

UNIVERSITE KASDI MERBAH OUARGLA
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département de l'Electronique et de télécommunications



Mémoire MASTER

Domaine : Sciences et technologies

Filière : Automatique

Spécialité : Automatique et Systèmes

Présenté par :

- ❖ **Beddoua Dia Eddine**
- ❖ **Khaled Douba**

Thème

**COMMANDE ADAPTATIVE FLOUE ROBUSTE :
APPLICATION A LA COMMANDE EN POURSUITE
D'UN ROBOT**

Le : 20 Septembre 2021

Devant le jury :

Mr	TIDJANI Zakaria	MAA	Président	UKM Ouargla
Mr	BENHELLAL Belkheir	MCB	Examineur	UKM Ouargla
Mm	KARA Fouzia	MAA	Promotrice	UKM Ouargla

Année universitaire : 2020/2021

Remerciements

En préambule de ce projet nous remercions **ALLAH**

Qui nous a aidés et nous a fourni patience et courage pendant ces années d'études.

Merci à Mm **KARA Fouzia**

Pour ses efforts et pour nous joindre pour les meilleurs résultats.

Nous remercions également **nos mères** et **nos pères** qui nous ont donné vie, tendresse et courage pour réussir. Quoi que nous puissions vous donner, vous ne pourrez pas exprimer l'amour et la reconnaissance que nous détenons.

Nous tenons à exprimer nos sincères remerciements aux personnes qui nous ont aidés et qui ont contribué à l'élaboration de ce message ainsi qu'à la réussite de cette belle année académique.

Table des matières

Remerciements	iii
Table des matières	i3
Liste des figures	6
Introduction générale	1

Chapitre 1: Généralités sur les systèmes flous adaptatifs et robustes

1.1. Introduction	3
1.2. Généralités sur les systèmes flous	3
1.2.1 Le fonctionnement d'un système flou	3
1.2.2. La fuzzification.....	4
1.2.3. La défuzzification.....	5
A) La moyenne des maximas revient.....	5
B) la méthode du centre de gravité.....	5
1.2.4. Fonctions d'appartenances.....	6
1.2.4.1. Fonction triangulaire.....	6
1.2.4.2. Fonction trapézoïdale.....	6
1.2.4.3. Fonction gaussienne	6
1.2.4.4. Fonction sigmoïdale	6
1.2.5. Opérateurs de la logique floue.....	7
1.2.5.1. L'opérateur ET.....	7
1.2.5.2. L'opérateur OU	7
1.2.5.3. L'opérateur NON.....	7
1.3. La commande floue.....	7
1.3.1. Introduction.....	8
1.3.2. Les différentes parties de la commande floue	9

1.3.3. But de la Commande Floue.....	10
1.3.4. Avantages des systèmes flous	11
1.3.5. Différents domaines d'applications.....	12
1.3.6. Règles linguistiques.....	13
1.3.7. Conception d'un contrôleur flou.....	14
1.3.8. Les différents types de la commande à base de la logique floue	15
1.3.9. Approximation floue	16
1.4. La commande adaptative.....	17
1.4.1. Définition	18
1.4.2. Commande adaptative directe et indirecte	19
1.4.2.1. Commande adaptative directe.....	20
1.4.2.2. Commande adaptative indirect.....	21
1.4.3. Technique de la commande adaptative.....	22
1.4.4. Commande adaptative à modèle de référence(MRAC).....	24
1.4.4.1. Commande adaptative indirecte (STC)	24
1.4.4.2. La différence entre M RAC et STC.....	24
1.4.5. Les différents domaines de la commande adaptative	25
1.4.5.1. La commande adaptative floue.....	26
A) Commande adaptative directe floue	26
B) Commande adaptative indirecte floue	26
1.4.6. La loi de commande	27
1.4.6.1. La méthode du gradient	28
1.5. La commande robuste.....	29
1.5.1. Asservissement nominale.....	30
1.5.2. Commande robuste par Lyapunov	31
1.5.3. Stabilité nominale.....	32
1.5.4. Méthodes de Lyapunov	33
A) Méthode directe de Lyapunov	33
B) Méthode indirecte de Lyapunov	33

1.6. Conclusion.....	34
----------------------	----

Chapitre 2 : Généralités sur les robots

2.1. Introduction.....	35
2.2. Le robot.....	36
2.3. Historique de l'évolution de la robotique et les robots.....	37
2.4. L'histoire du développement des robots.....	38
2.4.1. L'horloge	39
2.4.2. L'automate d'Al Jazari	40
2.4.3. Canard mécanique	41
2.4.4. Le premier bras télé-opéré.....	42
2.5. Les types des robots	43
2.6. Classification des robots	44
A) Les manipulateurs	45
B) Les télémanipulateurs	46
C) Les robots	47
2.7. Programmation des robots Classiquement	48
2.8. Classification des robots	49
2.9. Le robot industriel.....	50
2.9.1. Industrie	51
2.9.2. Domaine militaire	52
2.9.3. Santé.....	53
2.10. Conclusion	54

Chapitre 3 : Application a la commande en poursuite d'un robot

3.1. Introduction.....	55
3.2. Position du problème	56
3.3. Système flou (F.L.S.)	57
3.4. Mise en œuvre de la commande	58
3.5. Simulation et résultats	59
3.6. Conclusion	60
3.7. Conclusion générale.....	61
3.8. Résumé	62

3.9. Références63**Liste des figures**

Figure 1.1: Fonction d'appartenance de la température à la variable linguistique «chaud»	1
Figure 1.2. Fonction d'appartenance de la température à la variable linguistique «chaud».....	2
Figure 1.3: Fonction d'appartenance	3
Figure 1.4: Fonction d'appartenance du rapport cyclique.....	4
Figure 1.5: Fonction d'appartenance du rapport cyclique	5
Figure 1.6: Fonction d'appartenance du rapport cyclique	6
Figure 1.7: Contrôleur flou	11
Figure 1.8: Commande adaptative directe	15
Figure 1.9 : Modèle d'erreur	18
Figure 2.1: L'horloge.....	23
Figure 2.2: L'automate d'Al Jazari.....	24
Figure 2.3: Canard mécanique	24
Figure 2.4: L'utilisation de robots dans l'industrie automobile.....	28
Figure 2.5: L'utilisation de robots dans le domaine des premiers secours pendant les guerres.....	28
Figure 2.6: L'utilisation de robots dans le domaine du diagnostic médical.....	29
Figure 3.1 : Le schéma de commande de l'approche proposée.....	30
Figure 3.2 : Bras de robot à 2 degrés de liberté.....	31
Figure 3.3 : La réponse de $q_1(t)$ (trait plein) et la valeur désirée $q_1(t)=\sin(t)$ (trait pointillé)..	32
Figure 3.4 : La réponse de $q_2(t)$ (trait plein) et la valeur désirée $q_2(t)=\cos(t)$ (trait pointillé)..	33
Figure 3.5 : La tension U_1 au niveau de la 1 ^{ème} articulation).....	34
Figure 3.6 : La tension U_2 au niveau de la 2 ^{ème} articulation).....	35
Figure 3.7 : L'erreur quadratique au niveau de la 1 ^{ère} articulation	36
Figure 3.8 : L'erreur quadratique au niveau de la 2 ^{ème} articulation).....	3

Introduction générale

La logique floue repose sur la théorie des ensembles flous développée par Zadeh en 1965. A côté d'un formalisme mathématique fort développé, nous préférons aborder la présentation de manière intuitive. Les notions de température moyenne ou de courant faible sont relativement difficiles à spécifier de manière précise. En logique floue, Le concept de variable linguistique, comme son nom le suggère, joue un rôle très important. C'est une variable dont les valeurs sont des mots ou des phrases au lieu de nombres. Par exemple, 'Température' est une variable linguistique et l'ensemble des valeurs qu'elle peut prendre est appelé l'ensemble de termes (nuances), soient par exemple : Basse, Moyenne, Elevée. D'autres termes peuvent être construits à l'aide de modificateurs linguistiques tels que (très, plus ou moins, un peu...etc.). Chaque valeur linguistique (e.g., faible) est caractérisée par sa fonction d'appartenance faible qui associe à chaque valeur numérique x de 'Température', un degré qui reflète la vérité du sens faible.

