Articulation prismatique :

Il s'agit d'une articulation de type glissière, notée **P** réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe. (Voir figure suivante)[7]



Figure I.4: symbole de l'articulation prismatique

❖ Exemple d'une simple translation :

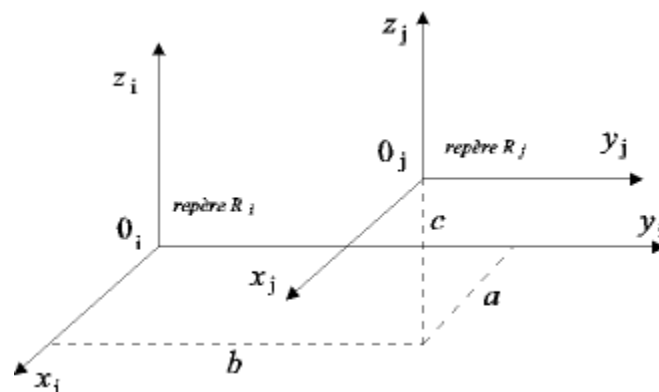


Figure I.5 : une simple translation

**I.2.5 Coordonnées homogènes :****I.2.5.1 Représentation d'un point :**

Soit **S** un point de coordonnées (x, y, z) et soit **O** l'origine du repère **R** ( $\vec{o}, \vec{i}, \vec{j}, \vec{k}$ )

Tel que **O** = (a, b, c). [1]

Le vecteur  $\vec{OS} = x\vec{i} + y\vec{j} + z\vec{k}$

$$P = \begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix}$$

**I.2.5.2 Représentation d'une direction:**

Une direction est représentée par quatre composantes la quatrième indique un point à l'infini. Soit les coordonnées cartésiennes (Vx, Vy, Vz) d'un vecteur unitaire V qui peut s'écrire en coordonnées homogènes comme suit :

$$V = \begin{bmatrix} Vx \\ Vy \\ Vz \\ 0 \end{bmatrix}$$

**I.2.5.3 Représentation d'un plan:**

Soit le plan dont son équation est représenté par la vectrice ligne S:

$\alpha x + \beta y + \gamma z + \delta = 0$  : équation du plan

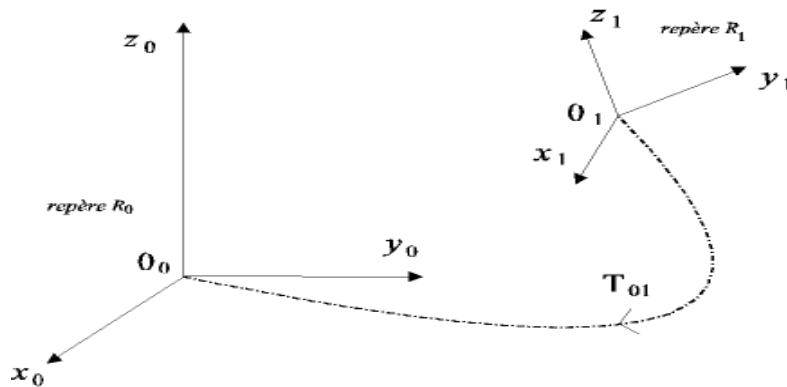
$$S = [\alpha \ \beta \ \gamma \ \delta]$$

$\forall n \in \text{au plan } S \Leftrightarrow S * n = 0$  (produit de matrice).

$$S * n = [\alpha \ \beta \ \gamma \ \delta] \begin{bmatrix} nx \\ ny \\ nz \\ 0 \end{bmatrix} = 0$$

## I. 2.6 Transformation homogène:

### I.2.6.1 Transformation des repères:



Soit la transformation quelconque (rotation ou translation) du repère  $R_i$  au repère  $R_j$  [Cette transformation est définie par la matrice homogène  ${}^i T_j$  de dimension  $4 \times 4$  telle que :

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

${}^i s_j, {}^i n_j, {}^i a_j, {}^i p_j$  : désignent les vecteurs unitaires selon les axes  $X_j, Y_j, Z_j$

On peut écrire la matrice  ${}^i T_j$  sous la forme :

$${}^i T_j = \begin{bmatrix} i s_j & i n_j & i a_j & i p_j \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### I.2.6.2 Matrice de rotation :

$$\text{Sur l'axe X} \quad \text{ROT}(X, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta & S\theta & 0 \\ 0 & S\theta & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Sur l'axe Y} \quad \text{ROT}(Y, \theta) = \begin{bmatrix} C\theta & 0 & S\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta & 0 & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Sur l'axe Z} \quad \text{ROT}(Z, \theta) = \begin{bmatrix} C\theta & S\theta & 0 & 0 \\ -S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### I.2.7 Domaines d'applications :

De la robotique industrielle à la robotique de service :

Les robots sont souvent utilisés dans des chaînes de fabrication, pour construire des automobiles ou des ordinateurs, par exemple. Ils effectuent une série de travaux répétitifs en peu de temps et sont aussi capables de détecter des produits défectueux.

Les robots sont également utilisés dans des endroits dangereux : par exemple, ils interviennent dans les centrales nucléaires, dans la lutte contre l'incendie ou dans le désamorçage de bombes.

Dans les hôpitaux, les chirurgiens se servent parfois de robots au cours d'opérations, lorsqu'ils doivent effectuer des tâches nécessitant une très grande précision. Ces opérations peuvent même se dérouler à distance : en effet, le 7 septembre 2001, un chirurgien de New York a piloté un robot se trouvant à Strasbourg et opéré un patient on parle alors de téléchirurgie.

Enfin, l'homme envoie des robots dans des endroits où il ne peut pas aller, comme l'espace ou le fond des océans. En 1997, le robot Sojourner piloté à distance par des ingénieurs de l'agence spatiale américaine (NASA) — a parcouru la surface de la planète Mars et envoyé des images vers la Terre pendant plusieurs semaines

### I.3 Description géométrique des robots :

Un robot est constitué d'une succession de segments solides mobiles les uns par rapport aux autres. On dit qu'on a affaire à un système mécanique articulé(SMA).

Le but poursuivi pour ce modèle est le suivant : à l'aide d'un langage graphique de description, faire de la génération automatique du modèle par programmation.

Les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement. [7]

Pour analyser le comportement de la structure mécanique d'un bras manipulateur, il est nécessaire de lier un repère orthonormé à chacun de ces corps et un référentiel (R0) attaché à la base. [7]

Pour étudier les mouvements des corps on va étudier les mouvements des repères.

Soient les repères {R0, R1,....., Rn} et un point P défini dans le repère {Rn} par :

$$\overrightarrow{OnP} = \begin{pmatrix} X_n \\ Y_n \\ Z_n \end{pmatrix}$$

Soit la relation :  ${}^0\check{P}_p = {}^0\tilde{T}_n * {}^0P_p$  tel que :

$${}^0\tilde{T}_n = {}^0\tilde{T}_1, {}^1\tilde{T}_2, \dots, {}^{n-1}\tilde{T}_n$$

La connaissance de la matrice de transformation  ${}^0\tilde{T}_n$  conduit à connaître la position de P.

Des méthodes ont été proposées pour connaître la position de l'orientation de chaque repère  $\{R_i\}$   $i=1 \dots n$  par rapport au repère  $\{R_{i-1}\}$  : la notation de **Denavit Hartenberg modifié**

### I. 3.1 Formalisme de Denavit Hartenberg modifié:

#### I. 3.1.1 Principe général :

L'articulation numéro  $i$  lie les segments  $S_{i-1}$  et  $S_i$  segments sont numérotés de 0 à  $n$ , et les articulations de  $1, 2, \dots, n$ . A chaque segment  $S_i$  sont associés à l'articulation numéro  $i$ , les paramètres  $(\theta_i, a_i, d_i, \alpha_i)$  ainsi que le repère  $R_i(O_i, x_i, y_i, z_i)$ .

Le passage du repère  $R_{j-1}$  au repère  $R_j$  se fait à travers 4 transformations élémentaires :

$d_i$  : translation le long de  $Z_i$

$a_i$  : la translation le long de  $X_i$

$\theta_i$  : la rotation de  $x_{i-1}$  et  $x_i$  autour de  $z_i$

$\alpha_i$  : la rotation de  $z_{i-1}$  et  $z_i$

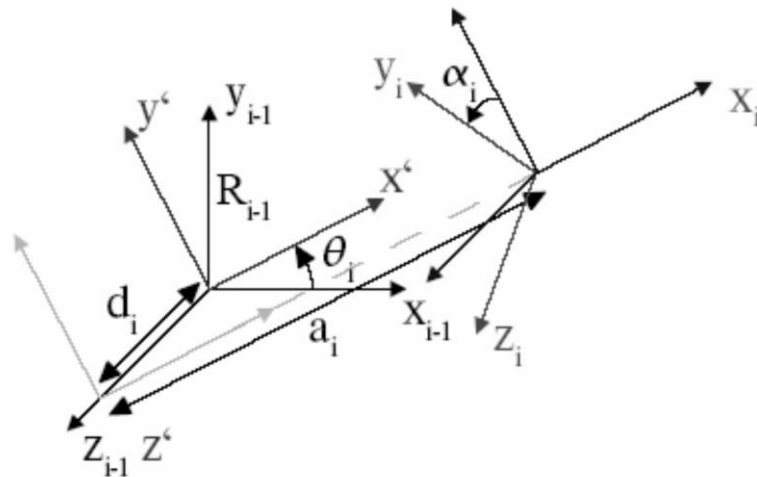


Figure I.6 : changement de repère

- **La matrice de passage** : c'est le produit de quatre matrices homogènes tel que :

$${}^i T_{i-1} = \text{rot}(z_{i-1}, \theta_i) * \text{trans}(z_{i-1}, d_i) * \text{trans}(x_i, a_i) * \text{rot}(x_i, \alpha_i)$$

Sous forme matricielle cette relation s'écrit :

$${}^i T_{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\sin\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cette méthode valable pour les deux types d'articulations.

**Remarque :**

Si l'articulation est rotoïde la variable articulaire est  $\theta_i$  et les autres paramètres sont constants ( $\mathbf{a}_i$ ,  $\mathbf{d}_i$ ,  $\alpha_i$ ) par contre, si l'articulation est prismatique la variable est  $\mathbf{d}_i$  et les autres sont constants.

**I.3.1.2 Règle de choix des repères :**

Les étapes à suivre sont :

- L'origine  $\mathbf{O}_i$  : placé à l'intersection de l'axe de rotation ( $\mathbf{i}$ ) et de la perpendiculaire communes aux axes de rotation  $\{\mathbf{i}, \mathbf{i}+1\}$
- Si les axes sont parallèle  $\mathbf{O}_i$  est placé de manière à rendre  $\mathbf{d}_i$  nulle.
- L'axe  $\mathbf{Z}_i$  aligné avec l'axe de rotation ( $\mathbf{i}+1$ )  
L'axe  $\mathbf{X}_i$  aligné avec la normale commune aux axes de rotation  $\{\mathbf{i}, \mathbf{i}+1\}$  il est orienté de l'axe ( $\mathbf{i}$ ) vers l'axe ( $\mathbf{i}+1$ ) (si les axes sont parallèle) sinon c'est-à-dire concourants  $\overline{\mathbf{X}_i}$  est parallèle ou antiparallèle au vecteur  $\overline{\mathbf{Z}_i} - \overline{\mathbf{1}} \wedge \overline{\mathbf{Z}_i}$
- L'axe  $\mathbf{Y}_i$  :  $\overline{\mathbf{Y}_i} = + (\overline{\mathbf{Z}_i} - \overline{\mathbf{1}} \wedge \overline{\mathbf{Z}_i})$
- Repère  $\mathbf{R}_0$  : son axe  $Z_0$  aligné avec l'axe de rotation (1) et son origine  $\mathbf{O}_0$  confondu avec  $\mathbf{O}_1$ .
- Repère  $\mathbf{R}_n$  : son origine confondue avec celle de  $\mathbf{R}_{n-1}$

**I.3.1.3 Le modèle géométrique direct :**

Le modèle géométrique direct exprime la situation de l'organe terminal (cordonnées opérationnelles) en fonction des variables articulaires du mécanisme. Mathématiquement, ce modèle peut être défini par la relation suivante [7] :

$$\mathbf{X} = \mathbf{f}(\mathbf{q}) \quad (\text{I.1})$$

$\mathbf{X}$  : cordonnées opérationnelles

$\mathbf{q}$  : variables articulaires

La position d'un solide  $\mathbf{S}_1$  dans le repère  $\mathbf{R}_1$  peut être représentée par le vecteur de position  $\overline{\mathbf{P}}_{01}$  dans le repère  $\mathbf{R}_0$  et son orientation représentée par la matrice  $\mathbf{M}_{01}$  par rapport à un solide  $\mathbf{S}_0$ . Pour cela on définit deux transformations :

- La transformation en translation exprimée par  $\overline{\mathbf{T}}_0$  :

$$\overline{\mathbf{P}}_{01} = \overline{\mathbf{P}}_{00} + \overline{\mathbf{T}}_0$$

Les composantes du solide  $\mathbf{S}_0$  dans le repère  $\mathbf{R}_0$  exprimées sous forme matricielle :

$$\mathbf{P}_{01} = \mathbf{P}_{11} + \mathbf{t}_0$$

$P_{11}$  : représente la matrice des composantes de la vectrice position du S1 dans le repère  $R_1$ .

$P_{01}$  : représente la matrice des composantes de la vectrice position du S1 dans le repère  $R_0$  correspond au solide  $S_0$

$t_0$  : la matrice des composantes de translation de  $S_0$  vers  $S_1$ .

L'objectif des coordonnées homogènes est de mettre les transformations (rotation et translation) sous forme d'une seule matrice tel que :

$$\begin{pmatrix} P_{01} \\ 1 \end{pmatrix} = \begin{pmatrix} M_{01} & t_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P_{11} \\ 1 \end{pmatrix}$$

${}^0K = \begin{pmatrix} P_{01} \\ 1 \end{pmatrix}$  : matrice de bloc du vecteur de position en coordonnées homogènes du  $S_1$  dans le repère  $R_0$  associé au  $S_0$

Pour simplifier on met  ${}^1K = \begin{pmatrix} P_{11} \\ 1 \end{pmatrix} \Rightarrow {}^0K = M_{01} {}^1K$

$M_{01}$  : une matrice de rotation de dimension 3\*3

$t_0$  : une matrice de rotation de dimension 3\*1 (matrice colonne)

### I.3.1.4 Exemple d'un bras d'un robot manipulateur à 2 degrés de liberté :

Dans notre exemple on va étudier la modélisation d'un manipulateur à deux degrés de liberté.

Pour ce modèle on utilise la méthode de «DH »modifié.

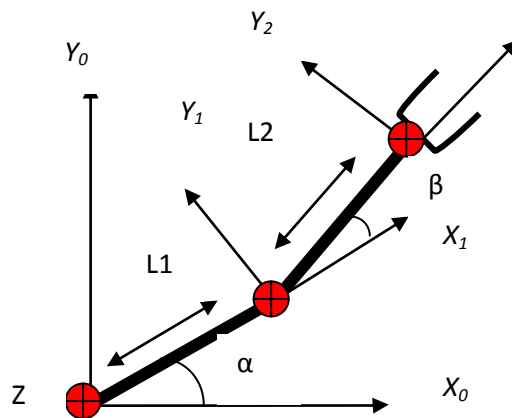


Figure I.7 : bras manipulateur à deux degré de liberté

On considère que les articulations sont rotoïde  $\Rightarrow \delta_i = 0$

Les paramètres de Denavit –Hartenberg sont donnés par le tableau suivant :

Corps	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\alpha$	0	$L_1$	0
2	$\beta$	0	$L_2$	0

Tableau I.1 : paramètres de Denavit –Hartenberg

$d_i=0 \Leftrightarrow$  pas de translation suivant l'axe Z

$\alpha_i=0 \Leftrightarrow$  pas de rotation autour de l'axe X

Le passage du repère  $R_0$  au repère  $R_2$  est donné par les matrices suivantes :

D'après la relation :

$${}^{i-1}\mathbf{T}_i = \text{rot}(Z_{i-1}, \theta_i) \times \mathbf{T}(Z_{i-1}, d_i) \times \mathbf{T}(X_i, a_i) \times \text{rot}(X_i, \alpha_i)$$

On obtient :

$${}^1\mathbf{T}_0 = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & L_1 \cos \alpha \\ \sin \alpha & \cos \alpha & 0 & L_1 \sin \alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

${}^1\mathbf{T}_0$  : la matrice de passage du repère  $R_0$  au repère  $R_1$ .

$${}^2\mathbf{T}_1 = \begin{bmatrix} \cos \beta & -\sin \beta & 0 & L_2 \cos \beta \\ \sin \beta & \cos \beta & 0 & L_2 \sin \beta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

${}^2\mathbf{T}_1$  : la matrice de passage du repère  $R_1$  au repère  $R_2$ .

$${}^2\mathbf{T}_0 = {}^1\mathbf{T}_0 \times {}^2\mathbf{T}_1$$

$${}^2\mathbf{T}_0 = \begin{bmatrix} \cos(\alpha + \beta) & -\sin(\alpha + \beta) & 0 & L_2 \cos(\alpha + \beta) + L_1 \cos \alpha \\ \sin(\alpha + \beta) & \cos(\alpha + \beta) & 0 & L_2 \sin(\alpha + \beta) + L_1 \sin \alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

D'où le modèle géométrique direct :

$$\left\{ \begin{array}{l} X_2 = L_2 \cos(\alpha + \beta) + L_1 \cos \alpha \\ Y_2 = L_2 \sin(\alpha + \beta) + L_1 \sin \alpha \\ Z_2 = 0 \end{array} \right.$$

**A.N :**

$$L_1 = 2 ; L_2 = 2.5$$

$$\alpha = \frac{\pi}{3} ; \beta = \frac{\pi}{6}$$

$$\left\{ \begin{array}{l} X_2 = 2.5 * \cos \frac{\pi}{2} + 2 * \cos \frac{\pi}{3} = \mathbf{0.80} \\ Y_2 = 2.5 * \sin \frac{\pi}{2} + 2 * \sin \frac{\pi}{3} = \mathbf{1.59} \\ Z_2 = \mathbf{0} \end{array} \right.$$

### I.3.1.4 Le modèle géométrique inverse :

Ce modèle consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles constitue ce que l'on appelle le modèle géométrique inverse (MGI) qui s'écrit sous forme :

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{X}) \quad (\text{I.2})$$

Il existe trois méthodes pour le calcul du MGI :

- La méthode de **PAUL** qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels.
- La méthode de **PIEPER** qui permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïde axes concourants ou trois articulations prismatiques.
- La méthode générale de **RAGHVAN** et **ROTH**, donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

#### Exemple d'un robot manipulateur à 2 degrés de liberté :

On prend le schéma précédent :

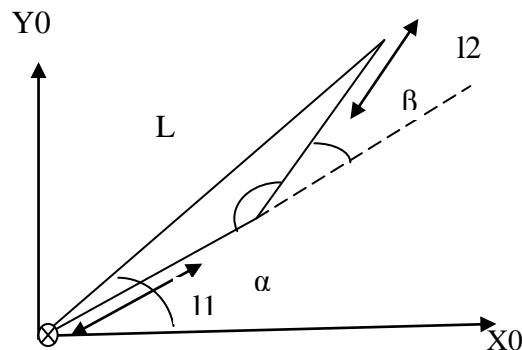


Figure I.7 : Exemple d'un robot manipulateur à 2 degrés de liberté

$$X_2 = l_2 \cos(\alpha + \beta) + l_1 \cos \alpha$$

$$Y_2 = l_2 \sin(\alpha + \beta) + l_1 \sin \alpha$$

Soit le système à deux équations et deux inconnus :

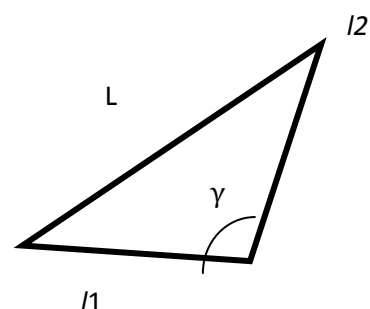
Pour obtenir le **MGI** on suit les étapes suivantes :

Utilisant le théorème de **Pythagore généralisé** :

$$L^2 = l_1^2 + l_2^2 - 2l_1l_2 \cos \gamma$$

Tel que :  $\gamma = \pi + \beta$

$$\text{D'où : } X_2^2 + Y_2^2 = l_1^2 + l_2^2 - 2l_1l_2 \cos \beta$$



**Rappel :**

$$\cos(x)=y \Rightarrow x=\arccos(y)$$

$$\cos(\beta) = (X_2^2 + Y_2^2 - (l_1^2 + l_2^2)) / 2l_1l_2$$

$$\Rightarrow \beta = \pm \arccos((X_2^2 + Y_2^2 - (l_1^2 + l_2^2)) / 2l_1l_2) \text{ sachant que :}$$

$$-1 \leq (X_2^2 + Y_2^2 - (l_1^2 + l_2^2)) / 2l_1l_2 \leq 1 \dots\dots(1)$$

(1) indique que le point P(X<sub>2</sub>, Y<sub>2</sub>) doit être atteignable.

**D'autre part :**

On développe les expressions  $\cos(\alpha + \beta)$  et  $\sin(\alpha + \beta)$  dans **MGD** on obtient :

$$\begin{cases} (l_1 + l_2 \cos(\beta)) \cos(\alpha) - l_2 \sin(\beta) \sin(\alpha) = X_2 \\ l_2 \sin(\beta) \cos(\alpha) + (l_1 + l_2 \cos(\beta)) \sin(\alpha) = Y_2 \end{cases}$$

On remarque qu'on peut écrire le système sous forme **Ax=b**

$$\begin{bmatrix} l_1 + l_2 \cos \beta & -l_2 \sin \alpha \\ l_2 \sin \beta & l_2 \cos \beta \end{bmatrix} \begin{bmatrix} \cos \alpha \\ \sin \alpha \end{bmatrix} = \begin{bmatrix} X_2 \\ Y_2 \end{bmatrix}$$

$$A = \begin{bmatrix} l_1 + l_2 \cos \beta & -l_2 \sin \alpha \\ l_2 \sin \beta & l_2 \cos \beta \end{bmatrix} ; x = \begin{bmatrix} \cos \alpha \\ \sin \alpha \end{bmatrix} ; b = \begin{bmatrix} X_2 \\ Y_2 \end{bmatrix}$$

Calcul de  $\cos(\alpha)$  et  $\sin(\alpha)$  par la méthode de **Cramer** :

❖ Calcul de déterminant :

$$\begin{vmatrix} l_1 + l_2 \cos \beta & -l_2 \sin \alpha \\ l_2 \sin \beta & l_2 \cos \beta \end{vmatrix} = l_1^2 + l_2^2 + 2l_1l_2 \cos \beta = X_2^2 + Y_2^2$$

$$\cos \alpha = \frac{\begin{vmatrix} X_2 & -l_2 \sin \beta \\ Y_2 & l_2 \cos \beta \end{vmatrix}}{\begin{vmatrix} l_1 + l_2 \cos \beta & -l_2 \sin \alpha \\ l_2 \sin \beta & l_2 \cos \beta \end{vmatrix}} = \frac{x_2 l_2 \cos \beta + y_2 l_2 \sin \beta}{X_2^2 + Y_2^2}$$

$$\sin \alpha = \frac{\begin{vmatrix} l_1 + l_2 \cos \beta & X_2 \\ l_2 \sin \beta & Y_2 \end{vmatrix}}{\begin{vmatrix} l_1 + l_2 \cos \beta & -l_2 \sin \alpha \\ l_2 \sin \beta & l_2 \cos \beta \end{vmatrix}} = \frac{l_1 + l_2 \cos \beta Y_2 - l_2 \sin \beta X_2}{X_2^2 + Y_2^2}$$

$$\begin{cases} \alpha = \text{Arctg} \left( \frac{l_1 + l_2 \cos \beta Y_2 - l_2 \sin \beta X_2}{x_2 l_2 \cos \beta + y_2 l_2 \sin \beta} \right) \\ \beta = \pm \text{Arccos} \left( (X_2^2 + Y_2^2 - (l_1^2 + l_2^2)) / 2l_1l_2 \right) \end{cases}$$

## I.4 Description cinématique :

### I.4.1 Définition

Ce modèle amélioré porte au moins trois noms, modèle cinématique, modèle variationnel, modèle pour la commande en vitesse.

Il consiste à remplacer le modèle géométrique dont une difficulté essentielle est la non-linéarité vis-à-vis des variables articulaires par une approximation linéaire qui sera validée si on se déplace par petits accroissements :

- **Positions :**

En pratique, les coordonnées d'un point de chaque solide sont choisies à l'extrémité avale de chaque solide  $R_0$ .

Les matrices des rotations par rapport au repère fixe.

- **Vitesses :**

La vitesse d'un point de chaque corps : en pratique, on choisit le centre de gravité du solide.

La vitesse instantanée de rotation du corps par rapport au repère fixe  $R_0$ .

### I.4.2 Description cinématique des robots :

Le modèle cinématique contient deux types de modèle :

#### I.4.2.1 Modèle cinématique direct :

Le modèle cinématique direct d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires.

Le MCD décrit les vitesses des coordonnées opérationnelles  $\dot{\mathbf{X}}$  en fonction des vitesses articulaires  $\dot{\mathbf{q}}$ :

$$\dot{\mathbf{X}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (\text{I.3})$$

Avec :  $\mathbf{J}(\mathbf{q})$  la jacobienne du mécanisme.

La matrice jacobienne intervient dans le calcul du modèle différentiel direct qui donne les variations élémentaires  $d\mathbf{X}$  des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires  $d\mathbf{q}$ .

#### I.4.2.2] Définition de la jacobienne :

La jacobienne  $\mathbf{J}$  et son inverse  $\mathbf{J}^{-1}$  ont des applications fondamentales dans l'analyse d'un mécanisme et dans sa commande. [1]

Supposons qu'on ait  $n$  fonctions  $f_1, \dots, f_n$  de  $m$  variables  $x_1, \dots, x_m$  :

$$\begin{aligned}
 Y_1 &= f_1(x_1, x_2, \dots, x_m) \\
 Y_2 &= f_2(x_1, x_2, \dots, x_m) \\
 &\vdots \\
 Y_n &= f_n(x_1, x_2, \dots, x_m) \\
 \mathbf{Y} &= \mathbf{F}(\mathbf{X})
 \end{aligned}
 \tag{I.4}$$

Les différentielles de  $y_i$  sont :

$$\begin{aligned}
 \frac{\partial f_1}{\partial x_1} * dx_1 + \frac{\partial f_1}{\partial x_2} * dx_2 + \dots + \frac{\partial f_1}{\partial x_m} * dx_m &= \partial y_1 \\
 \frac{\partial f_2}{\partial x_1} * dx_1 + \frac{\partial f_2}{\partial x_2} * dx_2 + \dots + \frac{\partial f_2}{\partial x_m} * dx_m &= \partial y_2 \\
 &\vdots \\
 \frac{\partial f_n}{\partial x_1} * dx_1 + \frac{\partial f_n}{\partial x_2} * dx_2 + \dots + \frac{\partial f_n}{\partial x_m} * dx_m &= \partial y_n
 \end{aligned}$$

$$\Leftrightarrow \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_m} \end{pmatrix}$$

On a :  $dY = JdX$

Si on divise sur dt on aura

$$\frac{dY}{dt} = J(X) \frac{dX}{dt} \Leftrightarrow \dot{Y} = J(X) \dot{X}$$

La jacobienne donne la vitesse de Y en fonction de la vitesse de X pour une position X.

