

وزارة التعليم العالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
جامعة فرحات عباس سطيف
UNIVERSITE FERHAT ABBAS — SETIF
UFAS (ALGERIE)

Mémoire

Présenté au département d'Electrotechnique
Faculté de Technologie
Pour obtenir le diplôme

De Master en Automatique

Option : commande des processus industriels

Par

MEZAACHE NOURDDINE

Thème

***Commande d'un bras manipulateur par logique floue
et les algorithmes génétiques***

Soutenue le 29/06/2011 devant la commission d'examen composée de :

<i>Dr. CHAIBA A.</i>	<i>M A. A. à l'Université de Sétif</i>	<i>Président</i>
<i>M. RAFOUFI S.</i>	<i>M A. A. à l'Université de Sétif</i>	<i>Encadreur</i>
<i>Mme. LATRECHE S.</i>	<i>M A. A. à l'Université de Sétif</i>	<i>Examineur</i>

REMERCIEMENTS

« A ceux qui m'ont conçu, fabriqué,
élevé et m'ont permis d'être ce que je
suis aujourd'hui ...»

Après louange à Allah, je tiens, tout d'abord, à exprimer mes vifs remerciements à tout le corps éducatif qui m'a permis de préparer ce travail avec la plus grande minutie.

Je voudrais aussi, exprimer mes remerciements les plus sincères à monsieur le président du jury ainsi qu'à ses membres pour avoir accepté d'examiner ce travail.

Enfin, j'adresse ma profonde gratitude à toute ma famille, ma mère, mon père, mes frères, et ma sœur, ainsi que mes amis et surtout Younès, Zahir, Walid, Moussa, Moustafa, Abouchi,...pour le soutien qu'ils m'ont apporté tout au long de mon cycle.

Merci à tout le Monde.

Liste des figures

Figure 1.1 Structure générale d'un robot manipulateur.	5
Figure 1.2 Représentation du robot à deux axes rotoïdes.	11
Figure 2.1 Représentation d'un sous-ensemble flou et principales caractéristiques.	19
Figure 2.2 Notions de spécificité et de précision représentée à l'aide de s-e flous.	19
Figure 2.3 La structure d'un contrôleur flou et le flux de données qui le traverse.	21
Figure 2.4 Mécanisme de Fuzzification.	22
Figure 2.5 Représentation d'une variable linguistique.	22
Figure 2.6 Inférence "MINIMUM" et "PRODUIT".	23
Figure 2.7 Compositions de valeurs floues issues de l'inférence.	24
Figure 3.1 Organigramme d'un AG standard.	27
Figure 3.2 Croisement simple.	30
Figure 3.3 Opération de mutation.	31
Figure 3.4 Opérateur de croisement en deux points.	33
Figure 3.5 Opérateur de croisement aléatoire.	33
Figure 3.6 Algorithme génétique adaptatif.	34
Figure 4.1 Positions des deux articulations.	35
Figure 4.2 Fonctions d'appartenance, RLF à 3 classes.	36
Figure 4.3 Poursuite à vide de la référence, RLF à 3 classes.	37
Figure 4.4 Poursuite de la référence pour un test de charge de 5 Kg, RLF (3×3).	38
Figure 4.5 Fonction objective pour optimisation des gains de l'erreur.	39
Figure 4.6 Poursuite à vide de la référence, RLF à 3 classes.	39
Figure 4.7 Poursuite de la référence pour un test de charge de 5 Kg, RLF (3×3).	40
Figure 4.8 Fonctions d'appartenances, RLF à 5 classes.	41
Figure 4.9 Poursuite à vide de la référence, RLF à 5 classes.	42
Figure 4.10 Poursuite de la référence pour un test de charge de 5 Kg, RLF (5×5).	43
Figure 4.11 Fonction objective pour optimisation des gains de l'erreur.	44
Figure 4.12 Poursuite à vide de la référence, RLF à 5 classes.	44
Figure 4.13 Poursuite de la référence pour un test de charge de 5 Kg, RLF (5×5).	45
Figure 4.14 Fonctions d'appartenances, RLF à 7 classes.	46
Figure 4.15 Poursuite à vide de la référence, RLF à 7 classes.	47
Figure 4.16 Poursuite de la référence pour un test de charge de 5 Kg, RLF (7×7).	48
Figure 4.17 Fonction objective pour optimisation des gains de l'erreur.	49
Figure 4.18 Poursuite à vide de la référence, RLF à 7 classes.	49
Figure 4.19 Poursuite de la référence pour un test de charge de 5 Kg, RLF (7×7).	50
Figure 4.20 Fonctions d'appartenances, RLF à 9 classes.	51
Figure 4.21 Poursuite à vide de la référence, RLF à 9 classes.	52
Figure 4.22 Poursuite de la référence pour un test de charge de 8 Kg, RLF (9×9).	53
Figure 4.23 Fonction objective pour optimisation des gains de l'erreur.	54
Figure 4.24 Poursuite à vide de la référence, RLF à 9 classes.	54
Figure 4.25 Poursuite de la référence pour un test de charge de 5 Kg, RLF (9×9).	55
Figure 4.26 Poursuite de la référence pour un test de charge de 8 Kg, RLF (9×9).	56

Liste des tableaux

Tableau 4.1	Base des règles, RLF (3×3).	36
Tableau 4.2	Base des règles, RLF (5×5).	41
Tableau 4.3	Base des règles, RLF (7×7).	46
Tableau 4.4	Base des règles, RLF (9×9).	51

Liste des abréviations

U : Vecteurs des couples moteurs.

q : Vecteur des positions articulaires.

\dot{q} : Vecteur des vitesses articulaires.

\ddot{q} : Vecteur des accélérations articulaires.

F : Forces extérieures exercées par l'organe terminal.

DDL : degré de liberté

D-H : Denavit-Hartenberg

FLC : contrôleur flou

AGs : algorithmes génétiques

PD : régulateur proportionnel dérivateur

RLF : régulateur flou

SOMMAIRE

Introduction générale	1
Chapitre I : Robot industriel	
1. Robots industriels	2
1.1. Introduction	2
1.2. Historique	2
1.3. Classifications Des Robots Industriels	2
1.4. Applications des robots industriels	3
2. Robot manipulateur	4
2.1. Structure générale d'un robot manipulateur	4
2.1.1. Structure mécanique	4
2.1.2. Actionneurs	4
2.1.3. Les systèmes de transmissions mécaniques	4
2.1.4. Les capteurs	5
2.1.5. Unité de contrôle	5
3. Commande des bras manipulateurs	5
3.1. Commande dans l'espace libre	6
3.2. Commande dans l'espace contraint	6
4. Modélisation d'un bras manipulateur	7
4.1. Modèle du robot manipulateur	7
4.2. Modèle cinématique	7
4.3. Modèle dynamique	7
4.3.1. Formalisme de Lagrange	7
4.3.2. Calcul de l'énergie cinétique	8
4.3.3. Calcul de l'énergie potentielle	10
4.4. Modèle du robot à deux degrés de liberté (ddl)	10
4.4.1. Modèle géométrique	11
4.4.2. Modèle cinématique	12
4.4.3. Paramètres de Denavit-Hartenberg(D-H)	12
4.4.4. Calcul de la matrice du passage	12
4.4.5. Calcul de la matrice d'inertie	13
4.4.6. Calcul du vecteur des termes de gravité	13
4.4.7. Calcul de la matrice $C(q, \dot{q})$	14
4.4.8. Modèle dynamique du robot à deux degrés de liberté	14
5. Conclusion	15
Chapitre II : Logique floue	
1. Introduction	16
2. Principe et historique de la logique floue	16
3. Applications	17
4. Notions nécessaires sur la logique floue	18
4.1. Les concepts	18
4.2. Sous-ensemble flou	18
4.3. Propositions et règle floue	19

5. Architecture classique d'un contrôleur flou (FLC)	20
5.1. Fuzzification	21
5.1.1. Base de connaissances	22
5.2. Mécanisme d'inférence	23
5.3. Défuzzification	24
6. Les étapes du contrôle flou	24
7. Conclusion	25

Chapitre III : Algorithmes génétiques

1. Introduction	26
2. Principe de fonctionnement DES AGs	26
3. Ce qui rend les AGs différents des méthodes classiques	28
4. Formulation du problème d'optimisation	28
5. LES MECANISMES D'UN AG	29
5.1. La sélection	29
5.1.1. La sélection proportionnelle (roue de loterie)	29
5.1.2. La sélection à reste stochastique	30
5.1.3. La sélection par tournoi stochastique	30
5.2. Le croisement	30
5.3. La mutation	31
6. Notion du codage	31
7. Notion du décodage	32
8. Amélioration des techniques de base des AGs	32
8.1. Amélioration des opérateurs de reproduction	32
8.1.1. La sélection	32
8.1.2. L'opérateur de croisement	32
8.2 Optimisation des paramètres de l'algorithme génétique	33
9. Conclusion	34

Chapitre IV : Simulation de la commande

1. Introduction	35
2. Application de la commande floue	35
2.1. Régulateur flou à trois classes	35
2.1.1 Avant l'utilisation des AGs	36
2.1.2 Après l'utilisation des AGs	38
2.2. Régulateur flou à 5 classes	40
2.2.1 Avant l'utilisation des AGs	41
2.2.2 Après l'utilisation des AGs	43
2.3. Régulateur flou à 7 classes	45
2.3.1 Avant l'utilisation des AGs	46
2.3.2 Après l'utilisation des AGs	48
2.4. Régulateur flou à 9 classes	50
2.4.1 Avant l'utilisation des AGs	51
2.4.2 Après l'utilisation des AGs	53
3. Conclusion	56

Conclusion générale

Introduction Générale

La robotique est l'ensemble des études et des techniques de conception et de mise en œuvre des robots. Le robot est un système automatique mécanisé capable d'effectuer une ou plusieurs tâches dans un environnement donné, de manière autonome et par l'exécution d'un programme. C'est une machine programmable capable d'exécuter diverses tâches répétitives, et en principe, d'adapter son comportement à certains aléas et événements perturbant le fonctionnement nominal prévu. Les tâches peuvent être de type point à point ou de type continu: suivi de courbes, suivi de surfaces. Les premières correspondent aux procédés les plus répandus dans l'industrie manufacturière dont le soudage par résistance, la palettisation, le perçage et la manutention. Tandis que les secondes, elles correspondent aux procédés de soudage à l'arc, collage, découpe, projection, peinture [1].

Les robots manipulateurs se substituent à l'homme ou prolongent son action en apportant précision, rapidité ou capacité à appliquer d'importants efforts. Ils sont généralement constitués d'un bras manipulateur plus ou moins anthropomorphe mis en mouvement par des actionneurs électriques, hydrauliques ou pneumatiques. Par rapport à leurs concurrents hydrauliques et pneumatiques, les actionneurs électriques présentent un certain nombre d'avantages parmi lesquels :

- une énergie facilement disponible, soit à partir du secteur, soit à partir de batteries pour les engins autonomes.
- une adaptation aisée de l'actionneur et de sa commande du fait de la nature électrique de l'ensemble des grandeurs [2].

1. Robots industriels

1.1. Introduction

Un robot industriel est un système poly articulé à l'image d'un bras humain souvent composé de 6 degrés de liberté, 3 axes destinés au positionnement et 3 axes à l'orientation permettant de déplacer et d'orienter un outil (organe effecteur) dans un espace de travail donné.

Jusqu'à présent, les robots industriels sont surtout employés pour servir que de générateur de positions et d'orientations d'un outil dans l'espace. Pour cela, ils ne sont modélisés que par trois modèles: Le modèle géométrique, Le modèle cinématique et Le modèle dynamique.

Les robots industriels sont désormais munis de systèmes de vision qui accroissent leur souplesse d'exécution et permettent de vérifier la qualité des produits fabriqués.

Les robots industriels ont été développés pour intervenir dans les milieux à risques, par exemple dans l'industrie nucléaire ou dans des environnements créant une forte corrosion.

Un robot industriel peut aussi servir au maniement d'objets lourds, ce qui est une autre utilisation très courante.

Les robots sont depuis longtemps utilisés dans les chaînes de montage de l'industrie Automobile où ils remplacent les ouvriers dans les tâches pénibles et dangereuses (peinture, soudage, emboutissage). [3].

1.2. Historique

L'origine du mot robot est issue du tchèque "robota" qui signifie travail forcé. Le terme de robotique est apparu en 1942 dans l'œuvre de l'écrivain Isaac ASIMOV.

Le premier robot manipulateur industriel, appelé Unimate, était un descendant direct des télémanipulateurs développés pour les besoins du nucléaire. Créé par George Devol dans les années 1950, il fut utilisé pour travailler sur les chaînes d'assemblage de General Motors à partir de 1961.

Parmi les robots industriels, le Kuka Famulus, qui date de 1973, fut le premier robot à 6 axes entraînés de façon électromécanique.

Le Robot Delta, imaginé en 1985 par Reymond Clavel, professeur à l'École Polytechnique fédérale de Lausanne (EPFL), possède un bras de manipulation formé de 3 parallélogrammes. Son brevet tombe dans le domaine public en 2007 et différents constructeurs devraient alors sortir leur propre robot delta.

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

1.3. Classifications Des Robots Industriels

Dans la robotique, on peut distinguer les robots industriels, composés d'un bras articulé muni d'un effecteur, et les robots mobiles qui peuvent se déplacer en autonomie dans un environnement.

Cependant, certains organismes, comme la JIRA «Association Japonaise des Robots Industriels» ou l'AFRI «Association Française de Robotique Industrielle», ont établi des classifications plus précises, basées sur la spécificité fonctionnelle des robots.

La classification élaborée par la JIRA est la suivante [4]:

–Classe 1 : TELEMANIPULATEURS.

Bras commandé par un opérateur humain.

–Classe 2 : MANIPULATEURS AVEC SEQUENCE FIXE.

Contrôle automatique, mais difficilement programmable.

–Classe 3 : MANIPULATEURS AVEC SEQUENCE VARIABLE.

Contrôle automatique, reprogramme mécaniquement.

–Classe 4 : ROBOTS « P LAY-BACK ».

Séquences qui sont exécutées à l'origine sous la supervision de l'être humain, mémorisées puis rappelées pour être rejouées.

–Classe 5 : ROBOTS AVEC CONTROLEUR NUMERIQUE.

Les positions des séquences sont contrôlées par des données numériques.

–Classe 6 : ROBOTS INTELLIGENTS.

Le robot peut gérer son environnement à des modifications arrivant durant l'exécution.

La classification élaborée par l'AFRI est la suivante :

–Classe A : TELEMANIPULATEURS.

Manipulateur maître/esclave.

–Classe B : MANIPULATEURS AUTOMATIQUES.

Manipulateurs automatiques avec séquences fixes.

Manipulateurs automatiques avec séquences variables (Machines à commandes numériques).

–Classe C : ROBOTS PROGRAMMABLES.

1ère génération de robots.

–Classe D : ROBOTS INTELLIGENTS.

2ème génération de robots, munis de système de vision.

1.4. Applications des robots industriels

Les applications des robots industriels sont multiples et vont de la mise en boîte (dans les domaines de l'agro-alimentaire ou de la pharmacie) à l'usinage de haute précision, à l'échelle du nanomètre, en passant par la prise de pièces pour le chargement d'automates.

Ainsi, les robots peuvent être utilisés en usine, pour le soudage, la manutention, la palettisation, l'empaquetage, la transformation et toutes sortes de tâches d'automatisation.

Parmi les robots industriels les plus répandus on distingue les bras manipulateurs industriels notés les manipulateurs.

Le bras manipulateur en milieu industriel réalise des tâches de façon répétitive, à hautes vitesses, pour satisfaire les temps d'exécution.

Les lois de commande utilisées utilisent des capteurs proprioceptifs pour mesurer l'état du robot et des capteurs extéroceptifs pour mesurer des variables liés à la tâche.

Lorsque l'espace de travail est sécurisé et la tâche connue à priori, les calculs de la trajectoire sont réalisés hors ligne pour minimiser les temps d'exécution [4].

2. Robot manipulateur

Un manipulateur est un robot industriel, dont la fonction est de réaliser des tâches précises et répétitives. Il est composé d'une mécanique à axes numériques et d'un système de commande. Le manipulateur est utilisé dans de nombreux secteurs d'activités comme l'automobile, la sidérurgie, la plasturgie....

Précis et rapide, le manipulateur est en outre capable de soulever de fortes charges et de déplacer les pièces les plus délicates.

Dans le domaine de la transformation des matières plastiques, le manipulateur est essentiellement utilisé pour garantir qualité et productivité des pièces fabriquées par les presses à injecter. Dotés d'un système de programmation par apprentissage, ils sont très simples à mettre en œuvre.

Un bras manipulateur est un système électromécanique capable d'interagir avec son environnement. Il est constitué de deux parties distinctes :

Un organe terminal : C'est un dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique, il regroupe les procédés destinés à manipuler des objets, ou à les transformer, il s'agit donc d'une interface permettant au robot d'interagir avec son environnement.

Une structure mécanique articulée : C'est une chaîne cinématique, à corps généralement rigides (segments), assemblés par des liaisons (articulations).

Les articulations sont des dispositifs se trouvant entre deux segments.

Ils lient deux corps successifs ; cette disposition permet le mouvement entre ces corps, ce qui a comme conséquence l'obtention d'un mouvement relatif des segments voisins.

Cette structure articulée supporte l'organe terminal à situer, son rôle est de l'amener à une position et une orientation donnée [3].

2.1. Structure générale d'un robot manipulateur

Description du robot manipulateur dans son environnement en 5 composantes interagissant

2.1.1. Structure mécanique

- membres rigides, liaisons articulées par des joints mécaniques
- à son extrémité l'outil ou effecteur

2.1.2. Actionneurs

- pour lutter contre la gravité, l'inertie, les forces extérieures;
- modifier la configuration;
- de types électriques / pneumatiques / hydrauliques

2.1.3. Les systèmes de transmissions mécaniques

- connecter les actionneurs à la structure mécanique
- transmettre et adapter les sources d'énergie aux charges

2.1.4. Les capteurs

- tactiles, optiques, électriques...
- capteurs proprioceptifs dont le but est de fournir l'information sur la configuration du manipulateur
- capteurs extéroceptifs dont le but est de donner des informations sur l'environnement

2.1.5. Unité de contrôle

- rôle d'information: collecter l'information venant des capteurs
- rôle de décision: partant d'une tâche définie et tenant compte des données du système et de l'environnement, établir des actions
- rôle de communication [3].

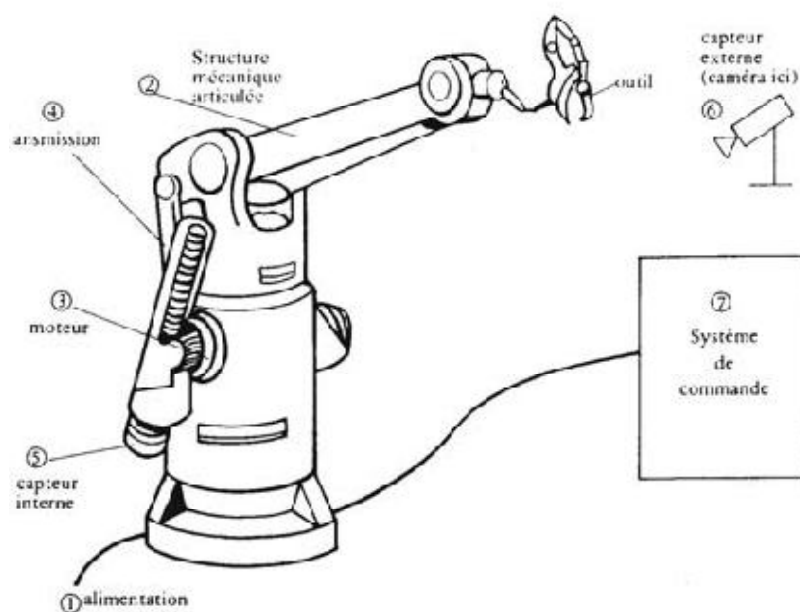


Fig 1.1 Structure générale d'un robot manipulateur.

3. Commande des bras manipulateurs

Le problème de la commande d'un robot manipulateur peut être formulé comme la détermination de l'évolution des forces généralisées (forces ou couples) que les actionneurs doivent exercer pour garantir l'exécution de la tâche tout en satisfaisant certains critères de performance.

Différentes techniques sont utilisées pour la commande des bras manipulateurs.

La conception mécanique du bras manipulateur a une influence sur le choix du schéma de commande. Un robot manipulateur est une structure mécanique complexe dont les inerties par rapport aux axes des articulations varient non seulement en fonction de la charge mais aussi en fonction de la configuration, des vitesses et des accélérations.

Deux types de mouvements apparaissent quand on parle de commande du bras manipulateur. Un premier type considère que les mouvements nécessaires pour la réalisation de la tâche sont exécutés dans l'espace libre.

Le deuxième type considère des mouvements spécifiques avec des forces de contact pour l'organe terminal qui se déplace dans un espace contraint [1].

3.1. Commande dans l'espace libre

– *Commande par articulation*

Cette technique est utilisée par des robots manipulateurs qui utilisent des servomoteurs avec de forts rapports de réduction.

Lorsque le système présente un comportement linéaire, l'asservissement du mouvement peut être réalisé par des techniques classiques de commande. Nous parlons alors d'une commande décentralisée de type PID.