La plupart des systèmes industriels sont, multi entrées multi sorties, complexes, difficilement modélisables, présentant de fortes non linéarités et entachés de perturbations structurelles et externes. Ces dernières années, la commande en poursuite des systèmes non linéaires a été un des problèmes auxquels les automaticiens se sont intéressés. Slotine et al [1] ont proposé, pour résoudre le problème de poursuite, un algorithme de commande adaptative, où les paramètres sont linéarisés puis ajustés. Afin d'appliquer cette méthode à un système robotique, par exemple, il faut considérer dix paramètres par articulation, ce qui a pour effet de rendre la mise en œuvre de ce type d'algorithme très longue et fastidieuse. D'autre part, dans les références [2-3], les auteurs ont proposé une méthode basée sur les réseaux de neurones, mais la preuve mathématique de la stabilité de la commande générée n'a pas été démontrée. De plus, la structure des réseaux ainsi que la loi d'adaptation sont déduites des différents tests et mesures réalisés sur le système. Ainsi le problème de convergence de l'erreur de poursuite et le choix des paramètres initiaux restent posés. Lam et al [4] ont proposé une commande adaptative floue basée sur l'approche de TakagiSugeno. Néanmoins, seuls les systèmes non

perturbés ont été étudiés, de plus la connaissance du modèle du système en des points de fonctionnement est nécessaire. Basé sur l'existence de l'approximateur universel [5], Hamzaoui et al. [6] ont proposé une commande adaptative floue robuste pour les systèmes SISO non linéaires incertains et perturbés, son extension aux systèmes MIMO a été donnée dans la référence [7]. L'approche proposée est basée sur des systèmes flous et génère une commande robuste. Néanmoins, la convergence de l'algorithme proposé ne peut être assurée que par un choix judicieux des paramètres sous entendant une connaissance minimale du système. Dans le cas où le modèle de référence (nominal) est connu, l'approche précédente peut être simplifiée. En effet, il suffit d'approcher les incertitudes sur le système par rapport au modèle de référence et de trouver une commande permettant de les éliminer ainsi que les perturbations externes ; c'est l'objectif de cet article. On présente donc une méthode permettant d'utiliser un algorithme adaptatif flou générant une commande qui permet d'éliminer les incertitudes sur le modèle dynamique du système. Une commande de type H_∞ est introduite pour atténuer l'effet des résidus d'élimination et des perturbations externes à un niveau donné par le concepteur. La robustesse et la stabilité de l'approche proposée sont basées sur la théorie de Lyapunov [8].

Dans ce contexte, le travail réalisé dans ce mémoire fait l'objet de trois chapitres :

Dans le premier chapitre, nous présentons une généralité sur système flous adaptatifs.

Le deuxième chapitre est consacré aux différents robots.

La mise en oeuvre de l'approche proposée et pour la validation de cette l'approche, la simulation d'un bras de robot à deux degrés de liberté est présentée dans le troisième chapitre.

Enfin, ce modeste travail sera clôturé par une conclusion générale.

Chapitre 1

Chapitre 1: Généralités sur les systèmes flous adaptatifs et robustes

1.1. Introduction

Dans ce chapitre, nous aborderons trois parties importantes de la logique floue. En général, nous parlerons d'abord des systèmes flous et de leur mécanisme de fonctionnement et les expliquerons en détail

Deuxièmement, nous expliquerons la commande floue et son fonctionnement, puis nous expliquerons la commande adaptative et robuste

1.2. Généralités sur les systèmes flous

1.2.1. Le fonctionnement d'un système flou

La logique floue est un système qui nous permet de convertir les valeurs mesurées par des capteurs, telles que la température, la distance et la pression en valeurs linguistiques.

Par exemple, l'un des capteurs mesure une distance que nous considérons 50 mètres lorsque nous convertissons cette langue en une valeur linguistique qui peut être dite à environ 30% proche, 50% moyenne, 20% loin.

Et il en va de même pour les utilisations en logique floue. [11]

1.2.2. La fuzzification

Cette étape, qui est l'étape de fuzzification, nous devons changer la valeur que nous obtenons du capteur en un langage approximatif que nous, les humains, pouvons comprendre et nous faciliter l'étude du système avec plus de précision.

Par exemple, la température provenant du capteur de mesure de température, on peut la convertir dans une langue, donc on dit de cette valeur, soit froide ou très froide ou modérée ou chaude ou très chaude. [11]

Chacun comment les diviser : une personne ne peut prendre que trois valeurs linguistiques et une autre les prend toutes et les analyse.

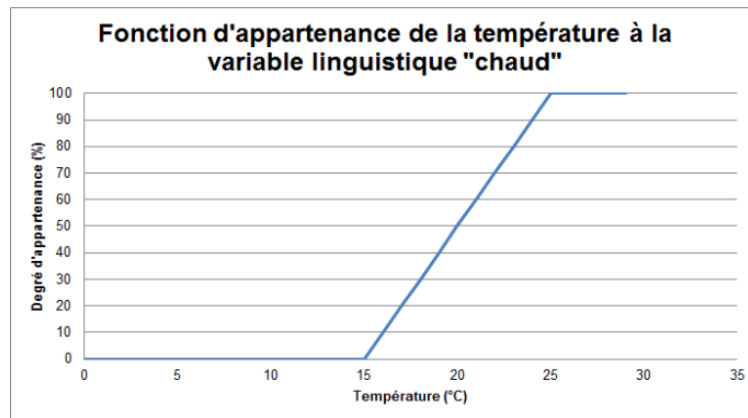


Figure 1.1: Fonction d'appartenance de la température à la variable linguistique «chaud»

Ici, si notre capteur de température nous indique 23°, la fuzzification nous dira que la température est chaude à 80%.

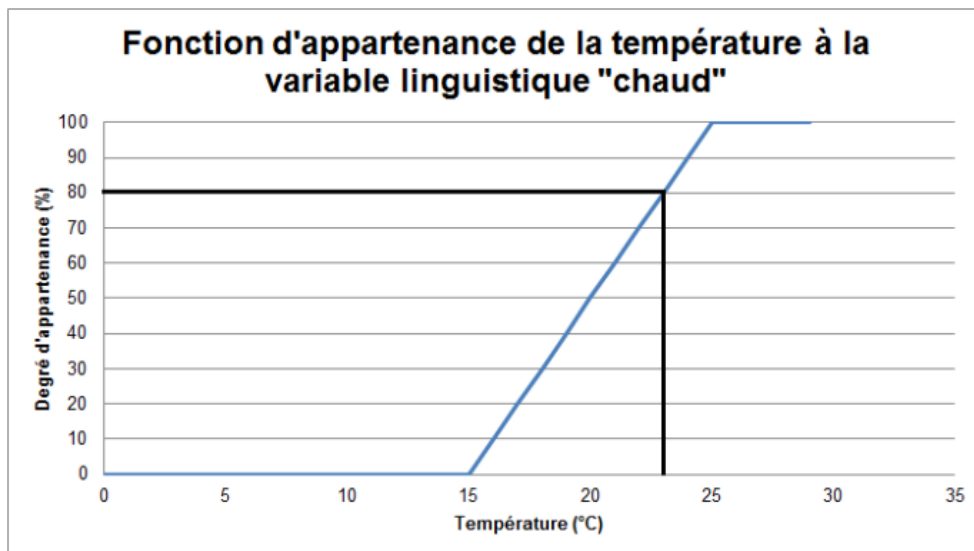


Figure 1.2: Fonction d'appartenance de la température à la variable linguistique «chaud»

Comme nous l'avons dit précédemment, la valeur mesurée peut être changée en plusieurs valeurs linguistiques et puisque toutes ces valeurs linguistiques représentent la même valeur mesurée par le capteur, nous pouvons la représenter sur la même courbe [1]