**Remarque :** Si la position X varie la jacobienne change.

Inversement, on calcul la vitesse de X en fonction de la vitesse de Y comme suit :

$$\dot{\mathbf{X}} = \mathbf{J}^{-1}(\mathbf{X}) \dot{\mathbf{Y}}
 \tag{I.5}$$

Où  $\mathbf{J}^{-1}(\mathbf{X})$  est la jacobienne inverse de F en X

### 4.5.2.3 Calcul de la jacobienne :

Il existe deux méthodes de calcul :

➤ **Méthode indirect** : Consiste à utiliser MGD :

$$\mathbf{X} = \mathbf{f}(\mathbf{q})
 \tag{I.6}$$

Sachant que, la matrice jacobienne est la matrice des dérivées partielles de f par rapport à coordonnées généralisées :

$$J_{ij} = \frac{\partial f_i(\mathbf{q})}{\partial q_j} \quad i=1 \dots m; j=1 \dots m$$

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

- **Méthode direct** : Cette méthode permet d'obtenir la matrice jacobienne par un calcul direct basé sur l'influence qui produit chaque articulation sur le repère terminal.

#### I.4.2.4 Exemple d'application :

$$\begin{cases} X_2 = L_2 \cos(\alpha + \beta) + L_1 \cos \alpha \\ Y_2 = L_2 \sin(\alpha + \beta) + L_1 \sin \alpha \end{cases}$$

$$\frac{dx}{dt} = \mathbf{J} \frac{dq}{dt}$$

En utilisant la méthode directe :

Avec :

$$\frac{dq}{dt} = \begin{bmatrix} \frac{d\alpha}{dt} \\ \frac{d\beta}{dt} \end{bmatrix}, \quad \frac{dx}{dt} = \begin{bmatrix} \frac{dx_2}{dt} \\ \frac{dy_2}{dt} \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x_2}{\partial \alpha} & \frac{\partial x_2}{\partial \beta} \\ \frac{\partial y_2}{\partial \alpha} & \frac{\partial y_2}{\partial \beta} \end{bmatrix}$$

$$\begin{cases} X_2 = L_2 \cos(\alpha + \beta) + L_1 \cos \alpha \\ Y_2 = L_2 \sin(\alpha + \beta) + L_1 \sin \alpha \end{cases}$$

$$\begin{cases} \dot{X}_2 = -L_2 (\dot{\alpha} + \dot{\beta}) \sin(\alpha + \beta) - L_1 \sin(\alpha) \dot{\alpha} \\ \dot{Y}_2 = L_2 \cos(\alpha + \beta) (\dot{\alpha} + \dot{\beta}) + L_1 \cos(\alpha) \dot{\alpha} \end{cases}$$

$$\begin{cases} \dot{X}_2 = [-L_2 \sin(\alpha + \beta) - L_1 \sin(\alpha)] \dot{\alpha} - L_2 \sin(\alpha + \beta) \dot{\beta} \\ \dot{Y}_2 = [L_2 \cos(\alpha + \beta) + L_1 \cos(\alpha)] \dot{\alpha} + L_2 \cos(\alpha + \beta) \dot{\beta} \end{cases}$$

$$\begin{bmatrix} \dot{X}_2 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} [-L_2 \sin(\alpha + \beta) - L_1 \sin(\alpha)] & -L_2 \sin(\alpha + \beta) \\ L_2 \cos(\alpha + \beta) + L_1 \cos(\alpha) & L_2 \cos(\alpha + \beta) \end{bmatrix} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix}$$

$$\mathbf{J} = \begin{bmatrix} -L_2 \sin(\alpha + \beta) - L_1 \sin(\alpha) & -L_2 \sin(\alpha + \beta) \\ L_2 \cos(\alpha + \beta) + L_1 \cos(\alpha) & L_2 \cos(\alpha + \beta) \end{bmatrix}$$

**I.4.2.5 Modèle cinématique inverse (MCI) :**

L'objectif de ce modèle est de calculer, à partir d'une configuration  $\mathbf{q}$  donné, les vitesses articulaires qui assurent au repère terminal une vitesse opérationnelle imposée. Cette définition est analogue à celle du modèle différentiel inverse : ce dernier permet de déterminer la différentielle articulaire  $d\mathbf{q}$  spécifiée. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires. La mise en œuvre peut être faite de façon analytique ou numérique.

**I.4.2.6 Exemple d'application :**

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \dot{\mathbf{x}} \quad (\text{I.7})$$

$$\mathbf{q}' = \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} ; \quad \dot{\mathbf{x}} = \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix}$$

AN:

$$L_1=1 ; L_2=0.5 \quad \alpha = \frac{\pi}{2} \quad \beta = \frac{\pi}{3}$$

Calcul de  $\mathbf{J}^{-1}$  :

$$\mathbf{J}^{-1} = \frac{1}{0.43} \begin{bmatrix} -0.43 & 0.25 \\ 0.43 & -1.25 \end{bmatrix}$$

$$\mathbf{J}^{-1} = \begin{bmatrix} 2.9 & -0.58 \\ -1 & -1 \end{bmatrix}$$

**I.5 Description dynamique des robots :**

Connaissant  $\mathbf{X}(t)$ , la position et l'orientation de l'organe terminal et étant capable de passer de  $\mathbf{X}(t)$  à  $\mathbf{O}(t)$ , on se pose la question de savoir quel couple on doit appliquer aux articulations pour que le robot exécute bien  $\mathbf{O}(t)$  donc  $\mathbf{X}(t)$ .

Compte tenu des types de robot et des tâches exécutées, l'existence de certains phénomènes physiques liés à la vitesse de fonctionnement ou des effets parasites ne peut plus être occultée, sous peine de conduire à des dégradations notables des performances.

Parmi les effets parasites nous citons :

- Les perturbations liées à l'environnement.
- L'élasticité des segments du robot.
- L'influence de la charge transportée.
- Les jeux, frottement au niveau des articulations.

Pour aboutir au modèle dynamique du robot, on essaie d'établir les équations du mouvement en relation avec les forces :

- La pesanteur.
- Les forces qui découlent des moteurs et qui s'appliquent aux articulations.
- Les forces qui viennent du mouvement des masses :
- Les forces inertielles.
- Les forces centrifuges.
- Les forces issues du couplage entre les différents segments.

### I.5.1 Modèle dynamique des robots :

Le modèle dynamique d'un robot à n degrés de liberté est décrit par un système d'équations différentielles de second ordre liant les couples ou forces exercées par les actionneurs, la position, la vitesse, et l'accélération des articulations du mécanisme.

### I.5.2 Forme générale des équations dynamiques :

$$\Gamma = A(q) \ddot{q} + C(q, \dot{q}) \dot{q} + Q(q) + F(\dot{q}) - H \text{signe}(\dot{q}) \quad (\text{I.8})$$

$\Gamma$ , efforts actionneurs

$A$ , matrice d'inertie

$C$ , efforts centrifuges et de Coriolis

$Q(p)$ , couple/forces de gravité

$F(\dot{q})$ , frottements visqueux

$H \text{signe}(\dot{q})$  frottements secs

Plusieurs formalismes ont été utilisés pour obtenir un modèle dynamique des robots .Les formalismes les plus souvent utilisés sont :

#### I.5.2.1 Formalisme Newton-Euler :

Il est basé sur l'expression des torseurs dynamiques (forces et moments) appliqués aux centres de gravités de chaque articulation.

Un algorithme itératif permet alors d'exprimer le modèle dynamique.

#### I.5.2.2 Formalisme de Lagrange :

Décrit les équations du mouvement en termes de travail et d'énergie du système (détermine  $A$ ,  $C$ ,  $Q$ ,  $F$ , et  $H$ ), ce qui traduit, lorsque l'effort extérieur sur l'organe terminal est supposé nul, par l'équation :

$$\Gamma_i = \frac{d}{dt} \times \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \quad i = 1, \dots, n \quad (\text{I.9})$$

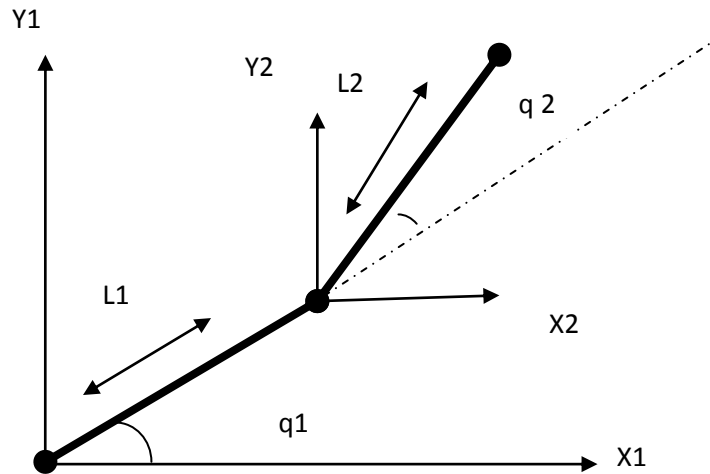
Avec :

- $\Gamma_i$ : couple/force articulaire engendré par l'actionneur (articulation)
- $L$  : lagrangien du système

$$\mathbf{L} = \mathbf{E} - \mathbf{U} \quad (\text{I.10})$$

- E : énergie cinétique totale du système,
- U : énergie potentielle totale du système.

**I.5.3 Exemple d'application :**



Pour obtenir des équations différentielles, on utilise le formalisme de LAGRANGE qui conduit directement aux systèmes d'équations non linéaires couplées du second ordre :

$$\Gamma \mathbf{i} + \mathbf{f}_i = \frac{d}{dt} \times \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} \tag{I.11}$$

Soit les centres de masses des deux articulations aux extrémités de ces dernières :

$$\Gamma \mathbf{i} + \mathbf{f}_i = \frac{d}{dt} \times \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}}$$

$$L = T - U$$

Où :  $\left\{ \begin{array}{l} T : \text{énergie cinétique totale du système} (T = T_1 + T_2) \\ U : \text{énergie potentielle totale du système} (U = U_1 + U_2) \end{array} \right.$

Soit les centres de masses des deux articulations aux extrémités de ces dernières :

$$T(m_1) = \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2) \tag{I.12}$$

$$\text{Avec: } \left\{ \begin{array}{l} X_1 = L_1 \cos q_1 \Rightarrow \dot{X}_1 = -L_1 \dot{q}_1 \sin q_1 \\ Y_1 = L_1 \sin q_1 \Rightarrow \dot{Y}_1 = L_1 \dot{q}_1 \cos q_1 \end{array} \right.$$

On remplace X1 et Y1 dans T1 :

$$\Rightarrow T_1(m_1) = \frac{1}{2} m_1 [(-L_1 \dot{q}_1 \sin q_1)^2 + (L_1 \dot{q}_1 \cos q_1)^2] = \frac{1}{2} (L_1 \dot{q}_1)^2$$

$$T_2(m_2) = \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2)$$

$$\left\{ \begin{array}{l} X_2 = L_1 \cos q_1 + L_2 \cos(q_1 + q_2) \\ Y_2 = L_1 \sin q_1 + L_2 \sin(q_1 + q_2) \end{array} \right.$$

$$\begin{cases} X_2 = -L_1 \dot{q}_1 \sin q_1 - L_2 (\dot{q}_1 + \dot{q}_2) \sin(q_1 + q_2) \\ Y_2 = L_1 \dot{q}_1 \cos q_1 + L_2 (\dot{q}_1 + \dot{q}_2) \cos(q_1 + q_2) \end{cases}$$

On remplace  $X_1$  et  $Y_1$  dans  $T_1$  :

$$\Rightarrow T_2(m_2) = \frac{1}{2} m_2 [(-L_1 \dot{q}_1 \sin q_1 - L_2 (\dot{q}_1 + \dot{q}_2) \sin(q_1 + q_2))^2 + (L_1 \dot{q}_1 \cos q_1 + L_2 (\dot{q}_1 + \dot{q}_2) \cos(q_1 + q_2))^2]$$

$$T_2(m_2) = \frac{1}{2} m_2 [(L_1 \dot{q}_1)^2 + (L_2 (\dot{q}_1 + \dot{q}_2))^2 + 2L_1 L_2 \dot{q}_1 (\dot{q}_1 + \dot{q}_2) \sin q_1 \sin(q_1 + q_2) + 2L_1 L_2 \dot{q}_1 (\dot{q}_1 + \dot{q}_2) \cos q_1 \cos(q_1 + q_2)]$$

$$T_2(m_2) = \frac{1}{2} m_2 [(L_1 \dot{q}_1)^2 + (L_2 (\dot{q}_1 + \dot{q}_2))^2 + 2L_1 L_2 \dot{q}_1 (\dot{q}_1 + \dot{q}_2) \cos q_2]$$

On a  $T = T_1 + T_2$

❖ Calcul de  $U$  :

$$U = U(m_1) + U(m_2)$$

$$U(m_1) = \int_0^{L_1 \sin q_1} m_1 g dy = m_1 g L_1 \sin q_1$$

$$U(m_2) = \int_0^{L_1 \sin q_1 + L_2 \sin(q_1 + q_2)} m_2 g dy = m_2 g [L_1 \sin q_1 + L_2 \sin(q_1 + q_2)]$$

$$\mathbf{L} = \mathbf{T} \cdot \mathbf{U}$$

$$L = \frac{1}{2} L_1 (m_1 + m_2) \dot{q}_1^2 + \frac{1}{2} m_2 L_2^2 \dot{q}_1^2 + \frac{1}{2} m_2 L_2^2 \dot{q}_2^2 + m_2 L_2^2 \dot{q}_1 \dot{q}_2 + m_2 L_1 L_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \cos(q_2) - (m_1 + m_2) g L_1 \sin q_1 - m_2 g L_2 \sin(q_1 + q_2)$$

$$\bullet \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{q}_1} \right) = L_1^2 (m_1 + m_2) \ddot{q}_1 + m_2 L_2^2 \ddot{q}_1 + m_2 L_1^2 \ddot{q}_2 + m_2 L_1 L_2 (2\dot{q}_1 + \dot{q}_2) \cos q_2$$

$$- m_2 L_1 L_2 (2\dot{q}_1 + \dot{q}_2) \dot{q}_2 \sin q_2$$

$$\bullet \frac{\partial L}{\partial q_1} = -(m_1 + m_2) g L_1 \cos q_1 - m_2 g L_2 \cos(q_1 + q_2)$$

$$\bullet \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{q}_2} \right) = m_2 L_2^2 \ddot{q}_2 + m_2 L_2 (L_2 + L_1 \cos q_2) \ddot{q}_1 - m_2 L_2 L_1 \dot{q}_1 \dot{q}_2 \sin q_2$$

$$\bullet \frac{\partial L}{\partial q_2} = m_2 L_1 L_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \sin q_2 + m_2 g L_2 \cos(q_1 + q_2)$$

Les forces extérieures qui existent sont :  $F$ ,  $U$ ,  $\tau$  telque :

$\tau$ : Le vecteur de commande (forces produites par les moteurs sur les articulations)

$F$ : les forces de frictions visqueuses

$U$  : le choc en retour

**Finalement :**

$$\tau_1 = [L_1^2 (m_1+m_2) + m_2 L_2 (L_2+2L_1 \cos q_2)] \ddot{q}_1 + [m_2 L_2^2 + m_2 L_1 L_2 \cos q_2] + (-2L_1 L_2 m_2 \dot{q}_2 \sin q_2) \dot{q}_1$$

$$+ (-L_1 L_2 m_2 \dot{q}_2 \sin q_2) \dot{q}_2 + (m_1+m_2) g L_1 \cos q_1 + m_2 g L_2 \cos (q_1+q_2) + f_1 + u_1$$

$$\tau_2 = m_2 L_2 (L_2+L_1 \cos q_2) \ddot{q}_1 + m_2 L_2^2 \ddot{q}_2 + (L_1 L_2 m_2 \sin q_2) q_1 \dot{q}_1 + g m_2 L_2 \cos (q_1+q_2) + f_2 + u_2$$

Ce système d'équations peut s'écrire sous forme matricielle :

$$\tau = M(q) \ddot{q} + H(q, \dot{q}) + F + U \quad (\text{I.13})$$

L'équation (1) représente un système d'équations différentielles non linéaires. Pour résoudre, on utilise le théorème de Rung- Kutta.

Pour cela on peut écrire L'équation (1) sous forme d'un modèle d'état :

$$(1) \Rightarrow \ddot{q} = M^{-1}[\tau - H(q, \dot{q}) - F - U].$$

$$\text{on pose : } \ddot{q}_1 = \dot{x}_1 \Rightarrow \dot{q}_1 = x_1 \Rightarrow q_1 = x_2$$

$$\ddot{q}_2 = \dot{x}_3 \Rightarrow \dot{q}_2 = x_3 \Rightarrow q_2 = x_4$$

$$\Rightarrow \begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_3 \end{bmatrix} = M^{-1}[\tau - H(q, \dot{q}) - F - U]. \\ \dot{x}_2 = x_1 \\ \dot{x}_4 = x_3 \end{cases}$$

Tel que :

$$M(q) = \begin{bmatrix} 2l_1 l_2 \cos(q_2) + l_2^2 m_2 + l_1^2 (m_1 + m_2) & l_2 m_2 + l_1 l_2 \cos(q_2) m_2 \\ l_2^2 m_2 + l_1 l_2 \cos(q_2) m_2 & l_2^2 m_2 \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} -2l_1 l_2 m_2 \dot{q}_2 \sin(q_2) & -l_1 l_2 m_2 \dot{q}_2 \sin(q_2) \\ l_1 l_2 m_2 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} g(m_2 l_2 \cos(q_1 + q_2) + (m_1 + m_2) l_1 \cos q_1) \\ m_2 l_2 g \cos(q_1 + q_2) \end{bmatrix}$$

## I.6 Conclusion :

Les performances d'une machine se mesurent lors de son utilisation, qu'on doit donc précisément connaître lors de la conception. Mais, par sa flexibilité, le robot peut s'adapter à plusieurs types de tâches non envisagées au stade conceptuel, la de mesurer à priori les performances d'un robot qui peuvent êtres excellentes dans une application et mauvaises dans une autre. Pour contourner cet écueil, il est préférable de se contenter de mesurer lorsque c'est possible, quelques grandes caractéristiques, parmi lesquelles le volume pouvant être atteint par l'extrémité du robot, les possibilités d'orientation de l'organe terminal, la précision en position, la résolution, la vitesse de déplacement de l'organe terminal.

## **II.1 Introduction :**

Dans la commande des systèmes non linéaires ou ayant des paramètres non constants, les lois de commande classique peuvent être insuffisantes surtout lorsque les exigences sur la précision et autres caractéristiques dynamiques des systèmes sont strictes. On doit faire appel à des lois de commande insensibles aux variations des paramètres, aux perturbations et aux non linéarités.

Dans cet objectif, plusieurs outils sont proposés dans la littérature, dont on cite la logique floue. Depuis les travaux de Mamdani (1974), le domaine de contrôle par logique floue est devenu très important, grâce à ces capacités de traiter les informations incertaines. L'algorithme de contrôle par logique floue est constitué d'un ensemble de règles de décisions, il peut être considéré comme un algorithme de contrôle non mathématique contrairement aux algorithmes de contrôle conventionnel.

Dans le domaine de l'asservissement, les contrôleurs flous peuvent concurrencer les correcteurs classiques.

En effet, ils offrent de bonnes caractéristiques en termes de temps de réponse, dépassement et surtout robustesse dans certaines applications.

De plus, ils n'ont besoin que d'une connaissance sommaire du système à contrôler, qui s'exprime sous forme d'un tableau (moteur d'inférence) où l'on enregistre le comportement du correcteur.

Le temps de mise au point est donc beaucoup plus court bien qu'il soit nécessaire d'utiliser soit un microcontrôleur, soit un composant dédié. Dans un souci d'optimiser encore la qualité de l'asservissement, on se propose de trouver un moyen de rendre le contrôleur flou dynamique.

Ainsi, la base devrait se modifier petit à petit en suivant les variations des paramètres du système et assurer un asservissement optimal.

Dans ce chapitre, nous allons présenter et analyser quelques structures intelligentes basées sur les concepts tels que la logique floue et les algorithmes génétiques. Cette démarche est menée afin de lever une contrainte en contrôle relative à la nécessité de disposer d'un modèle le plus fidèle possible, sachant que les erreurs de modélisation et les imperfections des modèles, contribuent à dégrader notablement les performances des lois de commandes conventionnelles [12]

Dans la première partie, nous étudierons en détail les principes de la logique floue et ses applications dans le domaine du réglage puis nous décrirons différentes solutions pour un organe de commande basé sur les principes flous.

Dans la deuxième partie de ce chapitre, nous allons présenter le pendule inversé et nous discuterons des algorithmes génétiques. Nous présenterons les propriétés des algorithmes génétiques, en tenant compte de leurs caractéristiques dans le traitement du problème posé.

## II.2 La logique floue et la commande floue [3] :

Dans la commande des systèmes non linéaires ou ayant des paramètres non constants, les lois de commande classique peuvent être insuffisantes car elles sont non robustes surtout lorsque les exigences sur la précision et autres caractéristiques dynamiques des systèmes sont strictes. On doit faire appel à des lois de commande insensibles aux variations des paramètres, aux perturbations et aux non linéarités.

Dans cet objectif, plusieurs outils sont proposés dans la littérature, dont on cite la logique floue qui représente est une théorie qui a connu un grand engouement depuis que **Zadeh** a introduit le concept de sous-ensembles flous en 1965. Par la suite, en 1974, **Mamdani** introduisait la commande floue pour la régulation de processus industriel. Dans les années 80, la commande floue connaît un essor considérable au Japon, notamment grâce aux travaux de **Sugeno**. La logique floue permet de faire le lien entre modélisation numérique et modélisation symbolique, ce qui a permis des développements industriels spectaculaires à partir d'algorithmes très simples de traduction de connaissances symboliques en entité numérique et inversement. Elle présente en effet l'avantage d'utiliser des règles linguistiques simples permettant de traduire facilement le savoir faire d'un expert pour répondre à une problématique spécifique.

### II.2.1 Principe de la logique floue [3] :

La notion de logique floue permet d'étendre la notion de logique classique, associée aux variables booléennes ne prenant que deux valeurs 0 et 1. Il est alors possible d'associer à des variables des coefficients d'appartenance à des sous-ensembles flous prenant des valeurs dans l'intervalle  $[0, 1]$  et quantifiant l'incertitude sur la variable. Un événement certain pour la variable se traduira par un coefficient d'appartenance au sous-ensemble flou, c'est-à-dire à la propriété, égal à 1 alors que la valeur sera inférieure à 1 en présence d'incertitudes. Il vient alors qu'une valeur nulle pour un coefficient d'appartenance indique que la possibilité d'appartenance au sous-ensemble sélectionné de la variable représentant la grandeur concernée est complètement rejetée.

### II.2.2 Les domaines d'application de la logique floue :

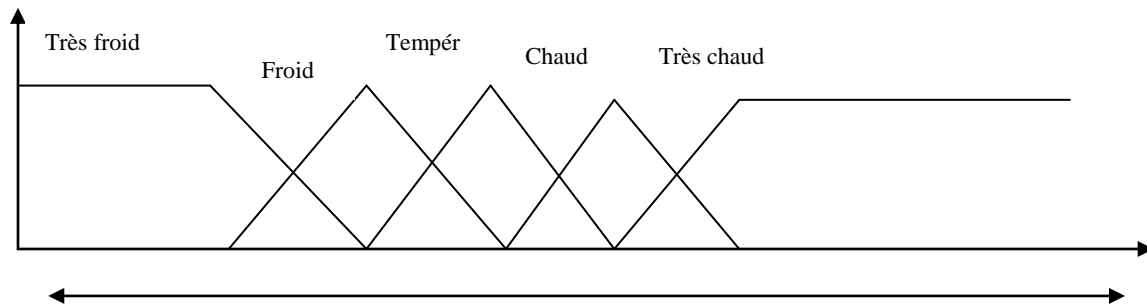
L'évolution de la technologie permet d'aborder des problèmes en temps réel, à dynamique rapide (plus rapide que dans les premières applications de type de conduite de processus lents : chaudière, fours...etc.), l'industrie japonaise a développée les produits grand public à base de la logique floue tel que les appareils électroménagers, systèmes audiovisuel, systèmes automobile embarqués, de transport, de contrôle et de commande dans la plupart des domaines industriels de production ;transformation, traitement de produit et déchet.