– *Commande jacobéenne*

Cette technique est utilisée depuis les travaux de Whitney [Whitney, 1969] et elle est appelée de cette façon lorsqu'elle utilise la matrice jacobéenne inverse du bras manipulateur pour calculer les vitesses de consigne aux articulations.

– *Commande passive*

Cette technique considère le robot comme un système passif, c'est à dire un système qui dissipe de l'énergie. De telles lois de commande permettent de modifier l'énergie naturelle du robot pour qu'il réalise la tâche. En utilisant le formalisme de Hamilton, la commande cherche à minimiser l'énergie du système en utilisant un bloc non linéaire passif dans la boucle de retour. La commande passive tend à être plus robuste que le découplage non linéaire, lorsque la technique ne recherche pas à annuler les non linéarités

– *Commande adaptative*

Ce type de techniques vise à corriger les déficiences de la commande par découplage non linéaire, comme la connaissance approximative des paramètres du modèle du robot ou pour s'adapter aux différentes conditions opératoires

– *Commande optimale*

Pour réaliser une tâche, il peut exister un grand nombre de solutions.

Dans ce cas, il peut être souhaitable de choisir une solution qui satisfasse un certain critère [2].

3.2. Commande dans l'espace contraint

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

Lorsque le robot est contraint par l'environnement suivant toutes les directions, aucun déplacement de l'organe terminal n'est possible et il ne peut qu'exercer des efforts sur l'environnement ; en revanche, lorsque le robot est dans l'espace libre, l'absence de tout contact empêche la génération d'efforts.

Ainsi, les notions de commande en position et de commande en effort s'excluent mutuellement : on ne peut contrôler simultanément une force et un déplacement suivant une même direction [1].

4. Modélisation d'un bras manipulateur

4.1. Modèle du robot manipulateur

Un système mécanique peut être traduit sous forme d'un modèle dynamique pour faciliter son étude grâce aux équations différentielles, qui existent entre les variables d'états du mécanisme, leurs dérivés et les forces extérieures agissant sur chaque corps.

La forme la plus générale est :

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = U \quad (1)$$

4.2. Modèle cinématique

Un robot à structure cinématique ouverte est constitué de plusieurs corps reliés entre eux par des axes.

Ces derniers réalisent généralement une liaison en rotation ou en translation.

L'ensemble des paramètres définissant les angles de rotation ou les longueurs de translation des axes sont appelés *coordonnées articulaires* et seront notées q .

On modélise un robot en attachant un repère à chaque axe. On considère le robot comme un chaînage de $n+1$ corps numérotés de 0 à n . Le corps 0 représente la base du robot tandis que l'organe terminal porte le numéro n .

La position du corps k est repérée par rapport au corps $(k-1)$ grâce à la matrice homogène $T_{(k-1)(k)}$. Cette matrice définit la transformation entre le repère R_{k-1} attaché au corps $(k-1)$ et le repère R_k attaché au corps k exprimée dans le repère R_{k-1} . La matrice $T_{(k-1)(k)}$ dépend de q_k , la coordonnée articulaire de l'axe k , on la notera donc $T_{(k-1)(k)}(q_k)$ [5].

Le modèle géométrique du robot est l'expression, en fonction des coordonnées articulaires, de l'attitude de l'organe terminal du robot par rapport à sa base.

Il est défini par la matrice homogène $T_{0n}(q)$ où $q = [q_1 \dots q_n]^T$ est le vecteur des coordonnées articulaires.

On a donc [5]:

$$T_{0n}(q) = T_{01}(q_1)T_{12}(q_2) \dots T_{(n-1)n}(q_n) \quad (2)$$

4.3. Modèle dynamique

Le modèle dynamique (3) exprime les couples (ou les forces) moteurs des actionneurs des différents bras du robot manipulateurs en fonctions des positions q , des vitesses \dot{q} , et des accélérations \ddot{q} , articulaires et des forces extérieures F à exercer sur l'organe terminal.

Il exprime l'équilibre entre les couples d'entraînement et le couple de freinage dus aux inerties, aux forces centrifuges et de Coriolis ainsi qu'aux forces de gravitation.

Ce modèle est aussi appelé, modèle dynamique inverse.

$$U = f(q, \dot{q}, \ddot{q}, F) \quad (3)$$

Où :

- $U \in \mathbb{R}^n$: Vecteurs des couples moteurs dont la dimension est égale au nombre de degrés de liberté du robot manipulateur.
- q : Vecteur des positions articulaires.
- \dot{q} : Vecteur des vitesses articulaires.
- \ddot{q} : Vecteur des accélérations articulaires.
- F : Forces ou moments extérieurs exercés par et/ou sur l'organe terminal.

4.3.1. Formalisme de Lagrange

Le modèle dynamique (3) peut être obtenu par plusieurs méthodes, la plus utilisée est celle de Lagrange qui décrit le comportement dynamique d'un système en termes de travail et d'énergie.

Le modèle dynamique est obtenu par les équations d'Euler-Lagrange (EL) suivantes :

$$\frac{d}{dt} \left[\frac{\delta L}{\delta \dot{q}_i} \right] - \frac{\delta L}{\delta q_i} = U_i \quad i=1 \dots n, \quad (4).$$

Avec L : désigne la fonction de Lagrange donnée par l'équation :

$$L(q, \dot{q}) = K(q, \dot{q}) - U(q). \quad (5).$$

$$\text{Avec : } \begin{cases} K(q, \dot{q}) : \text{Energie cinétique.} \\ U(q) : \text{Energie potentielle.} \end{cases}$$

4.3.2. Calcul de l'énergie cinétique

Soit un point de l'articulation « i » de vecteur ${}^i r$ des coordonnées dans le repère R_i . Les coordonnées de ce point dans le repère R_0 de la base du robot sont exprimées par le vecteur :

$$r = {}^0 T_i {}^i r.$$

Où :

$${}^0 T_i = {}^0 T_1 {}^1 T_2 \dots {}^{i-1} T_i.$$

${}^0 T_i$: Est la matrice $\in \mathbb{R}^{4 \times 4}$ de transition homogène du repère R_i dans le repère R_0 .

Le vecteur de vitesse V du point considéré par rapport au repère R_0 s'écrit :

$$V = \frac{dr}{dt} = \sum_{j=1}^n \left[\frac{\delta {}^0 T_i}{\delta q_j} \dot{q}_j \right] {}^i r. \quad (6).$$

L'énergie cinétique d'une masse infinitésimale « dm » en ${}^i r$ ayant un vecteur de vitesse

$V=[V_x \ V_y \ V_z]^T$ est définie par :

$$dK_i = \frac{1}{2} (V_x^2 + V_y^2 + V_z^2) dm = \frac{1}{2} \text{trace}(VV^T) dm. \quad (7).$$

On remplace V par son expression dans (7), on trouve :

$$dK_i = \frac{1}{2} \left[\sum_{j=1}^n \sum_{k=1}^n \frac{\delta {}^0T_i}{\delta q_j} {}^i r \ {}^i r^T dm \frac{\delta {}^0T_i^T}{\delta q_k} \dot{q}_j \dot{q}_k \right]. \quad (8).$$

L'énergie cinétique de l'articulation « i » est donnée par :

$$K_i = \int_{\text{articulation } i} dK_i. \quad (9).$$

L'intégrale n'agit que sur le terme ${}^i r \ {}^i r^T$, nous définissons ainsi une pseudo-matrice d'inertie $\in \mathbb{R}^{4 \times 4}$ pour l'articulation i :

$$I_i = \int_{\text{articulation } i} {}^i r {}^i r^T dm = \begin{bmatrix} \int x^2 dm & \int xy dm & \int xz dm & \int x dm \\ \int xy dm & \int y^2 dm & \int yz dm & \int y dm \\ \int xz dm & \int yz dm & \int z^2 dm & \int z dm \\ \int x dm & \int y dm & \int z dm & \int dm \end{bmatrix}. \quad (10)$$

L'énergie cinétique totale du robot s'écrit alors :

$$K = \sum_{i=1}^n K_i = \frac{1}{2} \sum_{i=1}^n \text{trace} \left[\sum_{j=1}^n \sum_{k=1}^n \frac{\delta {}^0T_i}{\delta q_j} I_i \frac{\delta {}^0T_i^T}{\delta q_k} \dot{q}_j \dot{q}_k \right]. \quad (11).$$

Puisque la trace d'une somme de matrices est égale la somme des traces individuelles de chaque matrice, K s'écrit alors :

$$K = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n m_{jk}(q) \dot{q}_j \dot{q}_k.$$

Où :

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q}.$$

Où l'élément $M_{jk}(q)$ de $M(q) \in \mathbb{R}^{4 \times 4}$ est défini par :

$$M_{jk}(q) = \sum_{i=1}^n \text{trace} \left[\frac{\delta T_i}{\delta q_j} I_i \frac{\delta {}^0T_i^T}{\delta q_k} \right]. \quad (12).$$

Nous pouvons conclure que la matrice d'inertie $M(q)$ est symétrique [$M(q)=M(q)^T$], et définie positive $M(q) > 0$.

Alors l'énergie cinétique K dépend des variables articulaires q et \dot{q} .

4.3.3. Calcul de l'énergie potentielle

L'énergie potentielle d'un robot manipulateur est donnée par :

$$U(q) = \sum_{j=1}^n U_j(q) \quad (13).$$

où : $U_j(q)$: désigne l'énergie potentielle du corps C_j , et elle est définie par:

$$U_j(q) = -m_j {}^0\mathbf{g}^T {}^0\mathbf{p}_j(q).$$

Avec:

${}^0\mathbf{g}^T$: Vecteur $\in \mathbb{R}^3$ de l'accélération de la gravitation, exprimé dans le repère R_0 de la base du robot.

${}^0\mathbf{p}_j$: Vecteur $\in \mathbb{R}^3$ des coordonnées du centre de masse du corps C_j par rapport repère R_0 .

Étant donné que le vecteur ${}^0\mathbf{p}_j$ est en fonction des positions articulaires q , l'énergie potentielle U dépend alors de la variable q .

4.4. Modèle du robot à deux degrés de liberté (ddl)

Nous allons modéliser un robot à deux degrés de liberté et qui a deux articulations rotoïdes (Figure 1.2), c'est-à-dire qui se déplace dans le plan vertical.

m_1 :Masse du corps 1.

m_2 : Masse du corps 2

G_1 : Centre de masse du corps 1.

G_2 : Centre de masse du corps 2.

L_1 : Longueur du corps 1.

L_2 : Longueur du corps 2.

L_{c1} : position du centre de masse G_1 par rapport à O_1 .

L_{c2} : positions du centre de masse G_2 par rapport à O_2 .

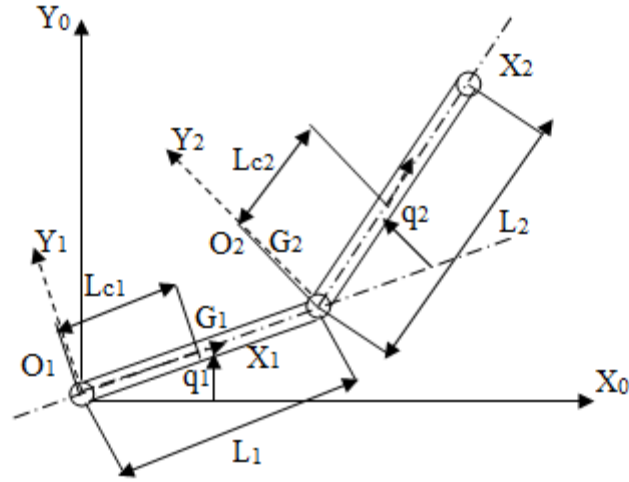


Fig 1.2 Représentation du robot à deux axes rotoïdes.

Ce bras est constitué de deux axes de masses respectives m_1 et m_2 de longueurs respectives L_1 et L_2 .

Les vecteurs numériques utilisés pour la simulation sont celles des axes 2 et 3 du robot Puma 560.

Soit :

$X = \begin{bmatrix} x \\ y \end{bmatrix}$: Vecteur des coordonnées opérationnelles donnant la position de l'organe terminal.

$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$: Vecteur des coordonnées généralisées.

4.4.1. Modèle géométrique

Les positions des masses m_1 , m_2 sont données par :

$$\begin{bmatrix} x_{c1} \\ y_{c1} \end{bmatrix} = \begin{bmatrix} L_{c1} \cdot \cos q_1 \\ L_{c1} \cdot \sin q_1 \end{bmatrix}$$

$$\begin{bmatrix} X_{c2} \\ Y_{c2} \end{bmatrix} = \begin{bmatrix} L_1 \cdot \cos q_1 + L_{c2} \cdot \cos(q_1 + q_2) \\ L_1 \cdot \sin q_1 + L_{c2} \cdot \sin(q_1 + q_2) \end{bmatrix}$$

Avec $L = L_1/2$.

4.4.2. Modèle cinématique

Le modèle cinématique appelé aussi modèle variationnel permet de calculer les vitesses ou la variation de l'organe terminal en fonction des vitesses articulaires.

$$V_{C1} = \begin{bmatrix} -L_{c1} \dot{q}_1 \sin q_1 \\ L_{c1} \dot{q}_1 \cos q_1 \end{bmatrix}$$

$$V_{C2} = \begin{bmatrix} -L_1 \dot{q}_1 \sin q_1 - L_{c2} (\dot{q}_1 + \dot{q}_2) \sin(q_1 + q_2) \\ L_1 \dot{q}_1 \cos q_1 + L_{c2} (\dot{q}_1 + \dot{q}_2) \sin(q_1 + q_2) \end{bmatrix}$$

4.4.3. Paramètres de Denavit-Hartenberg(D-H)

Nous définissons les paramètres (D-H) de ce robot manipulateur à deux degrés de liberté comme suit.

$$d_1 = \vec{O_0 O_1} \cdot \vec{X}_0 = 0, \alpha_2 = \text{Angle}(\vec{Z}_1, \vec{Z}_2) = 0, \alpha_1 = \text{Angle}(\vec{Z}_0, \vec{Z}_1) = 0,$$

$$\theta_2 = \text{Angle}(\vec{X}_1, \vec{X}_2) = q_2, \theta_1 = \text{Angle}(\vec{X}_0, \vec{X}_1) = q_1, d_2 = \vec{O_1 O_2} \cdot \vec{X}_1 = L_1,$$

$$r_2 = \vec{O_1 O_2} \cdot \vec{Z}_2 = 0, r_1 = \vec{O_0 O_1} \cdot \vec{Z}_0 = 0,$$

4.4.4. Calcul de la matrice du passage

Les matrices de passage intermédiaires sont 0T_1 et 1T_2 .

Exprimées suivant la forme (3).

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

$${}^1T_2 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & L_1 \\ \sin(q_2) & \cos(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donc la matrice de passage globale est donnée par :

$${}^0T_2 = {}^0T_1 {}^1T_2 = \begin{bmatrix} \cos(q_1 + q_2) & -\sin(q_1 + q_2) & 0 & L_1 \cos(q_1) \\ \sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 & L_1 \sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14).$$

4.4.5. Calcul de la matrice d'inertie

Afin de calculer la matrice d'inertie $M(q)$, on calcule d'abord ses éléments $m_{jk}(q)$, on a donc :

$$M(q) = \frac{1}{2} \text{trace} \left[\frac{\delta T_1}{\delta q_1} I_1 \frac{\delta {}^0 T_1^T}{\delta q_1} \right] + \frac{1}{2} \text{trace} \left[\frac{\delta {}^0 T_2}{\delta q_1} I_2 \frac{\delta {}^0 T_2^T}{\delta q_1} \right] \quad (15).$$

$$I_1 = \begin{bmatrix} \int x^2 dm_1 & 0 & 0 & m_1 L_{c1} \\ 0 & \int y^2 dm_1 & 0 & 0 \\ 0 & 0 & \int z^2 dm_1 & 0 \\ m_1 L_{c1} & 0 & 0 & m_1 \end{bmatrix}$$

$$I_2 = \begin{bmatrix} \int x^2 dm_2 & 0 & 0 & m_2 L_{c2} + m_p L_2 \\ 0 & \int y^2 dm_2 & 0 & 0 \\ 0 & 0 & \int z^2 dm_2 & 0 \\ m_2 L_{c2} + m_p L_2 & 0 & 0 & m_2 \end{bmatrix}$$

Après le développement de la formule (15) on obtient :

$$m_{11}(q) = j_1(\theta_1) + j_2(\theta_2) + m_2 L_1^2 + 2(m_2 L_1 L_{c2} + m_p L_1 L_2) \cos(q_2).$$

$$\text{Avec : } \begin{cases} j_1(\theta_1) = \int (x^2 + y^2) dm_1 \\ j_2(\theta_2) = \int (x^2 + y^2) dm_2 \end{cases}$$

Telle que $j_i(\theta_i)$ représente l'inertie par rapport à l'axe (O_i, Z_i) du corps C_i .

De la même manière l'application permet d'écrire

$$m_{22}(q) = j_2(\theta_2).$$

$$m_{21}(q) = m_{12}(q) j_2(\theta_2) + (m_2 L_{c2} L_1 + m_p L_1 L_2) \cos(q_2).$$

4.4.6. Calcul du vecteur des termes de gravité

L'énergie potentielle de chaque corps C_i ($i=1,2$) est :

$$\begin{cases} U_1 = -g m_1 L_{c1} \cos(q_1) \\ U_2 = -g(m_2 L_1 + m_p L_1) \cos(q_1) - g(m_2 L_{c2} + m_p L_2) \cos(q_1 + q_2) \end{cases}$$

D'où l'expression de l'énergie potentielle totale

$$U = -g[(m_1 L_{c1} + m_2 L_1 + m_p L_1) \sin(q_1) + (m_2 L_{c2} + m_p L_2) \cos(q_1 + q_2)]$$

Et d'après la relation $G(q) = \frac{\delta U}{\delta q_i}$ on trouve

$$G_1(q) = \frac{\delta U}{\delta q_1} = g \left[(m_1 L_{c1} + m_2 L_1 + m_p L_1) \sin(q_1) + (m_2 L_{c2} + m_p L_2) \sin(q_1 + q_2) \right]$$

$$G_2(q) = \frac{\delta U}{\delta q_2} = g \left[(m_2 L_{c2} + m_p L_2) \sin(q_1 + q_2) \right]$$

4.4.7. Calcul de la matrice $C(q, \dot{q})$

A partir de la relation du modèle dynamique et la matrice $M(q)$, on peut calculer les éléments de la matrice $C(q, \dot{q})$, on a donc :

$$C_{11}(q, \dot{q}) = \frac{1}{2} \left[\dot{q}^T \frac{\delta m_{11}(q)}{\delta q} + \left(\frac{\delta m_{11}(q)}{\delta q_1} - \frac{\delta m_{11}(q)}{\delta q_1} \right) \dot{q}_1 + \left(\frac{\delta m_{12}(q)}{\delta q_1} - \frac{\delta m_{12}(q)}{\delta q_1} \right) \dot{q}_2 \right]$$

$$C_{11}(q, \dot{q}) = -(m_2 L_1 L_{c2} + m_p L_1 L_2) \dot{q}_2 \sin(q_2). \quad (16).$$

$$C_{12}(q, \dot{q}) = \frac{1}{2} \left[\dot{q}^T \frac{\delta m_{12}(q)}{\delta q} + \left(\frac{\delta m_{11}(q)}{\delta q_2} - \frac{\delta m_{21}(q)}{\delta q_1} \right) \dot{q}_1 + \left(\frac{\delta m_{22}(q)}{\delta q_2} - \frac{\delta m_{22}(q)}{\delta q_1} \right) \dot{q}_2 \right]$$

$$C_{12}(q, \dot{q}) = -(m_2 L_1 L_{c2} + m_p L_1 L_2) (\dot{q}_1 + \dot{q}_2) \sin(q_2). \quad (17).$$

$$C_{21}(q, \dot{q}) = \frac{1}{2} \left[\dot{q}^T \frac{\delta m_{21}(q)}{\delta q} + \left(\frac{\delta m_{21}(q)}{\delta q_1} - \frac{\delta m_{11}(q)}{\delta q_2} \right) \dot{q}_1 + \left(\frac{\delta m_{22}(q)}{\delta q_1} - \frac{\delta m_{12}(q)}{\delta q_2} \right) \dot{q}_2 \right]$$

$$C_{21}(q, \dot{q}) = (m_2 L_1 L_{c2} + m_p L_1 L_2) \dot{q}_1 \sin(q_2). \quad (18).$$

$$C_{22}(q, \dot{q}) = \frac{1}{2} \left[\dot{q}^T \frac{\delta m_{22}(q)}{\delta q} + \left(\frac{\delta m_{21}(q)}{\delta q_2} - \frac{\delta m_{21}(q)}{\delta q_2} \right) \dot{q}_1 + \left(\frac{\delta m_{22}(q)}{\delta q_2} - \frac{\delta m_{22}(q)}{\delta q_2} \right) \dot{q}_2 \right]$$

$$C_{22}(q, \dot{q}) = 0. \quad (19).$$

4.4.8. Modèle dynamique du robot à deux degrés de liberté

D'après les paragraphes précédents, les matrices et le vecteur des termes des gravités du modèle dynamique du bras manipulateur sont exprimés comme suit:

Matrice d'inertie :

$$M(q) = \begin{bmatrix} j_1(\theta_1) + j_2(\theta_2) + m_2 L_1^2 + 2(m_2 L_1 L_{c2} + m_p L_1 L_2) \cos(q_2) & j_2(\theta_2) + (m_2 L_1 L_{c2} + m_p L_1 L_2) \cos(q_2) \\ j_2(\theta_2) + (m_2 L_1 L_{c2} + m_p L_1 L_2) \cos(q_2) & j_2(\theta_2) \end{bmatrix} \quad (20).$$

Matrice des forces centrifuge et Coriolis :

$$C(q, \dot{q}) = \begin{bmatrix} -(m_2 L_1 L_{c2} + m_p L_1 L_2) \dot{q}_2 \sin(q_2) & (m_2 L_1 L_{c2} + m_p L_1 L_2) (\dot{q}_1 + \dot{q}_2) \sin(q_2) \\ (m_2 L_1 L_{c2} + m_p L_1 L_2) \dot{q}_1 \sin(q_2) & 0 \end{bmatrix} \quad (21).$$

Vecteur des efforts gravitationnels.

$$G(q) = \begin{bmatrix} g(m_1 L_{c1} + m_2 L_1 + m_p L_1) \sin(q_1) + (m_2 L_{c2} + m_p L_2) \sin(q_1 + q_2) \\ (m_2 L_{c2} + m_p L_2) \sin(q_1 + q_2) \end{bmatrix} \quad (22).$$

Les paramètres réels du robot

$$m_2 = 5,04 \text{ Kg} \quad m_1 = 17,40 \text{ Kg} \quad ,$$

Paramètres géométriques

$$L_1 = 431,8 \text{ mm} \quad , L_2 = 433,07 \text{ mm}$$

Paramètre d'inertie

$$I_1 = \frac{1}{3} m_1 L_1^2 \quad I_2 = \frac{1}{3} m_2 L_2^2 .$$

$$Im_1 = 4,71 \text{ Kg } m^2 \quad Im_2 = 0,83 \text{ Kg } m^2 .$$

5. Conclusion

Nous avons présenté dans ce chapitre le robot industriel, le robot manipulateur industriel et ses classifications ainsi que la commande et les principaux composants qui constituent le robot manipulateur.