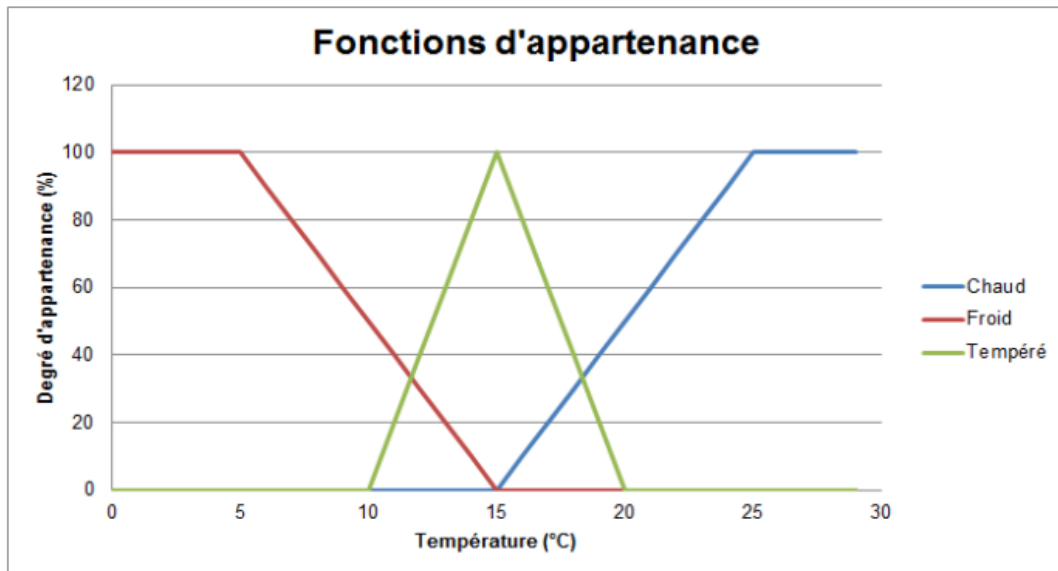


Figure 1.3: Fonction d'appartenance

Si le capteur nous renvoie 17°C, après fuzzification, la température sera chaude à 20%, tempérée à 60% et froide à 0%. [11]

1.2.3. La défuzzification

Maintenant, comme dernière étape pour obtenir un système flou, les valeurs linguistiques doivent être converties en données numériques pour supprimer la confusion sur le système et résoudre le problème à résoudre, et cette étape se fait en deux étapes:

Premièrement: fusionner les variables linguistiques qui s'écoulent en un seul sens, en utilisant la logique du concepteur du système.

Par exemple, si on a trois règles qui génèrent la variable linguistique accélère fortement à 20%, 25% et 35%. Il en résultera que la variable accélère fortement aura pour valeur finale 35%.

Deuxièmement: cette étape est maintenant sensible. Nous avons plusieurs variables linguistiques qui représentent les mêmes données, par exemple, quatre variables linguistiques qui représentent une température, et voici le problème en lui-même. Par conséquent, pour éliminer la confusion sur le système, la meilleure valeur doit être choisi en fonction de ce système.

Par exemple, on peut avoir trois variables linguistiques: accélération forte à 35%, accélération moyenne à 80% et accélération faible à 0% qui qualifie l'accélération.

Il existe plusieurs méthodes pour défuzzifier. Parmi les plus utilisés, on peut citer la méthode de la moyenne des maximas et la méthode du centre de gravité. [11]

A) La moyenne des maximas revient

Cette méthode vise à prendre la moyenne des valeurs linguistiques qui ont la même valeur. Nous prenons dans cette méthode 80% pour exprimer le maximum et cela s'exprime dans la relation suivante:

$$valeur = \frac{\int_S x dx}{\int_S dx} \text{ avec } S = \{x, \mu(x) = \sup(\mu(x))\} \quad (1.1)$$

B) La méthode du centre de gravité

Nous prenons les coordonnées du centre de gravité et sont exprimées dans la relation suivante:

$$valeur = \frac{\int_S \hat{\mu}(x).x.dx}{\int_x \hat{\mu}(x).dx} \quad (1.2)$$

Pour illustrer ces deux méthodes de défuzzification, prenons un exemple. L'on souhaite régler le rapport cyclique d'un moteur à l'aide d'un système flou. On fournit au système des données brutes et un ensemble de règles d'inférence. Le système calcul automatiquement les sorties sous forme de variables linguistiques. On suppose que les variables linguistiques de sorties

sont rapport cyclique faible à 0%, rapport cyclique moyen à 80% et rapport cyclique fort à 30%. [11]

L'on considère que l'on a choisi les fonctions d'appartenances suivantes pour chacune des trois variables linguistiques :

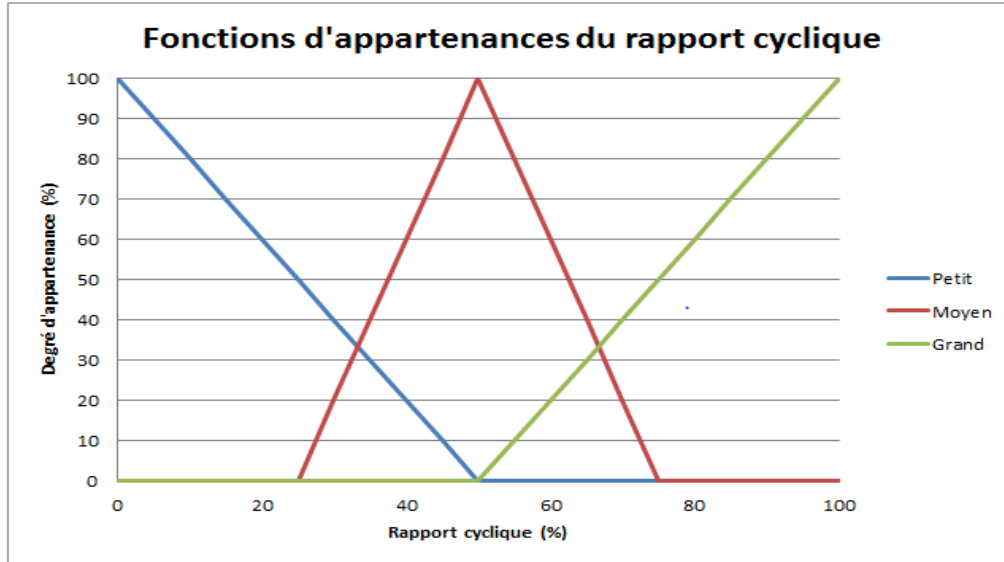


Figure 1.4: Fonction d'appartenance du rapport cyclique

Avec les données en sortie du moteur d'inférence, on est capable de générer la nouvelle fonction d'appartenance :