**II.2.3 Quelques concepts sur la logique floue :**

**II.2.3.1 Les variables floues :**

Contrairement aux variables binaires qui sont définies par les deux états « vrai=1 » ou « faux=0 », les variables floues présentent toute gradation entre la valeur « 0 » et la valeur « 1 ». on définit les notions suivantes :

❖ **Exemple sur les variables floues :**



*Figure 1.1 : exemple sur les variables floues*

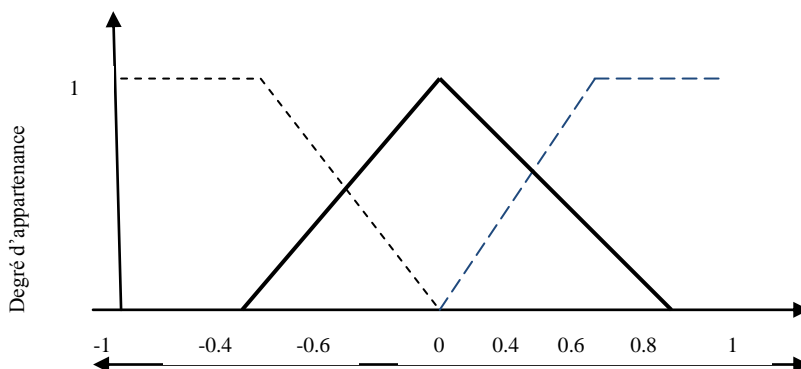
**Univers du discours :** gamme de température de 0°C à 200 °C

**Variable linguistique :** la température

**Valeur linguistique :** « très froid » « froid » « tempéré » « chaud » « très chaud »

**II.2.3.2 Univers de discours [11]**

L'univers de discours représente l'ensemble de référence ou le domaine de variation de la variable linguistique du processus dans le cas de réglage. Il est divisé en plusieurs sous-ensembles au moyen des fonctions d'appartenance triangulaires et trapézoïdales comme illustré sur la figure II.2



*Figure II.2: Univers de discours normalisé uniforme de 3 ensembles.*

### II.2.3.3 Les sous ensembles flous :

Dans la théorie classique des ensembles, un sous-ensemble  $A$  de  $B$  est défini par une fonction d'appartenance  $\mu_A(x)$  qui caractérise tout élément  $x$  appartenant à  $B$ . Cette fonction prend la valeur 1 si  $x$  appartient à  $A$  et la valeur 0 dans le cas contraire:

$$\mu_A(x) = \begin{cases} 1 & \text{Si } x \in A \\ 0 & \text{Si } x \notin A \end{cases}$$

En logique floue, un sous-ensemble flou  $A$  de  $B$  est défini par une fonction d'appartenance  $\mu_A(x)$  pouvant prendre différentes valeurs comprises entre 0 et 1, selon le degré d'appartenance de l'élément  $x$  au sous ensemble  $A$ .

$$\mu_A(x) \in [0, 1]$$

#### ❖ Les opérations sur les ensembles flous [11] :

**Egalité** :  $A=B$  si  $\forall x \in X : \mu_A(x) = \mu_B(x)$

**Inclusion** :  $A \subset B$  si  $\forall x \in X : \mu_A(x) \leq \mu_B(x)$

**Intersection** :  $A \cap B = C$  tel que :  $\forall x \in X : \mu_C(x) = \min(\mu_A(x), \mu_B(x))$

**Union** :  $A \cup B = D$  tel que :  $\forall x \in X : \mu_D(x) = \max(\mu_A(x), \mu_B(x))$

**Complément d'un sous ensemble** :  $\forall x \in X : \mu_{A^c}(x) = 1 - \mu_A(x)$

Cependant, il existe d'autres opérateurs utilisables pour définir ces opérateurs. Les plus connus sont les normes triangulaires pour l'intersection, les conformes triangulaires pour l'union et les négations pour le complément.

### II.2.3.4 Les fonctions d'appartenance [4]:

Définissent le degré de vérité ou d'appartenance d'une variable floue à un ensemble flou. Ils peuvent avoir diverses formes selon leur définition **Triangulaire, Trapézoïdale et gaussienne..... etc.**

- **Fonction triangulaire [4]** : Elle est définie par trois paramètres  $\{a, b, c\}$ , qui déterminent les coordonnées des trois sommets.

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

- **Fonction trapézoïdale** : Elle est définie par quatre paramètres  $\{a, b, c, d\}$  qui déterminent les coordonnées des quatre sommets.

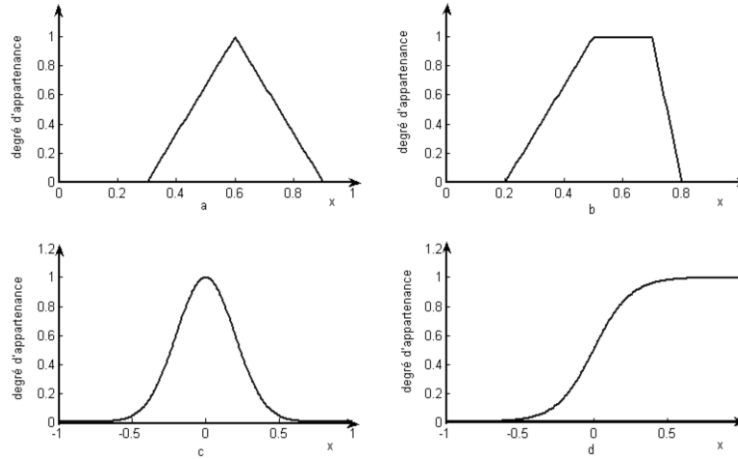
$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-b}\right), 0\right)$$

- **Fonction sigmoïde** : Elle est définie par deux paramètres  $\{a, b\}$

$$\mu(x) = \frac{1}{1 + \exp(-a(x - c))}$$

- **Fonction gaussienne** : Elle est définie par deux paramètres  $\{\sigma, m\}$

$$\mu(x) = \exp(-(x-m)^2/2\sigma^2)$$



**Figure II.3:** Formes des fonctions d'appartenance usuelle

### Choix de fonction d'appartenance :

La notion « d'univers de discours » se conçoit aisément : reprenons notre concept de température. L'utilisateur pourra décrire la variable <température> par un certain nombre de mots : par exemple <chaud>, <froid>, <tiède>, ou <très chaud>, <assez chaud>, <tiède>, <assez froid>, <très froid>. Pour chacun de ces prédicats, on pourra donner une fonction d'appartenance. L'univers de discours sera considéré comme le domaine de fonctionnement du processus.

Les problèmes qui peuvent se poser sont :

- Combien de prédicats sont nécessaires à la commande et comment les choisir ?
- Est-il nécessaire de choisir chaque prédicat ?

En général, on peut donner un nombre de règles de commande important, mais l'intérêt de la commande floue est que seul un petit nombre de règles est nécessaire. Le nombre de prédicats dépend essentiellement de la manière dont l'expert peut décrire le processus et de la précision souhaitée. Par exemple, en commande floue classique, 5 prédicats (grand, très grand, moyen, petit, très petit) est un bon compromis ; parfois trois peuvent suffire (dans le cas extrêmes, on peut aller jusqu'à 7). Les trois prédicats <petit>, <moyen>, <grand> ou <négatif>, <positif>, et <proche de zéro> servent à la régulation ; les autres prédicats servent à la poursuite. La commande floue permet d'assurer une transition harmonieuse entre les deux modes de fonctionnement.

L'univers de discours d'une variable couvrira l'ensemble des valeurs prises par cette variable. En pratique, l'intersection entre deux prédicats consécutifs est non nulle, de façon à pouvoir exercer une

pondération sur la commande. Il en résulte un chevauchement des variables qui doit être suffisant pour permettre une description continue des variables mais pas trop important pour limiter l'imprécision.

Il est notamment préférable d'éviter que les fonctions d'appartenance de deux prédicats voisins soient simultanément égales à 1.

Les fonctions d'appartenance seront choisies triangulaires et/ou trapézoïdales, symétriques et équidistantes.

**II.3 Réglage par logique floue:**

Le contrôleur par logique floue est chargé de calculer à chaque instant la commande réelle  $u$  (non flou) à appliquer au système afin de réaliser l'objectif de la commande qui est l'annulation de l'erreur de réglage  $e=y-r$  (la sortie mesurée – la référence)

Les entrées de réglage sont en général :

- La valeur réelle de l'erreur  $e$
- La dérivée de l'erreur  $\dot{e}$

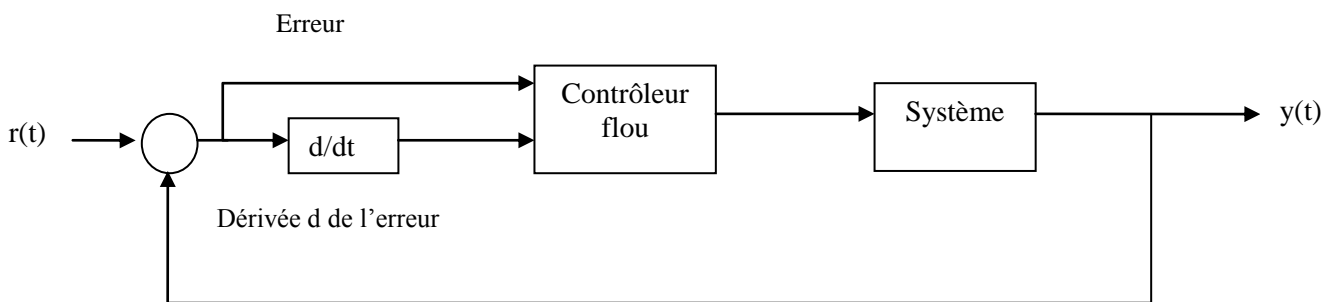


Figure II.4: Structure d'un réglage par logique floue

**II.3.1 La structure interne d'un contrôleur flou :**

Un système flou est un système à base de connaissances particulières, dont l'architecture de base illustrée par la figure ci-dessous, se compose essentiellement de trois étapes : fuzzification, raisonnement flou, défuzzification.

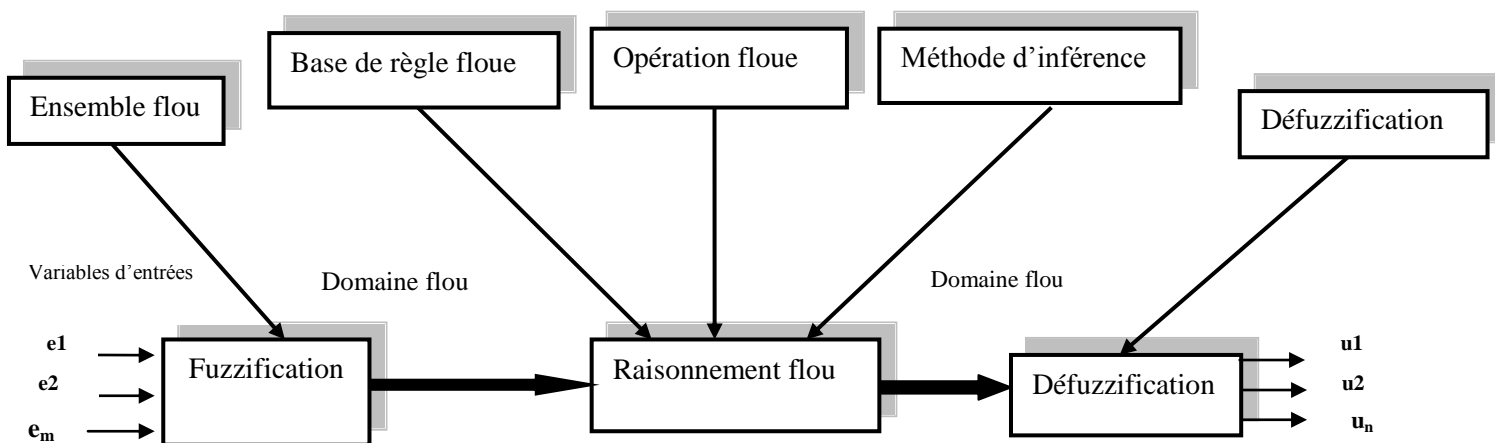
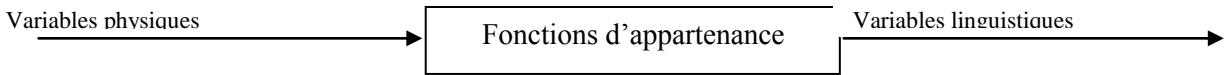


Figure II.5: Architecture de base d'un système flou

**II.3.1.1 Fuzzification :**



La fuzzification est un module de codage a pour rôle de convertir les données physiques réelles provenant des capteurs en un label caractérisant la variable linguistique associée pouvant être manipulés par le contrôleur, en définissant des fonctions d'appartenance pour différentes variables d'entrée. Le choix de la fonction d'appartenance dépend de la nature des données et du problème.

Supposons que l'ensemble des symboles relatifs à l'entrée x soit défini par  $L(x) = \{n, z, p\}$  où n= négative, z= zéro, p= positive et que l'univers de discours associé X, soit défini par  $X = [-0.4, 0.4]$ . La signification floue des différents symboles est donnée par  $\mu$ . Ainsi, la figure suivante montre la description floue d'une entrée  $x = 0.3$ .

$$D_{L(x)}(0.3) = 0/n + 0.25/z + 0.75/p$$

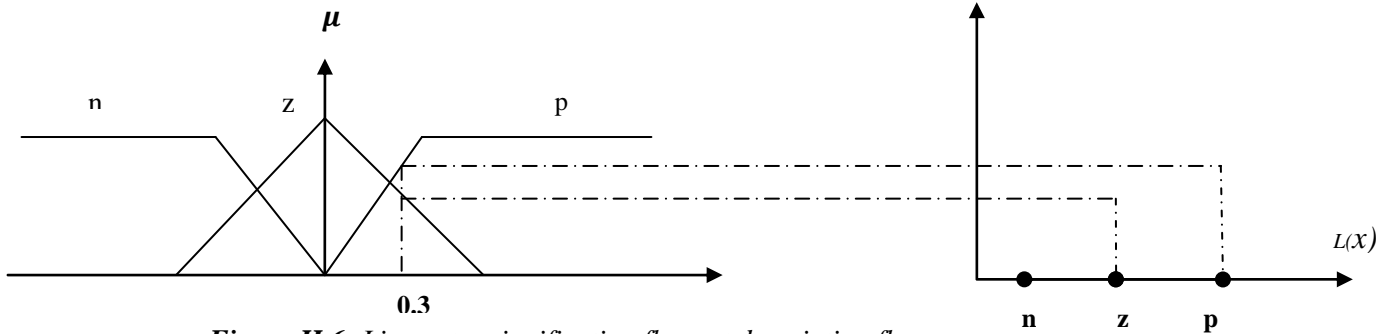


Figure II.6: Lien entre signification floue et description floue

**II.3.1.2 Raisonnement flou :**

C'est une base de connaissance qui contient les règles floues décrivant le comportement du système et les opérateurs logiques et les méthodes d'inférences. Il transforme à l'aide des opérateurs flous, la partie floue issue de la fuzzification en une partie floue conformément aux règles floues et une méthode d'inférence choisie, et l'agrégation des règles floues par l'opérateur flou ou.

**II.3.1.3 Les règles floues :**

Les règles floues permettent de décrire sous forme de règles linguistiques, une fonction de transfert entre les variables d'entrée et celles de sortie d'un système donné. Ces derniers à logique floue utilisent une expertise exprimée sous forme d'une base de règle du type : Si.....Alors.....

$$\text{Si}(x \text{ est } A) \text{ alors } (y \text{ est } B)$$

❖ **Exemple d'application :**

Règle de conduite automobile à l'approche d'un carrefour contrôlé par les feux tricolores résumés dans le tableau suivant [9]:

Si le feu est rouge ...	Si ma vitesse est élevée...	Et si le feu est proche...	Alors je freine fort.
Si le feu est rouge ...	Si ma vitesse est faible...	et si le feu est loin...	Alors je maintiens la vitesse.
Si le feu est orange...	Si ma vitesse est moyenne....	Si le feu est loin...	Alors je freine doucement.
Si le feu est vert...	Si ma vitesse est faible....	Et si le feu est proche...	Alors j'accélère.

Tableau II.1 : les règles floues en langage naturel

II.3.1.4 Les opérateurs flous :

Les règles floues font appel aux opérateurs flous présentés dans le tableau ci-dessous qui s'appliquent aux variables floues.

Opérateurs flous	Opération sur le degré de vérité des variables
<i>ET</i>	<i>Minimum</i>
	<i>Produit</i>
<i>OU</i>	<i>Maximum</i>
	<i>Valeur moyenne</i>
<i>ALORS (NON)</i>	<i>Complément à un</i>

Tableau II.2 : les operateurs flous

Les opérations minimum et maximum présentent l'avantage de la simplicité du calcul, par contre elles privilégient l'une des variables par rapport aux autres.

- **Operateur NON** : La propriété peut être caractérisée de façon évidente par la fonction d'appartenance :

$$\mu_B(x) = 1 - \mu_A(x)$$

A noter qu'il s'agit de l'opérateur **NON**, appelé aussi <complément>, <négation >ou <inverse>.

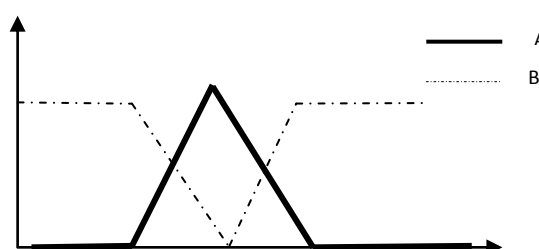


Figure II.7: Operateur de complémentation de la fonction d'appartenance A

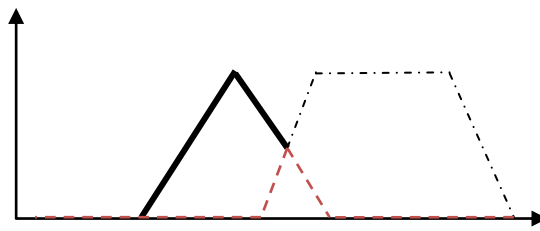
- **Opérateur ET :** La solution la plus simple et la plus utilisée pour caractériser la satisfaction simultanée de deux propriétés est de poser selon :

**Mamdani :**  $\mu_E(x) = \mu_A \text{ et } \mu_B(x) = \min \{ \mu_A(x), \mu_B(x) \}$

**Sugeno :**  $\mu_E(x) = \mu_A \text{ et } \mu_B(x) = \mu_A(x) \cdot \mu_B(x)$

On parle alors d'opérateur minimum.

Quant l'opération produit et l'opération valeur moyenne, elles sont les plus complexes à calculer, mais produisent un résultat qui tient compte des valeurs de toutes les variables.

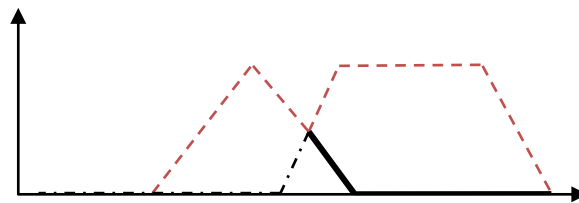


*Figure II.8: Fonction d'appartenance d'intersection*

- **Opérateur OU :** Il fait l'union des ensembles flous défini par la fonction d'appartenance:

**Mamdani :**  $\mu_E(x) = \mu_A(x) \text{ ou } \mu_B(x) = \max \{ \mu_A(x), \mu_B(x) \}$

**Sugeno :**  $\mu_E(x) = \mu_A(x) \text{ ou } \mu_B(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$



*Figure II.9: Fonction d'appartenance de l'union*

### II.3.1 .5 Critères de choix d'une méthode d'inférence [6]:

On utilisera en général la méthode **max-min**, celle-ci provoque une commande avec des sauts mais ce n'est en général pas ou peu contraignant. Il est à noter que la méthode somme-prod permet de développer un calcul analytique de la sortie permettant, si l'on possède un modèle du système, de calculer les caractéristiques temporelles de la sortie.

#### Remarque :

Dans la méthode de Mamdani, les prémisses et les conclusions des règles sont floues. L'inférence floue est réalisée à l'aide de l'opérateur min (minimum) ou prod (produit) et l'agrégation des règles à l'aide de l'opérateur max (maximum). La défuzzification est généralement effectuée par la méthode du centre de gravité.

Pour TSUKAMOTO, la fonction d'appartenance de chaque conclusion est une fonction monotone croissante. L'inférence floue est réalisée par l'opérateur min ou prod. La sortie numérique est une moyenne pondérée des valeurs issues des fonctions monotones.

La méthode de TAKAGI-SUGENO, est très efficace du point de vue de calcul car elle privilégie l'analyse mathématique, elle fonctionne bien avec les techniques linéaires, l'optimisation et les techniques dites adaptatives. L'inférence floue est réalisée avec l'opérateur min ou prod. Les conclusions des règles sont des valeurs numériques calculées à partir des entrées et la valeur finale est obtenue en effectuant une moyenne pondérée des conclusions.

**II.3.1.6 Méthodes d'inférences [6]:**

C'est étape où l'on établit les règles floues qui permettent d'aboutir à la commande en fonction des valeurs de l'erreur et de sa variation.

En général, la commande dépend non seulement de l'erreur mais aussi de sa variation. Dans le cas du pendule, il faut le ramener à la position de consigne d'autant plus énergiquement qu'il s'en éloigne et que sa vitesse est grande.

Les règles floues lient la variable de sortie aux variables d'entrée afin de tirer la conclusion floue.

**II.3.1.6.1 Méthode d'inférence max-min :**

Avec la méthode d'inférence max-min le mot **ALORS** est représenté par l'opération minimum de la valeur de la règle et de la fonction d'appartenance considérée de la sortie. Ainsi la règle : Si  $\epsilon$  est P ET  $\Delta\epsilon$  est Z ALORS u est P est traduite par la surface hachurée de la figure qui représente la fonction d'appartenance partielle.

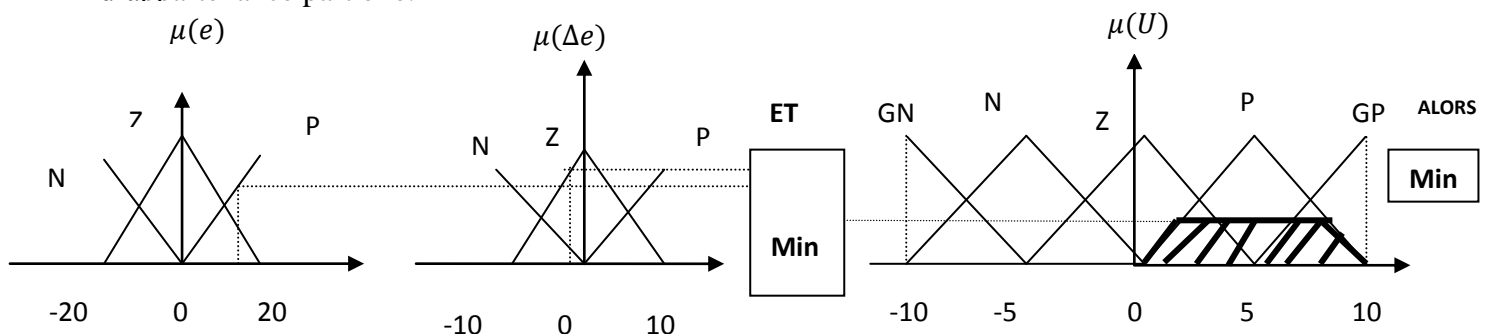


Figure II.10 : méthode d'inférence max-min

Par ailleurs l'interaction entre les règles sera nommée OU ainsi, on considère que les deux règles suivantes [6] :

- Si  $\epsilon$  est Z ET  $\Delta\epsilon$  est N ALORS u est N
- Si  $\epsilon$  est P ET  $\Delta\epsilon$  est Z ALORS u est P
- Si  $\epsilon$  est Z ET  $\Delta\epsilon$  est N ALORS u est N OU
- Si  $\epsilon$  est P ET  $\Delta\epsilon$  est Z ALORS u est P

Et ce **OU** se traduit par l'opération max.



**II.3.1.6.2 Méthode d'inférence max-prod :**

La différence avec la méthode précédente est la réalisation de « **ALORS** » qui se traduit par la multiplication de la fonction d'appartenance considérée par la valeur de la règle. Ce résultat est illustré sur la figure suivante

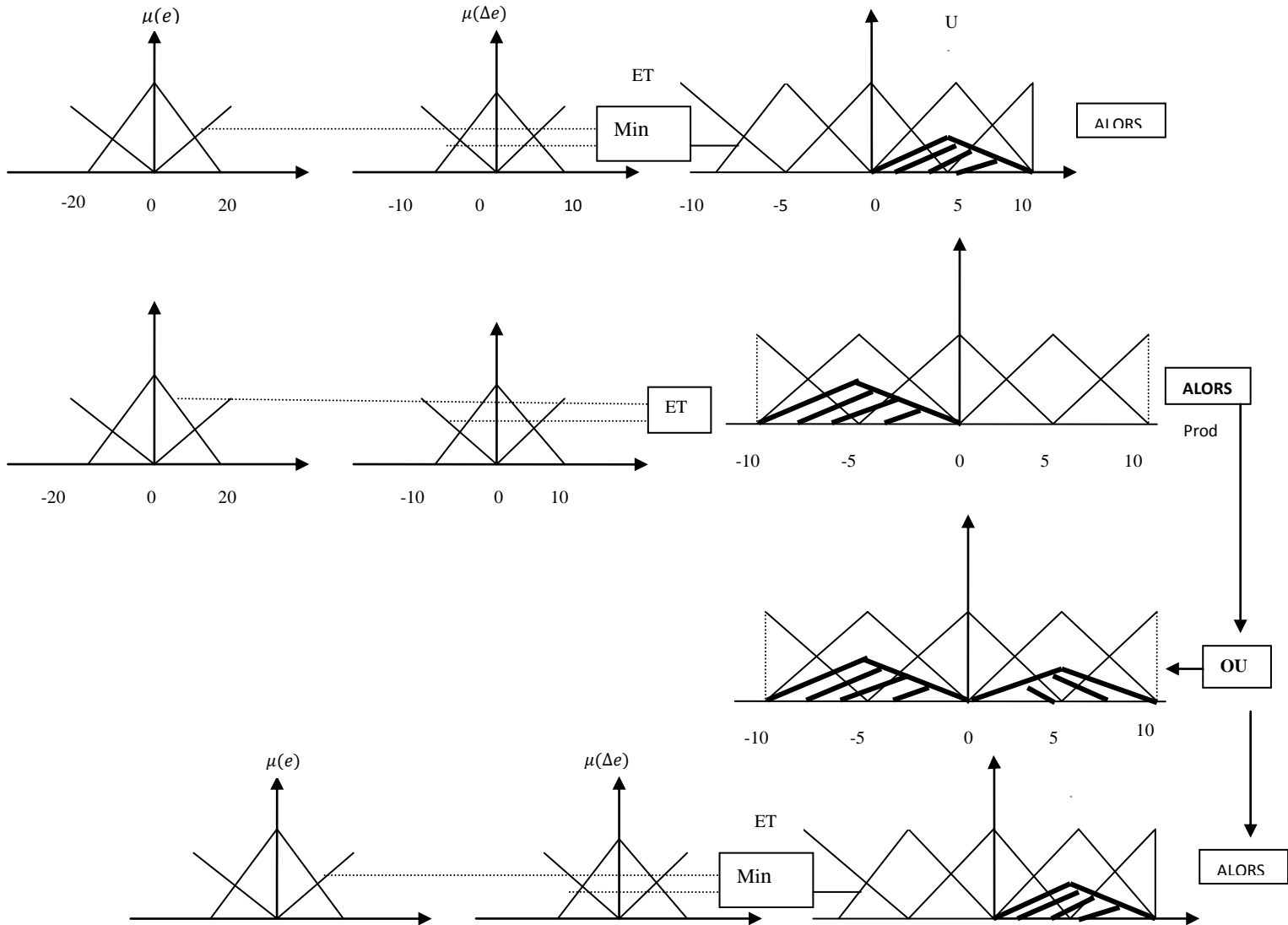
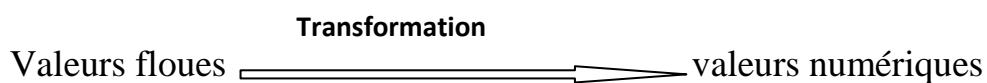


Figure II.13: Système d'inférence flou « max-prod » de Mamdani

**II.3.1.6.3 Méthode d'inférence somme-prod :**

Proche de la méthode précédente, il suffit de remplacer la traduction de **ALORS** par la somme des fonctions d'appartenance partielles].