Nous avons donné un aperçu de la modélisation cinématique et dynamique d'un robot manipulateur de 2 degrés de liberté en se basant sur les paramètres de Denavit-Hartenberg (D-H), afin de tester les différentes commandes.

1. Introduction

Les connaissances de l'univers dans lequel nous évoluons sont généralement imparfaites dans la mesure où elles peuvent souffrir d'incertitudes et/ou d'imprécisions, ne serait-ce qu'à travers la perception que nous en avons [6].

Or, nous pouvons constater que l'homme intègre naturellement ces imperfections dans la vie de tous les jours, en particulier au niveau du raisonnement et de la décision.

L'idée de Zadeh, en 1965, à travers le nouveau concept ensembliste d'appartenance graduelle d'un élément à un ensemble, a été de définir une logique multivaluée permettant de modéliser ces imperfections (prendre en compte les états intermédiaires entre le tout et le rien).

En ce qui concerne le contrôle d'un processus quelconque, la logique floue permet une approche novatrice par rapport à l'automatique classique.

En automatique, en général, on s'attache à modéliser le processus au travers d'un certain nombre d'équations différentielles, cette modélisation est rendue difficile à mesure que la complexité des processus à contrôler augmente.

D'une manière radicalement opposée, un contrôleur flou va décrire non pas le processus mais la façon de le contrôler, tout comme le ferait un expert humain à travers des règles intégrant naturellement imprécisions et incertitudes.

2. Principe et historique de la logique floue

La logique floue est une logique qui substitue à la logique binaire une logique fondée sur des variables pouvant prendre, outre les valeurs «vrai» ou «faux», les valeurs intermédiaires «vrai» ou «faux» avec un certain degré [7]. Ce qui caractérise le raisonnement humain qui est basé sur des données imprécises ou incomplètes.

En effet, déterminer si une personne est de petite ou de grande taille est aisé pour n'importe lequel d'entre nous, et cela sans nécessairement connaître sa taille.

Supposons que la limite soit de 1.65m, et je mesure 1.63m. Suis-je vraiment petit ?

Bien que dans l'esprit de tout le monde le mot «flou» soit de connotation négative, il n'en est rien en réalité.

Venant à l'origine du mot «duvet» (en anglais «fuzzy», c'est-à-dire le duvet qui couvre le corps des poussins), le terme «fuzzy» signifie (indistinct, brouillé, mal défini ou mal focalisé), qui se traduit par «flou» en français et se prononce « aïmaï » en japonais [8].

Dans le monde universitaire et technologique, le mot «flou» est un terme technique représentant l'ambiguïté ou le caractère vague des intuitions humaines plutôt que la probabilité.

Voici un Bref historique de la logique floue [9]:

- En 1965, le concept flou apparut grâce au professeur Loft Zadeh (Université de Berkley en Californie).

Il déclara qu'un contrôleur électromécanique doté d'un raisonnement humain serait plus performant qu'un contrôleur classique», et il introduit la théorie des «sous-ensembles flous».

- En 1973, le professeur Zadeh publie un article (dans l'IEEE Transactions on Systems, Man and Cybernetics), il y mentionne pour la première fois le terme de variables linguistiques (dont la valeur est un mot et non un nombre).
- En 1974, Mamdani (Université de Londres) réalise un contrôleur flou expérimental pour commander un moteur à vapeur.
- En 1980, Smidth et Co.A/S (au Danemark), mettent en application la théorie de la logique floue dans le contrôle de fours à ciment. C'est la première mise en oeuvre pratique de cette nouvelle théorie.
- Dans les années 80, plusieurs applications commencent à immerger (notamment au Japon).
- En 1987, 'explosion du flou' au Japon (avec le contrôle du métro de Sendai) et qui atteint son apogée en 1990.
- Aujourd'hui, une vaste gamme de nouveaux produits ont une étiquette «produit flou» (Fuzzy).

3. Applications

Tandis que son application au niveau des systèmes de réglage et de commande est relativement récente, depuis quelques années la commande par la logique floue a connu, essentiellement au Japon à partir de 1985 un essor appréciable. En effet, elle a été appliquée dans des problèmes industriels pour résoudre des problèmes de régulation aussi divers, liés à l'énergie, le transport, les machines outils, et la robotique, etc... [10].

Les principaux domaines de recherche et d'application de la logique floue sont les suivants :

- Terrain : applications majeures
- Instrumentation : capteurs, instruments de mesure, reconnaissance de voix et de caractère,...
- Conception/Jugement : consultation, investissement et développement, horaires de train, ...
- Ordinateurs : opérateurs, unités arithmétiques, micro-ordinateurs, ...
- Traitement d'informations : base de données, recherche d'information, modélisation des systèmes, ...
- Automatisation : production du fer et de l'acier, purification, de l'eau, chaînes et robots de fabrication, ...

Les champions toutes catégories de la logique floue sont les Japonais (avec 80 % du marché mondial dans les années 1990). De plus, beaucoup d'entreprises japonaises disposent de laboratoires de recherche. Voici quelques réalisations mises sur le marché :

- En 1987, le métro de Sendai (et peut-être bientôt celui de Tokyo)
- Une machine à laver "intelligente"
- Une photocopieuse "intelligente"
- Une caméra vidéo "intelligente"

4. Notions nécessaires sur la logique floue

4.1. Les concepts

Le concept de la théorie des sous-ensembles flous (et par extension, la logique floue), s'appuie sur la notion de degré d'appartenance d'un élément à un sous-ensemble.

Tandis que les ensembles traditionnels sont caractérisés par une fonction d'appartenance, notée X (également appelée fonction caractéristique) définie sur $\{0,1\}$, les sous-ensembles flous sont, eux, caractérisés par une fonction d'appartenance, notée μ , définie sur $[0, 1]$.

En d'autres termes, dans le langage ensembliste classique, un élément appartient ou n'appartient pas à un ensemble tandis qu'un élément appartient à un sous-ensemble flou avec un certain degré (éventuellement nul).

En résumé, pour un sous-ensemble A défini sur un univers de discours U , on peut écrire :

A sous-ensemble classique : fonction caractéristique $X_A : U \longrightarrow \{0,1\}$

A sous-ensemble flou : fonction d'appartenance $\mu_A : U \longrightarrow [0,1]$

Par extension, ce nouveau concept définit une logique multivaluée qui apparaît comme une généralisation de la logique binaire.

Nous allons maintenant définir un certain nombre de termes propres au domaine de la logique floue auxquels nous pourrions nous référer, si nécessaire, dans la suite de ce travail.

4.2. Sous-ensemble flou

Nous venons de voir ce que l'on entend par sous-ensemble flou, d'un point de vue formel. Un sous-ensemble flou A sur un univers de discours U , est représenté comme dans la (Figure 2.1) à travers sa fonction caractéristique μ_A .

Il peut également être décrit par un certain nombre de caractéristiques comme :

-- son support :

$$\text{Support}(A) = \{\chi \in U / \mu_A(\chi) \neq 0\}. \dots\dots\dots (6)$$

--Sa hauteur :

$$\text{Hauteur}(A) = \sup_{\chi \in U} \{\mu_A(\chi)\}. \dots\dots\dots (7)$$

--Son noyau :

$$\text{Noyau}(A) = \{\chi \in U / \mu_A(\chi) = 1\}. \dots\dots\dots (8)$$

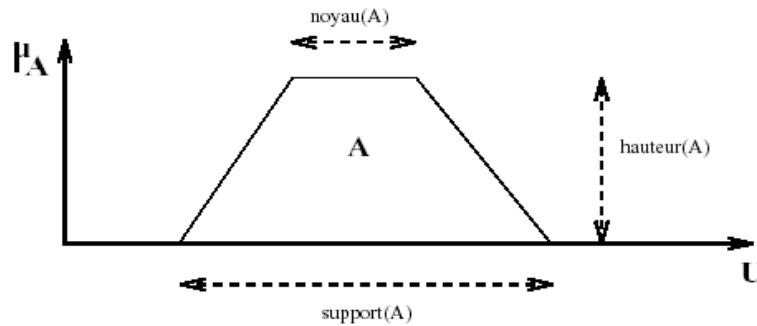


Fig 2.1 Représentation d'un sous-ensemble flou et principales caractéristiques.

Un sous-ensemble flou permet de représenter différentes notions, en particulier la spécificité et la précision [11].

Ces différentes notions apparaissent dans la (Figure 2.2).

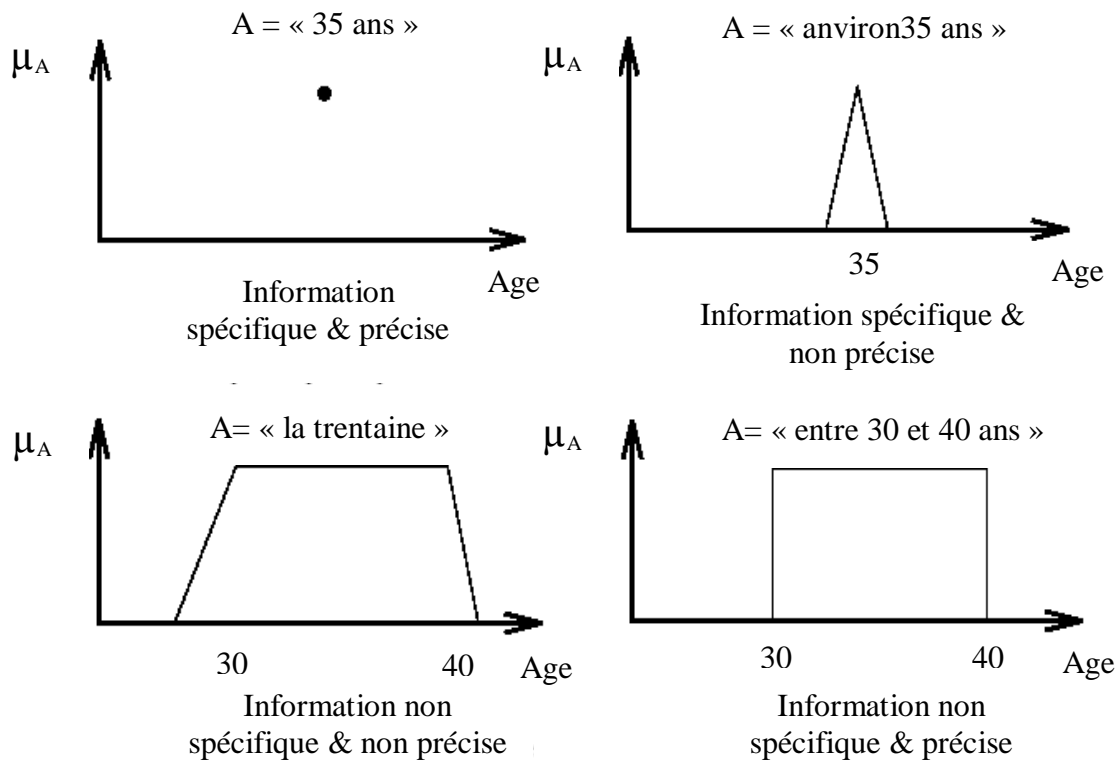


Fig 2.2 Notions de spécificité et de précision représentée à l'aide de sous-ensembles flous.

4.3. Propositions et règle floue

Une proposition floue élémentaire est définie à partir d'une variable linguistique (V ; U ; T_V) par la qualification " V est A ", avec A appartenant à T_V , par exemple "taille de Nabil est MOYENNE" est une proposition élémentaire définie à partir de la variable linguistique (taille de Nabil, {tailles}, { . . . , MOYENNE, . . }).

La valeur de vérité d'une proposition élémentaire " V est A " est égale à $\mu_A(v)$ ou " v " correspond à la valeur numérique exacte de V .

Une proposition floue générale est définie à partir de propositions élémentaires et d'opérateurs logiques binaires (et, ou, implique) ou complémentarité (non).

Il existe plusieurs méthodes pour calculer la valeur de vérité de telles propositions. Nous ne donnons ici que les plus communément utilisées.

Conjonction : $(V_1 \text{ est } A_1) \text{ et } (V_2 \text{ est } A_2)$ (10)

$$\begin{aligned} \text{Min } (\mu_{A_1}(v_1), \mu_{A_2}(v_2)). & \quad (\text{Logique de Zedah}). \\ \text{Max } (\mu_{A_1}(v_1) + \mu_{A_2}(v_2)) \Longleftrightarrow (1,0). & \quad (\text{Logique de Lukasiewicz}). \\ \mu_{A_1}(v_1) \cdot \mu_{A_2}(v_2). & \quad (\text{Logique de probabiliste}). \end{aligned}$$

Disjonction : $(V_1 \text{ est } A_1) \text{ ou } (V_2 \text{ est } A_2)$ (11)

$$\begin{aligned} \text{Max } (\mu_{A_1}(v_1), \mu_{A_2}(v_2)). & \quad (\text{Logique de Zedah}). \\ \text{Min } (\mu_{A_1}(v_1) + \mu_{A_2}(v_2)), 1). & \quad (\text{Logique de Lukasiewicz}). \\ \mu_{A_1}(v_1) + \mu_{A_2}(v_2) \Longleftrightarrow \mu_{A_1}(v_1) \cdot \mu_{A_2}(v_2). & \quad (\text{Logique de probabiliste}). \end{aligned}$$

Implication : $(V_1 \text{ est } A_1) \text{ implique } (V_2 \text{ est } A_2)$ (12)

$$\begin{aligned} \text{Min } (1 \Longleftrightarrow \mu_{A_1}(v_1) + \mu_{A_2}(v_2)), 1). & \quad (\text{Lukasiewicz}). \\ \text{Min } (\mu_{A_1}(v_1), \mu_{A_2}(v_2)). & \quad (\text{Mamdani}). \\ \mu_{A_1}(v_1) \cdot \mu_{A_2}(v_2). & \quad (\text{Larsen}). \end{aligned}$$

Complémentation : non $(V \text{ est } A)$ (13)

$$1 \Longleftrightarrow \mu_A(v).$$

Dans ce qui précède, V , V_1 et V_2 correspondent à des instanciations numériques réelles des variables et V_2 .

Une règle floue est une proposition floue générale utilisant une implication entre deux propositions floues quelconques.

Par exemple :

$(V_1 \text{ est } A_1) \text{ et } (V_2 \text{ est } A_2) \text{ implique } (V_3 \text{ est } A_3)$ flou sous une forme plus linguistique :

Si :

$(V_1 \text{ est } A_1) \text{ et } (V_2 \text{ est } A_2)$

Alors :

$(V_3 \text{ est } A_3)$ est une règle floue.

La partie $(V_1 \text{ est } A_1) \text{ et } (V_2 \text{ est } A_2)$ est appelée prémisse de la règle et $(V_3 \text{ est } A_3)$ conclusion.

5. Architecture classique d'un contrôleur flou (FLC)

L'architecture classique d'un contrôleur flou (FLC) est illustrée dans la figure suivante :

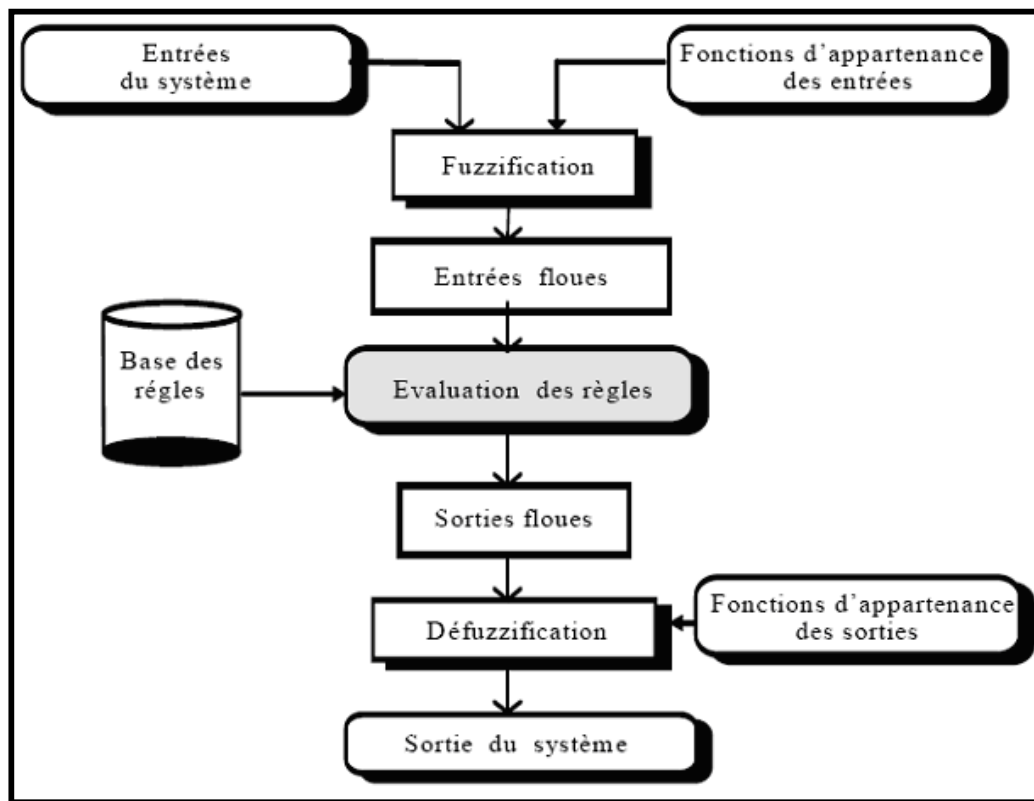


Fig 2.3 La structure d'un contrôleur flou et le flux de données qui le traverse.

On distingue classiquement trois parties dans la structure d'un contrôleur flou : la fuzzification, le mécanisme de décision et la défuzzification [12].

La fuzzification: est l'étape qui permet de transformer une grandeur mesurée sur le processus en un ensemble flou.

La base de connaissances (règles) : celle-ci est composée de :

- Variables linguistiques permettant de caractériser les états associés au processus contrôlé.

- Règles linguistiques codant la connaissance que l'on a sur le contrôle de ce processus.

Le mécanisme d'inférence : cette étape représente le raisonnement du contrôleur.

La défuzzification : est l'étape qui permet de transformer l'ensemble flou, obtenu par le calcul précédent, en une grandeur de commande à appliquer au processus.

Nous allons maintenant revenir en détail sur les différentes étapes dans le traitement des règles.

5.1. Fuzzification

Dans la littérature, l'étape de Fuzzification est qualifiée de transformation permettant de passer :

- d'un niveau discret à un niveau continu.
- d'un niveau numérique/quantitatif à un niveau symbolique/qualitatif [13].

Chaque instanciation d'une variable d'entrée réelle du contrôleur va être traduite en un label caractérisant la variable linguistique associée.

Par exemple, un âge de 30 ans sera traduit en JEUNE, label de la variable linguistique (âge, {âges}, {JEUNE, . . . , AGE}).

Ainsi une information ne sera plus caractérisée par une valeur numérique, mais par un terme linguistique comme JEUNE, si l'on considère un âge par exemple.

Cependant, le degré d'appartenance $\mu_A(v)$ d'une valeur réelle V à un sous-ensemble Flou A doit être mémorisé car celui-ci interviendra dans la détermination des valeurs de vérité des propositions utilisant A .

La (Figure 2.4) illustre le mécanisme de Fuzzification pour deux éléments appartenant au domaine des âges.

Ainsi deux personnes ayant respectivement 30 et 40 ans appartiennent toutes deux à la catégorie "JEUNE" avec cependant des degrés d'appartenance différents

($\mu_{JEUNE}(30) = 0.66$ ET $\mu_{JEUNE}(40) = 0.33$.)