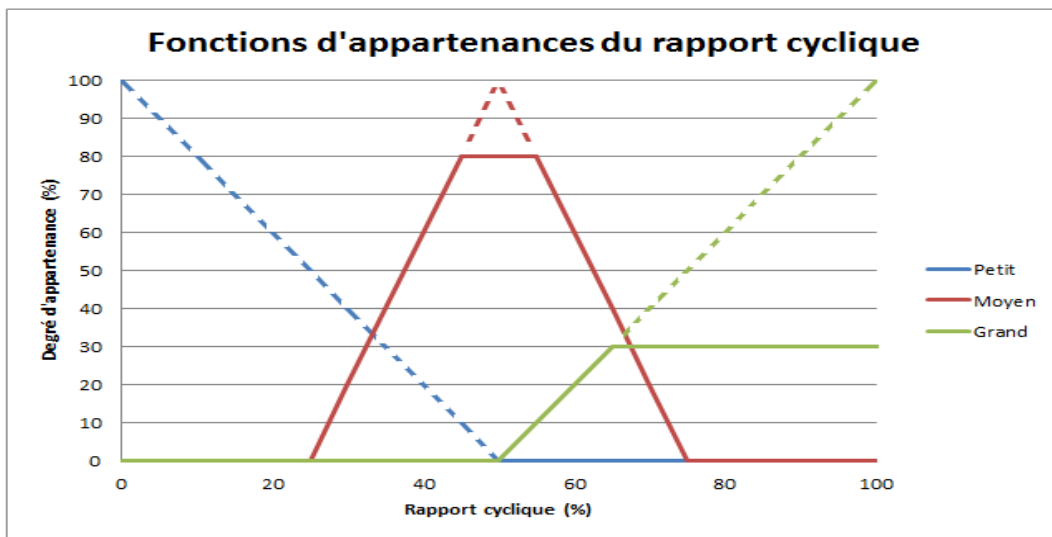


Figure 1.5: Fonction d'appartenance du rapport cyclique

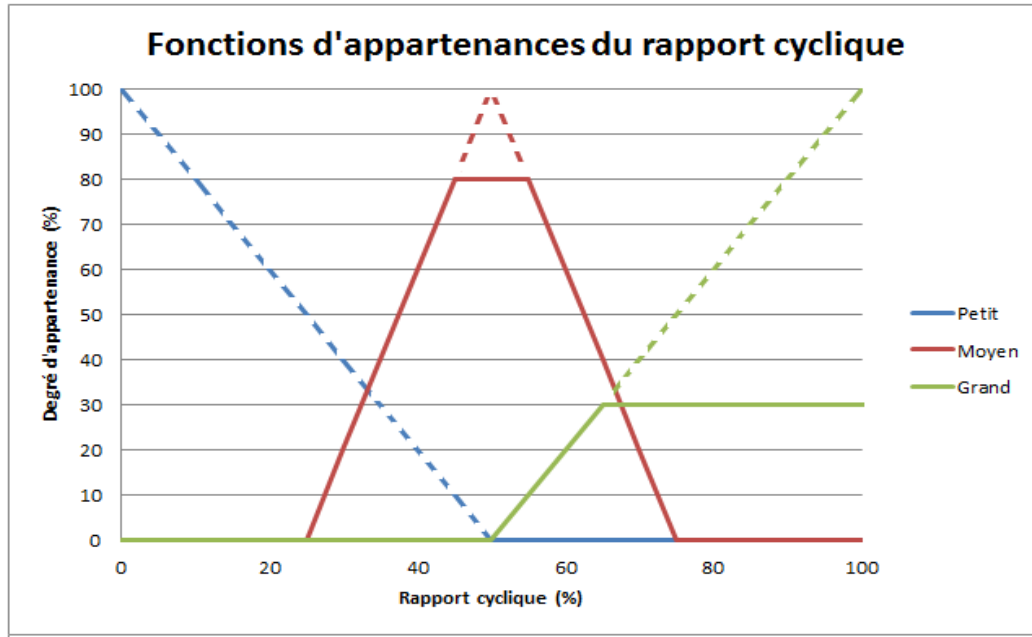


Figure 1.6: Fonction d'appartenance du rapport cyclique

Ainsi, pour défuzzifier, si l'on applique la première méthode de la moyenne des maxima, on remarque que la fonction d'appartenance résultante est maximale pour un rapport cyclique compris entre 40 et 60%. Il suffit de faire la moyenne sur l'intervalle [40,60] pour trouver le rapport cyclique résultant. Ici, le résultat de la défuzzification vaudra 50%. [11]

1.2.4. Fonctions d'appartenances

Dans ce contexte on associe à chaque valeur de la variable linguistique une fonction d'appartenance désignée par $U_a(x)$, qui sera désignée par le degré ou le facteur d'appartenance. Il est à noter que l'ensemble des éléments de x pour lesquels $U_a(x) > 0$, Est appelé «support de A ». Le plus souvent, on utilise pour les fonctions d'appartenance les fonctions suivantes. [13]

1.2.4.1. Fonction triangulaire

Elle est définie par trois paramètres $\{a, b, c\}$ qui déterminent les coordonnées des trois sommets.

$$U(x) = \begin{cases} \frac{x-a}{b-a} & \text{si } x \in [a, b]; \\ \frac{c-x}{c-b} & \text{si } x \in [b, c]; \\ 0 & \text{ailleurs} \end{cases} \quad (1.3)$$

1.2.4.2. Fonction trapézoïdale

Elle est définie par quatre paramètres {a, b, c, d}

$$U(x) = \max(\min(\frac{x-a}{b-a}, 1, \frac{c-x}{c-d}, 0) \quad (1.4)$$

1.2.4.3. Fonction gaussienne

elle est définie par deux paramètres {m,s}

$$U(x) = e^{-(x-\frac{m}{2s})^2} \quad (1.5)$$

m: Centre de la gaussienne

s: largeur

1.2.4.4. Fonction sigmoïdale

Elle est définie par deux paramètres {a, c} : [13]

$$U(x) = \frac{1}{e^{(a-(x-c))}} \quad (1.6)$$

1.2.5. Opérateurs de la logique floue

L'une des portes logiques les plus importantes utilisées dans la logique floue ET et OU, et elles sont utilisées sur la base des deux règles: MIN et MAX.

Par exemple, si le vent est fort, et que nous symbolisons 'x' et que la pièce est chaude, et que nous la désignons ici 'y', une des décisions doit être prise, soit de fermer la porte, soit de l'ouvrir pour éviter l'un des agents A, B montre ici l'importance d'utiliser les portes dans une logique floue.

1.2.5.1. L'opérateur ET

L'opérateur ET est réalisé dans la plupart des cas par la formation du minimum (intersection) des deux fonctions d'appartenance :

$$\mu(z) = \mu_{A(z) \text{ et } B(z)}(z) = \min(\mu_A(x), \mu_B(y)) \quad (1.7)$$

1.2.5.2. L'opérateur OU

L'opérateur OU est réalisé par la formation du maximum des deux fonctions d'appartenance :

$$\mu(z) = \mu_{A(z) \text{ et } B(z)}(z) = \max(\mu_A(x), \mu_B(y)) \quad (1.8)$$

1.2.5.3. L'opérateur NON

Il est caractérisé en logique floue par la fonction :

$$\hat{\mu}_{4_c}(z) = \hat{\mu}_{4_j}(x) = 1 - \hat{\mu}_{4_A}(x) \quad (1.9)$$

1.3. La commande floue

1.3.1 Introduction

La commande floue est le domaine dans lequel il existe le plus de réalisations effectives, en particulier industrielles. Son but est de traiter des problèmes de commande de processus, le plus souvent à partir des connaissances des experts ou d'opérateurs qualifiés travaillant sur le processus. On peut citer par exemple la commande de machines-outils, de groupes d'ascenseurs, d'appareils électroménagers, de caméras, de voitures ou d'hélicoptères sans pilote, de métro, dont des réalisations existent au Japon.

La commande floue a le même but qu'une commande réalisée en automatique classique, c'est-à-dire la gestion automatique d'un processus, en fonction d'une consigne donnée, par action sur les variables qui décrivent le processus. Elle en diffère cependant sur les points suivants :

La connaissance mathématique du fonctionnement du processus n'est pas nécessaire. C'est le savoir-faire de l'opérateur qualifié manipulant habituellement le processus ou les connaissances d'experts qui sont prises en compte pour mettre au point la commande floue. Si l'on se réfère à la conduite d'une voiture, par exemple, il n'est pas nécessaire de savoir comment fonctionne le moteur pour en réaliser une commande floue, il suffit de savoir comment agit un conducteur expérimenté.

Des variables caractérisées subjectivement sont utilisables. Les sens humains (toucher, vue...) peuvent par exemple être modélisés.

1.3.2. Les différentes parties de la commande floue

Un contrôleur flou est un système à base de connaissances particulier, utilisant un raisonnement en profondeur limité, dans une procédure de chaînage avant règles (activation des règles par les prémisses). On peut distinguer plusieurs étapes dans le traitement des règles.