**II.3.1.7 La défuzzification :**



La défuzzification est une interface « linguistique –numérique » qui transforme la partie floue issue du raisonnement flou en valeurs numériques directement exploitables par le processus. La méthode de

défuzzification la plus utilisée est celle du centre de gravité de la fonction d'appartenance résultante  $\mu_{\text{res}}(\mathbf{z})$ .

Dans ce contexte, il suffit de calculer l'abscisse  $\mathbf{z}^*$

L'abscisse du centre de gravité peut être déterminée à l'aide de la relation générale :

$$\mathbf{z}^* = \frac{\int \mathbf{z} \mu_{\text{res}}(\mathbf{z}) d\mathbf{z}}{\int \mu_{\text{res}}(\mathbf{z}) d\mathbf{z}}$$

**Remarque :** En commande floue, pour la défuzzification cette méthode est presque toujours utilisée. Elle prend en compte l'influence de l'ensemble des valeurs proposées par la solution floue.

#### II.4 Conception de contrôleur flou [2] :

Il y a cinq étapes nécessaires lors de la conception d'un contrôleur flou :

- Définition des entrées et des sorties du contrôleur :  
Nombres, noms, types, univers du discours.
- Subdivision de toutes les variables d'entrées et de sorties en sous ensembles flous :  
Nombre de subdivision, noms, types, paramètres.
- Définition de la base de règles :  
Nombres de règles, types, les combinaisons possibles, les résultats.
- Sélection de la méthode d'inférence.
- Sélection de la méthode de défuzzification.
- Fixation d'un ensemble de gains dit gains de normalisation (facteur d'échelle).

#### II.5 synthèse des contrôleurs flous :

##### II.5.1 Les systèmes experts:

Un système expert utilise la connaissance correspondante à un domaine spécifique afin de fournir une performance comparable à l'expert humain.

Les connaissances sont issues de l'expertise ou/et de la pratique. [2]

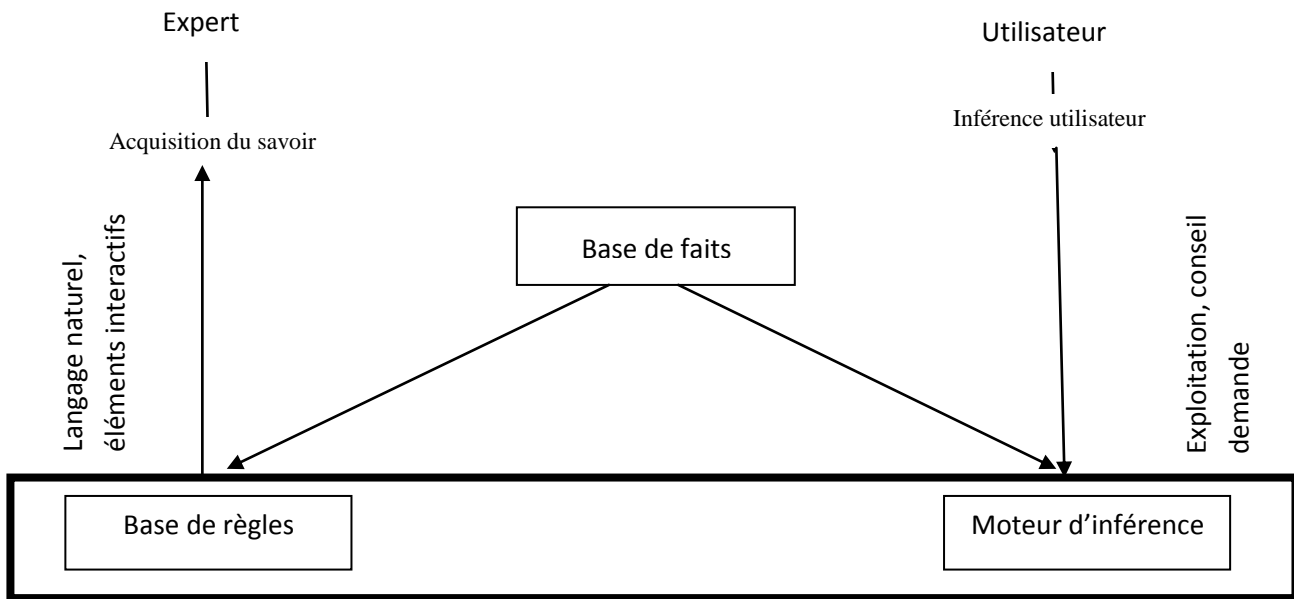


Figure II.14 : Structure d'un système expert

### II.5.2 Régulateur de type Mamdani :

De ce type de régulateur, les prémisses et les conclusions des règles floues sont floues. Le raisonnement flou est réalisé à l'aide de l'opérateur min (minimum) ou prod (produit), et l'agrégation des règles à l'aide de méthode de l'opérateur max (maximum) et la défuzzification est effectuée par la méthode du centre de gravité.

### II.5.3 Régulateur flou de type Sugeno :

Les régulateurs de Sugeno, est très efficace du point de vue calcul car il privilégie l'analyse mathématique, et fonctionne bien avec les techniques linéaires, adaptatives et l'optimisation. Le raisonnement flou est réalisé avec l'opérateur min ou prod. Les conclusions des règles sont des valeurs numériques calculées à partir des entrées, ainsi la valeur finale est obtenue en effectuant une moyenne pondérée des conclusions. Pour simplifier la compréhension et sans perte de généralité, concéderons un système à deux entrées  $x_1$  et  $x_2$  et une sortie  $y$ , modéliser par un système flou de type « **Sugeno** » composé de quatre règles suivantes:

- Si  $x_1$  est A1 et  $x_2$  est B1 alors  $y_1=f_1(x_1, x_2)=a_1x_1+b_1x_2+c_1$
- Si  $x_1$  est A1 et  $x_2$  est B2 alors  $y_2=f_2(x_1, x_2)=a_2x_1+b_2x_2+c_2$
- Si  $x_1$  est A2 et  $x_2$  est B1 alors  $y_3=f_3(x_1, x_2)=a_3x_1+b_3x_2+c_3$
- Si  $x_1$  est A2 et  $x_2$  est b2 alors  $y_4=f_4(x_1, x_2)=a_4x_1+b_4x_2+c_4$

La représentation de cette base de règle est donnée par la figure suivante :

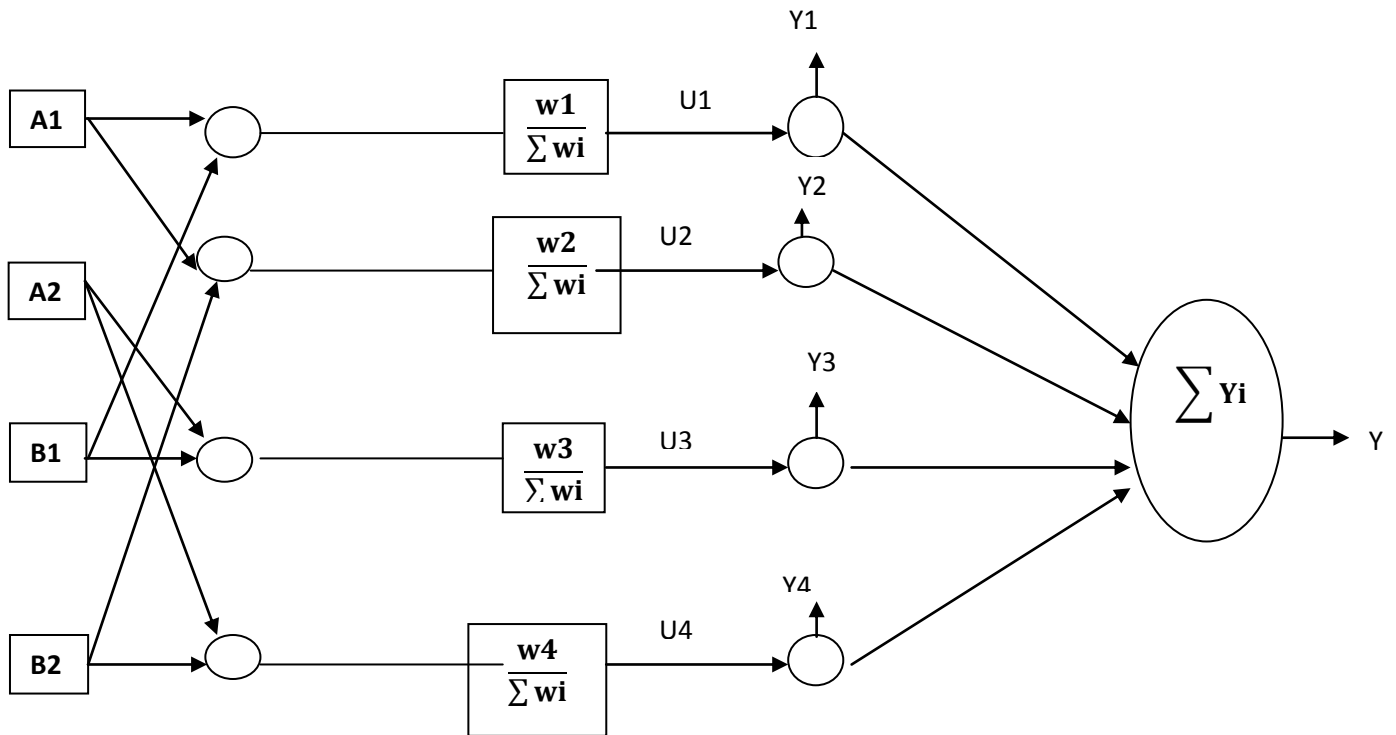


Figure I.15 : régulateur flou de type Sugeno

**II.6 Les avantage :**

**II.6.1 La méthode de Mamdani :**

- Elle est intuitive
- Elle a l'acceptation réponde
- C'est bien assorti à l'entrée humaine.

**II.6.2 La méthode de Sugeno :**

- Elle fonctionne bien avec les techniques linéaires
- Elle fonctionne bien l'optimisation et les techniques adaptatives
- Elle garantie la continuité de la surface de la sortie

**II.7 Détermination des paramètres d'adaptation d'un contrôleur flou en fonction des paramètres classiques :**

En premier lieu, il faut choisir le type de correcteur, puis en discrétisant le correcteur classique obtenu on détermine les relations donnant les paramètres d'adaptation du correcteur flou équivalent.

**II.7.1 Cas d'un régulateur proportionnel dérivé flou :**

Le principe du régulateur PD flou est de remplacer par la valeur défuzzification de la commande, la commande calculée grâce à un PD classique. L'algorithme de base d'un contrôleur PD classique est exprimé par :

$$U_{PD} = K_P e + K_d \dot{e} = K_P (e + T_d \dot{e})$$

Avec :

$T_d = K_d / K_P$  : la constante de temps dérivée. Si  $e$  et  $e'$  sont des variables floues alors  $U_{PD}$  de l'équation représentera la commande PD floue. On note  $K_e$  et  $K_d$  les gains des entrées du PD flou, alors que  $K_2$  représente le gain de la sortie. La sortie du PI flou est exprimée par :

$$U_{pd} = K_2 f \{ K_e e ; (K_d/T) \}$$

Le correcteur proportionnel dérivé flou a comme entrée : l'erreur «  $e$  » et sa dérivée «  $de$  » illustré par la figure :

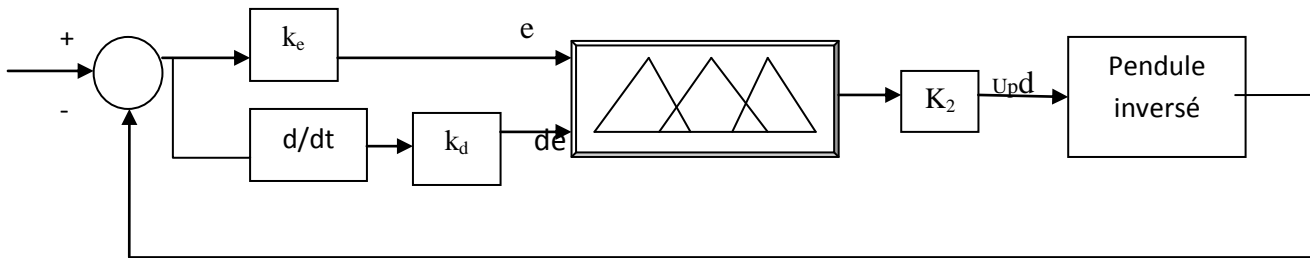


Figure II.16: structure d'un Pd flou

Dans ce cas le signal de commande  $u_{pd}$  est une fonction non linéaire en fonction de l'erreur «  $e$  » et sa dérivée «  $de$  ». L'expression de  $u_{pd}$  donnée comme suit :

$$U_{pd} = f(k_e * e, k_d * de) * k_2$$

Par identification à l'expression de proportionnel dérivé classique :

$k_e * k_2 = k_c$ ;  $k_d / k_e = T_d$  tel que  $k_c$  représente le gain d'un Pd classique et  $T_d$  la constante de temps.

## II.8 Avantages et désavantages du réglage par la logique floue :

Evidemment, le réglage par logique floue réunit un certain nombre d'avantage et de désavantage.

Les avantages essentiels sont :

- Le non nécessité d'une modélisation (cependant, il peut être utile de disposer d'un modèle convenable).
- La maîtrise du système à régler avec un comportement complexe (fortement non linéaire et difficile à modéliser).
- L'obtention fréquente de meilleures prestations dynamiques (régulateur non linéaire).
- La disponibilité de système de développement efficace, soit pour microprocesseur ou PC (solution logicielle), soit pour circuit intégrés (processeurs dédiés, fuzzy processors).

Par contre les désavantages sont :

- Le manque de directives précises pour la conception d'un réglage (choix des grandeurs à mesurer, détermination de la fuzzification, des inférences et de la défuzzification).

- L'approche artisanale et non systématique (implémentation des connaissances de l'opérateur est souvent difficile).
- L'impossibilité de la démonstration de la stabilité du circuit de réglage en toute généralité (en l'absence d'un modèle valable).
- La précision de réglage souvent peut élever.
- La cohérence des inférences non garantie a priori (apparition de règles d'inférence contradictoires possible).

## II.9 Exemple d'application :

On veut commander un pendule inversé par la logique floue à fin d'améliorer les performances du système mais en premier, on va le présenter et le modéliser.

### II.9.1 Présentation du système

Le pendule inversé est une plate forme d'essai qui pose un problème d'instabilité à l'angle  $\theta = 0$ , et qui se compose de deux parties, mécanique (chariot, pendule, ....etc.), et partie électrique (moteur à courant continue, amplificateur ... etc.)

Le pendule inversé utilisé est constitué d'un chariot mobile en translation sur un axe horizontal, et un pendule fixé sur ce chariot libre en rotation.

En exerçant une force horizontale  $F(t)$  sur le chariot, une translation de  $x$  mètres de ce système est obtenue avec une rotation de  $\theta$  radians du pendule.

Le pendule inversé est un système instable en boucle ouverte, non linéaire et multivariable. La commande de ce système doit réaliser, la régulation de l'angle  $\theta$ , en partant d'une condition initiale comprise dans l'intervalle

$]-\pi/2, +\pi/2[$ , et la poursuite sur la position  $x$ , en partant d'une condition initiale comprise dans l'intervalle  $[-0.5m, +0.5m]$ .

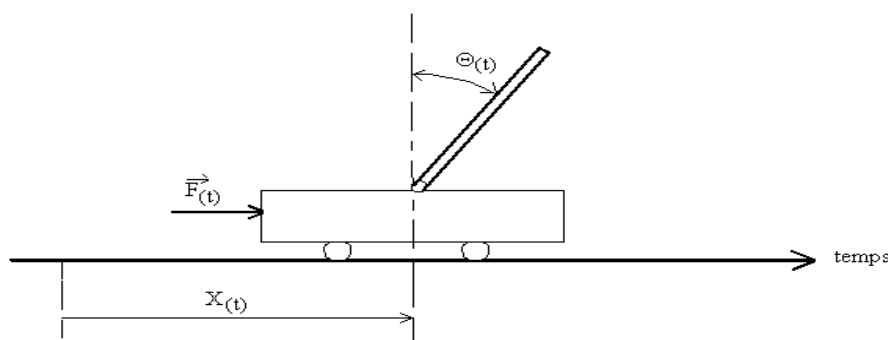


Figure II.17: Schéma simplifié d'un pendule inversé

Soit :

$m$  : masse de pendule

$M$  : masse de chariot

$l$  : demi longueur de pendule

$U$  : tension de commande du MCC

$f$  : frottements de déplacement du chariot

$r(t)$  : position de chariot

$\varphi(t)$  : l'angle de pendule

$g$  : intensité de pesanteur

Soit le vecteur d'état :

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} r \\ \varphi \\ \dot{r} \\ \dot{\varphi} \end{bmatrix}$$

L'équation de Lagrange de cette pendule inversée est la suivante :

$$L = \frac{1}{2}(m + M) \cdot \dot{X}(t)^2 + \frac{1}{2}ml^2 \dot{\theta}(t)^2 - lm\dot{X}(t)\dot{\theta}(t)\cos(\theta(t)) - mgl(1 + \cos(\theta(t)))$$

ET en résolvant le système d'équations différentielles suivant:

$$\begin{cases} \frac{d}{dt} \left( \frac{\delta L}{\delta \dot{X}} \right) - \frac{\delta L}{\delta X} = F(t) - f_x \dot{X} \\ \frac{d}{dt} \left( \frac{\delta L}{\delta \dot{\theta}} \right) - \frac{\delta L}{\delta \theta} = 0 \end{cases}$$

On obtient le modèle d'état suivant :

$$\begin{cases} \dot{x}_1 = x_3 \\ \dot{x}_2 = x_4 \\ \dot{x}_3 = \frac{a_{32} \sin(x_2) \cos(x_2) + a_{33}x_3 + a_{34}x_4 \cos(x_2) + a_{35}x_4^2 \sin(x_2) + b_3.u}{1 - a_{45}(\sin(x_2))^2} \\ \dot{x}_4 = \frac{a_{42} \sin(x_2) + a_{43}x_3 \cos(x_2) + a_{44}x_4 + a_{45}x_4^2 \sin(x_2) \cos(x_2) + b_4 \cos(x_2).u}{1 - a_{45}(\sin(x_2))^2} \end{cases}$$

Les coefficients  $a_{ij}$  et  $b_i$  [3] pour  $i = 3,4$  et  $j = 2, \dots, 5$  sont donnés par :

$$a_{32} = -1.93, a_{33} = -1.99, a_{34} = 0.009, a_{35} = 0.063,$$

$$a_{42} = 36.99, a_{43} = 6.26, a_{44} = -0.174, a_{45} = -0.1972,$$

$$b_1 = 0.5128, b_2 = -1.61.$$

**II.9.2 Commande du pendule par un proportionnel dérivé flou de type Mamdani:**

Dans cette application on veut ramener le pendule inversé à sa position verticale  $\varphi=0$  pour cela on a introduit le pd flou qui améliore la précision et la rapidité du système en suivant les étapes des contrôleurs flous:

**II.9.2.1 fuzzification:**

On a choisit comme entrées la position du chariot et sa vitesse, l'angle et la vitesse angulaire du pendule et pour la sortie le couple du moteur.

Pour les fonctions d'appartenances on a utilisé les fonctions de type **triangulaire** pour les entrées et même pour la sortie.

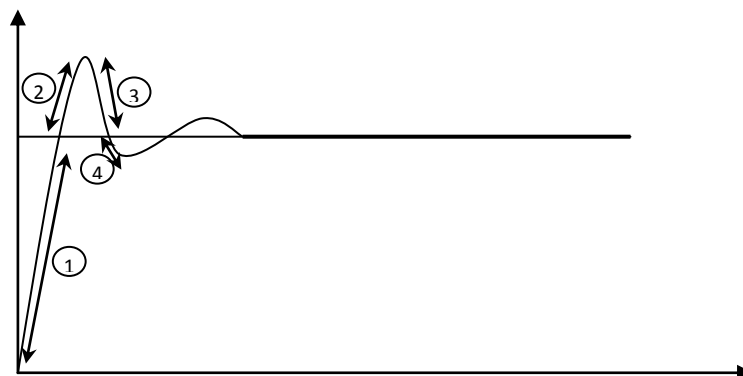
**II.9.2.2 Les règle d'inférence :**

Cette étape concerne l'élaboration de la base de règle du contrôleur. C'est de nouveau à un expert et à sa connaissance du problème que l'on se fiera le plus souvent. Dans le cadre de la régulation, nous utilisons fréquemment l'erreur et sa variation. À partir de ces deux entrées, traduites sous forme de variables floues, il est possible de déterminer les règles dans le domaine temporel et on peut construire une matrice comprenant toutes les possibilités linguistiques de ces règles.

L'analyse temporelle, qui doit conduire à établir les règles du contrôleur flou, peut par exemple consister à considérer la réponse à un échelon d'un processus à piloter en fonction des objectifs qui seront fixés en boucle fermé.

La forme générale de la réponse est représentée par la figure II.16, selon l'amplitude de  $e$  et le signe de  $\Delta e$ , cette réponse est divisée en quatre régions.

- 1) :  $e > 0$  et  $\Delta e < 0$
- 2) :  $e < 0$  et  $\Delta e < 0$
- 3) :  $e < 0$  et  $\Delta e > 0$
- 4) :  $e > 0$  et  $\Delta e > 0$



**Figure II.18:** Ecriture du jeu de règles grâce à une analyse temporelle.

Le problème de commande du pendule est loin d'être trivial, car deux variables sont à réguler mais l'on ne dispose que d'une seule action de commande. La dynamique de  $\varphi$  étant plus rapide que celle de  $r$ ,

la stratégie adoptée est la suivante : nous chercherons dans un premier temps à équilibrer le pendule (position verticale haute), puis à l'amener progressivement vers sa position de référence en le déséquilibrant du bon côté, c'est à dire vers celui qui entraînera une valeur de commande telle que le pendule se déplacera vers la référence.

Nous pouvons alors écrire les règles suivantes afin de commander le pendule à la fois horizontalement et verticalement.

$\Delta\phi$ \ / \ $\phi$	N	Z	P
N	N	N	P
Z	N	Z	P
P	N	P	P

**Tableau II.3:** Base de règle du contrôleur flou

Si l'angle est négative  $\phi$  et la vitesse angulaire  $\Delta\phi$  est négative alors le couple  $u$  négatif.

:

Si l'angle est positive  $\phi$  et la vitesse angulaire  $\Delta\phi$  est positive alors le couple  $u$  positif.

### II.9.2.3 déffusification :

Choix d'une méthode de défuzzification, du mécanisme d'inférence, ainsi des valeurs des facteurs d'échelle.

La méthode de défuzzification utilisée dans notre travail est la méthode du centre de gravité.

Pour le mécanisme d'inférence, nous avons opté pour le mécanisme d'inférence MAX-MIN.

Le choix des facteurs d'échelle résulte après différents essais de simulation afin d'avoir les meilleurs résultats.

#### Limitation physique

Les limitations physiques du pendule inversé ici considéré sont  $r \in [-0.5, +0.5]$  mètres et  $u \in [-10, +10]$  volts.

### II.10 Optimisation par les algorithmes génétiques :

Les algorithmes génétiques sont une méthode d'optimisation utile dans les cas non linéaires (évidemment, cette méthode fonctionne également pour les cas linéaires, mais dans ce cas, est inutilement lourde en temps et en calcul). Cette technique part du principe évolutif de la sélection naturelle de Darwin. Celle-ci énonçait que les individus les plus aptes à survivre (les « meilleurs ») se reproduiront plus souvent et auront plus de descendants. Ainsi, la qualité du pool génétique de la population sera augmentée, les gènes plus efficaces deviendront plus fréquent; la population s'améliore. Selon le même

principe, un algorithme génétique part d'une population de solutions initiales, les fait se reproduire (les meilleures solutions ont plus de chances de se reproduire), créant ainsi la nouvelle génération de solutions. En répétant ce cycle plusieurs fois, on obtient une population composée de solutions meilleures. On utilise généralement les algorithmes génétiques pour trouver une solution, la meilleure solution après un certain nombre de générations.

### II.10.1 Terminologie :

- Les chromosomes : sont les éléments à partir desquels sont élaborées les solutions (mutation et croisement génétiques).
- La population : (génération) est l'ensemble des chromosomes.
- La reproduction : est l'étape de combinaison des chromosomes.

### II. 10.2 Principe :

Pour utiliser les techniques génétiques il faut faire deux opérations :

- 1- Une fonction de codage de données en entrée sous forme d'une séquence de bits.
- 2- Trouver une fonction  $U(x)$  pour pouvoir calculer l'adaptation d'une séquence de bits  $x$ .

Après avoir trouvé ces deux fonctions on peut appliquer l'AG :

- a- Générer aléatoirement quelques séquences de bits pour composer la soupe initiale.
- b- Mesurer l'adaptation de chacune des séquences présentes.
- c- Reproduction des séquences en fonction de son adaptation.
- d- Faire l'opération de croisement aléatoirement de quelque paire de séquences, et ce sera :
  - une séquence est composée de la 1<sup>ère</sup> partie de la 1<sup>ère</sup> séquence et de la 2<sup>nd</sup> partie de la 2<sup>nd</sup> séquence.
  - et une séquence est composée de la 2<sup>nd</sup> partie de la 1<sup>ère</sup> séquence et de la 1<sup>ère</sup> partie de la 2<sup>nd</sup> séquence.
- e- Faire l'opération de mutation d'un bit choisi aléatoirement dans une ou plusieurs séquences.
- f- Retour à l'étape 2 (mesurer l'adaptation à nouveau).

### II.10. 3 But :

Les AG ont deux objectifs principaux :

- Expliciter les théories sur les adaptations des systèmes naturels.
- Résoudre des problèmes dont les AG sont particulièrement adaptées. Les AG présentent à la fois robustesse et adaptation. C'est à dire qu'ils ont la capacité d'évoluer (fitness) dans le but d'une recherche satisfaisante. Mais ils sont aussi capables d'être attirés par des solutions et d'y rester.

### II.10.4 Applications :

Les applications des AG sont multiples :

- optimisation de fonctions numériques difficiles (discontinues...)