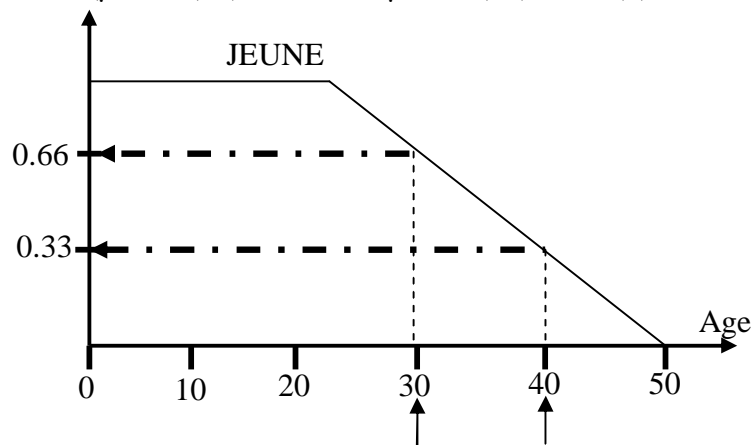


Fig 2.4 Mécanisme de Fuzzification.

5.1.1. Base de connaissances

a. Variable linguistique

Une variable linguistique est définie par un triplet $(V; U; T_V)$ ou V représente une variable classique (âge, température, . . .) définie sur l'univers de discours U .

T_V est l'ensemble des instanciations possibles de la variable V : il s'agit de sous ensembles flous repérés par leur label A_i : on écrit ainsi

$$T_V = \{A_1, A_2, \dots, A_n\} \quad (9)$$

Graphiquement, une variable linguistique peut être représentée comme dans (Figure 2.5).

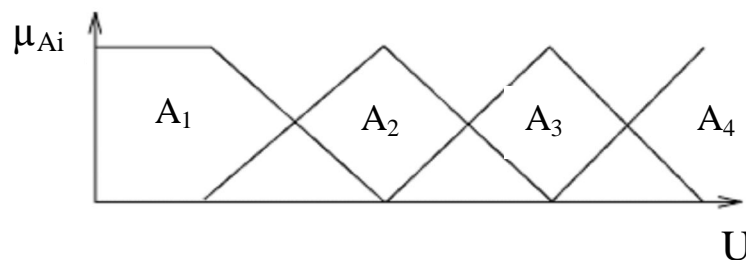


Fig 2.5 Représentation d'une variable linguistique.

b. Règles linguistiques

Les règles linguistiques sont des règles floues dont la forme générale est la Suivante:

"Si prémisses alors conclusion"

Où les prémisses portent sur les entrées du contrôleur flou et les conclusions sur les sorties.

Comme nous l'avons vu précédemment, ces règles sont en fait des propositions floues incluant un opérateur d'implication.

Les conclusions de règles peuvent être de deux sortes [12]:

- les règles de type Mamdani où les conclusions, comme les prémisses, sont des propositions floues :

Si x est A_1 et y est B_1 alors z est C_1 (14)

- les règles de type Sugeno où dans les conclusions, une sortie est en fonction des entrées :

Si x est A_1 et y est B_1 alors $z = f(x; y)$ (15)

5.2. Mécanisme d'inférence

Le mécanisme d'inférence consiste à déterminer les règles floues activées (i.e. les règles dont le degré d'activation $\mu_{\text{prémisse}}$ issu des prémisses est non nul).

Au niveau de chacune de ces règles, ce degré va permettre de déterminer une valeur floue pour la (les) variable(s) de sortie apparaissant dans la partie "conclusion".

Ce degré d'activation est calculé en utilisant les formules décrites précédemment sur la détermination des valeurs de vérité des propositions floues.

Les deux méthodes d'inférence les plus couramment utilisées sont le MINIMUM et le PRODUIT (Figure 2.6).

Si (x est X_1) et (y est Y_1) alors (z est Z_1)

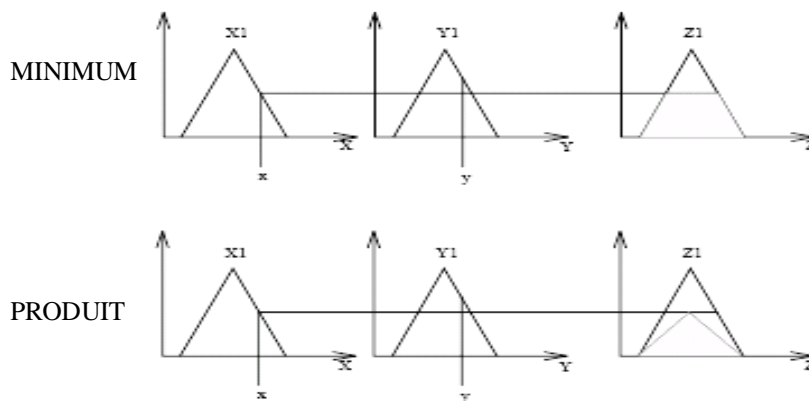


Fig 2.6 Inférence "MINIMUM" et "PRODUIT".

L'inférence MINIMUM consiste à tronquer à la valeur $\mu_{\text{prémisse}}$ la caractérisation floue associée à une variable de sortie, alors que l'inférence PRODUIT revient à affecter à celle-ci un facteur d'échelle correspondant à $\mu_{\text{prémisse}}$.

5.3. Défuzzification

Une fois la phase d'inférence terminée, pour chaque variable linguistique de sortie du contrôleur, il s'agit de regrouper les caractérisations floues issues de l'inférence pour en obtenir une seule par variable, comme méthodes de composition.

On peut citer [13] en particulier les compositions MAXIMUM (en général couplée avec l'inférence MINIMUM) et SOMME (en général couplée avec l'inférence PRODUIT).

La première consiste à prendre le maximum de chacune des caractérisations floues impliquées dans la phase d'inférence pour une variable linguistique déterminée.

La seconde consiste à faire la somme des caractérisations floues issues de l'inférence (Figure 2.7).

On notera que la seconde méthode ne garantit pas une normalisation de la caractérisation floue résultant de la composition.

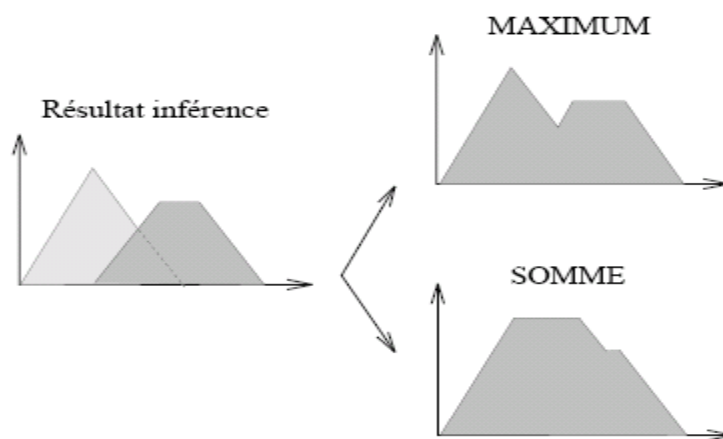


Fig 2.7 Compositions de valeurs floues issues de l'inférence.

6. Les étapes du contrôle flou

On procède tout d'abord à la partition en ensembles flous des différents univers de discours (ou référentiels) que le système impose.

Ensuite on détermine la base de règles qui va caractériser le fonctionnement désiré du système.

Puis il faut transformer les variables réelles, c'est-à-dire celles qui ont une réalité physique, en variables floues.

On appelle cette étape la fuzzification (de fuzzy=flou en anglais). On utilise alors ces variables floues dans un mécanisme d'inférence qui crée et détermine les variables floues de sortie en utilisant les opérations sur les fonctions d'appartenance.

Enfin, on opère à la defuzzification qui consiste à extraire une valeur réelle de sortie à partir de la fonction d'appartenance du sous-ensemble flou de sortie établi par le mécanisme d'inférence.

7. Conclusion

La logique floue, de par la quantité des travaux de recherche et des réalisations existantes, présente un certain nombre de points forts.

En premier lieu, elle simplifie la conception d'un contrôleur dans la mesure où l'on décrit directement le fonctionnement de celui-ci sur le système contrôlé, dans le cas de systèmes complexes, cela représente un avantage indéniable par rapport à la plupart des autres méthodes qui s'attachent à modéliser les systèmes.

De plus, un contrôleur flou est relativement ouvert à travers l'utilisation de règles linguistiques écrite en langage naturel, alors il est possible de gérer de manière assez souple ces règles.

Cependant, le problème majeur réside dans le fait que l'on ne puisse pas prouver la stabilité d'un système flou, contrairement aux modèles mathématiques classiques (fonction de transfert, fonctions de Lyapunov), cela peut présenter des problèmes certains dans des domaines fortement contraints.

1. Introduction

Le désir humain de perfection trouve son expression dans la théorie de l'optimisation qui étudie comment décrire et atteindre ce qui est meilleur, une fois que l'on connaît comment mesurer et modifier ce qui est bon et mauvais.

Dans la pratique un grand nombre de fonctions à optimiser ne sont pas dérivables et souvent même pas continues, ainsi que le monde réel à explorer est envahi de discontinuités, d'espaces de recherches bruités et multi-modaux, ce qui le rend moins adapté au calcul.

Il n'est donc pas surprenant que les méthodes d'optimisation soumises aux contraintes de continuité et de dérivabilité ne soient adaptées qu'à une classe de problèmes très limitée, pour cette raison, et à cause du caractère intrinsèque local de leur exploration, nous devons rejeter ces méthodes classiques. Récemment, une nouvelle classe de méthodes d'optimisation est apparue employant les principes d'évolution et d'hérédité de la nature et présentant une probabilité importante de convergence vers un optimum global de la fonction à optimiser, ce sont des méthodes pseudo-aléatoires appelées "LES ALGORITHMES GENETIQUES".

Les algorithmes génétiques (AGs) ont été développés par John Holland à l'université du Michigan, ses recherches avaient comme objectif de [14] :

- mettre en évidence et expliquer rigoureusement les processus d'adaptation des systèmes naturels.
- concevoir des systèmes artificiels possédant des propriétés importantes des systèmes naturels.

Les algorithmes génétiques sont des algorithmes d'exploration fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils utilisent à la fois les principes de la survie des structures les mieux adaptées, et les échanges d'information pseudo-aléatoires, pour former un algorithme d'exploration qui possède certaines caractéristiques de l'exploration humaine. A chaque génération, un nouvel ensemble de créatures artificielles (des chaînes de caractères) est créé en utilisant des parties des meilleurs éléments de la génération précédente.

Bien qu'utilisant le hasard, les algorithmes génétiques ne sont pas purement aléatoires, ils explorent efficacement l'information obtenue précédemment pour spéculer sur la position de nouveaux points à explorer, avec l'espoir d'améliorer la performance [15].

2. Principe de fonctionnement DES AGs

Les AGs effectuent le procédé d'optimisation en agissant sur une population de créatures artificielles (chaînes de caractères) analogues aux chromosomes en nature.

Chaque créature ou individu représente un point de recherche dans l'espace des solutions à qui on associe une valeur de fonction coût dont on veut obtenir la valeur maximale.

Une description abstraite d'un AG de base peut se faire selon les étapes suivantes [16] :

1. Initialiser aléatoirement une population de chromosomes (individus).
2. Evaluer chaque chromosome dans la population, à chaque chromosome est associé une fonction coût ou fonction d'aptitude déterminant son rang dans la population.

Cette fonction est l'arbitre final décidant la vie ou la mort de chaque individu.

3. Créer de nouveaux chromosomes en appliquant la sélection et les opérateurs de reproduction.

4. Evaluer les nouveaux chromosomes (les descendants) et les insérer dans la population pour construire une nouvelle génération.

Ce processus se répète jusqu'à la satisfaction du critère d'arrêt de l'AG qui est généralement spécifié par un nombre maximal de générations. L'organigramme suivant donne une description générale de l'AG standard [17].

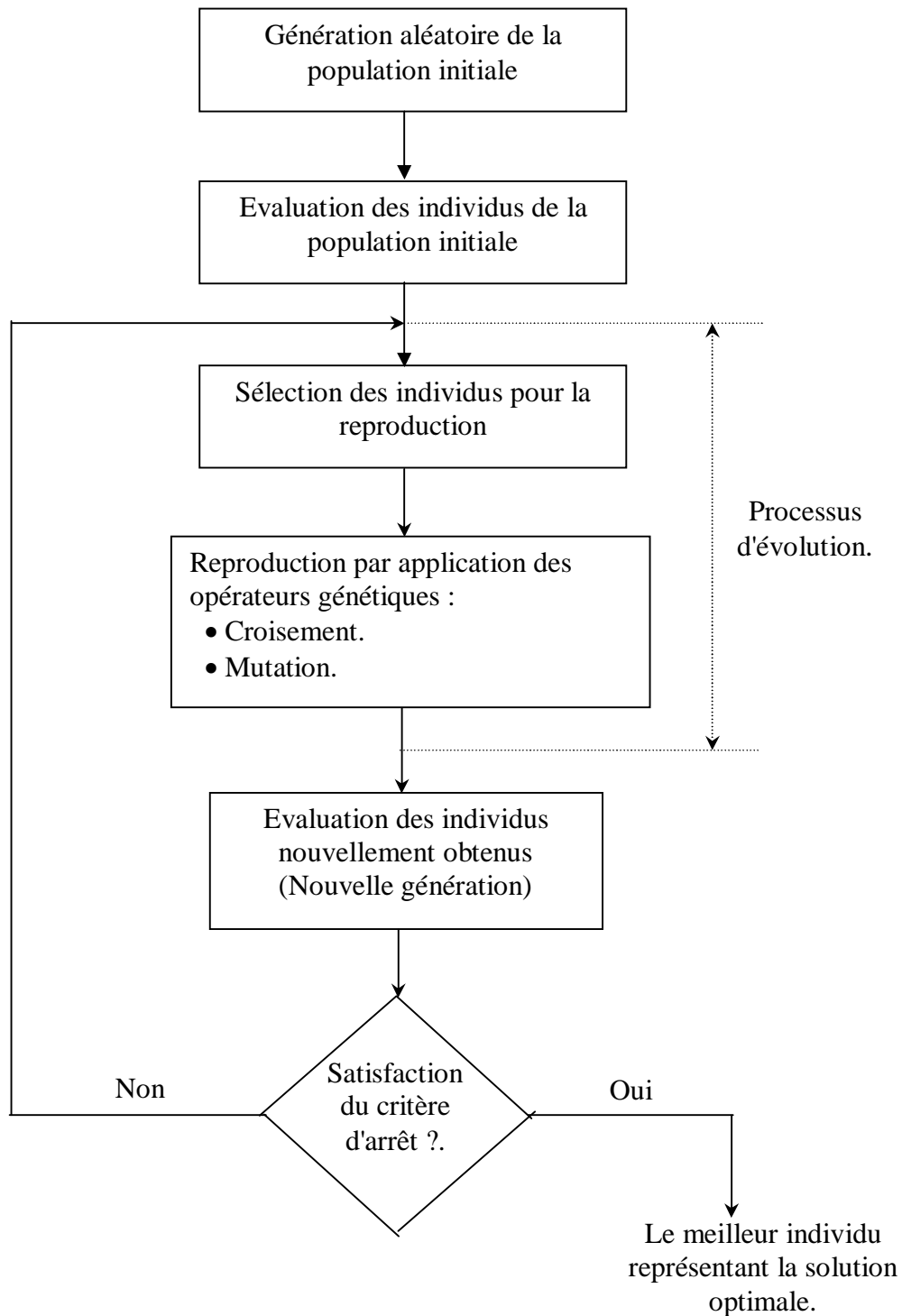


Fig 3.1 Organigramme d'un AG standard.

3. Ce qui rend les AGs différents des méthodes classiques

Pour qu'ils puissent surpasser leurs cousins plus classiques dans la quête de la robustesse, les algorithmes génétiques doivent être fondamentalement différents. Ils le sont en fait selon trois axes principaux [16] :

1. Les AGs utilisent un codage des paramètres, et non pas les paramètres eux-mêmes.
2. Les AGs travaillent sur une population de points, au lieu d'un point unique.
3. Les AGs n'utilisent que les valeurs de la fonction étudiée, pas sa dérivée ou une autre connaissance utilisent des règles de transition probabilistes, et non déterministes.

4. Formulation du problème d'optimisation

En raison de l'analogie avec la théorie de l'évolution naturelle (survie des individus les mieux adaptés à leur environnement), l'algorithme génétique est naturellement formulé en terme de maximisation. Etant donnée une fonction f réelle à une ou plusieurs variables, le problème d'optimisation sur l'espace de recherche E s'écrit de la manière suivante :

$$\max_{x \in E} f(x) \quad (3.1)$$

De plus, la fonction à optimiser par un algorithme génétique doit avoir des valeurs positives sur l'ensemble du domaine E . Dans le cas contraire, il convient d'ajouter aux valeurs de f une constante positive f_{\min} :

$$\max_{x \in E} f(x) + f_{\min} \quad (3.2)$$

Dans beaucoup de problèmes, l'objectif est exprimé sous la forme de minimisation d'une fonction coût g donnée par :

$$\min_{x \in E} g(x) \quad (3.3)$$

Le passage du problème de minimisation à un problème de maximisation est obtenu par transformation de la fonction g . La transformation souvent utilisée est :

$$\max_{x \in E} h(x) \quad (3.4)$$

Avec :

$$h(x) = \begin{cases} G_{\max} - g(x) & \text{si } g(x) \leq G_{\max} \\ 0 & \text{si non} \end{cases} \quad (3.5)$$

La fonction $h(x)$ n'est pas unique, en effet, toute composition de la fonction g par une fonction quelconque conduit à un problème de maximisation équivalent à (3.3). On rencontre notamment dans la littérature la fonction de transformation suivante :

$$h(x) = \frac{1}{1+g(x)} \quad (3.6)$$

5. LES MECANISMES D'UN AG

Les mécanismes d'un algorithme génétique de base sont étonnamment simples et ne mettent en jeu rien de plus compliqué que des copies de chaînes et des échanges de morceaux de chaînes.

Les raisons pour lesquelles ce procédé simple fonctionne sont bien plus complexes et subtiles. La simplicité de mise en œuvre et l'efficacité constituent deux des caractéristiques les plus attrayantes de l'approche proposée par les algorithmes génétiques.

Un algorithme simple composé des opérateurs suivants :

La sélection, le croisement et la mutation, conduit généralement à de bons résultats pour un grand nombre d'applications [18].

5.1. La sélection

La sélection est un procédé dans lequel chaque chaîne est copiée en fonction des valeurs de la fonction à optimiser f (cette fonction est appelée aussi la fonction d'adaptation).

Intuitivement cette fonction peut être envisagée comme une mesure de profit, d'utilité ou de qualité, que l'on souhaite maximiser.

Copier des chaînes en fonction des valeurs de leurs fonctions d'adaptation revient à donner aux chaînes dont la valeur est plus grande une probabilité plus élevée de contribuer à la génération suivante, en créant au moins un descendant.

Cet opérateur est bien entendu une version artificielle de la sélection naturelle, la survie darwinienne des chaînes les plus adaptées.

Dans les populations naturelles, l'adaptation est déterminée par la capacité d'une créature à survivre aux prédateurs, aux maladies et aux autres obstacles à franchir pour atteindre l'âge adulte et la période de reproduction.

Dans notre environnement indéniablement artificiel, la fonction à optimiser est l'arbitre final décidant la vie ou la mort de chaque chaîne créature.

L'opérateur de sélection peut être mis en œuvre sous forme algorithmique de différentes façons :

5.1.1. La sélection proportionnelle (roue de loterie)

Le mode de sélection des parents consiste à dupliquer chaque individu de la population proportionnellement à son adaptation dans son milieu.

Soit :

f_j : la valeur de la fonction coût associée au $j^{\text{ème}}$ individu,

f_s : la somme des valeurs de cette fonction,

La sélection pouvant être faite en utilisant le rapport (f_j/f_s) pour réaliser une roulette pondérée où chaque individu occupe une surface proportionnelle au rapport précédent, des tirages aléatoires sur cette roulette donneront les chaînes qui participeront à la prochaine génération.

De cette façon, les chaînes bien adaptées ont un grand nombre de descendants dans les générations suivantes.

5.1.2. La sélection à reste stochastique

Dans ce mode de sélection, le nombre de copies $n(a_j^t)$ d'un individu a_j^t est directement fixé par le rapport (f_j/f_{moy}) où f_{moy} est la valeur moyenne de la fonction coût. Dans un premier temps, on reproduit chaque individu (partie entière de (f_j/f_{moy})) fois :

$$n(a_j^t) = \text{partie entière de } [f_j/f_{\text{moy}}] \quad (3.7)$$

Ensuite la population est complétée par tirage au sort en associant à chaque individu a_j^t une probabilité $P_s(a_j^t)$:

$$p_s(a_j^t) = f_j/f_{\text{moy}} - \text{partie entière de } [f_j/f_{\text{moy}}] \quad (3.8)$$

5.1.3. La sélection par tournoi stochastique

Dans cette méthode de sélection, les probabilités de sélection sont calculées normalement et des paires successives d'individus sont tirées au sort grâce à la sélection par roue de loterie.