Un schéma représentatif peut être le suivant : Les variables caractéristiques du système à commander et les consignes définissent les variables d'entrée du contrôleur flou. Les variables

caractéristiques sont en général, les grandeurs de sortie du processus et, le cas échéant, d'autres mesures déterminantes pour saisir l'évolution dynamique du processus. Les variables de sortie du contrôleur flou sont les commandes à appliquer au processus. La base de connaissances est composée d'une base de données et d'une base de règles. La base de données regroupe :

- Les ensembles flous associés aux variables d'entrée et de sortie du contrôleur flou,
- Les facteurs d'échelle (gains) en entrée (normalisation) et en sortie (dénormalisation). [12]

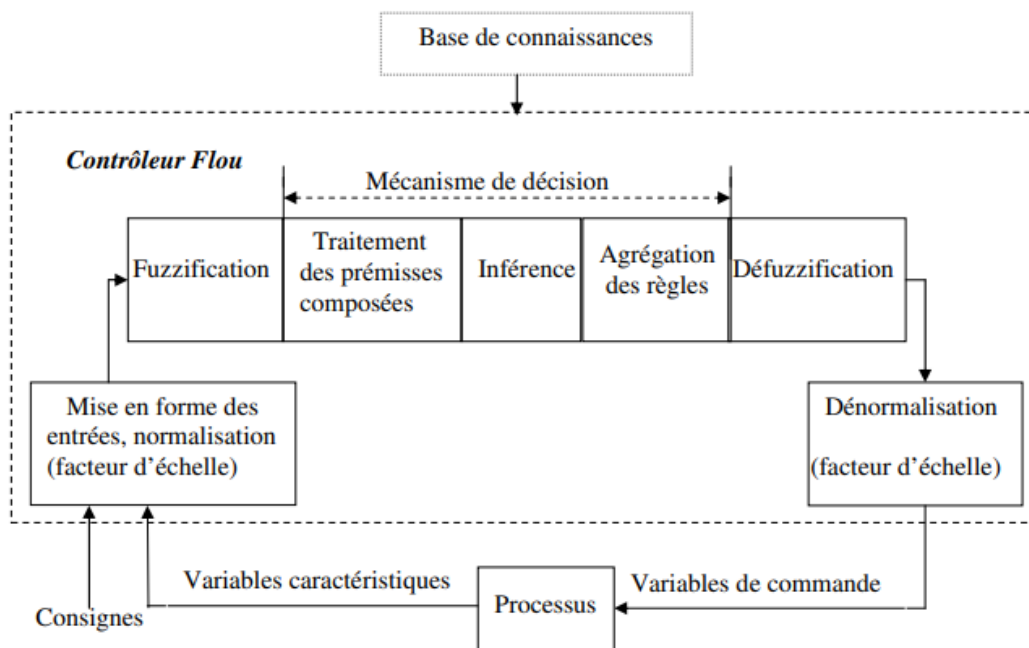


Figure 1.7: Contrôleur flou

1.3.3. But de la Commande Floue

Son but est de résoudre le problème des systèmes complexes, en contrôlant ces systèmes avec des commandes linguistiques que les humains peuvent comprendre. [13]

1.3.4. Avantages des systèmes flou

❖ Le système flou permet d'inclure simplement des informations qualitatives dans la Conception du contrôleur.

❖ Les systèmes flous ne nécessitent pas l'existence de modèles d'analyse de processus. Contrôle, il y a peu d'informations suffisantes pour réaliser une boucle Commandée.

❖ Le système flou est un système non linéaire, il est donc plus approprié pour le contrôle des processus non linéaires. [12]

1.3.5. Différents domaines d'applications

Les produits de consommation sont développés sur la base d'une logique floue, tels que : Appareils électroménagers (machines à laver, aspirateurs, autocuiseurs, etc.).

❖ Système audiovisuel (caméra autofocus, caméra stabilisée Image, copieur,...). Systèmes embarqués (BVA, ABS, suspension, climatisation...),

❖ Système autonome mobile. □ Système de transport (train, métro, ascenseur,...).

❖ Système de climatisation des chambres.

❖ Système d'aide à la décision, diagnostic et identification.

❖ La plupart des systèmes de contrôle/commande industriels Production, transformation et traitement des produits et déchets. [17]

1.3.6. Règles linguistiques

Les règles de langage précisent le fonctionnement du contrôleur à travers des liens Condition, prendre des mesures sur le processus. Notez que la syntaxe est très proche. En langage naturel, la représentation générale est la suivante : SI "la condition A satisfait "ALORS" le résultat B. [17]

1.3.7. Conception d'un contrôleur flou

La commande floue est l'étude du développement et de l'expérimentation basés sur le système Concernant les règles exprimées par catégories linguistiques, par exemple :

si (x est A_1) ET (y est A_2) alors ET (u est B) où A_1 , A_2 et B sont des prédicats motivés, par exemple : positif, grand, zéro, négatif . La structure de base du système flou est illustrée dans la figure ci-dessous :

1.3.8. Les différents types de la commande à base de la logique floue

Basé sur différentes règles et méthodes de contrôle flou Générer, utiliser des méthodes de contrôle de logique floue peuvent être classés Dans les catégories suivantes :

- * La commande floue de type Mamdani
- * La commande floue du type PI ou PID
- * La commande neuro-floue
- * La commande par mode glissant flou
- * La commande floue adaptative
- * La commande floue de type T-S

1.3.9. Approximation floue

Sélection des méthodes de fuzzification, des stratégies de raisonnement et la défuzzification permet la mise en place de différents types de systèmes flous. Dans ces classes, on utilise la fuzzification singleton, la défuzzification du centre de gravité et produit d'inférence, qui représente toutes les fonctions de la forme suivante:

$$y(x) = \frac{\sum_{l=1}^k \bar{y}^l (\prod_{i=1}^n u_{F_i^l}(x_i))}{\sum_{l=1}^k (\prod_{i=1}^n u_{F_i^l}(x_i))} \quad (1.10)$$

1.4. La commande adaptative

Le contrôle adaptatif est un ensemble de concepts et de techniques utilisés Pour un réglage automatique via Internet pour maintenir le niveau de performance en cas d'entrées variables dans le temps ou inconnues.

Les systèmes neuronaux sont utilisés pour approximer la dynamique

Pour le système où certains de ses paramètres sont adaptés sur Internet afin d'assurer la stabilité

À partir du système en boucle fermée, l'objectif principal de cette thèse est de développer des lois

De contrôles adaptatifs stables à une poursuite neurologique

Catégories de systèmes non linéaires SISO et MIMO.

1.4.1. Définition

La commande adaptative est un ensemble de technique utilisée pour l'ajustement automatique en ligne et en temps réel des régulateurs des boucles de commande afin de réaliser ou maintenir un certain niveau de performance quand les paramètres du procédé à commander sont soit inconnus soit/et varient dans le temps. [10]

1.4.2. Commande adaptative directe et indirecte

Le principe de la commande adaptative consiste en l'ajustement des paramètres du régulateur. Cet ajustement se fait en une seule étape (commande adaptative directe), soit en deux étapes (commande adaptative indirecte). Le schéma de la commande adaptative que soit direct ou indirect contient deux boucle :

- Une boucle a contre
- Réaction formée par le processus avec le régulateur.
- Une boucle d'adaptation qui permet d'ajuster les paramètres du régulateur. [10]

1.4.2.1. Commande adaptative directe

L'idée est le calcul direct des entrées du système, de sorte que nous ayons un contrôle direct du système, comme le montre la figure suivante, c'est-à-dire en testant à chaque fois pour obtenir un bon retour sur le système, ou plus clairement, il n'est pas possible de prédire le comportement du système en fonction de l'entrée. [10]

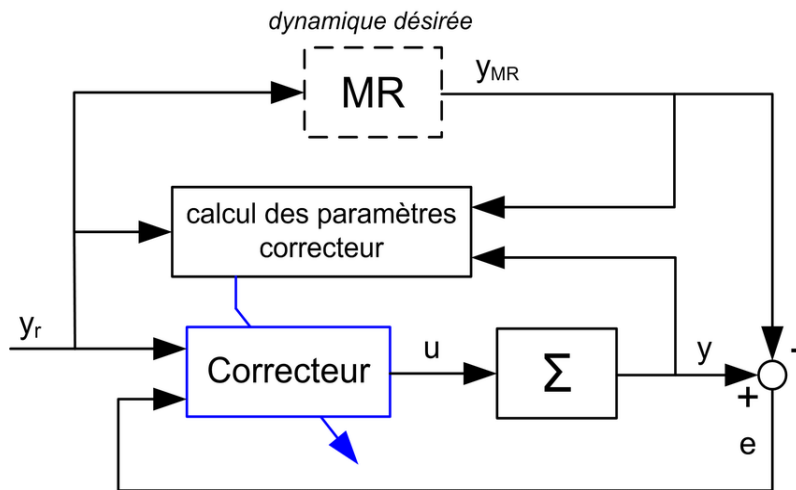


Figure 1.8: Commande adaptative directe

1.4.2.2. Commande adaptative indirecte

Cela dépend principalement de la recherche des entrées du système, puis de leur saisie dans le système pour obtenir le résultat souhaité, ce qui signifie que nous pouvons prédire le contraire direct. [10]

1.4.3. Technique de la commande adaptative

Habituellement, nous trouvons dans les systèmes plusieurs schémas visant à atteindre les performances souhaitées, et ces schémas existent en cas de processus inconnus ou de changement avec le temps.