- traitement d'image (alignement de photos satellites, reconnaissance de suspects...)
- optimisation d'emplois du temps, optimisation de design, contrôle de systèmes industriels - apprentissage des réseaux de neurones etc.
- Les AG peuvent être utilisés pour contrôler un système évoluant dans le temps (chaîne de production, centrale nucléaire...) car la population peut s'adapter à des conditions changeantes. Ils peuvent aussi servir à déterminer la configuration d'énergie minimale d'une molécule.
- Les AG sont également utilisés pour optimiser des réseaux (câbles, fibres optiques, mais aussi eau, gaz...),
- des antennes .Ils peuvent être utilisées pour trouver les paramètres d'un modèle petit-signal à partir des mesures expérimentales

**II.10.5 les opérateurs :**

**II.10.5.1 Codage binaire et réel des variables :**

Pour résoudre un problème il faut d'abord coder les paramètres, un gène Correspond à une variable d'optimisation  $X_i$ , et un ensemble de gène correspond à un chromosome un individu à un ou plusieurs chromosomes et une population c'est un ensemble des individus.

Dans l'informatique nous utilisons un codage binaire (0 et 1), par exemple un gène est un entier long (32 bits).

Les cinq niveaux d'organisation schéma du codage de variable d'optimisation  $X_i$  de notre AG

Un des avantages du codage binaire est que l'on peut facilement coder les objets : réels, des entiers, des chaînes de caractères ...etc.

Pour résoudre il nous faut un espace de recherche fini :

$$0 < g_i < g_{max} \quad i = 1 \text{ à } n$$

Chaque gène est codé par 32 bits.



Entier long (32 bits) 
$$g_i = \sum_{j=0}^{31} b_j \cdot 2^j$$

Les formules de codage et de décodage sont :

$$g_i = (x_i - x_{imin} / x_{imax} - x_{imin}) g_{max}.$$

$$x_i = x_{imin} + (x_{imax} - x_{imin}) g_i / g_{max}$$

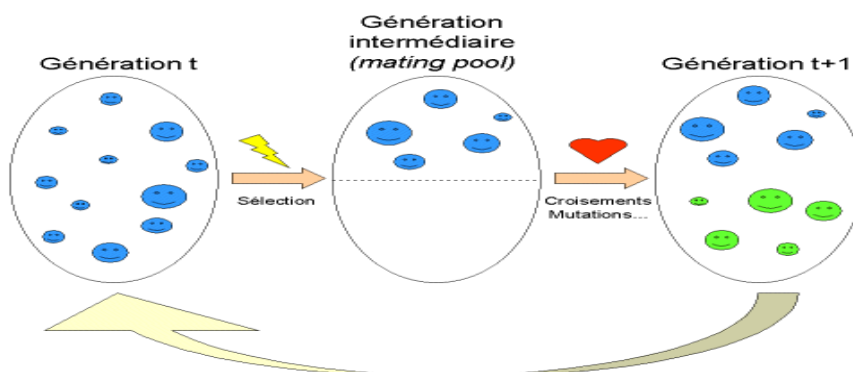
Contrairement au codage binaire, dans le codage réel il n'y a pas d'opération de conversion vers le réel et le binaire, et alors l'AG soit plus dépendant du problème, chacune des composantes correspond à une inconnue du problème. Et pour l'insertion des chromosomes soient :

- ✓ Par l'insertion des chromosomes directement dans la population de la génération suivante.

Par "l'élitisme" : c'est l'insertion du meilleur chromosome dans la population de la génération suivante et de compléter l'individu par des chromosomes d'une manière traditionnelle.

- ✓ Par "la population sans double" : c'est l'insertion du chromosome dans la population de la génération suivante à condition qu'il soit différent de tous les chromosomes de la nouvelle génération.

### II.10.5.2 Sélection :



**Figure II.19 :** Représentation schématique du fonctionnement de l'AG

Il y a plusieurs méthodes de sélection, citons quelques-unes :

#### II.10.6 .2.1 Roulette de casino :

C'est la sélection naturelle la plus employée pour l'AG binaire. Chaque chromosome occupe un secteur de roulette dont l'angle est proportionnel à son indice de qualité. Un chromosome est considéré comme bon aura un indice de qualité élevé, un large secteur de roulette et alors il aura plus de chance d'être sélectionné.

#### II.10.6.2.2 Elitisme " :

Les individus sont triés selon leur fonction d'adaptation, seul la moitié supérieure de la population correspondant aux meilleurs composants est sélectionnée, nous avons constaté que la pression de sélection est trop forte, il est important de maintenir une diversité de gènes pour les utiliser dans la population suivante et avoir des populations nouvelles quand on les combine.

#### II.10.6.2.3] "par tournoi" :

Choisir aléatoirement deux individus et on compare leur fonction d'adaptation (combattre) et on accepte la plus adaptée pour accéder à la génération intermédiaire, et on répète cette opération jusqu'à

remplir la génération intermédiaire ( $N/2$  composants). Les individus qui gagnent à chaque fois on peut les copier plusieurs fois ce qui favorisera la pérennité de leurs gènes.

### II.10.6.3 Croisement :

Le phénomène de croisement est une propriété naturelle de l'ADN, et c'est analogiquement qu'on fait les opérations de croisement dans les AG.

#### II.10.6.3.1 croisement binaire :

##### II.10.6.3.1.1 " croisement en un point " :

On choisit au hasard un point de croisement, pour chaque couple (fig. II.18). Notons que le croisement s'effectue directement au niveau binaire, et non pas au niveau des gènes. Un chromosome peut donc être coupé au milieu d'un gène.

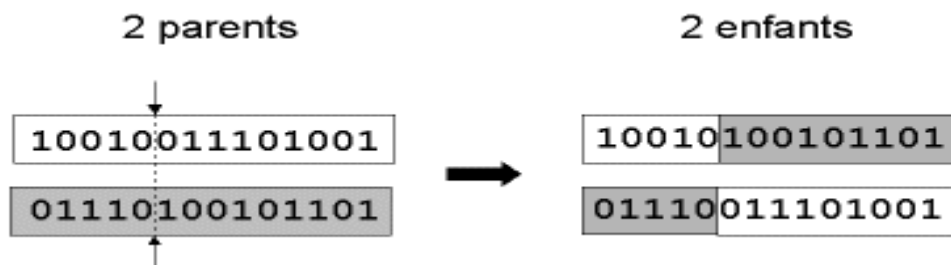


Figure II. 20 : représentation schématique du croisement en 1 point.

Les chromosomes sont bien sûr généralement beaucoup plus longs.

##### II.10.6.3.1.2 " croisement en deux points " :

On choisit au hasard deux points de croisement (Fig. II.19). Par la suite, nous avons utilisé cet opérateur car il est généralement considéré comme plus efficace que le précédent [Beasley, 1993b]. Néanmoins nous n'avons pas constaté de différence notable dans la convergence de l'algorithme.

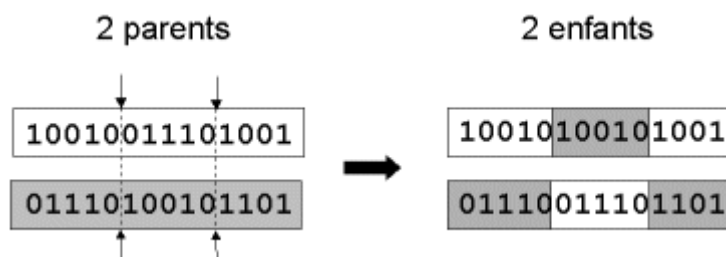


Figure II.21: représentation schématique du croisement en 2 points.

Notons que d'autres formes de croisement existent, du croisement en  $k$  points jusqu'au cas limite du croisement uniforme.

#### II.10.6.3.2 Croisement réel :

Le croisement réel ne se différencie du croisement binaire que par la nature des éléments qu'il altère : ce ne sont plus des bits qui sont échangés à droite du point de croisement, mais des variables réelles.

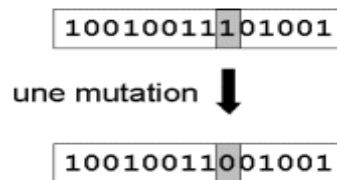
### II.10.6.3.3 Croisement arithmétique :

Le croisement arithmétique est propre à la représentation réelle. Il s'applique à une paire de chromosomes et se résume à une moyenne pondérée des variables des deux parents. Soient  $[a_i, b_i, c_i]$  et  $[a_j, b_j, c_j]$  deux parents, et  $p$  un poids appartenant à l'intervalle  $[0, 1]$ , alors les enfants sont  $[pa_i + (1-p)a_j, pb_i + (1-p)b_j, pc_i + (1-p)c_j]$  ...

Si nous considérons que  $p$  est un pourcentage, et que  $i$  et  $j$  sont nos deux parents, alors l'enfant  $i$  est constitué à  $p\%$  du parent  $i$  et à  $(100-p)\%$  du parent  $j$ , et réciproquement pour l'enfant  $j$ .

### II.10 6.4 Mutation :

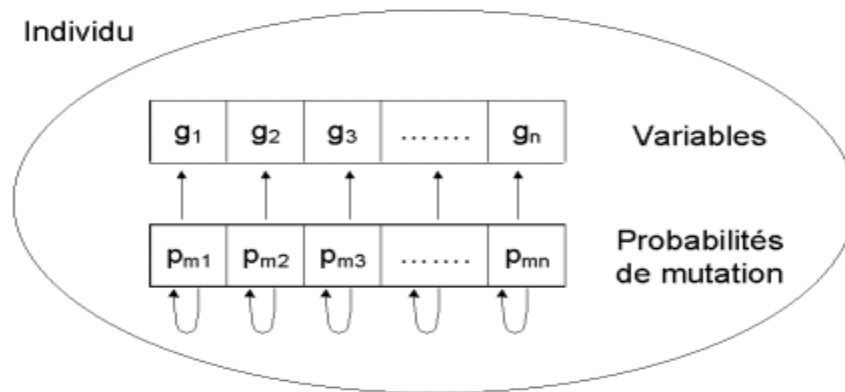
Nous définissons une mutation comme étant l'inversion d'un bit dans un chromosome (Fig. II.20). Cela revient à modifier aléatoirement la valeur d'un paramètre du dispositif. Les mutations jouent le rôle de bruit et empêchent l'évolution de se figer. Elles permettent d'assurer une recherche aussi bien globale que locale, selon le poids et le nombre des bits mutés. De plus, elles garantissent mathématiquement que l'optimum global peut être atteint.



**Figure II. 22:** représentation schématique d'une mutation dans un chromosome.

D'autre part, une population trop petite peut s'homogénéiser à cause des erreurs stochastiques : les gènes favorisés par le hasard peuvent se répandre au détriment des autres. Cet autre mécanisme de l'évolution, qui existe même en l'absence de sélection, est connu sous le nom de dérive génétique. Du point de vue du dispositif, cela signifie que l'on risque alors d'aboutir à des dispositifs qui ne seront pas forcément optimaux. Les mutations permettent de contrebalancer cet effet en introduisant constamment de nouveaux gènes dans la population

Comment réaliser notre opérateur mutation ? De nombreuses méthodes existent. Souvent la probabilité de mutation  $p_m$  par bit et par génération est fixée entre 0,001 et 0,01. On peut prendre également  $p_m=1/l$  où  $l$  est le nombre de bits composant un chromosome. Il est possible d'associer une probabilité différente de chaque gène. Et ces probabilités peuvent être fixes ou évoluer dans le temps.



*Figure II. 23: principe de l'auto-adaptation.*

A chaque variable est associée sa propre probabilité de mutation, qui est elle-même soumise au processus d'évolution.

L'individu possède donc un second chromosome codant ces probabilités.

Après divers essais, ils ont abouti à la méthode d'auto-adaptation des probabilités de mutation. Si dans un environnement stable il est préférable d'avoir un taux de mutation faible, la survie d'une espèce dans un environnement subissant une évolution rapide nécessite un taux de mutation élevé permettant une adaptation rapide. Les taux de mutation d'une espèce dépendent donc de leur environnement.

Pour prendre en compte cette formulation biologique et l'adapter à notre cas, ils ont introduit dans chaque individu (dispositif) un second chromosome (ensemble de paramètres) dont les gènes (paramètres) représentent les probabilités de mutation de chaque gène du premier chromosome (Fig. II.21). Ce second chromosome est géré de façon identique au premier, c'est-à-dire qu'il est lui-même soumis aux opérateurs génétiques (croisement et mutation). Cela revient à fixer les probabilités assurant la modification des valeurs des paramètres du composant en fonction des valeurs d'un ensemble d'autres paramètres (les probabilités de mutation).

Lors de la genèse, les probabilités de mutation sont posées égales à 0,1 (valeur qui a paru la meilleure après plusieurs essais). Au cours du déroulement de l'algorithme, les gènes et les individus ayant des probabilités de mutation trop élevées ont tendance à disparaître. De même, les gènes ayant des probabilités de mutation trop faibles ne peuvent pas évoluer favorablement et tendent à être supplantés. Les probabilités de mutation dépendent donc du gène considéré et de la taille de la population. De plus, elles évoluent au cours du temps. Il y a donc auto-adaptation des probabilités de mutation.

#### **II.10 5.4.1 Mutation binaire :**

La mutation binaire s'applique à un seul chromosome. Un bit du chromosome est tiré au hasard. Sa valeur est alors inversée.

Il existe une variante où plusieurs bits peuvent muter au sein d'un même chromosome. Un test sous le taux de mutation est effectué non plus pour le chromosome mais pour chacun de ses bits : en cas de succès, un nouveau bit tiré au hasard remplace l'ancien.

#### **II.10. 6.4.2 Mutation réelle :**

La mutation réelle ne se différencie de la mutation binaire que par la nature de l'élément qu'elle altère : ce n'est plus un bit qui est inversé, mais une variable réelle qui est de nouveau tirée au hasard sur son intervalle de définition.

#### **II.10. 6.4.3 Mutation non uniforme :**

La mutation non uniforme possède la particularité de retirer les éléments qu'elle altère dans un intervalle de définition variable et de plus en plus petit. Plus nous avançons dans les générations, moins la mutation n'écarte les éléments de la zone de convergence. Cette mutation adaptative offre un bon équilibre entre l'exploration du domaine de recherche et un affinement des individus. Le coefficient d'atténuation de l'intervalle est un paramètre de cet opérateur

### **II.10.7 Explications des éléments d'un algorithme génétique [15] :**

#### **II.10.7.1 Individu :**

Les individus correspondent aux « solutions » du problème à optimiser. Ces solutions doivent être « codées » pour que le traitement puisse être effectué par l'algorithme génétique. Cette représentation codée d'une solution est appelée chromosome, et est composée de gènes. Chaque gène peut représenter une variable, un élément de la solution, ou encore une partie plus abstraite. La manière la plus utilisée de codage par algorithme génétique est le codage en vecteurs. Chaque solution est représentée par un vecteur. Ce vecteur peut être binaire ou encore de n'importe quel type discret dénombrable (entier, caractères, etc.). On pourrait également utiliser un type continu (ex : nombres réels), mais dans ce cas, il faut également revoir les opérations qui modifient le contenu des chromosomes (la fonction qui crée aléatoirement les chromosomes et les opérateurs génétiques). La simplicité veut que les chromosomes soient uniformes, c'est-à-dire que tous les gènes sont du même type. Cependant, si on tient compte encore une fois des opérations qui modifient le contenu des chromosomes, on peut assez aisément construire des vecteurs d'éléments de type différents. On demande habituellement que les chromosomes soient tous de même longueur, basés sur la même architecture, les gènes homologues étant au même endroit sur leur chromosome respectif. De fait, le codage par vecteur est si utilisé (très grande simplicité comparée aux autres méthodes de codage de chromosome) que les algorithmes sont souvent identifiés comme étant des méthodes de traitement vectoriel. Cette affirmation n'est pas tout à fait vraie car d'autres types de codage existent, bien que n'étant pas très fréquents. Par exemple, l'utilisation des

algorithmes génétiques pour faire de la programmation génétique utilise un codage en arbre (ce qui permet entre autres d'avoir des chromosomes de longueurs différentes).

#### **II.10.7.2 Population :**

C'est l'ensemble des individus, ou encore l'ensemble des chromosomes d'une même génération. Habituellement, la taille de la population reste constante tout au long de l'algorithme génétique.

#### **II.10.7.3 Générer :**

Habituellement, au départ d'un algorithme génétique, il faut créer une population d'individus. Ces individus sont générés par une fonction simple. Cette fonction affecte à chaque individu qu'elle crée une valeur aléatoire pour chacun de ses gènes. L'algorithme génétique peut également utiliser comme population de départ une population déjà créée a priori.

#### **II.10.7.4 Qualité, *fitness*, d'un individu :**

Le calcul de la qualité d'un individu est essentiel aux algorithmes génétiques. Cette fonction donne, en valeur numérique (habituellement réelle), la qualité d'un individu. C'est selon cette valeur numérique que sont calculées les chances de sélection de cet individu. La fonction de *fitness* doit avoir 0 comme plancher, pour ne pas fausser le calcul des pourcentages. Les algorithmes génétiques étant une technique d'optimisation, ils cherchent la qualité maximale, donc l'optimisation de la fonction de qualité. Si on cherche plutôt à minimiser une fonction, il faudra la modifier de sorte que la fonction de qualité se maximise. Il serait bien entendu possible de conserver une fonction de qualité qui fonctionne à l'envers et de modifier à la place le calcul des probabilités, mais ceci rendrait l'algorithme beaucoup plus difficile à décoder pour les utilisateurs externes.

#### **II.10.7.5 Sélection :**

Selon la qualité des individus, chacun se voit attribuer un pourcentage de chances d'être choisi pour la reproduction, qui correspond à l'importance relative de la qualité de l'individu par rapport à la qualité totale de la population.

#### **II.10.7.6 Reproduction :**

La reproduction s'effectue généralement en croisant deux individus, ce qui produit deux nouveaux individus à placer dans la nouvelle population. De la manière classique, la reproduction consiste à appliquer les opérateurs génétiques sur les deux chromosomes sélectionnés et mettre les deux chromosomes résultant dans la nouvelle population. Les deux opérateurs génétiques courants sont l'enjambement (*cross-over*) et la mutation. Le premier est la véritable reproduction. Chaque chromosome enfant reçoit environ la moitié des gènes de chacun de ses parents. La probabilité d'enjambement est presque toujours de 50% (effectue de cette manière un mélange plus efficace). Selon cette probabilité, on échange les gènes homologues des deux chromosomes. De cette façon, après l'enjambement, on obtient deux enfants qui sont complémentaires par rapport à leurs parents (si un enfant

a un gène d'un tel parent, l'autre enfant tiendra ce même gène de l'autre parent). L'opérateur de mutation, bien qu'ayant une probabilité bien moindre (habituellement entre 0,5% et 5%) joue un rôle très important. Une reproduction utilisant uniquement l'enjambement est une méthode de **hill-climbing**, méthode qui est limitée par l'atteinte de maxima locaux. En effet, les gènes des enfants sont limités par les gènes des parents, et si un gène n'est pas présent dans la population initiale (ou s'il disparaît à cause des reproductions), il ne pourra jamais se développer chez les descendants. L'opérateur de mutation est là pour contourner ce problème. Chaque gène possède une faible probabilité de muter, c'est-à-dire d'être aléatoirement remplacé par une autre incarnation de ce gène. Cette précaution permet de conserver ce qu'on appelle la diversité génétique. Habituellement, la mutation crée des individus faibles, peu aptes à survivre. Cependant, une certaine mutation pourrait se révéler « géniale » et permet d'augmenter grandement l'évolution de la population. Ces opérateurs génétiques peuvent habituellement s'accommoder de la plupart des situations sans trop de modifications. Toutefois, afin de conserver la cohérence des individus, on doit parfois modifier grandement, voire changer totalement les opérateurs génétiques. Il n'y a pas de limite quant au nombre d'opérateurs génétiques. D'une manière générale, on considère comme opérateur génétique, n'importe quelle fonction qui modifie le contenu génétique d'un ou de plusieurs chromosomes. Il parle aussi quelques fois de l'opérateur génétique de clonage. Celui-ci existe (un individu est passé directement, sans modifications, à la génération suivante) et peut être incorporé dans un algorithme génétique. Cependant, le potentiel évolutif de cet opérateur est secondaire. On assiste également à un clonage lorsque les autres opérateurs génétiques n'ont pas d'effet (cas très rare).

#### **II.10.7.7 Validité et cohérence :**

Selon la méthode de codage et sa signification, on doit être toujours certain que les individus de notre population soient valides. En effet, que ferait-on d'une solution donnant une bonne note de qualité, mais n'ayant aucun sens pratique une fois interprété. Il faut donc s'assurer que la fonction de création des individus crée toujours des individus valides, et que les opérateurs génétiques conservent la validité des individus traités. Ceci vise à conserver la cohérence générale de l'algorithme.

#### **II.10.7.8 Critère de terminaison :**

Généralement, un algorithme génétique se termine après un certain nombre de générations, mais on peut également terminer l'exécution de l'algorithme lorsqu'une certaine condition soit atteinte, par exemple lorsque la qualité d'un individu dépasse un certain seuil.

### **II. 11 Conclusion:**

Ce chapitre contient les éléments de base de la commande par logique floue. L'avantage principal de la commande floue par rapport aux autres types de commande est la possibilité d'incorporer des connaissances symboliques d'une manière naturelle et flexible. Ainsi, les règles symboliques " Si - Alors "

remplacent les équations différentielles, et le raisonnement exact, rigide, laisse la place à un raisonnement approximatif flexible, qui peut utiliser des connaissances imparfaites, entachées d'imprécisions et d'incertitudes.

Il apparaît donc que cette technique de correction, appliquée ici à un système relativement simple, outre le fait qu'elle se prête bien à la régulation des systèmes qui pourraient être plus complexes (en particulier non linéaires), offre l'avantage d'une mise en œuvre très facile avec une auto adaptation à la variation des paramètres du système. Ce type de correcteur est donc très robuste tout en gardant une excellente stabilité quelque soit le point de fonctionnement.

Les algorithmes génétiques sont proposés comme outil d'optimisation. Cette méthode basée sur le principe de la sélection naturelle. Cette dernière cherche à partir d'un espace de solutions, la solution optimale en utilisant des procédés aléatoires. En créant de nouvelles générations de solutions, elle essaie d'en trouver la meilleure.

### III.1 Introduction

Dans le domaine de l'asservissement, les contrôleurs flous peuvent concurrencer les correcteurs classiques.

En effet, ils offrent de bonnes caractéristiques en termes de temps de réponse, dépassement et surtout robustesse dans certaines applications.

De plus, ils n'ont besoin que d'une connaissance sommaire du système à contrôler, qui s'exprime sous forme d'un tableau (moteur d'inférence) où l'on enregistre le comportement du correcteur.

Le temps de mise au point est donc beaucoup plus court bien qu'il soit nécessaire d'utiliser soit un microcontrôleur, soit un composant dédié. Dans un souci d'optimiser encore la qualité de l'asservissement, on se propose de trouver un moyen de rendre le contrôleur flou dynamique.

Ainsi, la base devrait se modifier petit à petit en suivant les variations des paramètres du système et assurer un asservissement optimal. (Voir figure III.1)

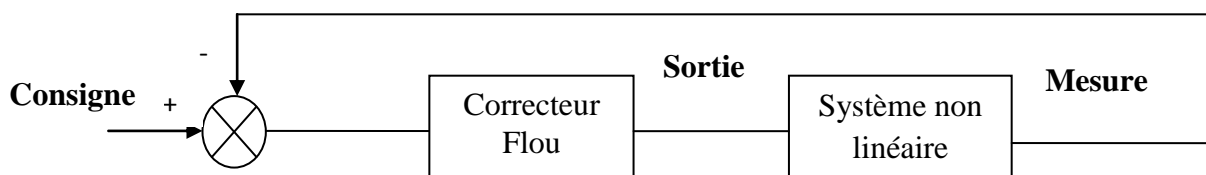


Figure III.1: Schéma d'un système commandé par un contrôleur flou

### III.2 Utilisation du contrôleur flou en Simulink :

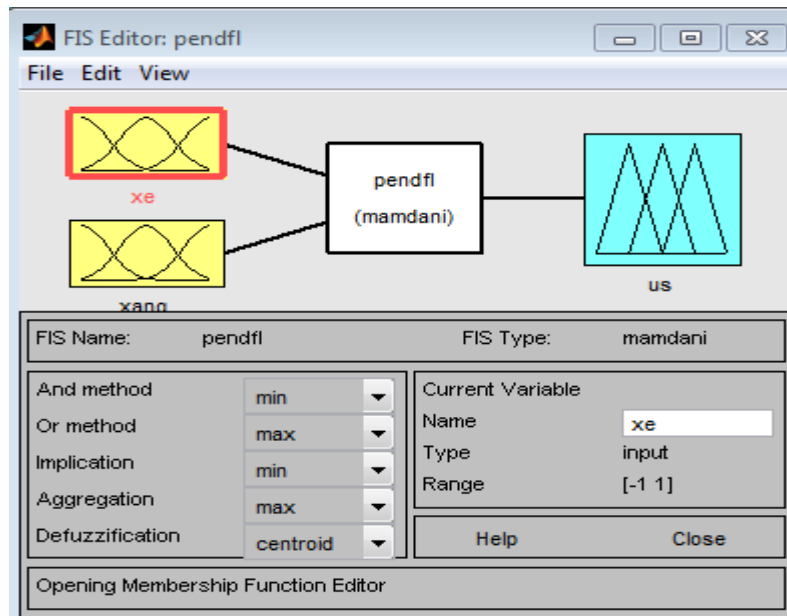
Le développement d'un contrôleur en logique floue pour intégration dans Simulink est très simple si on possède la boîte à outils *Fuzzy Logic* de Matlab. Cette boîte à outils ajoute les composantes nécessaires dans Simulink.

Ce guide sert seulement pour l'intégration de la logique floue dans Simulink; ce n'est pas un guide sur le design de contrôleurs PD.

Pour commencer Simulink, il suffit de taper la commande **Simulink** à la ligne de commandes de Matlab. Le contrôleur à logique floue est situé sous les options *Fuzzy Logic Toolbox*, comme montré à la figure ci-dessous :



Le design du contrôleur flou se fait de la même façon que tous les systèmes flous avec Matlab. On commence en tapant **fuzzy** à la ligne de commande. On obtient alors le *FIS Editor* de Matlab, montré à la figure suivante :



**Figure III.4 :** fenêtre fis editor

Pour le système simple sous étude, on n'utilise que l'erreur comme variable de contrôle. On a donc un système très simple, avec une entrée, une sortie. Dans plusieurs systèmes flous, on utilise aussi la dérivée de l'erreur, et on obtient alors un système à deux entrées, une sortie. Il faut utiliser un peu de jugement dans le design du contrôleur flou. Il ne faut pas simplement créer des fonctions d'appartenance de toute façon. Il faut quand même étudier le système en détail.