Après avoir tiré ces paires de chaînes, la chaîne ayant l'adaptation la plus élevée est déclarée vainqueur, elle est ajoutée à la nouvelle population, et une nouvelle paire est tirée.

Ce processus continue jusqu'à ce que la population soit remplie.

5.2. Le croisement

Après la sélection, une opération de croisement peut avoir lieu en deux étapes. Premièrement, les éléments nouvellement produits par la sélection sont appariés.

Deuxièmement, chaque paire de chaînes subit un croisement comme suit : un entier k représentant une position sur la chaîne est choisi aléatoirement entre 1 et la longueur de la chaîne moins 1 (entre 1 et $L-1$).

Deux nouvelles chaînes sont créées en échangeant tous les caractères compris entre les positions $k+1$ et L incluse.

Par exemple, considérons les chaînes parent_1 et parent_2 de la population initiale :

parent_1 : 1 0 1 1 1 0 1

parent_2 : 1 0 0 0 0 1 0

Supposons qu'en choisissant au hasard un nombre entre 1 et 7, nous obtenions $k = 3$. Le croisement conduit à la production de deux nouvelles chaînes enfant_1 et enfant_2 comme le montre la figure 3.2.

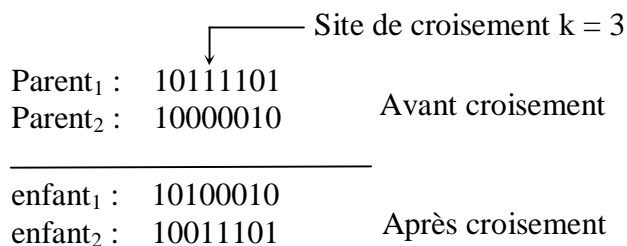


Fig 3.2 Croisement simple.

5.3. La mutation

Dans un algorithme génétique de base, la mutation est la modification aléatoire occasionnelle (de faible probabilité) de la valeur d'un caractère de la chaîne, elle est considérée comme une police d'assurance protégeant de la perte prématurée de notions importantes [16].

Par exemple, considérons la chaîne A de la population initiale :

A : 10111111,

Supposons qu'en choisissant au hasard un nombre entre 1 et L-1, nous obtenions k = 4, la mutation conduit à la production de la nouvelle chaîne A' (figure 3.3).

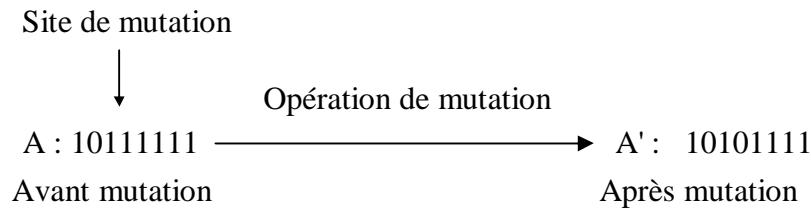


Fig 3.3 Opération de mutation.

Du rôle secondaire de l'opérateur de mutation dans les AGs, la fréquence de mutation conseillée pour obtenir de bons résultats se situe autour d'une mutation tous les 1000 bits (positions). [18] préconise plutôt la valeur suivante pour p_m :

$$p_m = \frac{1}{L} \quad (3.9)$$

Où L est la longueur du chromosome.

D'autres études ont conduit à une formule empirique qui exprime le taux optimal de mutation en fonction de la longueur du chromosome L et de la taille de la population N sous la forme suivante :

$$p_m = \frac{1}{N \cdot \sqrt{L}} \quad (3.10)$$

6. Notion du codage

L'utilisation des AGs nécessite, en première étape, le codage de l'ensemble des paramètres à optimiser en une chaîne de caractères de longueur finie, cette longueur est fixée par l'utilisateur selon :

- le domaine de variation de la chaîne,
- la précision demandée.

Les possibilités de codages sont nombreuses, on peut citer, entre autres, le codage binaire, le codage gray et le codage en virgule flottante si les paramètres à optimiser ont des valeurs réelles. Bien que ces codages aient donné une certaine flexibilité, ils ne constituent pas une méthode pratique pour le codage des fonctions multi-variables $f(x_1, x_2, \dots, x_n)$. L'une des méthodes de codage proposées, qui a été appliquée avec succès, est le codage concaténé et multi-paramétré qui se déroule de la façon suivante :

- Coder chaque variable x_i ($i = 1, 2, \dots, n$) selon le choix de sa longueur.

- Construire les chaînes en concaténant les différents codes, c'est à dire, juxtaposer ces derniers les uns après les autres pour construire les chaînes (chromosomes).

7. Notion du décodage

C'est l'opération inverse de la précédente, chaque chaîne doit être décodée pour pouvoir calculer la valeur de la fonction d'aptitude qui lui est associée.

Parmi les types de décodage existant, le décodage binaire est le plus utilisé :

$$X = U_{\min} + D(100\dots 1_B) \cdot \frac{U_{\max} - U_{\min}}{2^L - 1} \quad (3.11)$$

Avec U_{\max} , U_{\min} sont les bornes supérieure et inférieure du domaine de variation de la variable X , $D(100\dots 1_B)$ est la valeur décimale convertie du binaire et L est la longueur de la chaîne codant la variable X .

Le choix de la longueur dépend essentiellement de la précision voulue sur la variable X qui est donnée par :

$$\text{Précision} = \frac{U_{\max} - U_{\min}}{2^L - 1} \quad (3.12)$$

8. Amélioration des techniques de base des AGs

Les algorithmes génétiques ont démontré leur efficacité pour la résolution d'une large classe de problèmes difficiles et pour la recherche dans des espaces complexes, leur efficacité et leur performance peuvent être renforcées et leur utilisation peut être étendue à de nouvelles classes de problèmes en introduisant des modifications et des extensions au niveau de leurs stratégies et opérateurs génétiques de base[16].

Les efforts de recherche consacrés dans cette direction peuvent être résumés dans les points suivants:

8.1. Amélioration des opérateurs de reproduction

8.1.1. La sélection

Plusieurs stratégies sont possibles :

- a. La sélection par descendance
- b. La sélection par compétition
- c. Sélection en régime permanent
- d. Sélection par procréation sélective

8.1.2. L'opérateur de croisement

a. Opérateur de croisement en deux points

Le principe de cette opération, comme on peut le voir sur la figure (3.4), est le même que celui en un seul point, seulement au lieu de choisir un seul point de croisement, on doit choisir deux points et on échange l'ensemble de caractères existants entre ces deux points.

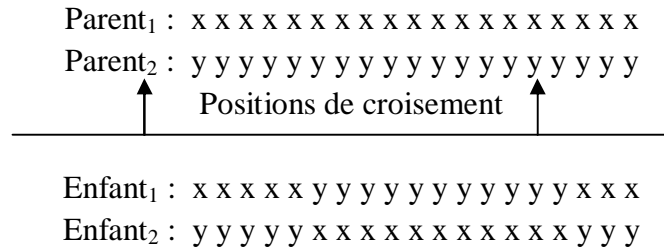


Fig 3.4 Opérateur de croisement en deux points.

b. Opérateur de croisement aléatoire

Cet opérateur combine deux chromosomes selon une chaîne binaire aléatoire, à chaque position de cette chaîne aléatoire, les caractères correspondants sont échangés si et seulement si la position contient un "1".

Aucun changement ne se produit si la position contient un "0" (Figure 3.5).

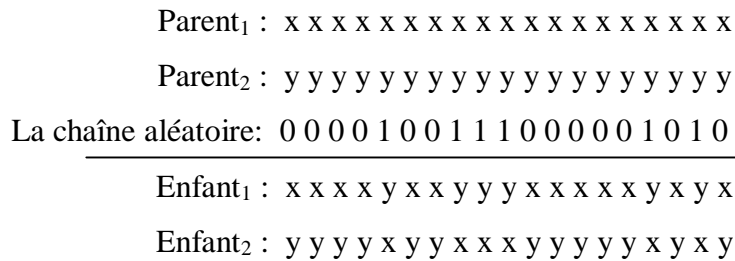


Fig 3.5 Opérateur de croisement aléatoire.

8.2 Optimisation des paramètres de l'algorithme génétique

Les principaux paramètres d'un algorithme génétique sont [17] :

La probabilité de mutation p_m , la probabilité de croisement p_c et la taille de la population N , le choix de ces paramètres joue un rôle très important dans l'emploi des AGs comme procédure d'optimisation.

Si par exemple, la taille de la population est choisie petite, cela implique une exécution rapide mais une convergence incertaine, si la taille est importante, le problème du temps de calcul apparaît.

La sélection des meilleurs individus pour la reproduction et la diversité dans la population sont deux facteurs très importants pour le déroulement de l'AG dans un sens positif, ces facteurs sont liés directement à la taille de la population et aux valeurs de p_m et p_c .

L'idée d'autorégulation des paramètres de l'AG a été développée par plusieurs chercheurs, cette idée peut être schématisée comme suit :

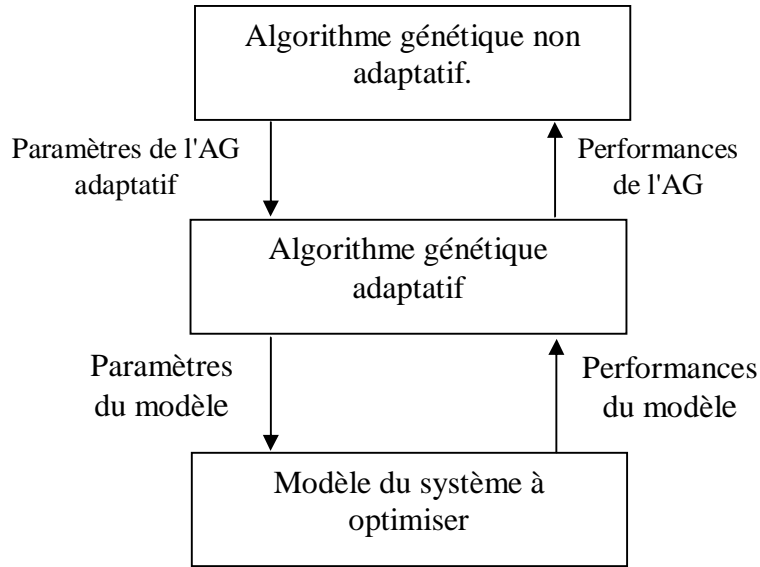


Fig 3.6 *Algorithme génétique adaptatif.*

De nombreuses études ont été faites sur l'effet d'adaptation des paramètres de l'AG sur son déroulement [18], le résultat de ces études est la formulation d'une relation permettant d'avoir des paramètres optimaux de l'AG, cette relation est donnée par :

$$\ln(N) + 0.93 \ln(p_m) + 0.456 \ln(L) = 0.56 \quad (3.26)$$

9. Conclusion

Les algorithmes génétiques sont des outils d'optimisation très puissants, ils explorent l'espace de recherche par une population et non pas par un point unique, ce qui réduit la probabilité d'atteindre un faux pic.

Les algorithmes génétiques réalisent une optimisation efficace en ne tenant compte que des valeurs de la fonction à optimiser.

Les autres méthodes utilisent beaucoup d'informations auxiliaires, si bien que dans les problèmes pour lesquels cette information n'est pas disponible ou difficile à obtenir, ces techniques sont bloquées.

Les algorithmes génétiques restent généraux en utilisant que l'information disponible dans tous les problèmes d'optimisation.

Les algorithmes génétiques traitent simultanément les similarités dans le codage sous-jacent et l'information qui permet d'ordonner les structures en d'information si généralement disponible, les AGs peuvent être appliqués virtuellement à tous les problèmes.

1. Introduction

Contrôler un système complexe, fortement non-linéaire ou difficile à modéliser présente une tâche très délicate, ainsi les performances désirées deviennent de plus en plus sévères, c'est pour ces raisons qu'apparaissent de nouvelles méthodes de contrôle plus sophistiquées. L'apparition du formalisme de la logique floue a donné naissance à une approche très efficace où les lois de commande conventionnelles sont remplacées par une série de règles linguistiques déterminant la stratégie et l'objectif de contrôle.

Cette approche exige la disponibilité de l'expertise humaine ce qui influence négativement sur les performances des systèmes flous conçus.

Ce problème a conduit au développement des systèmes hybrides permettant de réunir les capacités d'optimisation des algorithmes génétiques et celles de décision des systèmes flous.

L'utilisation conjointe de ces approches permet de réaliser un système de contrôle flou très efficace et robuste bien adapté à la complexité des systèmes réels (Robot manipulateur).

2. Application de la commande floue

Nous avons appliqué une commande floue dotée d'un régulateur de type Mamdani à un robot à deux degrés de liberté.

Les simulations ont pour objectif l'évaluation des erreurs de poursuite du robot, dans un environnement de performance décrit par la trajectoire.

Les graphes de la figure (4.1) représentent les positions des deux articulations du robot.

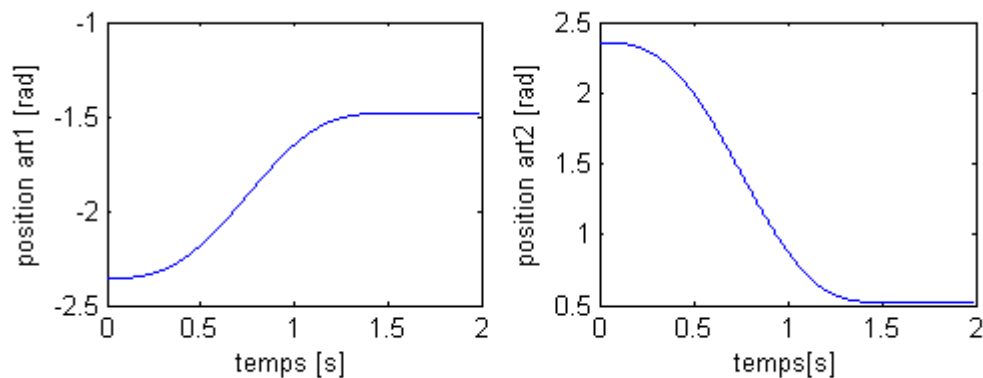


Fig 4.1 Positions des deux articulations.

2.1. Régulateur flou à trois classes

Dans cette étape nous avons utilisé un régulateur flou type PD qui contient trois fonctions d'appartenance pour l'erreur et la variation de l'erreur.

La base des règles :

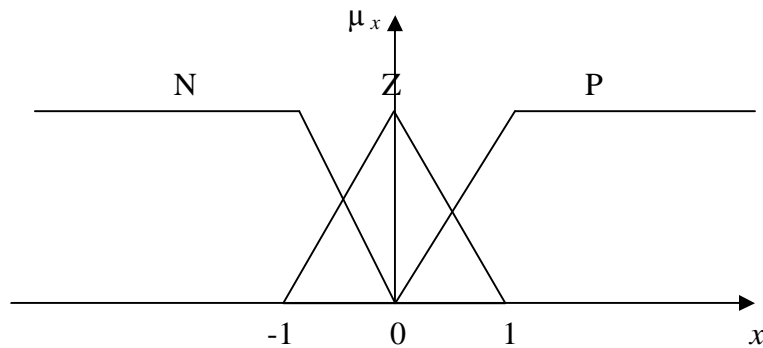
(N: négative, Z: zéro, P: positive).

Tableau 4.1 Base des règles, RLF (3×3).

ΔE \ E	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

Fonctions d'appartenance :

La forme des fonctions d'appartenance associée à l'erreur, la variation de l'erreur, et la commande, pour les deux régulateurs est donnée par la figure (4.2).

**Fig 4.2** Fonctions d'appartenance, RLF à 3 classes.

2.1.1 Avant l'utilisation des AGs

-Test à vide

Nous allons faire un test à vide pour observer l'état de l'erreur de position pour les deux articulations du robot.

Les graphes de la figure (4.3) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à vide.

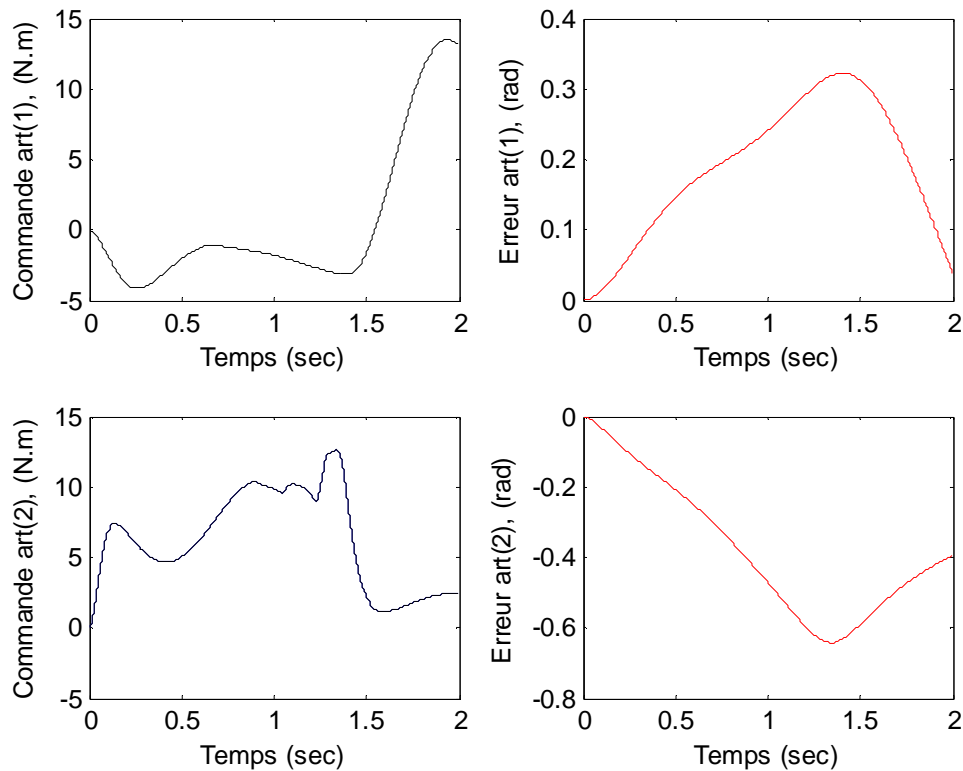


Fig 4.3 Poursuite à vide de la référence, RLF à 3 classes

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une très grande erreur maximale
- la commande est loin d'être acceptable.

-Test de la charge

Dans ce cas le FLC a été testé pour un changement de la charge portée par le robot. Ceci par l'introduction d'une charge, de 5 kg.

Les graphes de la figure (4.4) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à une charge de 5kg.

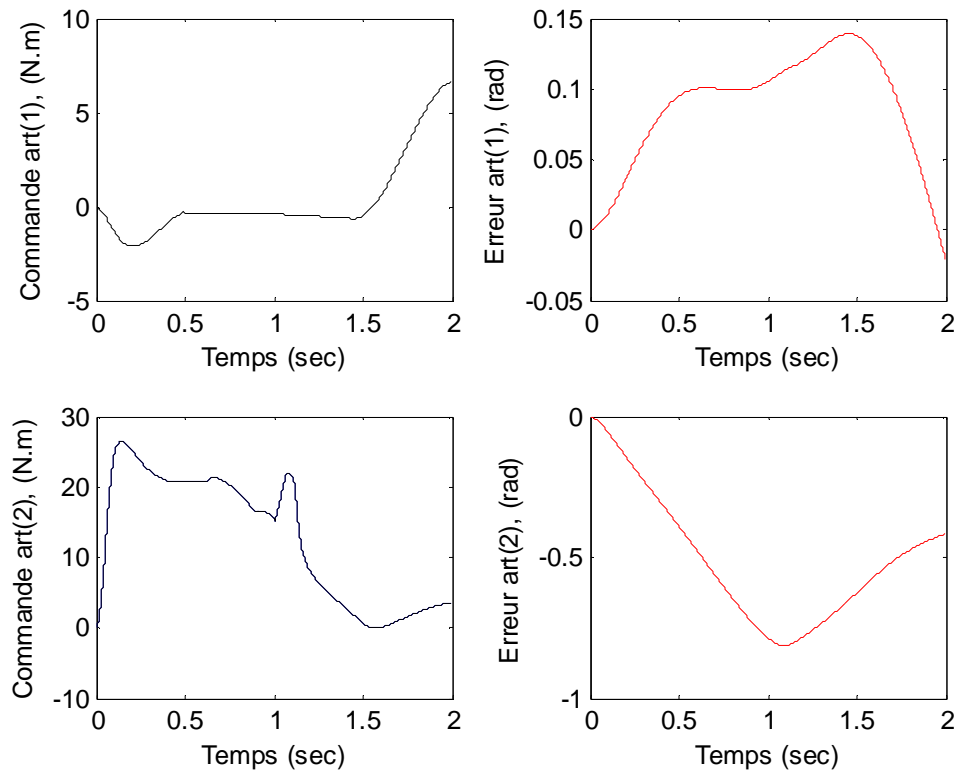


Fig 4.4 Poursuite de la référence pour un test de charge de 5 Kg, RLF (3×3).

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une très grande erreur maximale
- la commande est loin d'être acceptable.

2.1.2 Après l'utilisation des AGs

- Test à vide

Nous allons faire un test à vide pour observer l'état de l'erreur de position pour les deux articulations du robot.

Les graphes de la figure (4.5) représentent les deux gains d'optimisation de l'erreur.

$G1=75005.1406$

$G2=3800.2159$

Les graphes de la figure (4.6) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à vide.