En utilisant une boucle de contre réaction sur la mesure des performances et ceux qui utilisent une boucle ouverte pour la commande adaptative des systèmes.

1.4.4. Commande adaptative à modèle de référence(MRAC)

Le MRAC a été développé à l'origine par Whitaker et ses collègues en 1958.

Ces dernières années, c'est devenu un moyen très efficace de contrôler Systèmes avec des paramètres inconnus et/ou variables dans le temps. Le principe de cette approche est de réinitialiser les paramètres Modifiable ou travaillant sur le signal de commande ainsi qu'en utilisant un formulaire de référence pour décrire Performances souhaitées et comparables à tout moment avec des performances système réglables. Pour que l'erreur se situe entre le modèle Le point de référence et le système réglable sont inclinés asymptotiquement vers zéro. [10]

1.4.4.1. Commande adaptative indirecte (STC)

Cette approche s'effectue en deux étapes :

- Estimation des paramètres du modèle,
- Calcul des paramètres du contrôleur à partir des paramètres estimés. [11]

1.4.4.2. La différence entre M RAC et STC

Le problème avec le contrôle adaptatif (MRAC) avec un modèle de référence est Déterminer le mécanisme de réglage (MA) pour rendre le système stable et provoquer des erreurs Obtenez zéro. La différence entre MRAC et STC (Automatic Control System) Réglable ou réglable automatiquement) réside dans le mode de réglage du paramètre du régulateur. Pour le STC, l'ajustement se fait en deux étapes, à l'aide d'un algorithme explicite ou indirect. La modification du régulateur implique une identification en boucle fermée Paramètres inconnus du système, puis ajustez ces paramètres. Pour le MRAC, l'ajustement peut être effectué en une seule étape. Ajustable Les paramètres du contrôleur sont directement en boucle fermée, en utilisant des méthodes directes ou Implicite. [11]

1.4.5. Les différents domaines de la commande adaptative

1.4.5.1. La commande adaptative floue

La commande adaptative floue se compose donc d'un régulateur flou adaptatif.

La différence entre un régulateur adaptatif flou et un régulateur adaptatif conventionnel : C'est que nous pouvons prendre en compte les informations linguistiques pour maintenir les performances du système. [11]

A) Commande adaptative directe floue

Les systèmes flous sont utilisés comme contrôleurs.

B) Commande adaptative indirecte floue

Les systèmes flous sont utilisés comme modèles pour les systèmes. [17]

1.4.6. La loi de commande

L'ajustement des paramètres du régulateur peut être obtenu de deux manières en utilisant :

- La méthode du gradient (MIT)
- La théorie de stabilité (Lyapunov)

On va utiliser :

1.4.6.1. La méthode du gradient

Cette méthodologie dépend de l'entrée du système étant variable et elle est très lente par rapport aux autres systèmes.

Le système n'est pas nécessairement stable lors de l'application de cette méthodologie.

Elle est communément appelée la règle du MIT. [17]

Règle MIT

Supposons que nous allons ajuster les paramètres du régulateur de telle sorte que l'erreur entre les sorties du procédé et du modèle de référence soit amenée à zéro. Soit e l'erreur de poursuite, le paramètre à ajuster et soit le critère d'optimisation suivant

$$J(\theta) = \frac{1}{2} e^2 \tag{1.11}$$

Pour obtenir le min de J, il est raisonnable de changer les paramètres dans la direction négative du gradient de J

$$\begin{cases} \frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} \\ -\gamma \cdot e \frac{\partial e}{\partial \theta} \end{cases} \quad \text{ou } \gamma > 0$$

Cette règle MIT fonctionnera bien si le gain d'adaptation est faible.

$\frac{d\theta}{dt}$ représente la sensibilité du système, et détermine la vitesse d'adaptation des paramètres.

Cette La figure ci-dessous montre le modèle d'erreur [18]

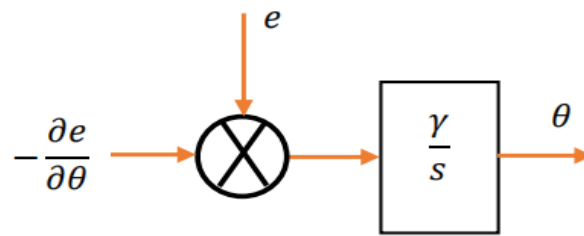


Figure 1.9 : Modèle d'erreur

Le choix du critère est arbitraire, si on pose :

$$J = |e| \text{ alors } \frac{\partial J}{\partial \theta} = -\gamma \frac{\partial J}{\partial \theta} \text{sign}(e) \tag{1.12}$$

Si le gain adaptatif est petit, la règle MIT est valide, mais sa valeur peut être Dépend de l'amplitude du signal de référence. Par conséquent, il est impossible Une frontière fixe de est donnée pour assurer la stabilité globale du système. Cela indique, Les règles du MIT saperont la stabilité des systèmes en boucle fermée. Peut obtenir Programme de gradient amélioré, où le taux d'ajustement ne dépend pas de Pour contrôler l'amplitude du signal, une possibilité consiste à effectuer la normalisation est Remplacez les règles du MIT par :

$$\frac{d\theta}{dt} = -\gamma \frac{e \frac{\partial e}{\partial \theta}}{\alpha + \left(\frac{d\theta}{dt}\right)^T \frac{d\theta}{dt}} \tag{1.13}$$

$$\frac{d\theta}{dt} = \gamma \text{sat} \left[\frac{e \frac{\partial e}{\partial \theta}}{\alpha + \left(\frac{d\theta}{dt}\right)^T \frac{d\theta}{dt}} \beta \right] \tag{1.14}$$

$$\text{Où } \begin{cases} -\beta & x < -\beta \\ x & |x| \leq \beta \\ x & x > \beta \end{cases}$$

1.5. La commande robuste

1.5.1. Asservissement nominale

Nous définissons au moyen d'un «système nominal» le modèle mathématique du système dans lequel nous opérons. Voulant passer commande, ce modèle mathématique ne prend pas en compte les paramètres négligés ou supprimés lors de la construction du modèle précédent. La conception nominale des servos consiste à trouver un correcteur $k(s)$ capable d'assurer les performances requises du système nominal.

Pour une bonne maîtrise de la conception du correcteur $k(s)$ est nécessaire

Comprendre les avantages des exigences de performance et connaître la formule des normes mathématiques. [17]

1.5.2. Commande robuste par Lyapunov

Une solution intéressante au problème de l'incertitude (paramétrique et/ou non paramétrique) est entièrement donnée par Lyapunov. Cette technique utilise une fonction de Lyapunov Système nominal pour calculer la commande ajoutée à la commande initiale afin d'améliorer le système en présence d'incertitude qui satisfait des paramètres inconnus. Trouvé dans la même équation d'état que la commande [17]

1.5.3. Stabilité nominale

La stabilité du système est nécessaire et nécessaire, et tout défaut mineur qui perturbe la stabilité de ce système peut provoquer des catastrophes et des problèmes humains et économiques. [17]

Afin d'assurer cette stabilité, nous clarifions deux théories:

1.5.4. Méthodes de Lyapunov

La méthode de Lyapunov permet d'étudier la stabilité des systèmes complexes décrits par le système différentiel. La stabilité d'un système est la capacité du système lorsqu'il dévie de sa

position d'équilibre à temps, il revient finalement à sa position d'équilibre. Il y en a deux Méthodes pour prouver directement et indirectement la stabilité du système. [18]

A) Méthode directe de Lyapunov

L'étude de la stabilité des systèmes différentiels $\dot{x} = f(x, t)$ est énoncée par le théorème suivant

Théorème

Soit $V(x, t)$ une fonction candidate de Lyapunov qui satisfait les conditions suivantes :

* $\forall x \neq 0, v(x) > 0, v(0, t) = 0$ est de classe C^1

* quand $\|x\| \rightarrow \infty, alors v(x) \rightarrow \infty$ fonctions indéfiniment croissante.