#### **Fuzzification :**

Il faut bien choisir les limites du système (limites de l'entrée et de la sortie).

Dans ce cas-ci, l'entrée utilisée est une entrée échelon unitaire. On choisit donc de développer les fonctions d'appartenance du contrôleur flou pour l'entrée entre -1 et +1.

On a choisit que 3 valeurs linguistiques possibles, soit *Négative*, *Zéro* et *Positive*. Il est généralement plus facile de commencer avec un nombre restreint de valeurs linguistiques possibles, pour simplifier le design initial. Par après, si la précision demandée est plus grande, on peut ajouter des valeurs linguistiques. Dans cet exemple, on a utilisé trois fonctions d'appartenance de type triangulaire pour les entrées (erreur et sa variation) et pour la sortie Défini sur un univers de discours [-1 ,1] comme le montre la figure suivante (fig. III.5) :

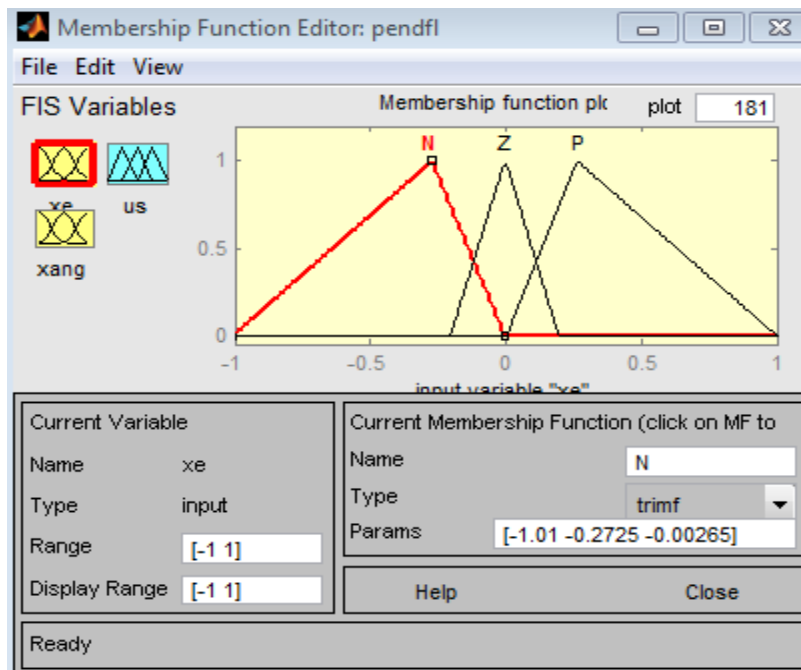


Figure III.5 : fenêtre des fonctions d'appartenance

**Inférence :**

C'est une base de connaissance qui contient les règles floues décrivant le comportement du système et les opérateurs logiques et les méthodes d'inférences. Il transforme à l'aide des opérateurs flous, la partie floue issue de la fuzzification en une partie floue conformément aux règles floues et une méthode d'inférence choisie, et l'agrégation des règles floues par l'opérateur flou **ET**. Dans notre cas on a utilisé la méthode d'inférence **max-min** (fig.III.6), l'opérateur min pour le raisonnement et l'opérateur max pour l'agrégation des règles.

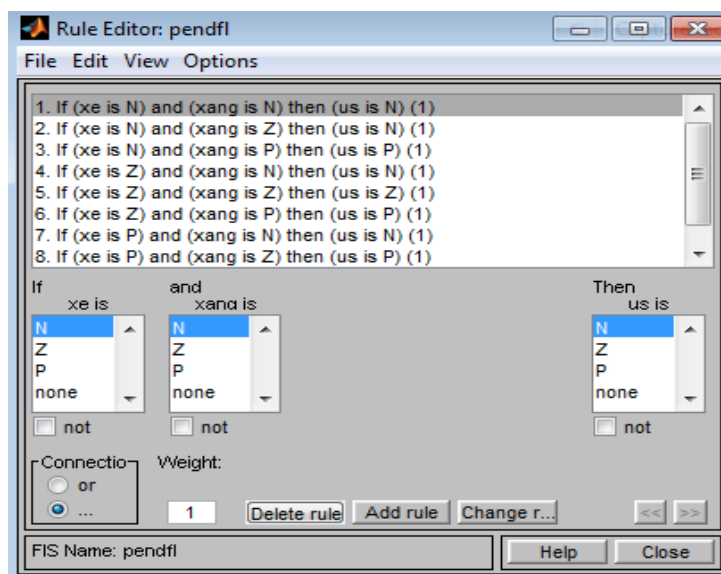
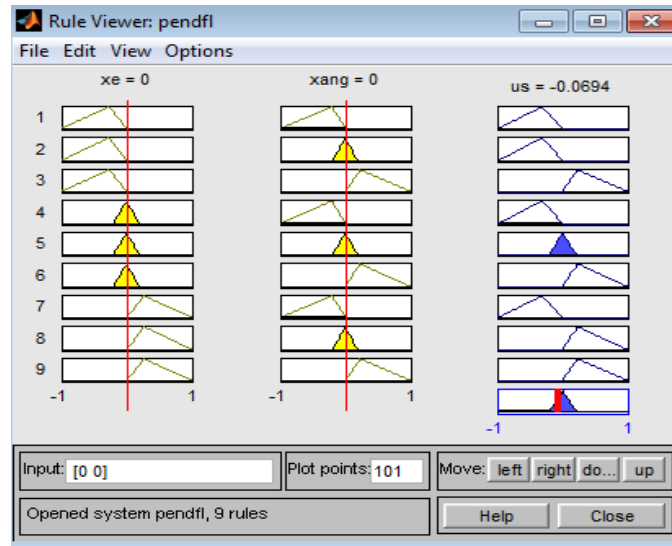


Figure III.6 : fenêtre de règles floues

**Défuzzification :**

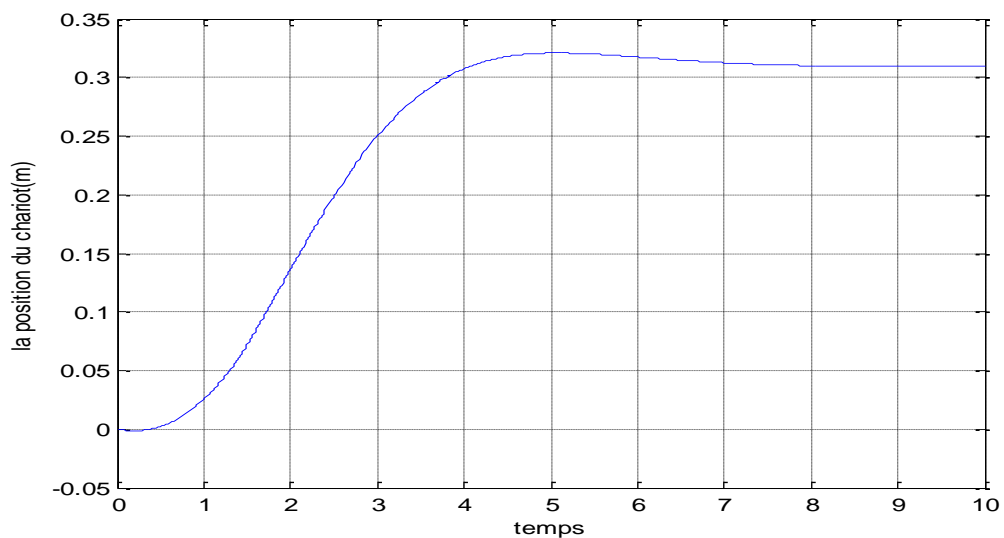
La défuzzification est une interface « linguistique –numérique » qui transforme la partie floue issue du raisonnement flou en valeurs numériques directement exploitables par le processus. La méthode de défuzzification la plus utilisée est celle du centre de gravité de la fonction d'appartenance résultante  $\mu_{res}(z)$ . (fig.III.7)



*Figure III.7 : fenêtre d'observation de la méthode centre de gravité*

La figure ci-dessus représente l'évolution respective de la position et vitesse du chariot par rapport à la consigne ainsi que celle de l'angle et de vitesse angulaire du pendule par rapport à la verticale par un régulateur Mamdani à 9 règles.

**III.3.1.1 Résultats de simulation:**



*Figure III.8 : la position du chariot*

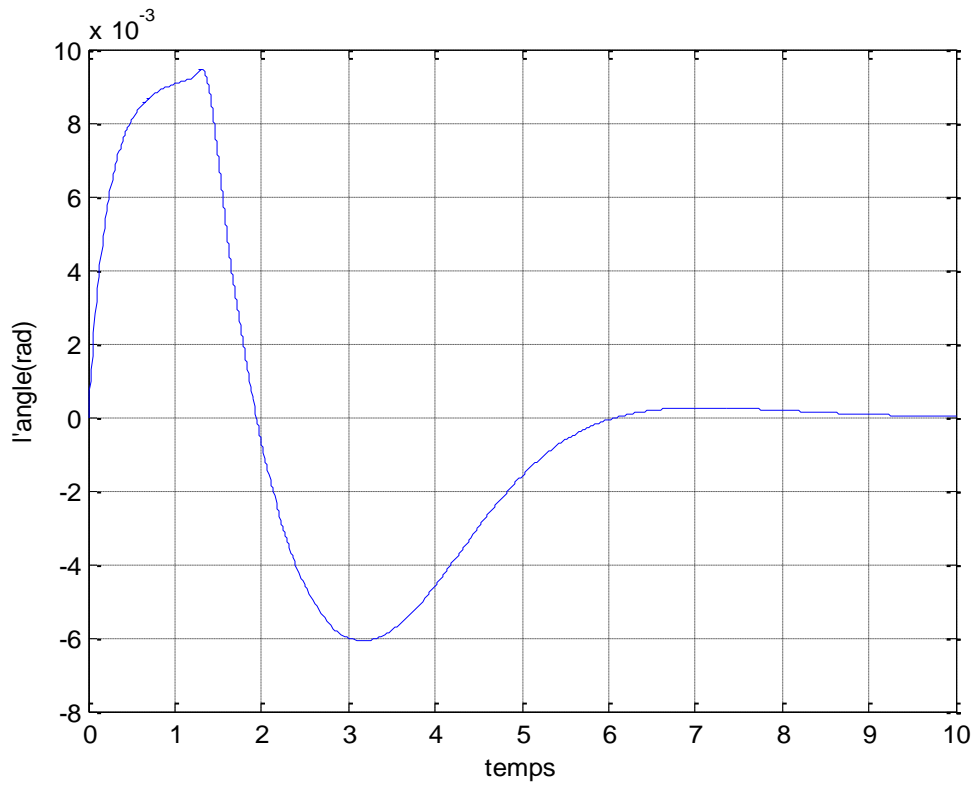


Figure III.9 : l'angle du pendule

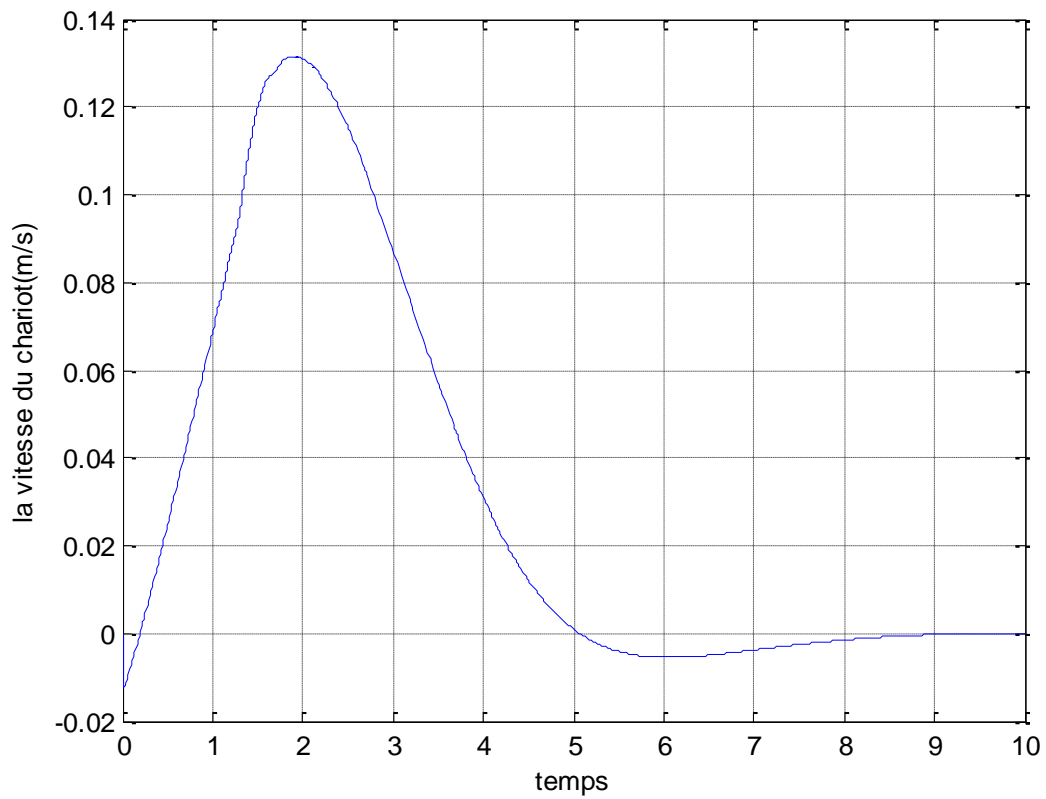
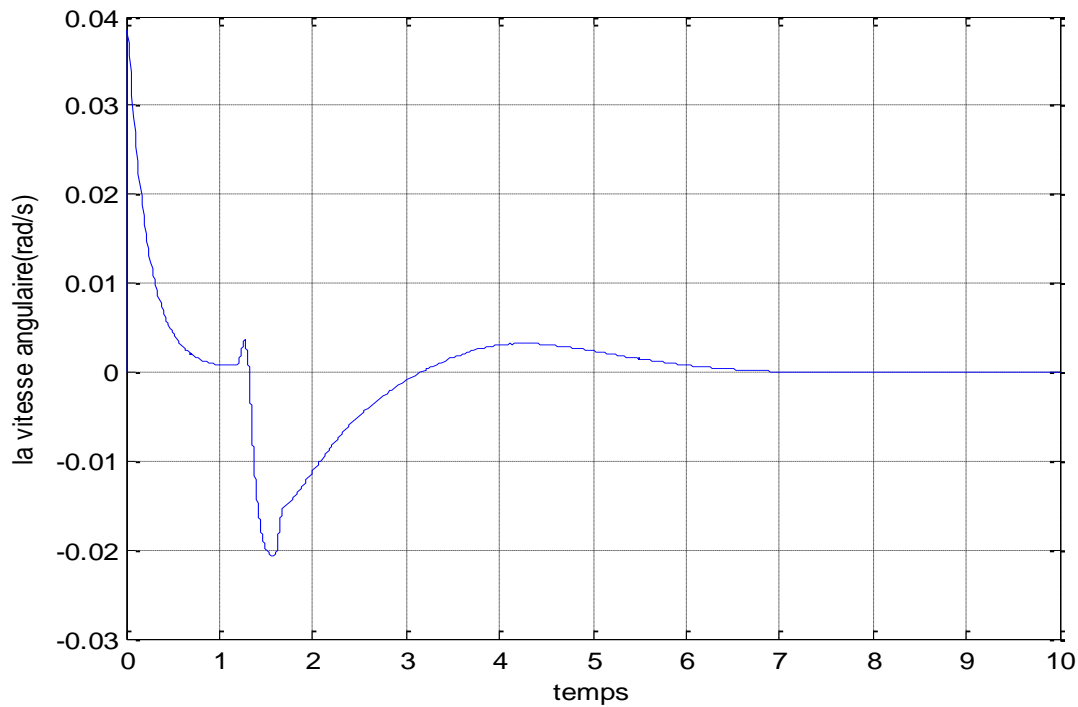
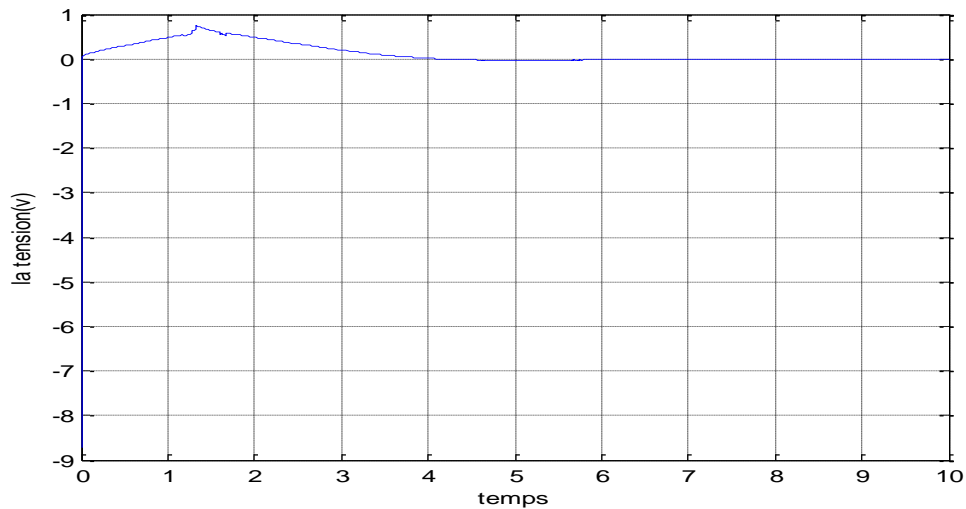


Figure III.10 : la vitesse du chariot



*Figure III.11 : la vitesse angulaire du pendule*



*Figure III.12: la commande (tension)*

*Les valeurs des facteurs d'échelle sont :*

**a=12;      b=2.5;      c=1.2;      d=2;      k=20;**

La figure ci-dessus représente l'évolution respective de la position et vitesse du chariot par rapport à la consigne ainsi que celle de l'angle et de vitesse angulaire du pendule par rapport à la verticale par un régulateur Mamdani à 9 règles floues.

### III.3.1.2 Interprétation des résultats :

D'après les résultats obtenus on constate que la position du chariot dépasse la référence à 4sec. Tandis que, l'angle du pendule et les vitesses (pendule, chariot) tendent vers zéro. A propos de la commande, on remarque qu'elle est insuffisante pour atteindre l'objectif voulu.

### III.3.2 Commande d'un bras manipulateur à 2ddl :

Pour cette application on a utilisé un proportionnel dérivé flou de type Mamdani à 4 règles floues pour la fuzzification des variables d'entrée (2 fonction de type trapézoïdal) et de sortie (4 fonction de type trapézoïdal), L'inférence floue est réalisée à l'aide de l'opérateur min (minimum) ou prod (produit) et l'agrégation des règles à l'aide de l'opérateur max (maximum). La défuzzification est généralement effectuée par la méthode du centre de gravité.

#### Sous Matlab :

On a réalisé cette configuration à l'aide des deux contrôleurs flous qui ont comme entrée l'erreur et sa variation, comme sortie les couples pour chaque articulation.

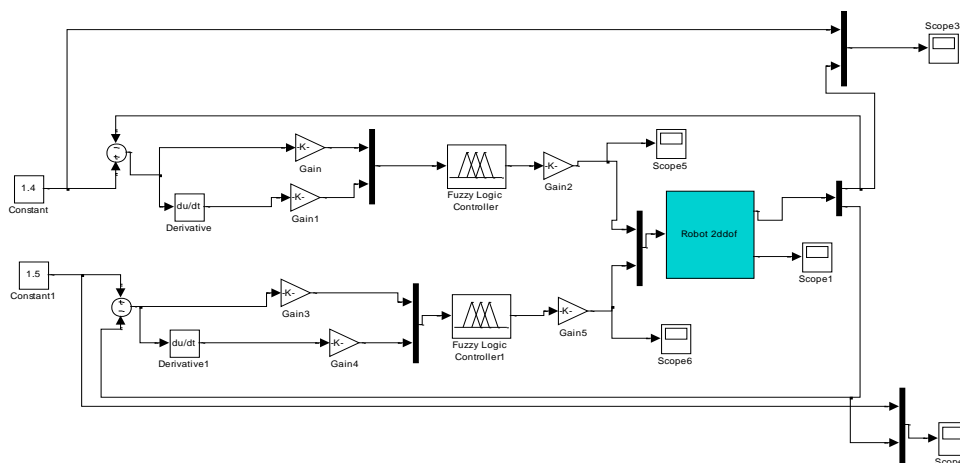
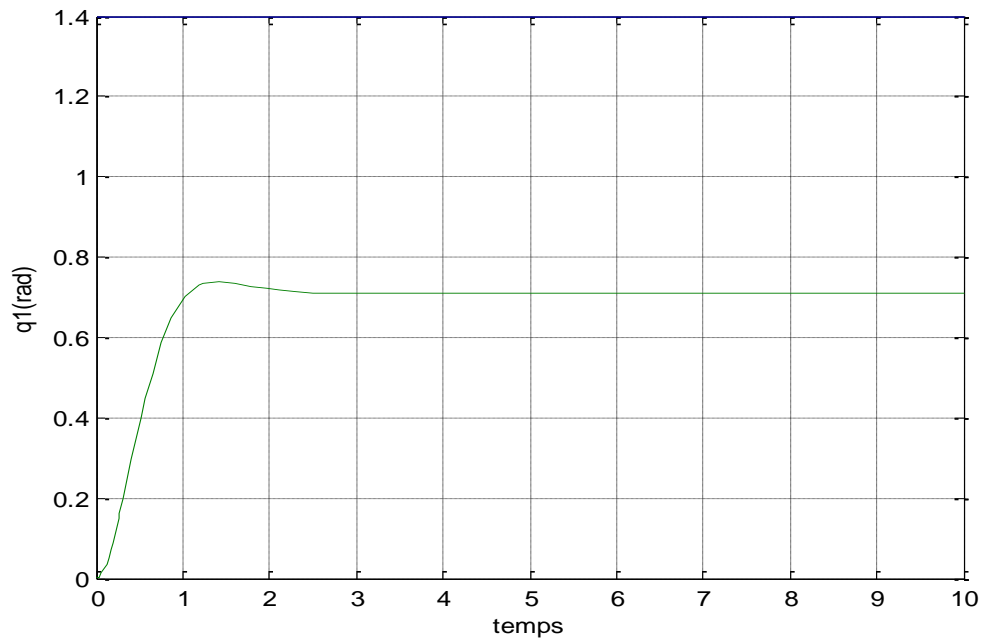
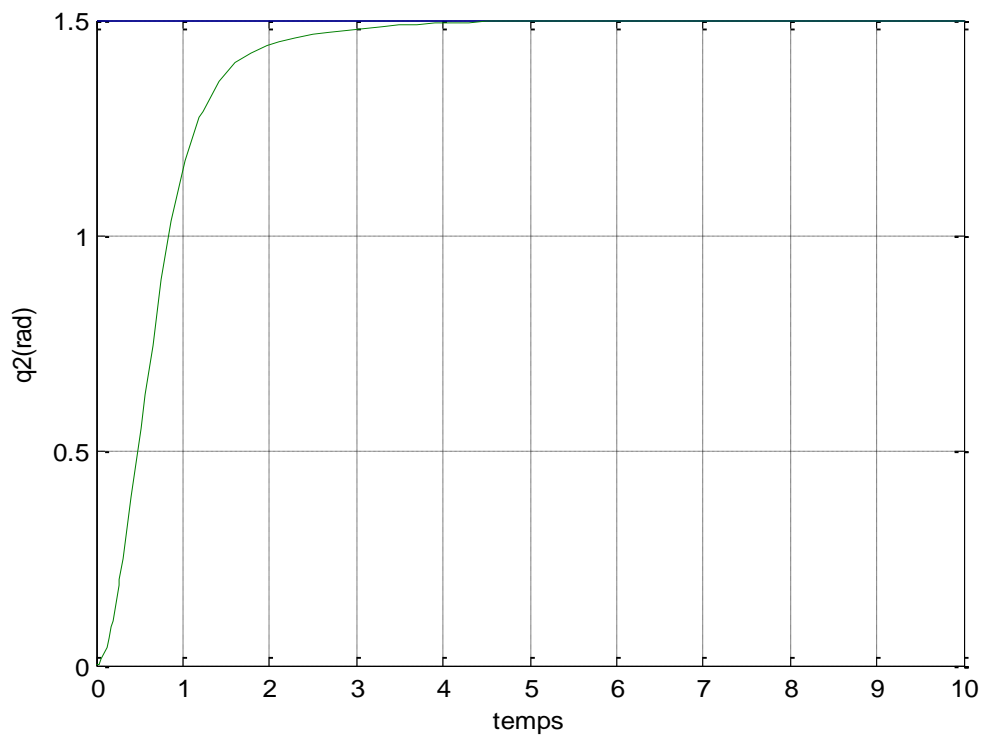


Figure III.13: schéma de simulation de stabilisation du pendule par la commande floue

Plusieurs essais ont été effectués pour régler la position et la vitesse des deux articulations d'un bras manipulateur mais nous n'avons pas pu aboutir aux résultats attendus à cause de non linéarité du système et le couplage des articulations.