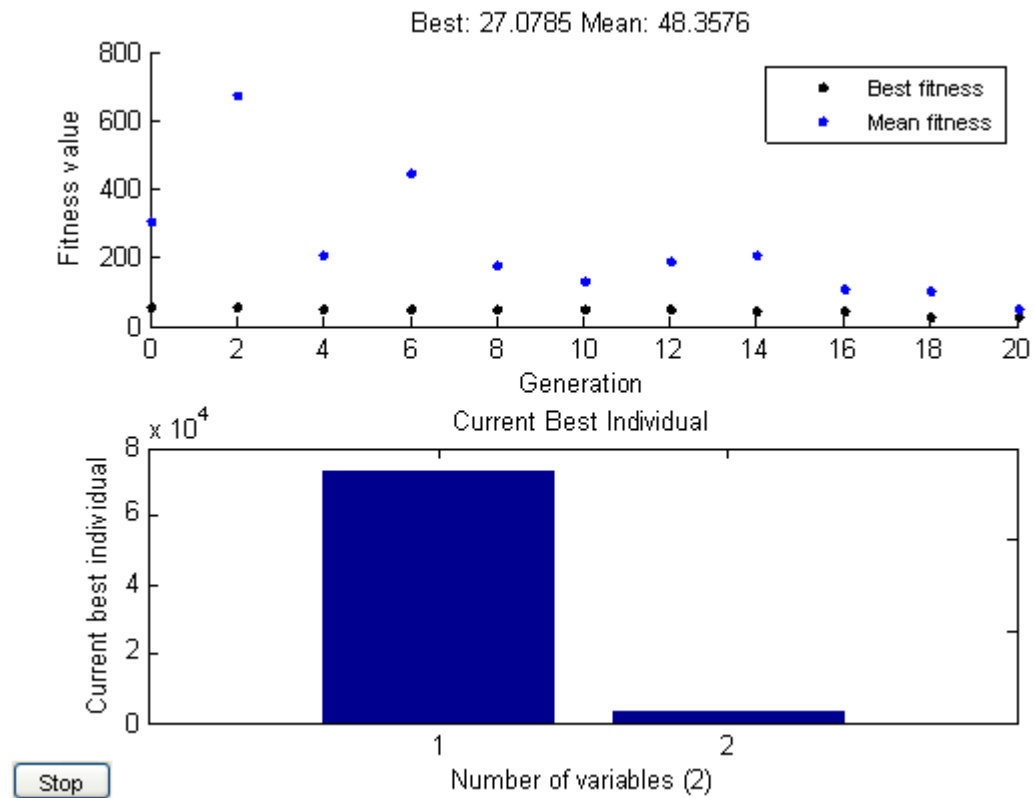


Fig 4.5 Fonction objective pour optimisation des gains de l'erreur.

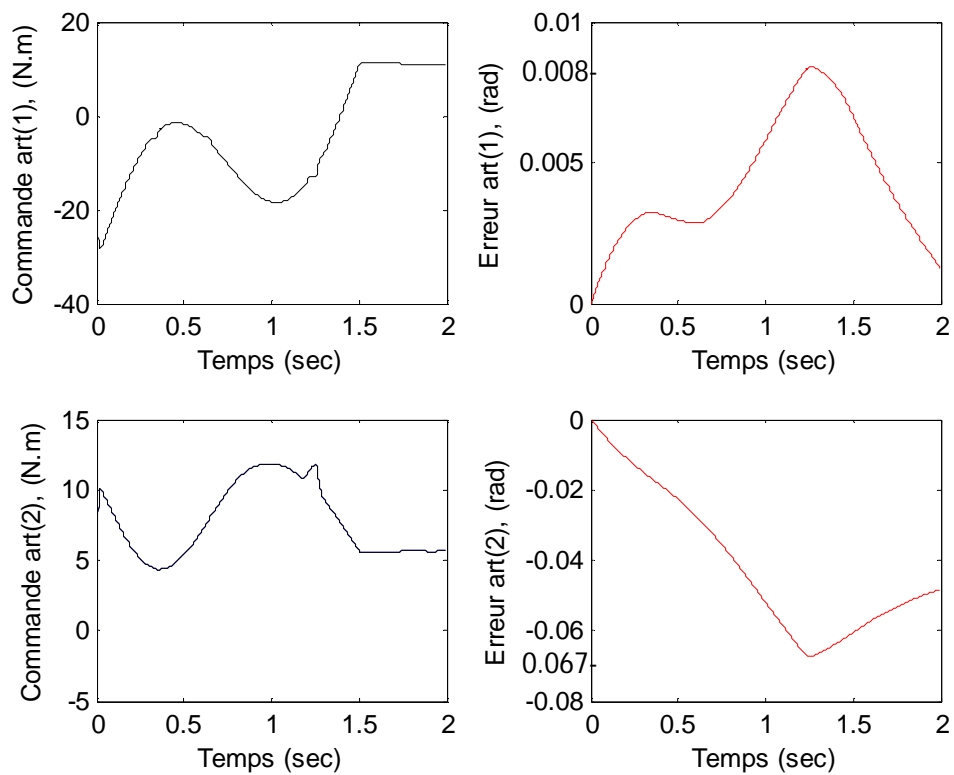


Fig 4.6 Poursuite à vide de la référence, RLF à 3 classes.

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une erreur maximale acceptable.
- la commande est lisse sauf une rupture à 1.2sec.

-Test de la charge

Dans ce cas le FLC a été testé pour un changement de la charge portée par le robot. Ceci par l'introduction d'une charge, de 5 kg.

Les graphes de la figure (4.7) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à une charge de 5kg.

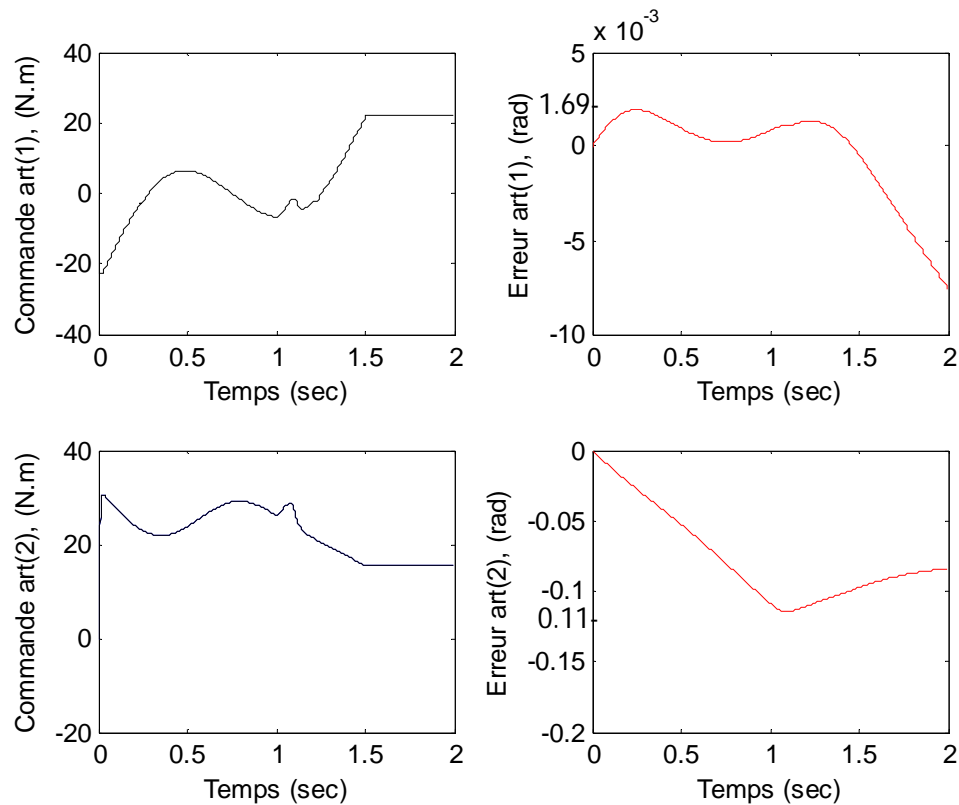


Fig 4.7 Poursuite de la référence pour un test de charge de 5 Kg, RLF (3×3).

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une erreur maximale acceptable.
- la commande est lisse sauf une rupture à 1.1sec.

2.2. Régulateur flou à 5 classes

Dans ce cas nous avons utilisé 5 fonctions d'appartenance, avec 25 règles dans la base de connaissance.

Les formes de la fonction d'appartenance et des gains de normalisation sont les mêmes que celles du régulateur à 3 classes.

La base des règles:

Tableau 4.2 Base des règles, RLF (5×5).

ΔE E	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NB	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	NE	PS	PB	PB
PB	ZE	PS	PB	PB	PB

Fonctions d'appartenance:

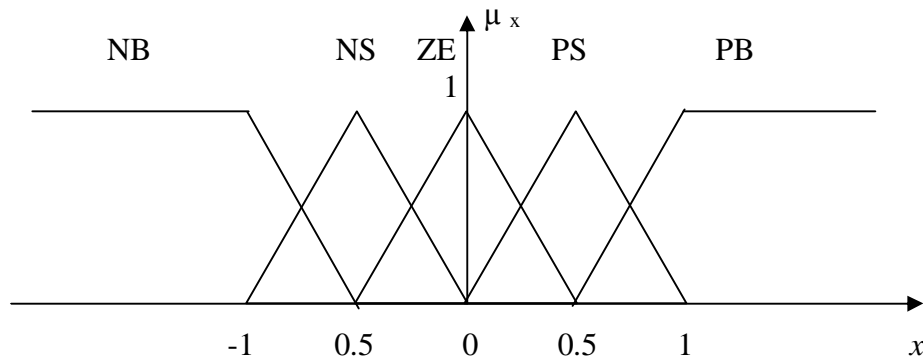


Fig 4.8 Fonctions d'appartenances, RLF à 5 classes.

Pour pouvoir comparer les performances des commandes synthétisées, nous avons fait les mêmes tests de performance pour le régulateur flou à trois classes.

2.2.1 Avant l'utilisation des AGs

-Test à vide

Nous allons faire un test à vide pour observer l'état de l'erreur de position pour les deux articulations du robot.

Les graphes de la figure (4.9) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à vide.

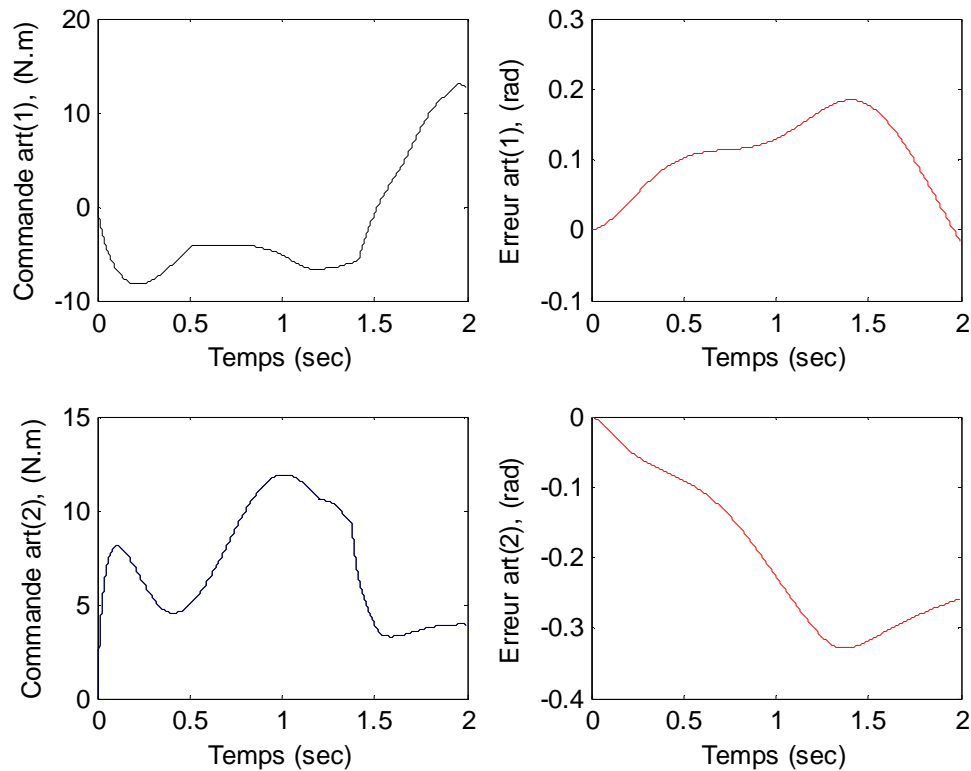


Fig 4.9 Poursuite à vide de la référence, RLF à 5 classes.

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une très grande erreur maximale
- la commande est loin d'être acceptable.

-Test de la charge

Dans ce cas le FLC a été testé pour un changement de la charge portée par le robot. Ceci par l'introduction d'une charge, de 5 kg.

Les graphes de la figure (4.10) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à une charge de 5kg.

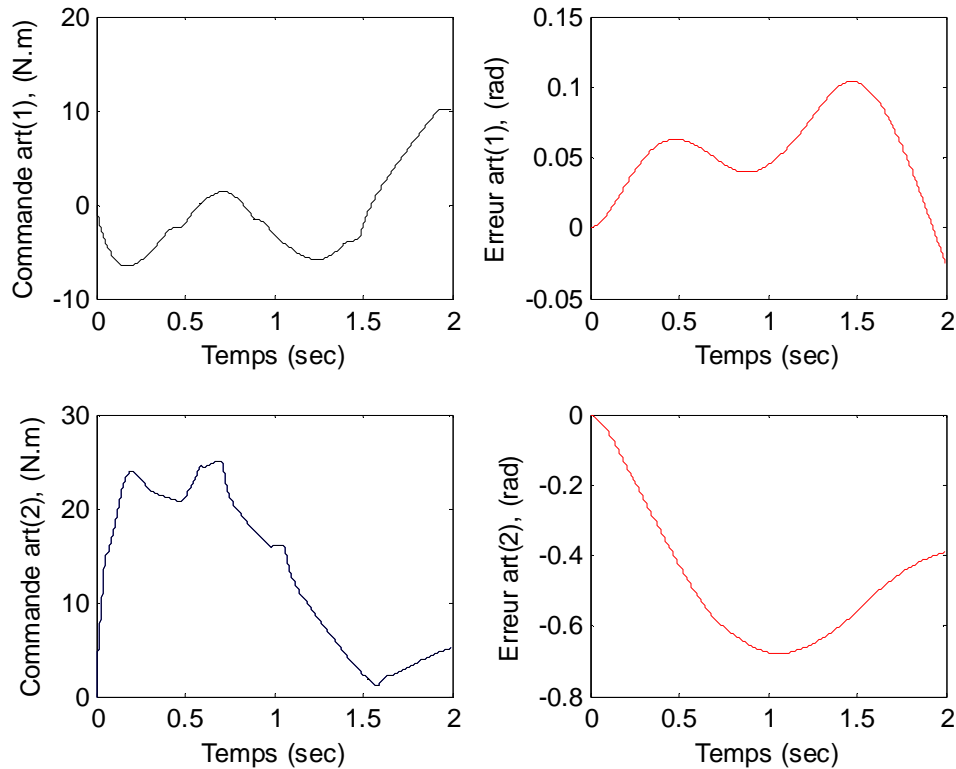


Fig 4.10 Poursuite de la référence pour un test de charge de 5 Kg, RLF (5×5).

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une très grande erreur maximale
- la commande est loin d'être acceptable.

2.2.2 Après l'utilisation des AGs

-Test à vide

Nous allons faire un test à vide pour observer l'état de l'erreur de position pour les deux articulations du robot.

Les graphes de la figure (4.11) représentent les deux gains d'optimisation de l'erreur.

G1=1702.1806

G2=1610.3104

Les graphes de la figure (4.12) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à vide.

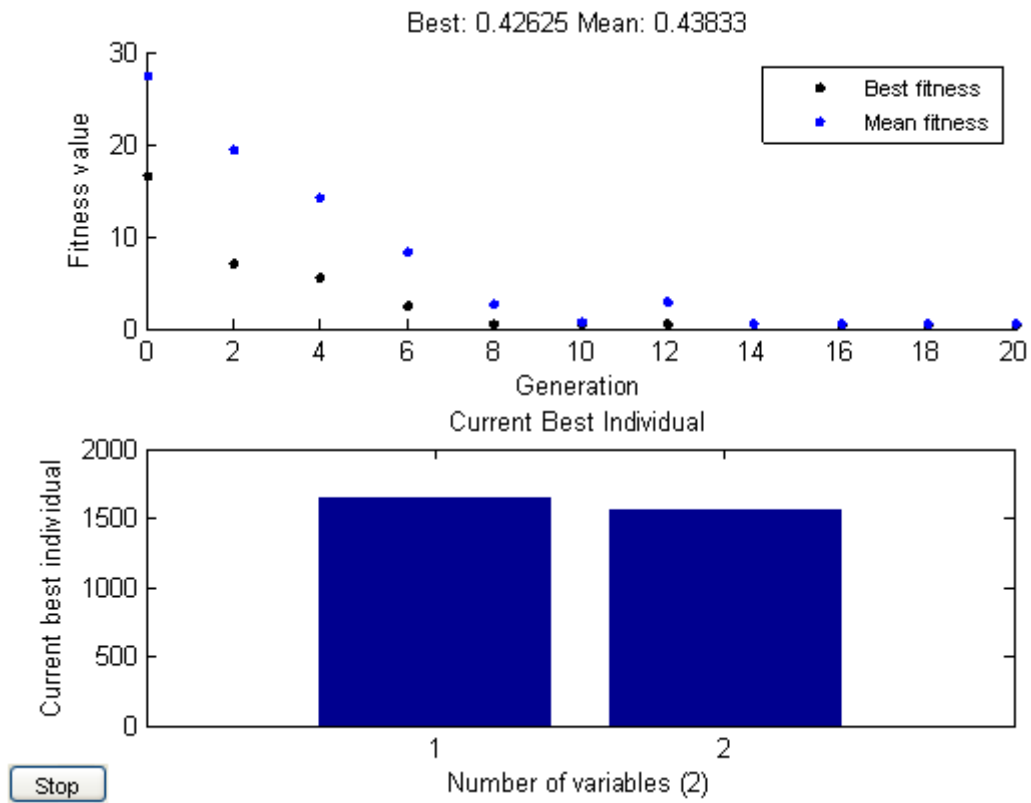


Fig 4.11 Fonction objective pour optimisation des gains de l'erreur.

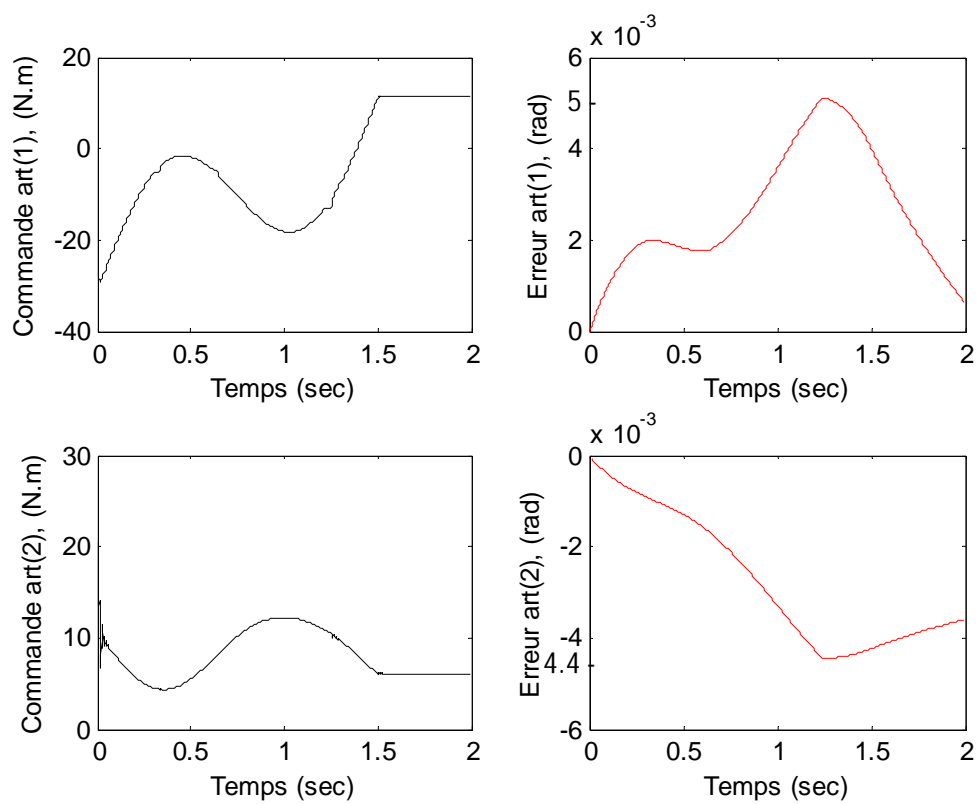


Fig 4.12 Poursuite à vide de la référence, RLF à 5 classes.

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une erreur maximale acceptable.
- la commande est presque tout lisse.

-Test de la charge

Dans ce cas le FLC a été testé comme le FLC trois classes, la charge portée par le robot est 5 kg.

Les graphes de la figure (4.13) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à une charge de 5kg.