On définit la dérivée de la fonction de Lyapunov par :

$$\dot{V}(x, t) = \frac{\partial V}{\partial x}(x, t) + \left[\frac{\partial V}{\partial x}(x, t) \right]^T f(x, t) \tag{1.15}$$

$$\left[\frac{\partial V}{\partial x}(x, t) \right] \text{ est le gradient de } \frac{v(x,t)}{x} = \begin{pmatrix} \frac{\partial V}{\partial x_1}(x, t) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial V}{\partial x_n}(x, t) \end{pmatrix}$$

si $\dot{V}(x) < 0$ alors le système est asymptotiquement stable.

B) Méthode indirecte de Lyapunov

Soit le système décrit par la représentation d'état suivante :

$$\dot{x} = Ax(t) + Bu(t) \tag{1.16}$$

On considère une fonction candidate de Lyapunov quadratique $V(x) = x^T Px$, alors

$$\dot{V}(x) = \dot{X}^T P X + X^T P \dot{X} = (X^T A^T + u^T B^T) P X + X^T P (A x + B u) \tag{1.17}$$

$$\dot{V}(x) = X^T (A^T P + P A) x + u^T B^T P x + x^T \tag{1.18}$$

$u(t) = -R^{-1} B^T P x(t)$ qui stabilise le système par retour d'état et minimise le critère

$J = \int_0^\infty (x^T Q x + u^T R u) dt$ par le Hamiltonien, alors :

$$\dot{V} = X^T (A^T P + P A) x - X^T P^T B R^{-1} B^T P x - x^T P B R^{-1} B^T P x \tag{1.19}$$

$$\dot{V}(x) = X^T (A^T P + P A - (P^T B R^{-1} + P B R^{-1}) B^T P) x \tag{1.20}$$

$$P^T = p \text{ alors } V(x) = X^T ((A^T P + P A) x - 2 P B R^{-1} B^T P) x \tag{1.21}$$

Théorème : Une condition nécessaire et satisfaisante pour que un système. $\dot{x} = Ax + Bu$

Soit asymptotiquement stable est que $\forall Q = Q^T > 0$ et $R > 0$, il existe solution unique P l'équation de Riccati, symétrique et définie positive

$$A^T P + PA - 2PBR^{-1}B^T P + Q = 0 \quad (1.22)$$

Remarque : Une autre forme de l'équation de Riccati est comme suit :

$$A^T P + PA - 2PBR^{-1}\left(\frac{1}{r} - \frac{1}{2\rho^2}\right)B^T P + Q = 0 \quad (1.23)$$

Avec $Q = Q^T > 0$ ET $\left(\frac{2}{r} - \frac{1}{\rho^2}\right)$ où r est une constante positive et ρ représente un niveau d'atténuation donné. Cette équation est utilisée généralement dans la robustification par H_∞ . [18]

1.6. Conclusion

Dans ce chapitre nous avons présenté les idées principales de la logique floue, en introduisant la notion de system flou, Un système flou est composé essentiellement de quatre étapes : la fuzzification, la base de règle, le moteur d'inférence et la defuzzification. Il est à noter que le concepteur d'un système flou doit faire un nombre de choix importants, et on a parlé sur la commande adaptative floue robuste avec le type H_∞ et aussi la théorie de Lyapunov.

Chapitre 2

Chapitre 2 : Généralités sur les robots

2.1. Introduction

L'origine du mot robot est issue du tchèque "robota" qui signifie travail forcé. Le terme de robotique est apparu en 1942 dans le cycle universellement connu rédigé par Isaac Asimov et intitulé "Les robots". Un robot est une machine programmable qui imite des actions d'une créature intelligente. Isaac Asimov a écrit les 3 lois fondamentales de la robotique. Celles-ci permettent d'éviter que cette science amène l'apocalypse avec elle

1ère loi : Un robot ne peut porter atteinte à un être humain ni, en restant passif, laisser cet être humain exposé au danger.

2ième loi : Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la première loi.

3ième loi : Un robot doit protéger son existence dans la mesure où cette protection n'est pas en contradiction avec la première ou la deuxième loi : Du temps des verbes. [20]

2.2. Le robot

C'est une machine qui peut gérer les choses dans l'environnement externe en faisant quelques commandes en fonction de la situation et soumet ces commandes à travers un programme que nous envoyons à la mémoire de cette machine et elle se compose de plusieurs sens, qui sont les détecteurs et un bouclier ou plus pour faire le mouvement Par exemple: S'il se trouve sur un robot en mouvement équipé d'une caméra U Pour se déplacer dans une pièce inconnue, il peut être programmé pour surmonter les obstacles qui pourraient obstruer sa trajectoire. [7]

2.3. Historique de l'évolution de la robotique et les robots

Durant plusieurs millénaires, les automates ont reposé sur des ressorts, engrenages et autres mécanismes, ce qui ne les a pas empêchés d'atteindre une grâce qui force l'admiration. Et puis

l'informatique est venue changer la donne en permettant de stocker de très larges quantités d'informations et de séquences d'actions dans une petite puce. Le robot est ainsi arrivé à une sophistication telle qu'il peut désormais tenter de trouver par lui-même la solution de certains problèmes [14]

2.4. L'histoire du développement des robots

2.4.1. L'horloge

En l'an 246 avant JC, les reliques du scientifique Ctésibios ont été trouvées, qui sont représentées dans sa conception et sa créativité en une heure très précise. [14]



Figure 2.1: L'horloge

2.4.2. L'automate d'Al Jazari

Le scientifique Al-Jazari a inventé des horloges à eau d'une longueur d'environ un mètre et 35 centimètres, et il s'est plié à travers le monde d'Archimède, et ces horloges ont fait un saut dans le monde de la robotique. [14]



Figure 2.2: L'automate d'Al Jazari

2.4.3. Canard mécanique

Jacques de Vaucanson a inventé le canard mécanique au XVIIIe siècle (1709), et il considérait cet âge comme le siècle d'or du passage aux automates. [14]



Figure 2.3: Canard mécanique

2.4.4. Le premier bras télé-opéré

C'est en Lorraine en 1954, à Argonne qu'un chercheur du nom de Raymond Goertz a l'idée de rationaliser les bras de télé-opération en assujettissant leurs articulations à des moteurs

électriques. Grâce à un tel système, l'opérateur peut désormais se trouver à plusieurs centaines de mètres du lieu [14]

2.5. Les types des robots

Il existe deux grandes familles de robots sont :

- Les robots manipulateurs.
- Les robots mobiles.

2.6. Classification des robots

On a 3 types de robot :

A) Les manipulateurs

- Les trajectoires sont non quelconques dans l'espace,
- Les positions sont discrètes avec 2 ou 3 valeurs par axe,
- La commande est séquentielle.

B) Les télémanipulateurs

Les appareils de manipulation à distance (pelle mécanique, pont roulant), apparus vers 1945 aux USA :

- Les trajectoires peuvent être quelconques dans l'espace,
- Les trajectoires sont définies de manière instantanée par l'opérateur, généralement à partir d'un pupitre de commande (joystick).

C) Les robots

- Les trajectoires peuvent être quelconques dans l'espace,
- L'exécution est automatique,
- Les informations extéroceptives peuvent modifier le comportement du robot. [15]

2.7. Programmation des robots classiquement

Deux étapes sont utilisées pour faire en sorte qu'un robot connaisse la tâche à exécuter.

* L'apprentissage

- Enregistrement dans une mémoire de la trajectoire à exécuter, sous contrôle d'un opérateur humain,

- **Pantin** : Structure mécanique identique à celle du robot, qui est déplacée et qui mémorise les points "pertinents",

- **Syntaxeur** : Un manche de pilotage (joystick) commande les déplacements de l'organe terminal,

- **Boîte à boutons** : Un interrupteur par actionneur.

*La génération de trajectoires et les opérations à réaliser le long de ces trajectoires, ce qui permet de définir la tâche à réaliser :

On fait appel à un logiciel qui, à partir du modèle du robot, et des trajectoires à réaliser, élabore la succession des commandes des actionneurs. Les langages de programmation les plus courants sont : WAVE, VAL (Unimate), LM (Hitachi).[15]

2.8. Classification des robots

Il existe plusieurs façons pour classer les robots, par exemple, sur la base de leur source d'énergie, sur la base de l'environnement ou sur la base du mouvement.