## III.3.2.1 Résultats de simulation :

*Figure III.14 : la position de l'articulation 1**Figure III.15 : la position de l'articulation 2*

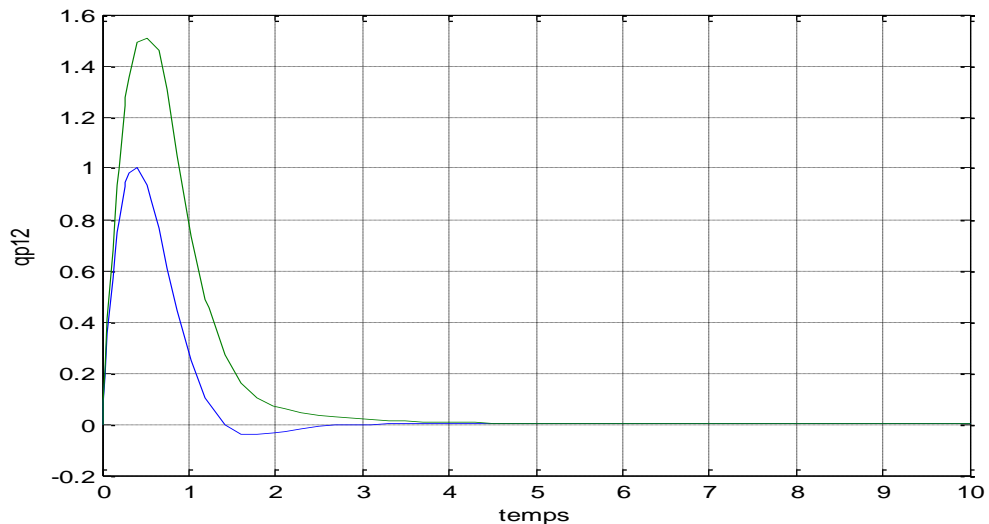


Figure III.16 : la vitesse des articulations

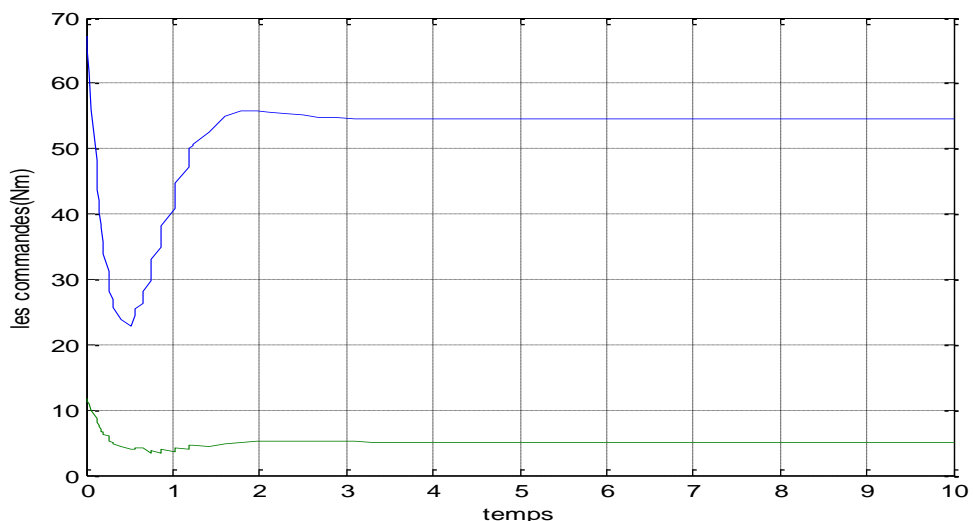


Figure III .17 : le signal de commande

**III.4 application de la commande classique sur un bras à 2ddl:**

Cette commande a été appliquée à un bras manipulateur à 2ddl en deux cas :

**III.4.1 En régulation :**

$$\Gamma = G(q) - k_p \tilde{q} - k_v \dot{\tilde{q}} \dots \dots \dots (III.1)$$

On veut positionner les différentes articulations du robot manipulateur autour d’une position désirée constante ; alors, il faut générer une loi de commande convenable. Pour cela on a choisi la loi de commande proportionnelle dérivée avec compensation de l’effet de gravité (point à point) défini par l’équation suivante :

Où  $k_p$  et  $k_v$  sont respectivement les matrices des gains proportionnels et dérivés de dimension  $(n \times n)$ , ces deux matrices sont diagonales et définies positives.  $\tilde{q} = q - q_d$  représente l'écart de position, avec  $q$  (mesurée) est la position angulaire, et  $q_d$  la position désirée.  $\dot{\tilde{q}} = \dot{q} - \dot{q}_d = \dot{q}$  est l'écart en vitesse.

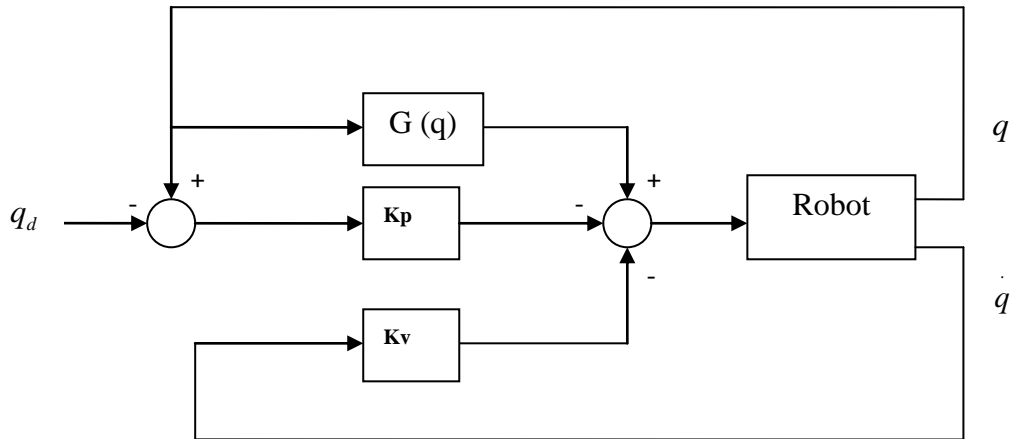


Figure III.18 : Schéma bloc d'une commande point à point

Avec :  $G(q) = 10 \begin{bmatrix} 7.6 \sin(q_1) + 0.63 \sin(q_1 + q_2) \\ 0.63 \sin(q_1 + q_2) \end{bmatrix}$

$q_d = [q_1 \quad q_2] = [1.5 \quad 1.6]$ ,

Sous Matlab :

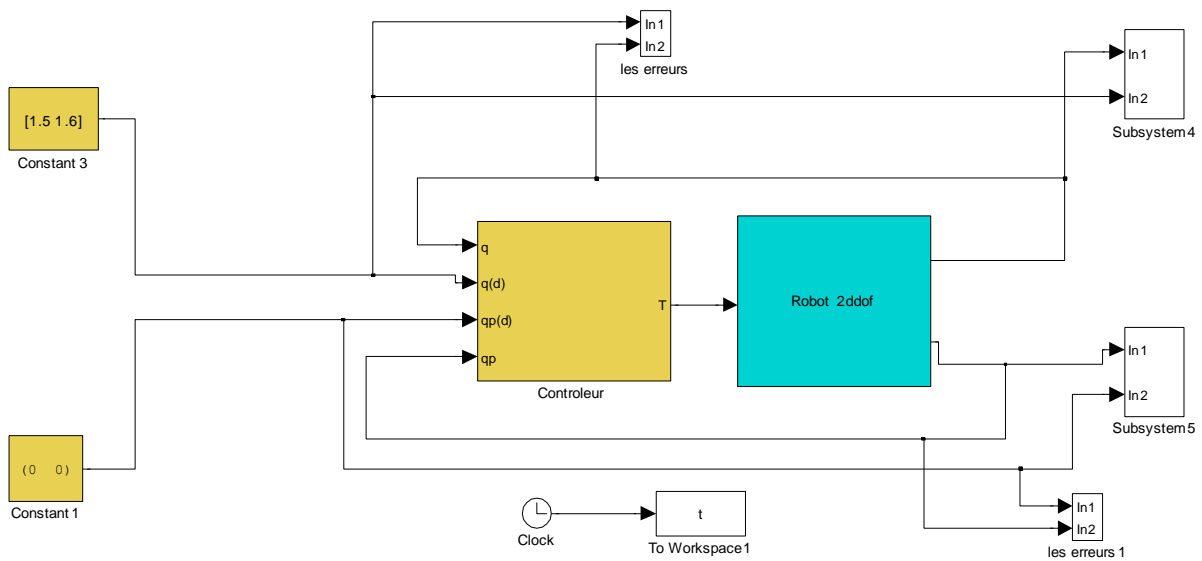


Figure III.19: Schéma de simulation de la régulation d'un bras manipulateur à 2ddl

III.4.1.1 Résultats de simulation :

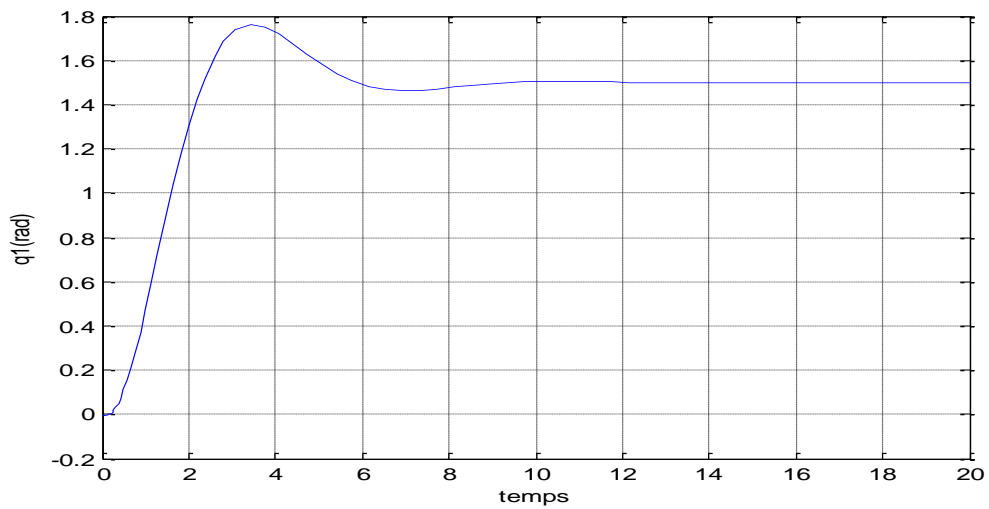


Figure III.20: la position de l'articulation 1

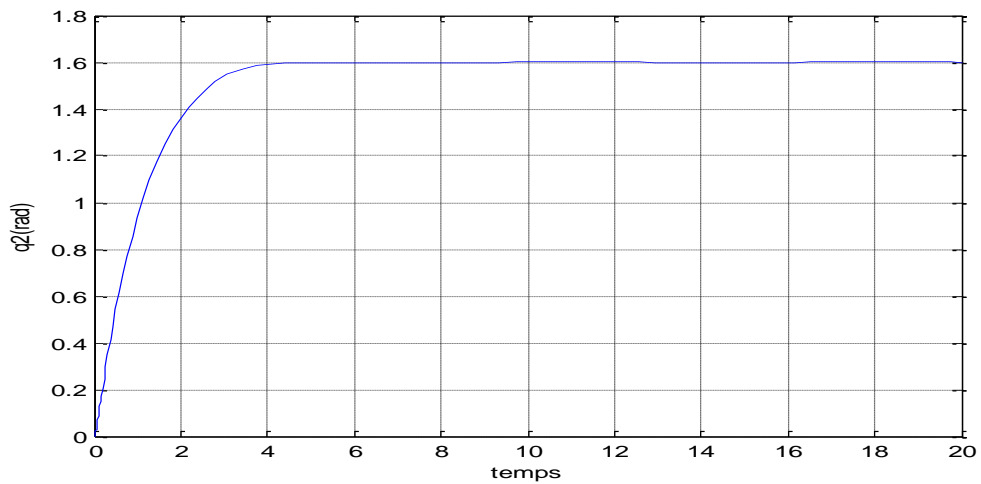


Figure III.21: la position de l'articulation 2

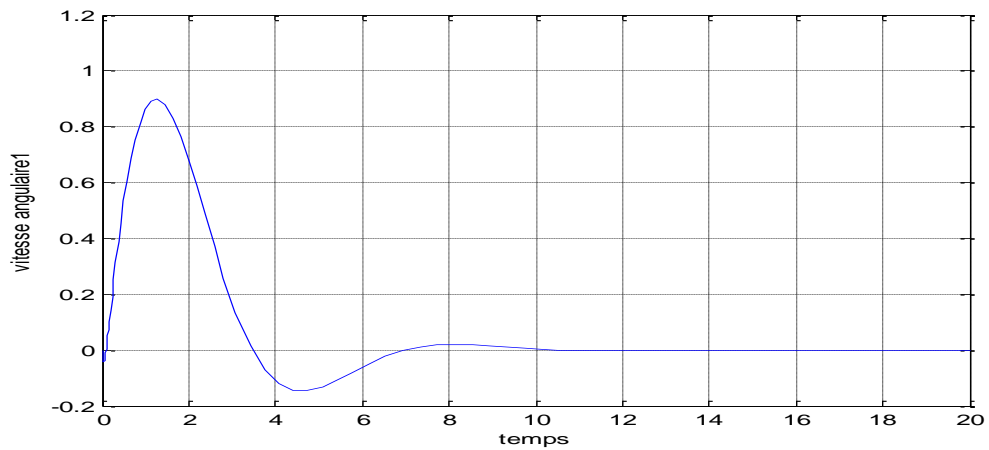


Figure III.22 : la vitesse de l'articulation 1

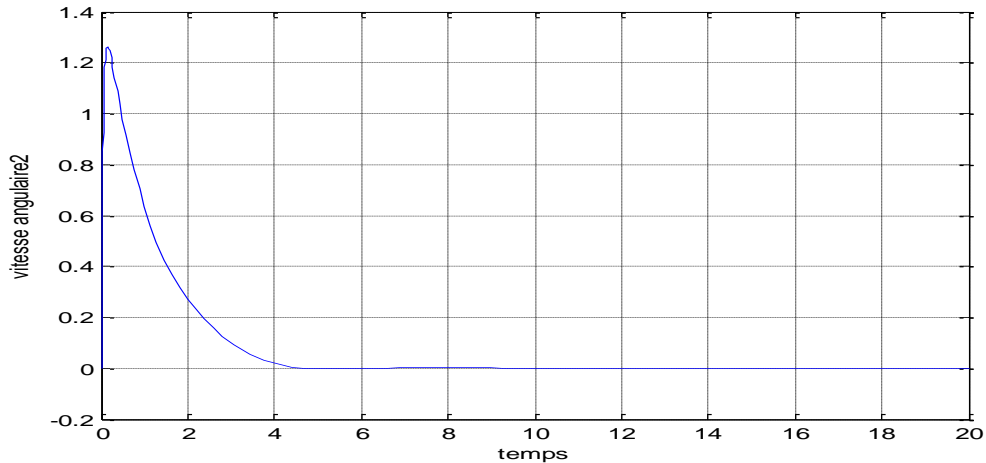


Figure III.23 : la vitesse de l'articulation2

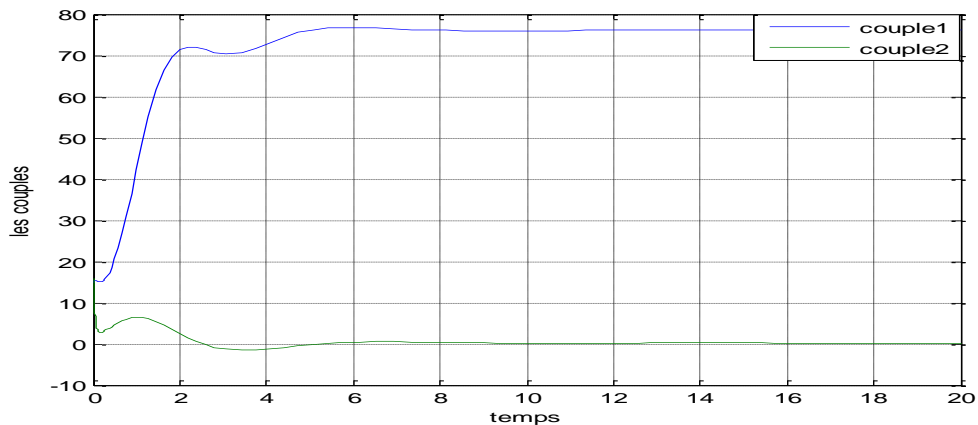


Figure III .24 : les commandes

Les valeurs des gains :

$$kp11=10; \quad kv11=10;$$

$$kp22=10; \quad kv22=10;$$

#### III.4.1.2 Interprétation des résultats:

Les résultats de la simulation sont présentés comme suit :

Les réponses en position des articulations du bras convergent vers la référence avec un dépassement (articulation1) .les vitesses convergent vers zéro et les commandes générées par le moteur sont suffisantes.

#### III.4.2 Commande en poursuite de trajectoire :

Afin d'étudier les lois de commandes il est nécessaire d'étudier les trajectoires de référence utilisées en robotique, ce qui nous permet d'analyser la validité d'une loi de commande.

D'une manière générale le problème de la génération de mouvement est de calculer pour chaque articulation les trajectoires de référence en position, vitesse et accélération qui sont fonction du temps et qui assurent le passage du robot par une trajectoire désirée d'une configuration initiale à une configuration finale désirée. Dans notre application on a choisi la loi de commande du couple calculé (**computed torque control**) qui est définie par la relation suivante :

$$\Gamma = M(q)[\ddot{q}_d + k_p \tilde{q} + k_v \dot{\tilde{q}}] + C(q, \dot{q})\dot{q} + G(q) \quad (III.2)$$

Avec :

$$\mathbf{K}_p = \begin{bmatrix} kp11 & 0 \\ 0 & kp22 \end{bmatrix} \quad \mathbf{K}_v = \begin{bmatrix} kv11 & 0 \\ 0 & kv22 \end{bmatrix}$$

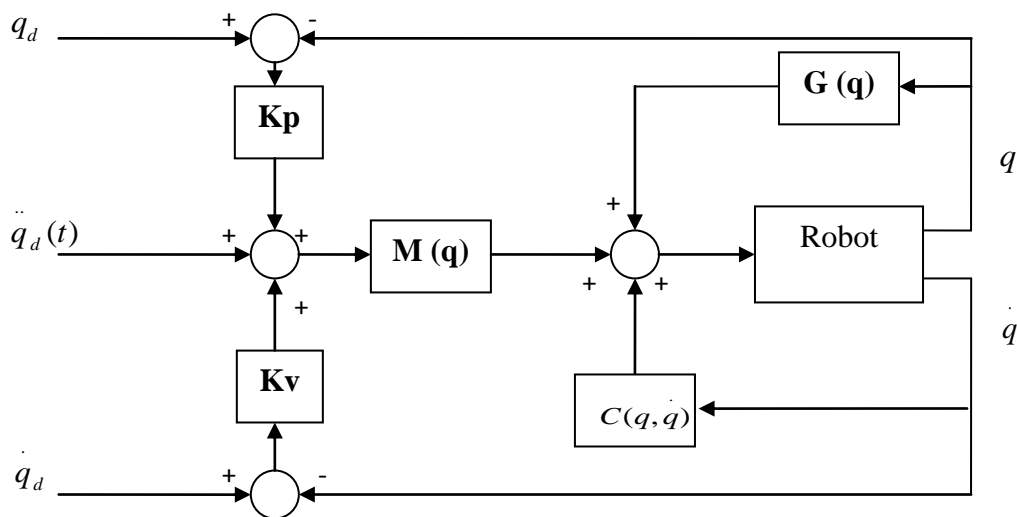


Figure III.25: structure de la loi de commande du couple calculé

Sous Matlab on a configuré cette loi comme suit :

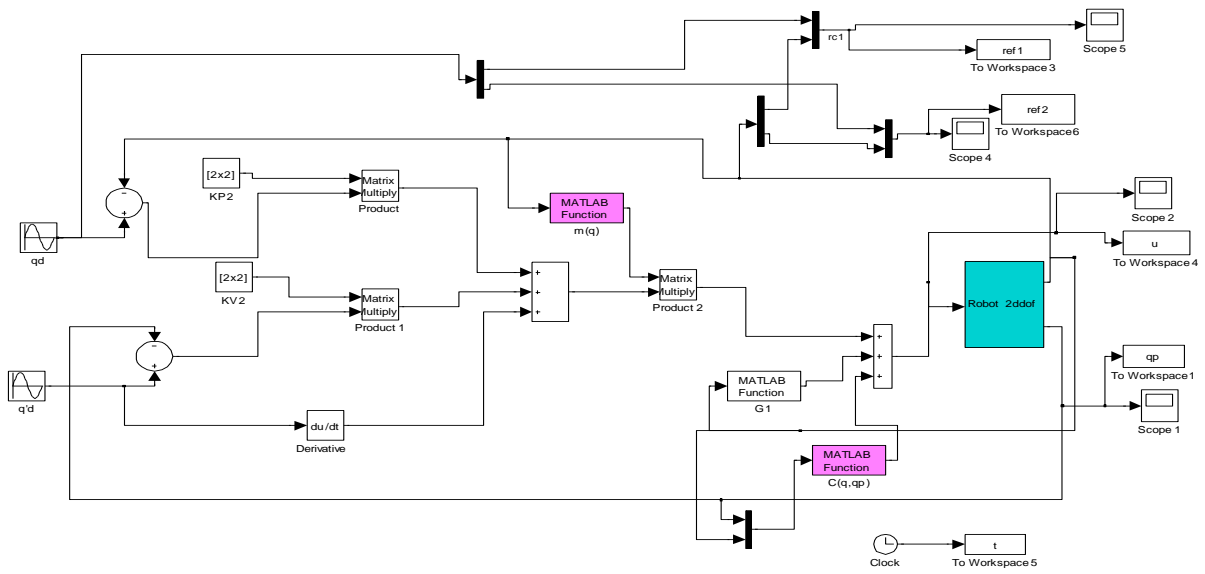


Figure III .26: Schéma de simulation de la poursuite d'une trajectoire sinusoïdale

III.4.2.1 Résultats de simulation :

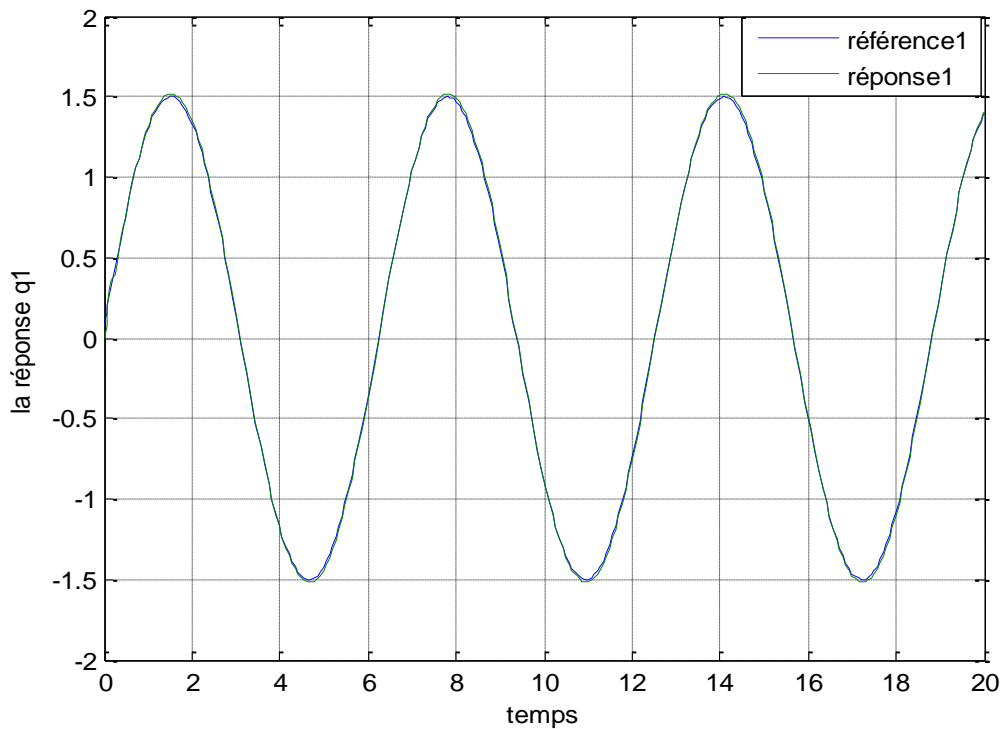


Figure III.27: la réponse de l'articulation 1

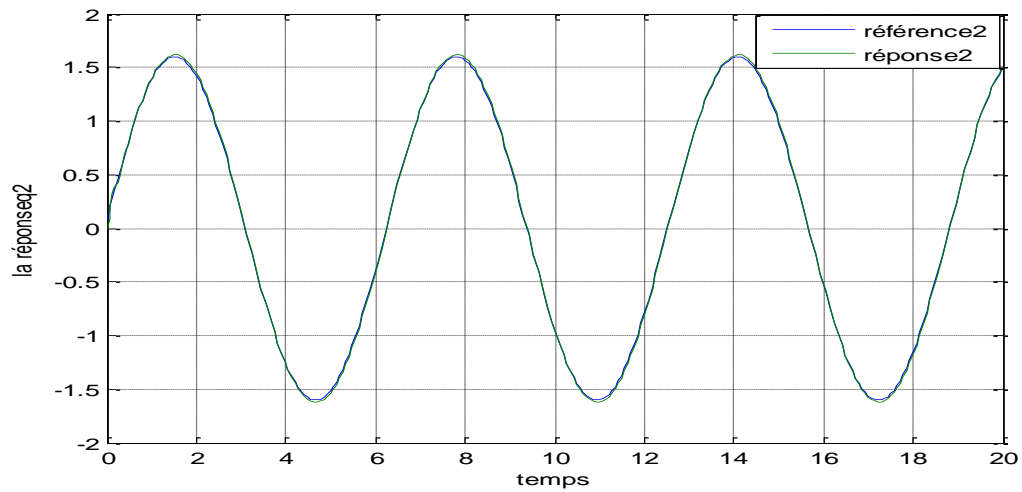


Figure III.28 : la réponse de l'articulation 2

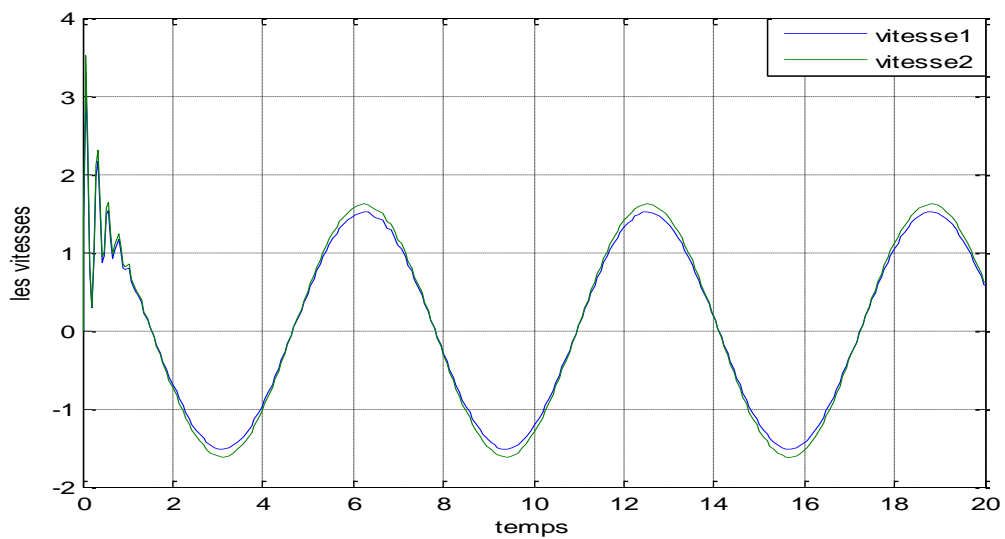


Figure III.29 : les vitesses des articulations

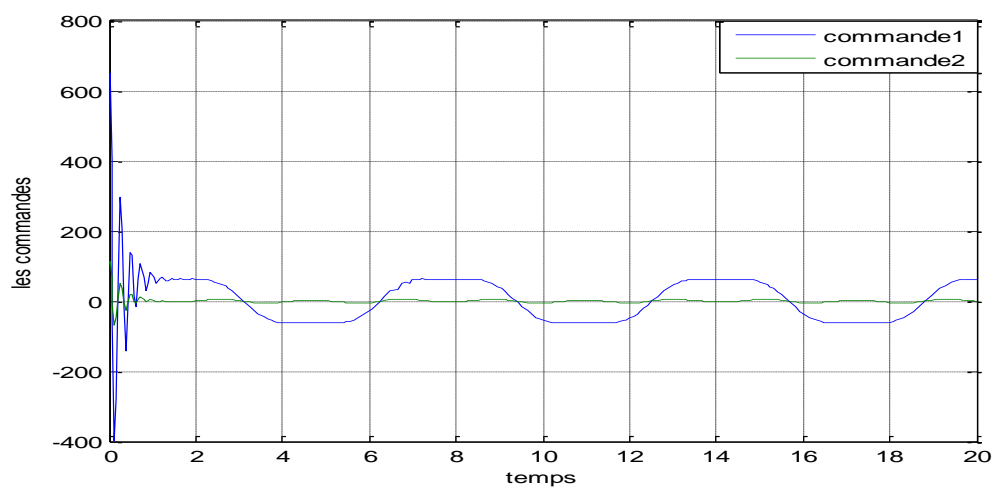


Figure III.30 : les commandes



Les résultats obtenus après l'optimisation:

**a**= 11.3800

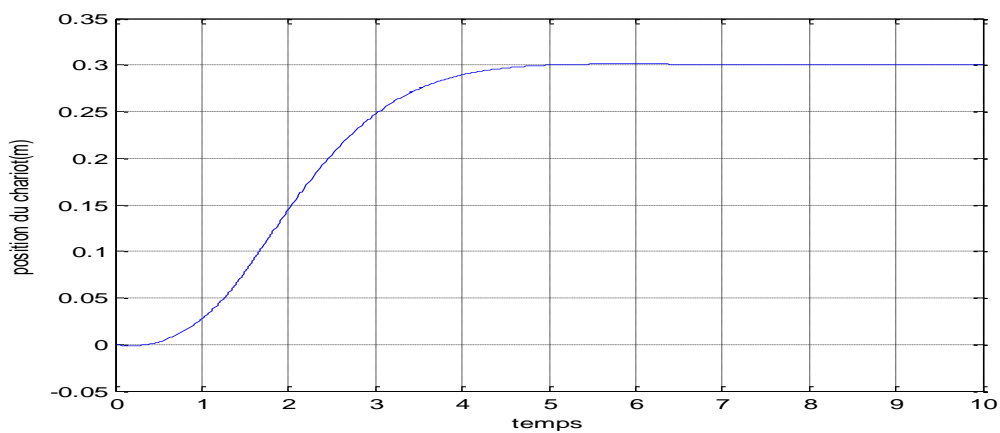
**b**=2.4034

**c**=1.4020

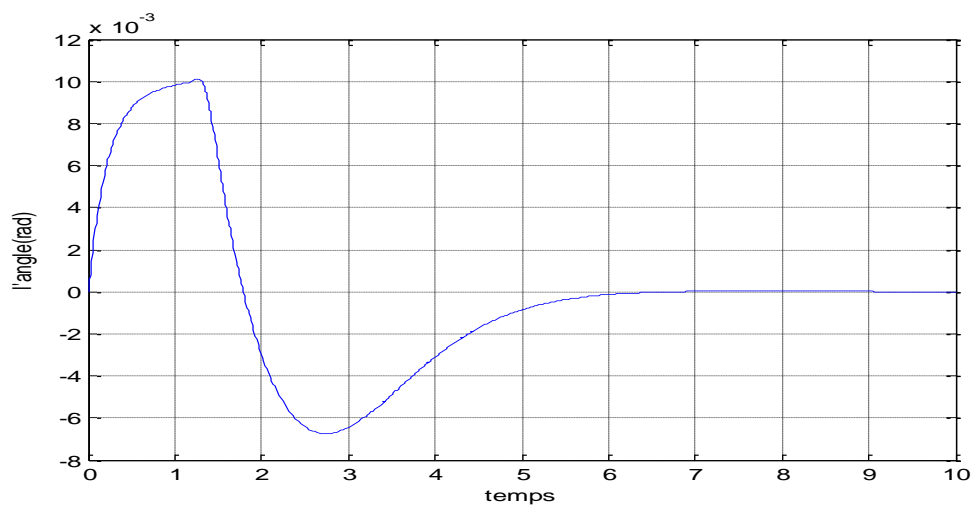
**d**=2.0219

**k**=17.6466

### III.5.1.1 Résultats de simulation :



*Figure III.31 : position du chariot après optimisation*



*Figure III.32 : l'angle du pendule après optimisation*

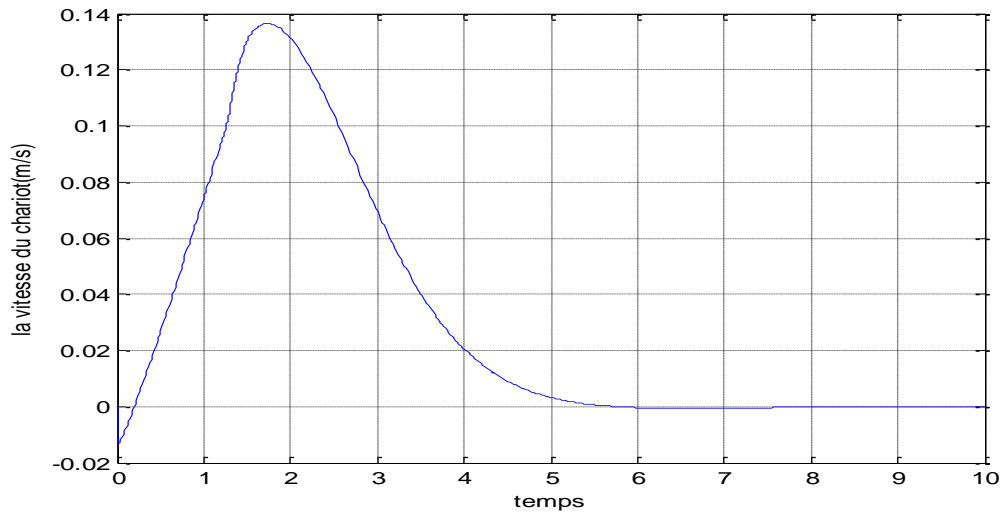


Figure III.33 : vitesse du chariot après optimisation

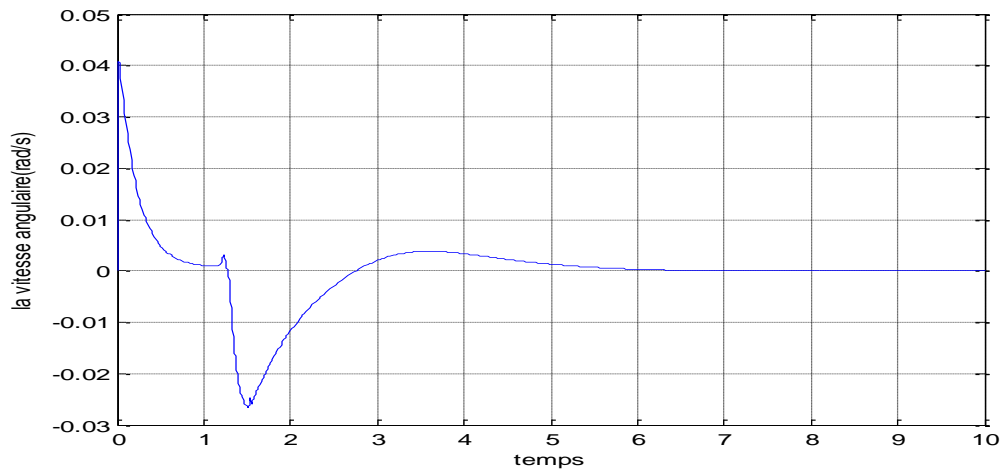


Figure III.34 : vitesse angulaire du pendule après optimisation

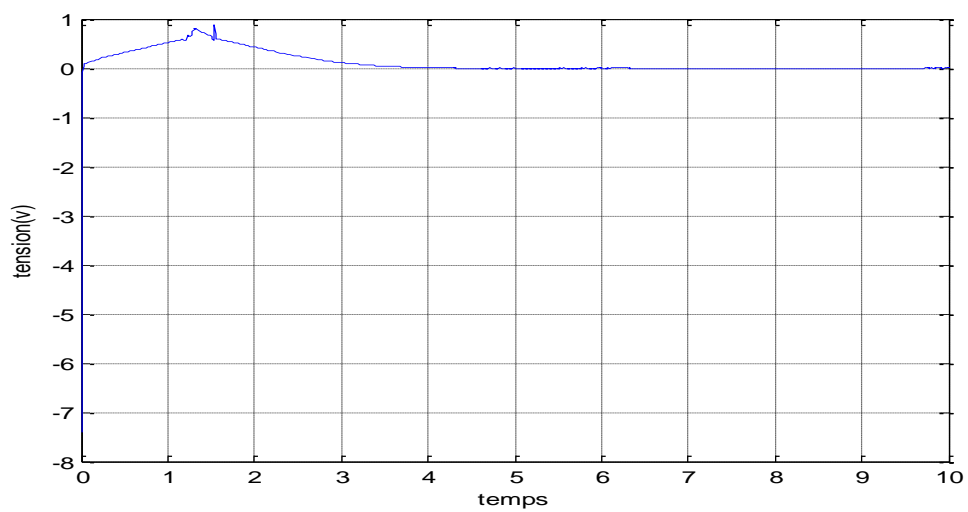
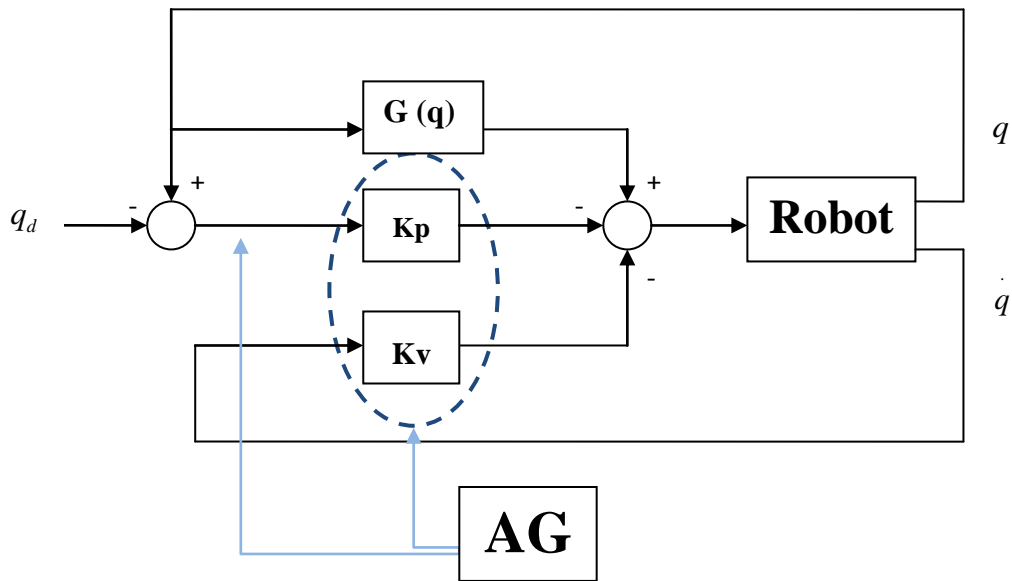


Figure III. 35: la commande après l'optimisation

**III.5.1.2 Interprétation des résultats:**

Après l’optimisation, nous avons obtenu des résultats acceptables concernant la position du chariot et l’angle du pendule ainsi que pour les vitesses et la commande (tension).

**III.5.2 application au contrôleur classique:** l’optimisation des gains proportionnels et dérivés sont illustrés par le schéma suivant :



Les opérateurs génétiques utilisés :

<i>Taille de la population</i>	<b>400</b>
<i>Nombre de génération</i>	<b>300</b>
<i>Coefficient de croisement</i>	<b>0.8</b>
<i>Coefficient de mutation</i>	<b>0.1</b>
<i>Type de sélection</i>	<b>Roulette</b>

*Tableau III .2 : les opérateurs génétiques*

**La valeur des gains après l’optimisation :**

**kp11=488.9578;**

**kp22= 488.7507;**

**kv11=115.0023;**

**kv22=84.6451;**

III.5.2.1 Résultats de simulation:

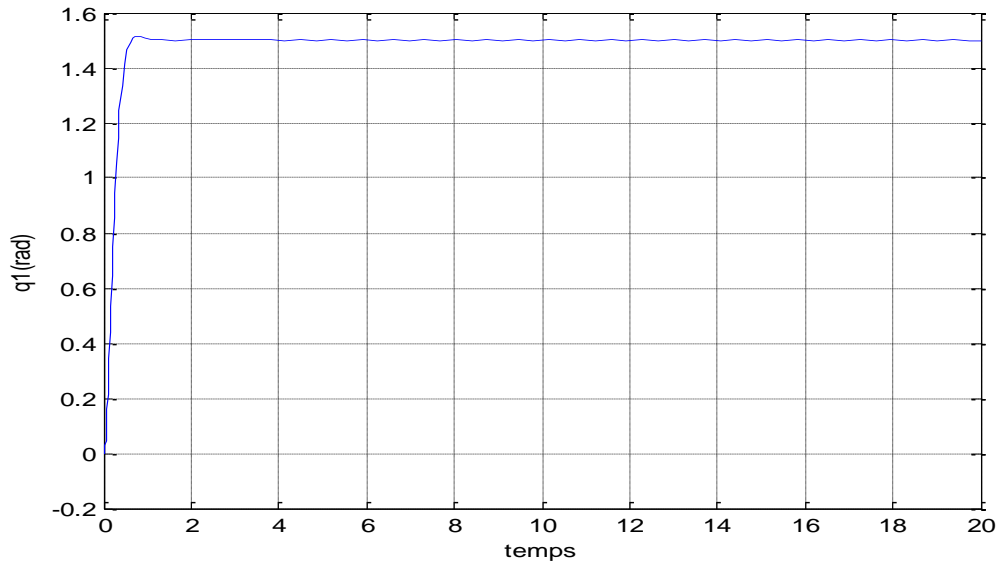


Figure III.36 : position de l'articulation 1 après optimisation

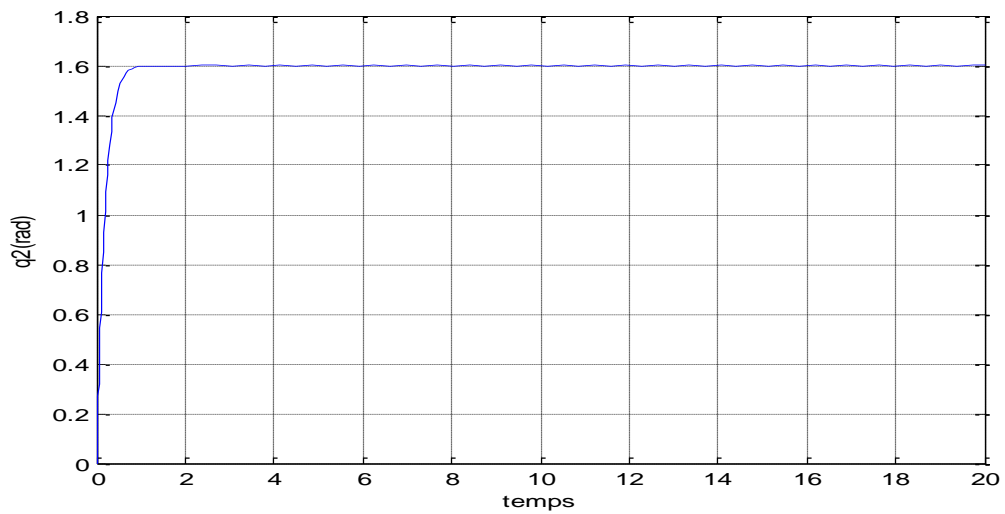


Figure III.37 : position de l'articulation 2 après optimisation

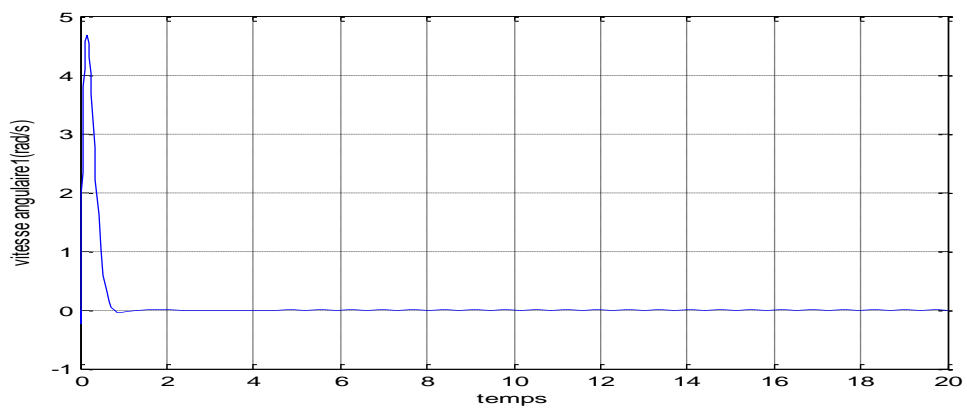
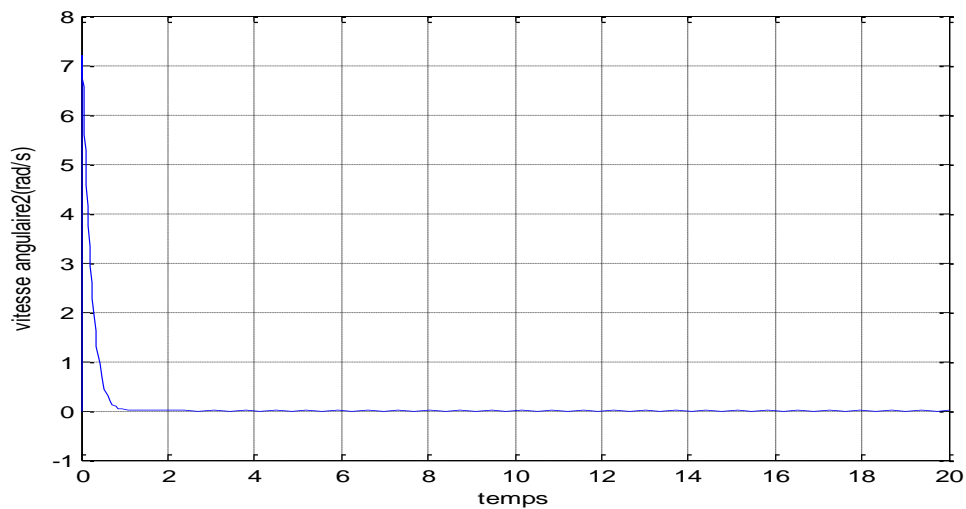
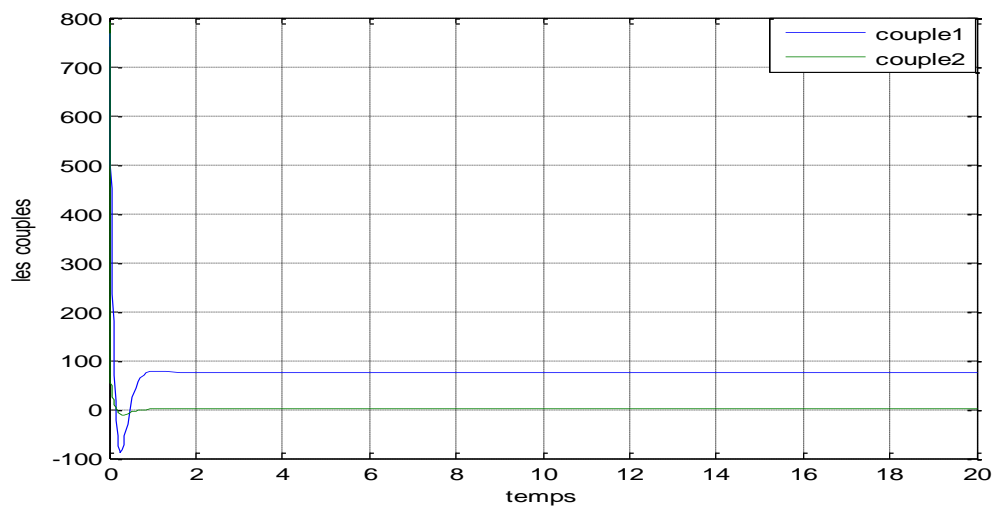


Figure III.38 : vitesse de l'articulation 1 après optimisation



*Figure III.39 : vitesse de l'articulation 2 après optimisation*



*Figure III.40: la commande en régulation après l'optimisation*

### III.5.2.2 Interprétation des résultats :

Après l'optimisation, nous remarquons que les réponses en position s'améliorent (sans dépassement), et le système devient plus rapide.

### III.6 Conclusion :

Dans ce chapitre, nous avons présenté les applications sous simulink (pendule inversé, bras manipulateur) utilisant la logique floue, des algorithmes génétiques ainsi la combinaison issus de la logique floue et les algorithmes.

Dans la deuxième partie nous avons présenté les lois de commandes classiques optimisés utilisant les AG ainsi leur structure.

Enfin, nous avons présenté les résultats de la simulation qu'ont donnée la commande floue et la commande classique avant et après optimisation respectivement. Nous avons vu que celles-ci manifestent de très bonnes performances.

Les simulations effectuées sur les modèles non linéaires systèmes pendule inversé et bras manipulateur montre l'efficacité de ces commandes.

Le contrôleur flou permet d'éviter le besoin d'un modèle mathématique exact et les algorithmes génétiques permettent d'éviter le besoin d'un expert.



### **Conclusion générale**

L'étude présentée dans ce mémoire a pour objectif d'aborder les techniques de l'intelligence artificielle qui constituent une alternative aux commandes classiques, généralement employées en automatique afin de répondre au mieux aux objectifs de performance et de robustesse pour la régulation d'un procédé. Les lois de commandes proposées sont appliquées à la stabilisation d'un pendule inversé et d'un bras manipulateur qui sont des systèmes instables, sous actionné et possède des non-linéarités non négligeables.

Pour y parvenir, nous sommes amenés d'abord à connaître la constitution de ce système et à comprendre son fonctionnement. La modélisation du bras manipulateur était nécessaire afin de synthétiser les lois de commandes classiques et intelligentes proposées dans notre travail.

Après avoir donné quelques rappels théoriques sur les algorithmes génétiques et rappels théorique de base de la logique floue et la représentation d'un régulateur flou et de ses différents constituants, ainsi que l'application de la logique floue dans la commande, nous avons appliqué la commande par logique floue et les algorithmes génétiques sur le pendule inversé et le bras manipulateur.

Les travaux décrits dans ce mémoire ouvrent un nombre de perspectives. Tout d'abord, il serait intéressant d'appliquer d'autres stratégies de commande afin d'apporter des améliorations tel que la Commande floue optimisée par des algorithmes génétiques, appliquer l'approche d'analyse de la commande utilisée dans ce mémoire à d'autres procédés.

Au terme de ce travail, nous pouvons conclure que chaque type de commande peut être avantageux dans un sens et désavantageux dans un autre, mais on peut affirmer que l'association de la logique floue et les algorithmes génétiques s'avère très efficace dans le contrôle de processus complexes.

## REFERENCES BIBLIOGRAPHIQUES

- [1] **BÜHLER Hansruedi** : Réglage par logique floue presses polytechnique et Université Romande, Larsane ,1994
- [2]**JELKOSANTINIS** : Introduction à la logique floue : les concepts fondamentaux et applications.
- [3] **FAUCHER. J**, «les plans d'expériences pour le réglage et commande à base logique floue».Thèse de Doctorat, Institut national Polytechnique de Toulouse. Spécialité : Génie Electrique et Automatique
- [4] **GACÔGNE, LOUIS**. «Eléments de logique floue». Paris : Hermès 1997.
- [5] **GERALD, H**. «processeurs flous». HE-ARC, BAPTISTE SAVOYE 26, CH-2610 STIMIER, Mai 2008.
- [6]**Gonzalo Cabodevila** : Commande et Identification Floue Inclus: IEEE Trans. On System, Man and Cybernetics, Vol. 23, No. 3, pp. 665-685, May/June, 1993.
- [7]**Khalil .W**: Modélisation identification et commande des Robots ,2 ème édition revue et augmentée, HERMES science publication, paris 1988 ,1999.
- [8] **LAURANT F. SYLVIE G. ANDRE T**. " Commande floue 1, de la stabilisation à la supervision " Paris, Hermès Sciences Publication, 2003.
- [9] **Lescieux Matthieu** : Introduction à la logique floue. « Fuzzy logic »Application à la commande floue.
- [10] **LUGER, George F.**, Artificial Intelligence: Structures & Strategies for complex problem solving, 4<sup>ème</sup> édition, Addison-Wesley, 2002, 856p.
- [11] **MADAOU M**. "Commande floue et neuro-floue d'un convertisseur direct AC/AC monophasé à structure minimale. Application à l'alimentation d'une machine asynchrone monophasé" Thèse de Magister, ENPA Alger Février 2007.
- [12] **MELLAH R.**, " Contribution la commande adaptative neuro floue. Application à la robotique" Thèse de Doctorat, Université des sciences et de la Technologie Houari Boumediene. Alger. Mai 2006.
- [13] **Nakoula, Y**. " Apprentissage des Modèles linguistiques flous, par Jeu de Règles Pondérées" Thèse de Doctorat, Université de Savoie, France, Juillet 1997.
- [14] **ZAIR** : Commande adaptative par les ajustements flous de la machine asynchrone (PFE), E.N.P, génie électrique 1999.

## **Résumé**

L'étude présentée dans ce mémoire a pour objectif d'aborder les techniques de l'intelligence artificielle qui constituent une alternative aux commandes classiques, généralement employées en automatique afin de répondre au mieux aux objectifs de performance et de robustesse pour la régulation d'un procédé. Les lois de commandes proposées sont appliquées à la stabilisation d'un pendule inversé et d'un bras manipulateur qui sont des systèmes instables, sous actionné et possède des non-linéarités non négligeables.

Mots clés : logique floue, algorithme génétique, robot, pendule inversé.