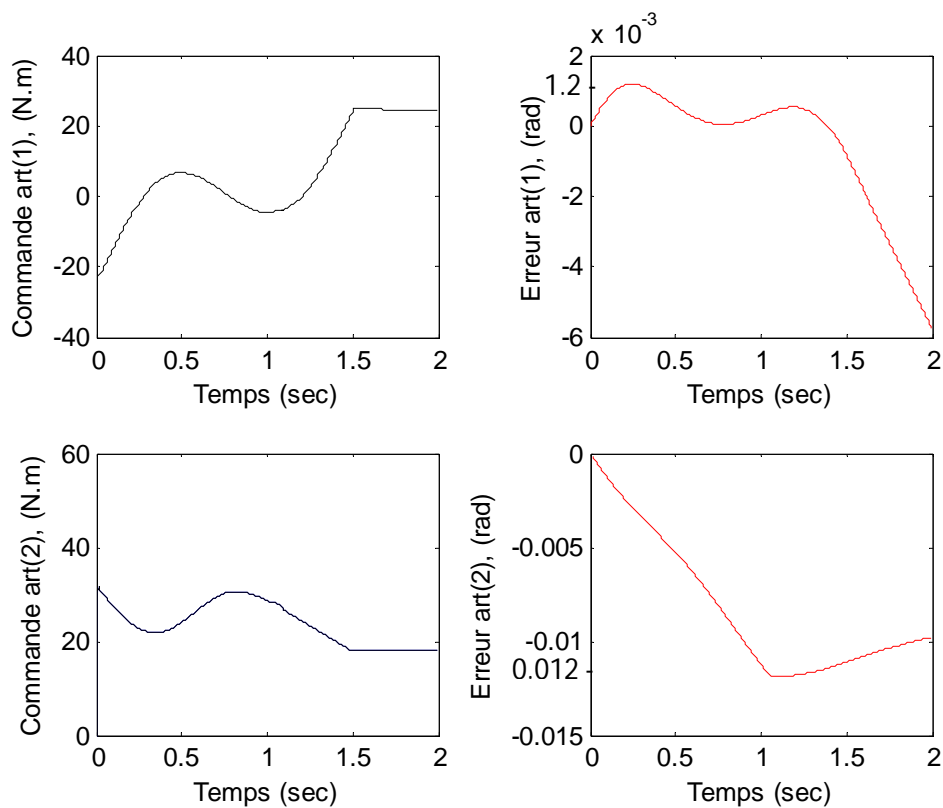


Fig 4.13 Poursuite de la référence pour un test de charge de 5 Kg, RLF (5×5).

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une bonne erreur maximale
- la commande est lisse.

2.3. Régulateur flou à 7 classes

Dans ce cas nous avons utilisé 7 fonctions d'appartenance, avec 49 règles.

Les formes de la fonction d'appartenance et des gains de normalisation sont les mêmes que celles du régulateur à 3 et 5 classes.

La base des règles:

Tableau 4.3 Base des règles, RLF (7×7).

ΔE \ E	NG	NM	NP	ZE	PP	PM	PG
NG	NG	NG	NG	NG	NM	NP	ZE
NM	NG	NG	NG	NM	NP	ZE	PP
NP	NG	NG	NM	NP	ZE	PP	PM
ZE	NG	NM	NP	ZE	PP	PM	PG
PP	NM	NP	ZE	PP	PM	PG	PG
PM	NP	ZE	PP	PM	PG	PG	PG
PG	ZE	PP	PM	PG	PG	PG	PG

Fonctions d'appartenance:

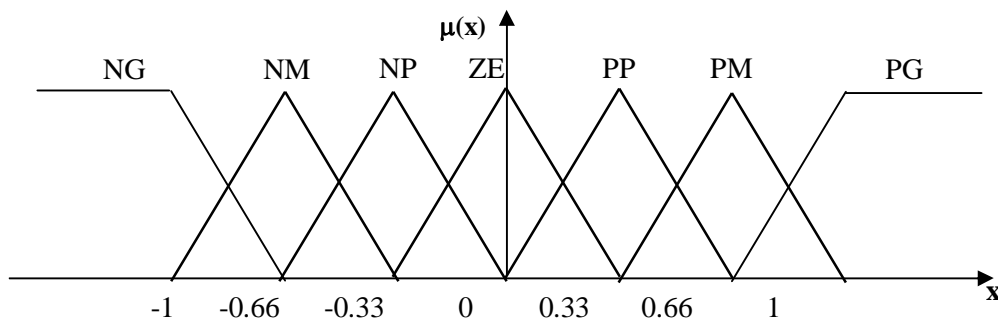


Fig 4.14 Fonctions d'appartenances, RLF à 7 classes.

Pour pouvoir comparer les performances des commandes synthétisées, nous avons fait les mêmes tests de performance pour le régulateur flou à 3 et 5 classes.

2.3.1 Avant l'utilisation des AGs

-Test à vide

Nous allons faire un test à vide pour observer l'état de l'erreur de position pour les deux articulations du robot.

Les graphes de la figure (4.15) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à vide.

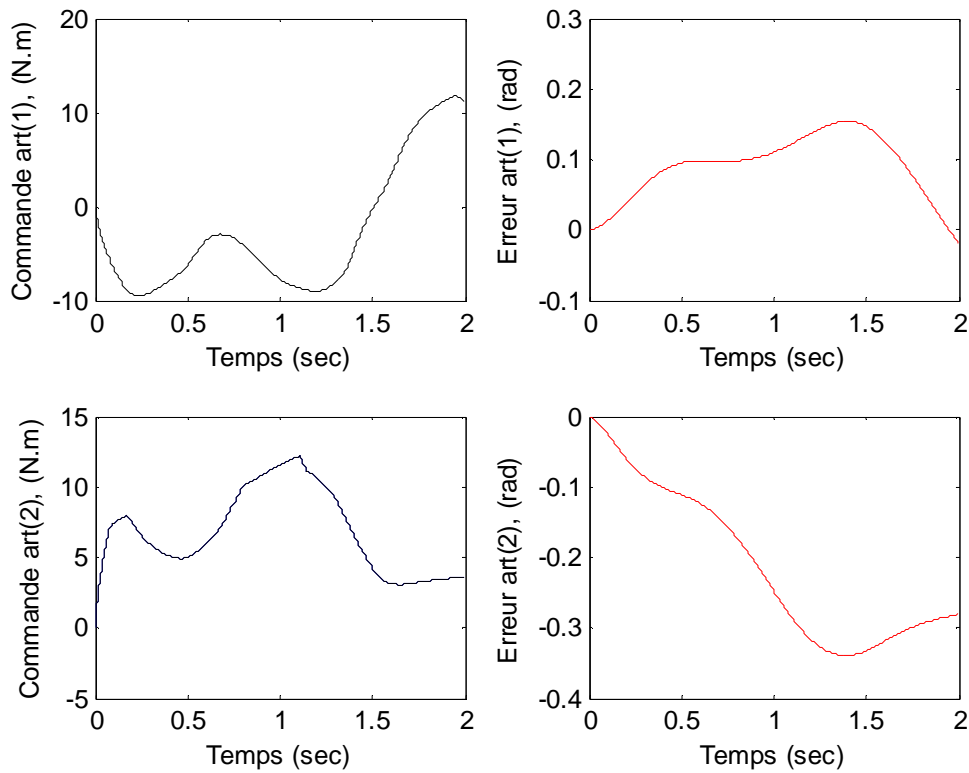


Fig 4.15 Poursuite à vide de la référence, RLF à 7 classes.

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une très grande erreur maximale
- la commande est loin d'être acceptable.

-Test de la charge

Dans ce cas le FLC a été testée pour un changement de la charge portée par le robot. Ceci par l'introduction d'une charge, de 5 kg.

Les graphes de la figure (4.16) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à une charge de 5kg.

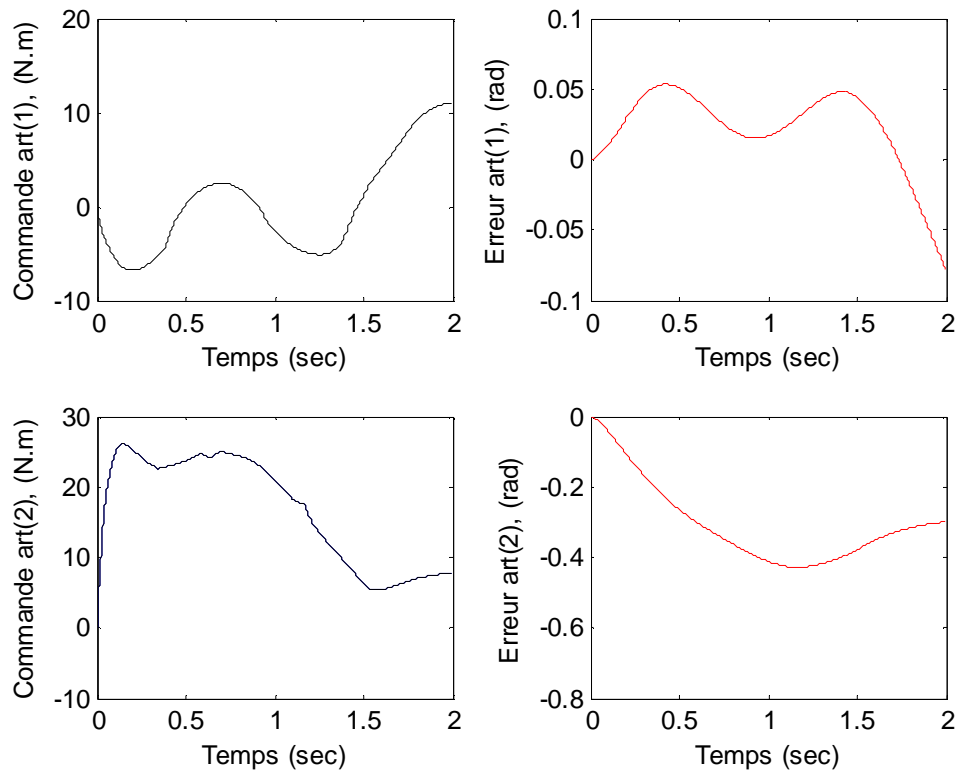


Fig 4.16 Poursuite de la référence pour un test de charge de 5 Kg, RLF (7×7).

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une très grande erreur maximale
- la commande est loin d'être acceptable.

2.3.2 Après l'utilisation des AGs

-Test à vide

Nous allons faire un test à vide pour observer l'état de l'erreur de position pour les deux articulations du robot.

Les graphes de la figure (4.17) représentent les deux gains d'optimisation de l'erreur.

$G1=1600.1409$

$G2=1207.1104$

Les graphes de la figure (4.18) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à vide.

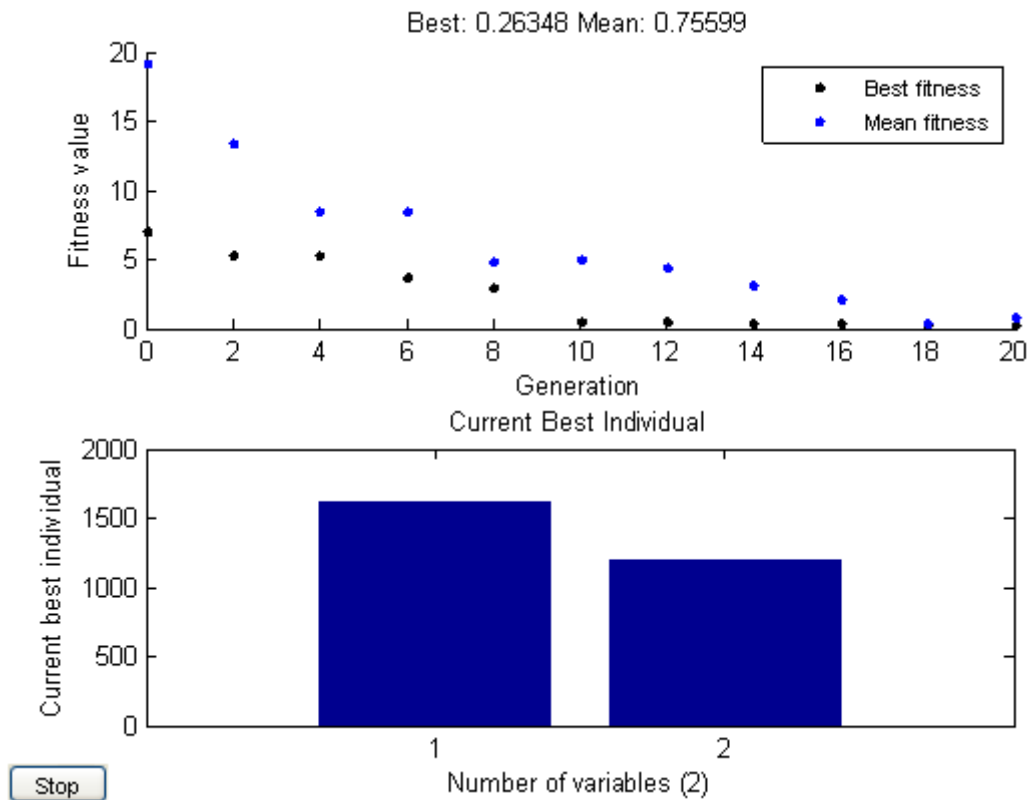


Fig 4.17 Fonction objective pour optimisation des gains de l'erreur.

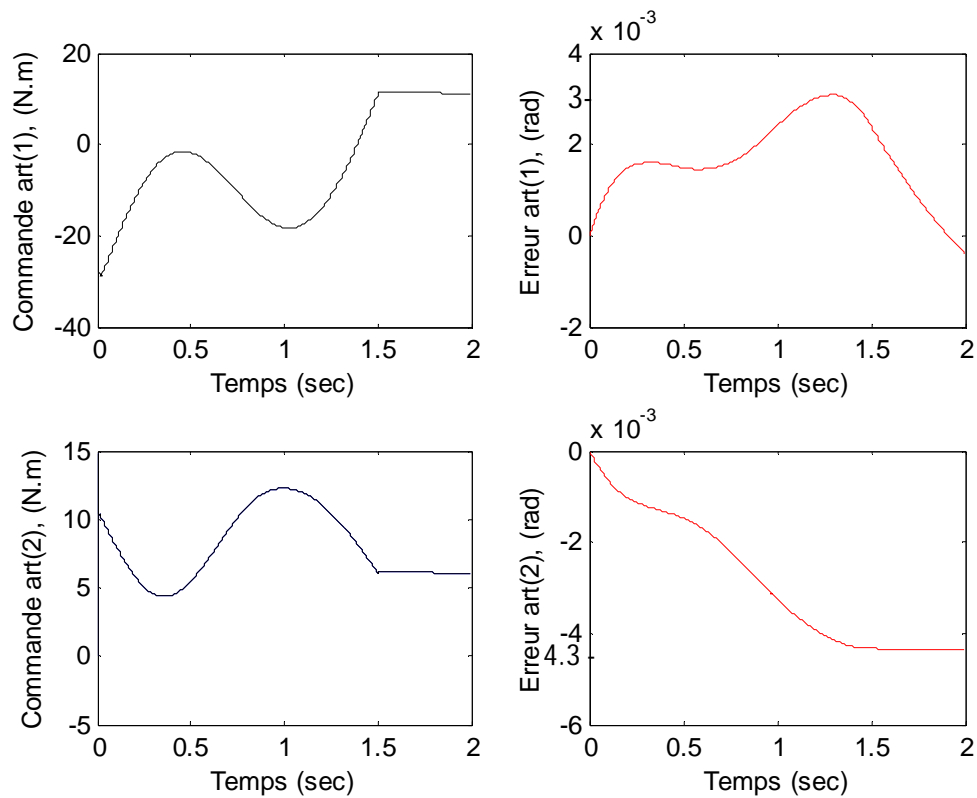


Fig 4.18 Poursuite à vide de la référence, RLF à 7 classes.

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une erreur maximale très bonne.
- la commande est lisse.

-Test de la charge

Dans ce cas le FLC a été testée comme le FLC 3 et 5 classes, la charge portée par le robot est 5 kg.

Les graphes de la figure (4.19) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à une charge de 5kg.

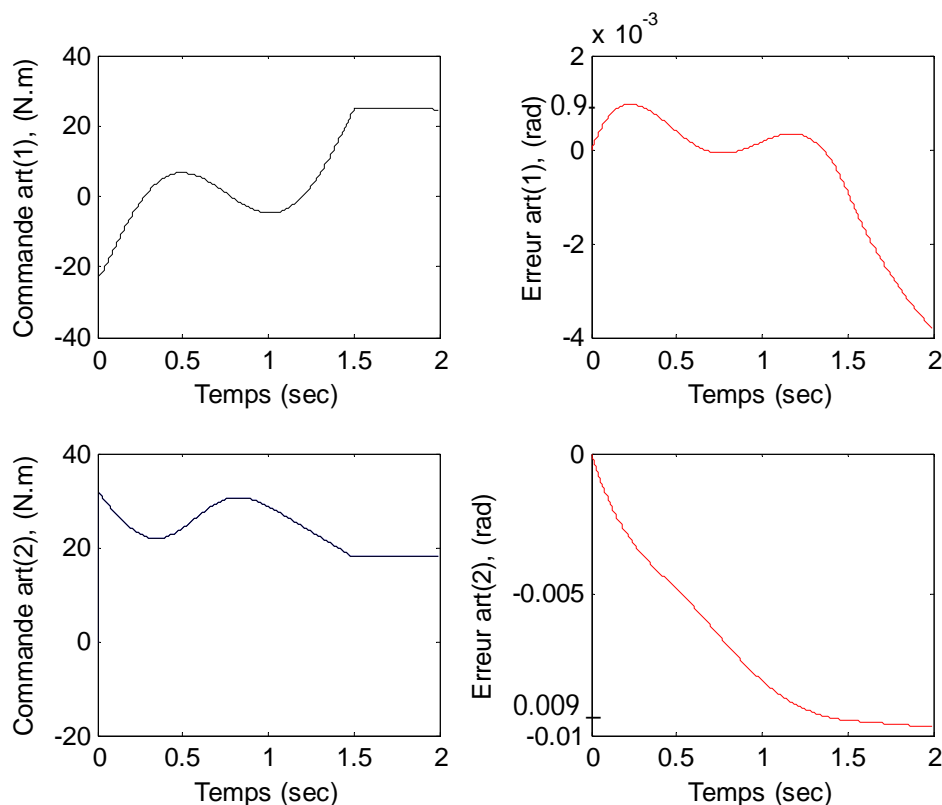


Fig 4.19 Poursuite de la référence pour un test de charge de 5 Kg, RLF (7×7).

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une erreur maximale très bonne.
- la commande est lisse.

2.4. Régulateur flou à 9 classes

Dans ce cas nous avons utilisé 9 fonctions d'appartenance, avec 81 règles dans la base de connaissance.

Les formes de la fonction d'appartenance et des gains de normalisation sont les mêmes que celles du régulateur à 3, 5 et 7 classes.

La base des règles:

Tableau 4.4 Base des règles, RLF (9×9).

ΔE E	NG	NN	NM	NP	ZE	PP	PM	PN	PG
NG	NG	NG	NG	NG	NG	NN	NM	NP	ZE
NN	NG	NG	NG	NG	NN	NM	NP	ZE	PP
NM	NG	NG	NG	NN	NM	NP	ZE	PP	PM
NP	NG	NG	NN	NM	NP	ZE	PP	PM	PN
ZE	NG	NN	NM	NP	ZE	PP	PM	PN	PG
PP	NN	NM	NP	ZE	PP	PM	PN	PG	PG
PM	NM	NP	ZE	PP	PM	PN	PG	PG	PG
PN	NP	ZE	PP	PM	PN	PG	PG	PG	PG
PG	ZE	PP	PM	PN	PG	PG	PG	PG	PG

Fonctions d'appartenance:

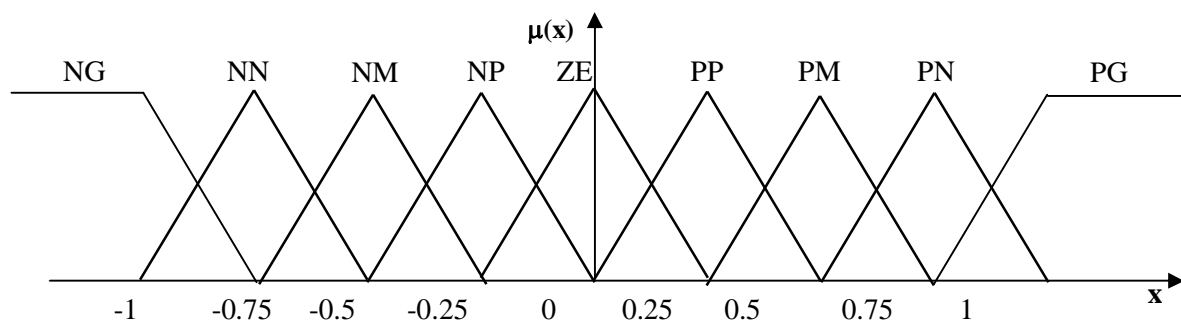


Fig 4.20 Fonctions d'appartenance, RLF à 9 classes.

Pour pouvoir comparer les performances des commandes synthétisées, nous avons fait les mêmes tests de performance pour le régulateur flou à 3, 5 et 7 classes.

2.4.1 Avant l'utilisation des AGs

-Test à vide

Nous allons faire un test à vide pour observer l'état de l'erreur de position pour les deux articulations du robot.

Les graphes de la figure (4.21) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à vide.

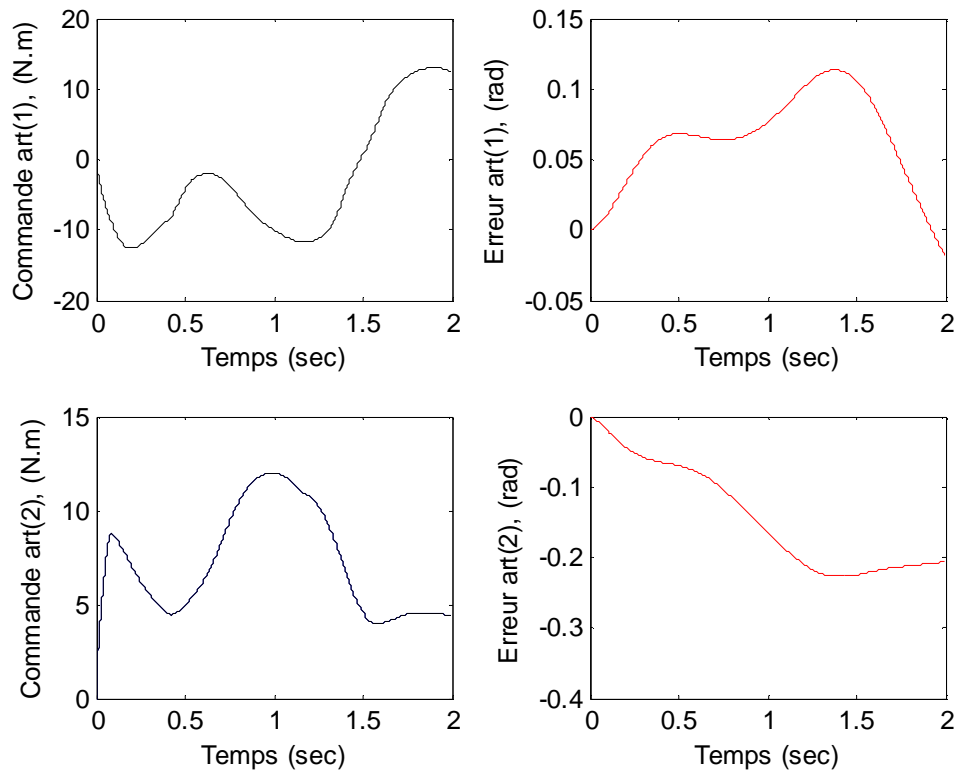


Fig 4.21 Poursuite à vide de la référence, RLF à 9 classes.

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une très grande erreur maximale
- la commande est loin d'être acceptable.

-Test de la charge

Dans ce cas le FLC a été testée pour un changement de la charge portée par le robot. Ceci par l'introduction d'une charge, de 8 kg.

Les graphes de la figure (4.22) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à une charge de 8 kg.

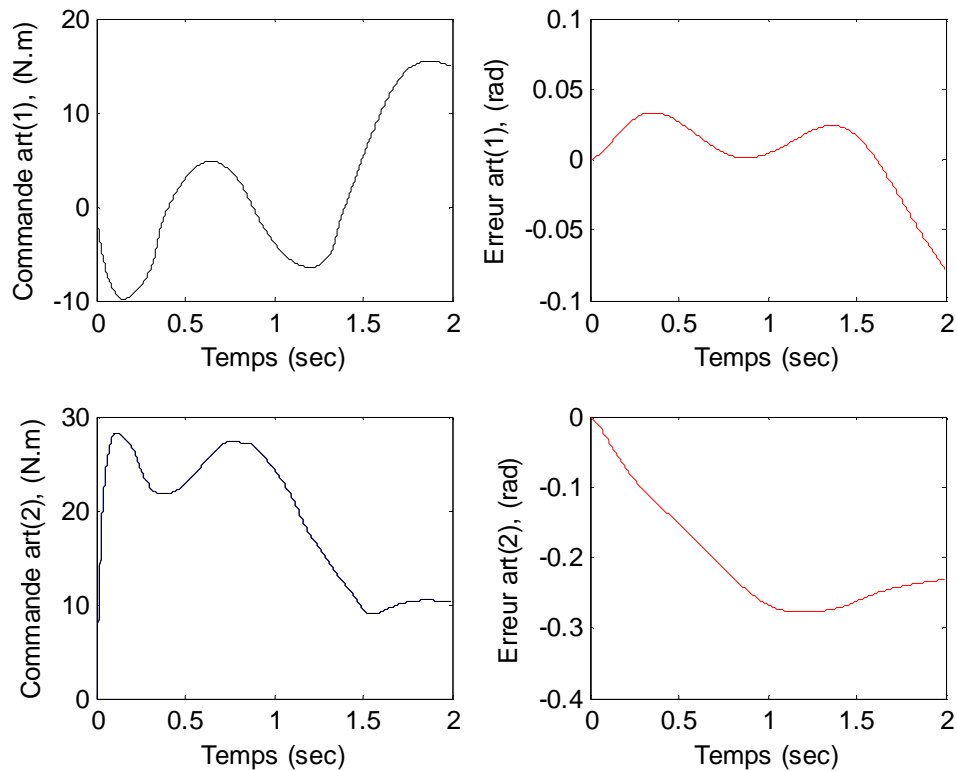


Fig 4.22 Poursuite de la référence pour un test de charge de 8 Kg, RLF (9×9).

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une très grande erreur maximale
- la commande est loin d'être acceptable.

2.4.2 Après l'utilisation des AGs

-Test à vide

Nous allons faire un test à vide pour observer l'état de l'erreur de position pour les deux articulations du robot.

Les graphes de la figure (4.23) représentent les deux gains d'optimisation de l'erreur.

$G1=2400.3601$

$G2=1412.8305$

Les graphes de la figure (4.24) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à vide.

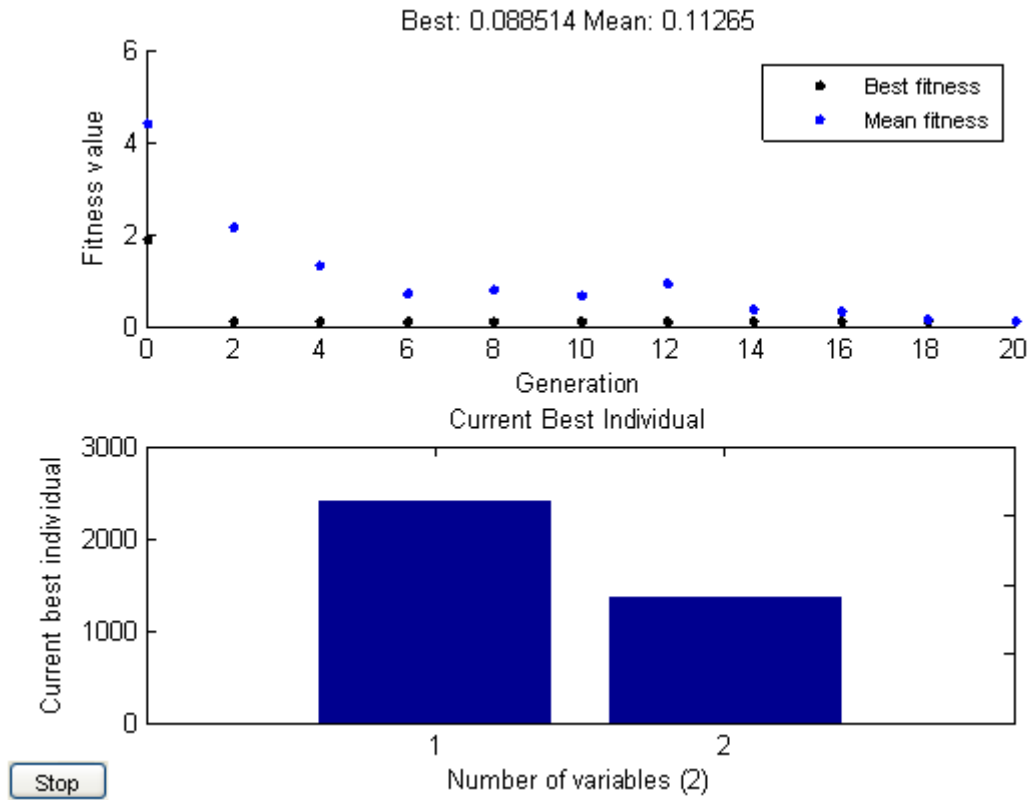


Fig 4.23 Fonction objective pour optimisation des gains de l'erreur.

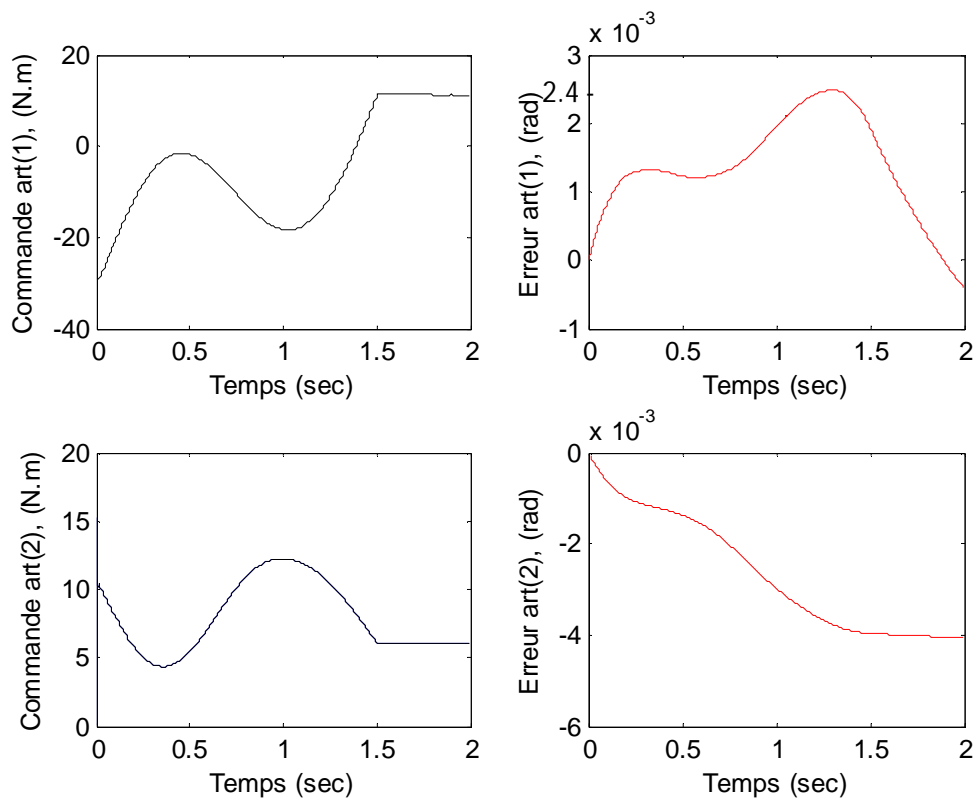


Fig 4.24 Poursuite à vide de la référence, RLF à 9 classes.

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une erreur maximale très excellente.
- une très bonne commande.

-Test de la charge

Dans ce cas le FLC a été testée comme le FLC 3, 5 et 7 classes, la charge portée par le robot est 5 kg et 8 kg.

Les graphes de la figure (4.25) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à une charge de 5kg.

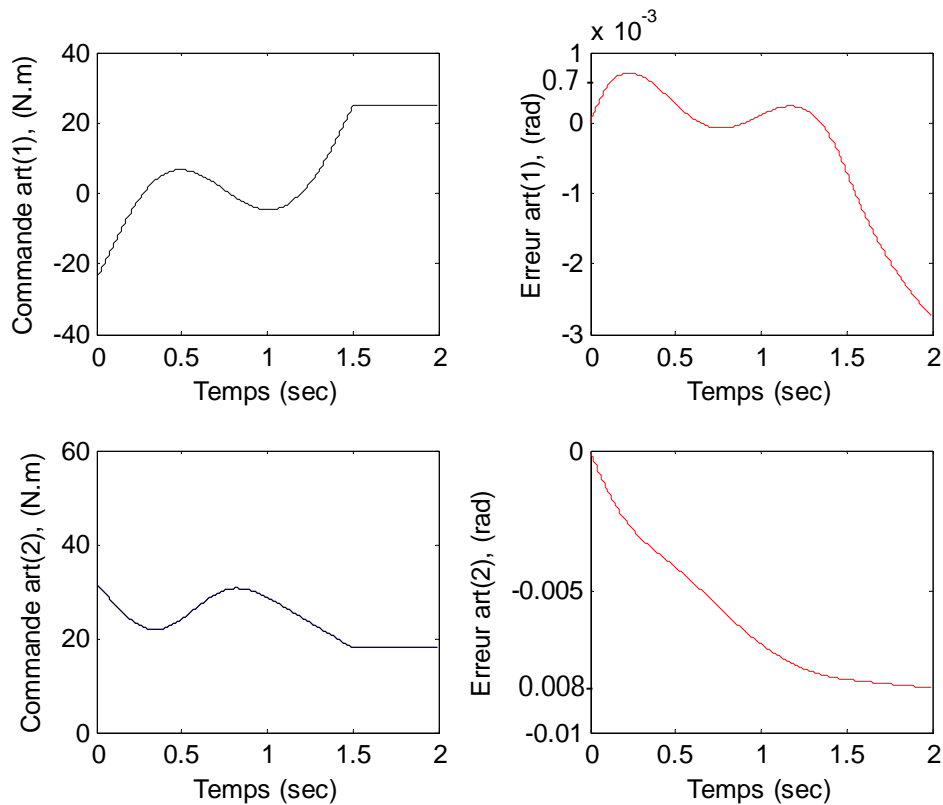


Fig 4.25 Poursuite de la référence pour un test de charge de 5 Kg, RLF (9×9).

Les graphes de la figure (4.26) représentent les deux articulations du robot, lors d'une poursuite de trajectoire à une charge de 8kg.

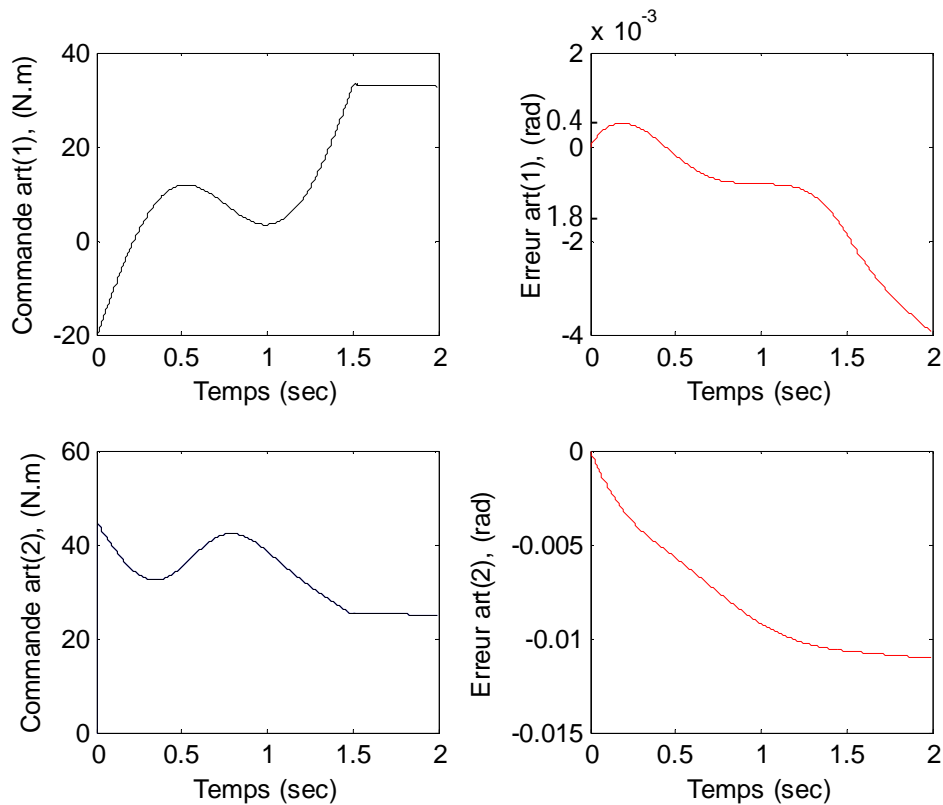


Fig 4.26 Poursuite de la référence pour un test de charge de 8 Kg, RLF (9×9).

D'après les résultats de la simulation obtenus on peut remarquer que:

- les articulations du robot suivent les trajectoires désirées, avec une erreur maximale très excellente.
- une très bonne commande.

3. Conclusion

Dans ce chapitre, un système de contrôle de mouvement des robots manipulateurs a été développé. Ce système est basé sur deux approches originales à savoir, la logique floue et les algorithmes génétiques. L'utilisation du contrôle flou a éliminé le besoin du modèle mathématique exact, et l'utilisation de l'algorithme génétique a éliminé le besoin d'un expert humain dans la phase de conception de tel contrôleur. Le système de contrôle flou ainsi développé a été appliqué à la commande décentralisée d'un bras manipulateur à 2 degrés de liberté.

Les résultats de simulation, envisagés ici, montrent l'efficacité de la technique de conception des contrôleurs flous par Les AGs.

Conclusion générale

L'objectif assigné à ce travail était d'élaborer une commande non linéaire dédiée à la commande de position d'un bras manipulateur.

Nous nous sommes intéressés à une commande non linéaire des robots manipulateurs.

Nous considérons une structure de commande nécessitant la connaissance exacte du modèle dynamique du robot, et nous avons essayé d'améliorer les performances vis-à-vis des incertitudes des paramètres du bras.

Afin d'améliorer les performances vis-à-vis de l'erreur de suivi nous avons proposé d'utiliser la combinaison des contrôleurs flous et des algorithmes génétiques, les résultats obtenus sont assez satisfaisants :

- Pour le problème de la commande de position, les résultats obtenus sont très encourageants.
- L'erreur de suivi de la trajectoire est acceptable.

L'étude envisagée ici, montre l'efficacité de la technique de conception des contrôleurs flous par les algorithmes génétiques pour accomplir un contrôle de poursuite rapide, précis et robuste dans des environnements de travail encombré.

Références Bibliographiques

- [1] :E. Dombre, "Programmation des robots" Techniques de l'ingénieur, Génie électrique.
- [2] :N.Seguy, "Système de commande d'un manipulateur". Techniques de l'ingénieur, Génie électrique.
- [3] :P. Duysinx, "Eléments de Robotique Aspects Mécaniques".
- [4] :E. Dombre, "Analyse et modélisation des robots manipulateurs".
- [5] : Jacques Gangloff. « [//eavr.u-strasbg.fr/thesis/gangloff/node34.html](http://eavr.u-strasbg.fr/thesis/gangloff/node34.html) ».1999.
- [6] : B. Bouchon-Meunier. La logique floue. Collection "Que sais-je?", Presses universitaires de France, 1994.
- [7] : Bernadette Bouchon Meunier, "La logique floue et ses applications", Edition Addison Wesley France, SA, 1995.
- [8] : E. H. Mamdani. Applications of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. IEEE Trans. Computer, C-26(12):1182-1191, 1977.
- [9] : E. H. Mamdani. Advances in the Linguistic Synthesis of Fuzzy controllers. International Journal of Man-Machine Studies, 8(6):669-678, 1976.
- [10] : M. Kantrowitz, E. Horstkotte, and C. Joslyn. Frequently Asked Questions about Fuzzy Logic and Fuzzy Expert Systems. Newsgroup comp.ai.fuzzy on Internet, March 1995.
- [11] : E. H. Mamdani. Applications of fuzzy algorithms for simple dynamic plants.
- [12] : H. R. Berenji. Fuzzy Logic Controller. An Introduction to Fuzzy Logic Applications in Intelligent Systems Kluwer academic publishers, 1990.
- [13] : M-C. Fritsch and E. Wendling. Commande de processus : les atouts de la logique floue. Technologies internationales, Avril 1994.
- [14] : C. Darwin, On the Origin of Species by Means of Natural Selection, John Murray, London, U.K, 1958
- [15] : J.H. Holland, Adaptation in Natural and Artificial Systems, Ann Arbor: The University of Michigan Press, USA, 1975
- [16] : T. Back, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms (Hardcover), Oxford University Press, USA, 1996
- [17] : D. Whitley, Foundations of Genetic Algorithms 2, US Edition, California, USA, 1993
- [18] : V. B. Gantovnik, An Improved Genetic Algorithm for the Optimization of Composite Structures, PhD thesis, PIS University Virginia, Blacksburg, Virginia, USA, 1999.

Résumé

Dans cette étude, un système de planification et de contrôle de mouvement des robots manipulateurs est développé en utilisant les algorithmes génétiques et la logique floue, le système proposé est destiné à la planification de mouvement des robots manipulateurs, il utilise un contrôleur flou optimisé par un algorithme génétique, il est chargé de contrôler en permanence le bon suivi des trajectoires des deux articulations générées par le planificateur de mouvement.

L'efficacité du système proposé est évaluée à travers la simulation d'un robot manipulateur à deux degrés de liberté.

Abstract

In this study, a system of planning and motion control of robot manipulators is developed using genetic algorithms and fuzzy logic, the proposed system is designed for motion control planning of robot manipulators, it uses a fuzzy controller optimized by a genetic algorithm, it is responsible to control in permanence the good follow-up of the two joints trajectories generated by the movement planner.

The effectiveness of the proposed system is evaluated through the simulation of a robot manipulator with two degrees of freedom.