Mais souvent, on classe structurellement les robots en fonction des systèmes de coordonnées dans lesquels ils travaillent :

Cartésien : trois axes de translation.

Cylindrique : deux axes de translation, un axe de rotation.

Sphérique : un axe de translation, deux axes de rotation.

Articulé : trois axes de rotation. [16]

2.9. Le robot industriel

Le terme robot a été introduit pour la première fois par l'écrivain tchèque Kapek dans sa pièce R.U.R (Rosum's Universal Robot) en 1920. Il est dérivé du terme Robota, qui signifie (force) travailleur. Un robot est un dispositif mécanique (combinant mécanique, électronique et traitement de données) permettant d'effectuer automatiquement diverses tâches. C'est une machine intelligente fonctionnelle qui nécessite un mouvement autonome. L'Organisation internationale de normalisation définit un robot comme un manipulateur à plusieurs degrés de liberté, à commande automatique, reprogrammable, multi-applications, mobile ou non mobile, et est destiné aux applications d'automatisation industrielle.

Structure générale d'un système robotique

Le terme robot a été introduit pour la première fois par l'écrivain tchèque Kapek dans sa pièce "RUR" en 1920. Ceux que nous avons détaillés ci-dessous.

Domaine d'utilisation. [17]

Ces dernières années, la robotique a été un domaine en plein essor. Le développement technologique continue de dépasser nos attentes, et il est désormais possible de réaliser des solutions techniques qui s'adaptent aux moindres problèmes. Par conséquent, la robotique est utilisée dans des domaines extrêmement stricts et exigeants. Nous explorerons ces différents domaines :

2.9.1. Industrie

Les robots sont largement utilisés dans l'industrie et continuent d'effectuer des tâches strictement répétitives dans l'industrie. Le premier objectif des robots est de remplacer les humains dans des activités dangereuses, pénibles, fastidieuses ou coûteuses pour les employeurs. Par conséquent, les robots ont commencé à être utilisés dans les chaînes de montage industrielles. [17]



Figure 2.4: L'utilisation de robots dans l'industrie automobile

2.9.2. Domaine militaire

Les robots militaires sont des robots autonomes ou télécommandés conçus pour des applications militaires. En effet, la miniaturisation d'aujourd'hui permet de fabriquer des robots discrets, mais équipés de nombreux capteurs, ils sont idéaux pour des tâches d'espionnage ou d'éclairage. [17]



Figure 2.5: L'utilisation de robots dans la domaine des premiers secours pendant les guerres

2.9.3. Santé

Les robots commencent à entrer de plus en plus dans le domaine médical, qu'il s'agisse de "simple" échographie ou de chirurgie plus sophistiquée. En effet, ces robots ne sont pas totalement autonomes, mais ils peuvent assister les médecins ou les chirurgiens et même permettre des opérations de télé-médecine (télé-médecine). Nous avons parlé de chirurgie (de

l'anglais « chirurgie » : chirurgie), c'est-à-dire tout ce qui introduit les derniers outils informatiques et robotiques dans la pratique de la chirurgie médicale. Cette approche de « chirurgie assistée » est en train d'émerger, si elle n'est pas courante, elle devient la chirurgie du futur. [17]

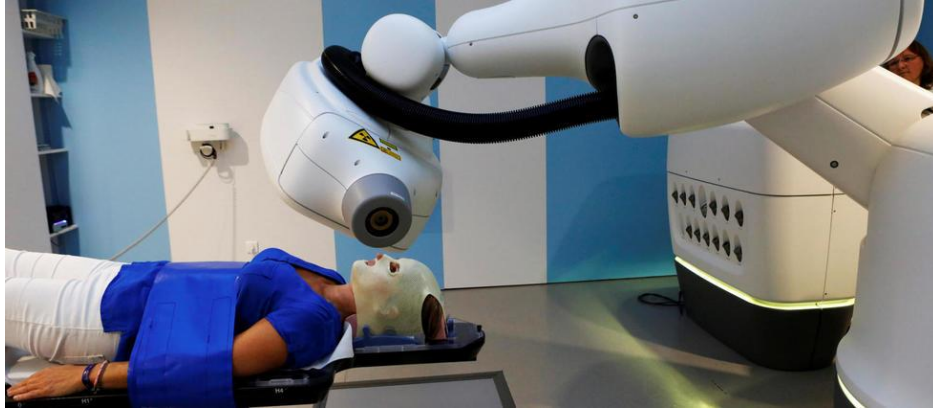


Figure 2.6: L'utilisation de robots dans la domaine du diagnostic médical

2.10. Conclusion

La robotique est très importante dans l'industrie, et la programmation est essentielle. Cette dernière montre les performances de nos machines. Dans ce chapitre, nous expliquons la généralité de la robotique et de ses opérations.

Chapitre 3

Chapitre 3: Application à la commande en poursuite d'un robot

3.1. Introduction

Dans ce chapitre, le problème de commande d'un système non linéaire incertain perturbé est formulé. La définition du système flou adaptatif utilisé est présentée dans la suite. La mise en oeuvre de l'approche proposée est donnée par les équations suites. Les lois d'adaptation et la stabilité du système sont déduites de la synthèse de Lyapunov. Cette commande développée sera appliquée pour la poursuite d'un bras de robot à deux degrés de liberté.

3.2. Position du problème

On considère un système non linéaire, d'ordre n , ayant p entrées et m sorties, donné sous sa forme canonique [1]

$$\begin{cases} \dot{x}^{(n)} = F(x) + G(x)U + d \\ y = x \\ X = [x^T, \dot{x}^T, \dots, (x^{(n-1)})^T]^T \end{cases} \quad (3.1)$$

F et G sont des fonctions non linéaires incertaines supposées bornées. U et y représentent, respectivement, les entrées et les sorties du système et d est le vecteur des perturbations externes. X est le vecteur d'état du système supposé mesurable. Sans nuire à la généralité, on peut considérer le nombre d'entrées égal au nombre de sorties ($p=m$). Le système de référence, qui peut correspondre au système étudié dans les conditions nominales, est écrit sous la forme suivante :

$$\begin{cases} \dot{x}^{(n)} = F_0(X) + G_0(X)U \\ y = x \end{cases} \quad (3.2)$$

Pour une trajectoire de référence donnée y_r l'erreur de poursuite est :

$e = y_r - y$ On définit le vecteur d'erreur $E = \left[e^T, \dot{e}^T, \dots, (e^{(n-1)})^T \right]^T$

Si l'on applique une commande, comme proposée dans la référence [9]

$$U = G_0^{-1} \left[-F_0 + y_r^{(n)} + K^T E \right] \quad (3.3)$$

avec K la matrice des gains de retour d'état calculée de façon que le polynôme

$H(s) = s^n + k_n s^{n-1} + \dots + k_1$ ait des racines à partie réelle négative. L'erreur de poursuite tendra vers zéro. Notre système (3.1) peut être réécrit sous la forme :

$$\dot{x}^{(n)} = (F_0(X) + \Delta F) + (G_0(X) + \Delta G)U + d \quad (3.4)$$

où ΔF et ΔG sont les incertitudes inconnues sur F et G , respectivement. La loi de commande à mettre en œuvre doit aussi atténuer les perturbations externes et s'écrit alors :

$U = G_0^{-1} \left[-F_0 + y_r^{(n)} + K^T E - U_h \right]$, où U_h est une commande de type H_∞ . L'équation dynamique de l'erreur de poursuite est donnée par :

$$\dot{e}^{(n)} = -K^T E - (\Delta F + \Delta G U) + U_h - d \quad (3.5)$$

ΔF et ΔG étant inconnues. Il est difficile d'obtenir la convergence asymptotique de l'erreur de poursuite vers zéro avec cette commande. L'idée est d'approcher le terme $(\Delta F(x) + \Delta G(x)U)$ et l'éliminer par une commande U_f . Le signal U_h doit en plus atténuer l'effet des erreurs d'approximation à un niveau donné. La loi de commande modifiée est donc :

$$U = G_0^{-1}(x) \left(-F_0(x) + y_r^{(n)} + K^T E - U_f - U_h \right) \quad (3.6)$$

Pour générer la commande U_f , on utilisera un système flou adaptatif.

3.3. Système flou (F.L.S.)

D'après le théorème de l'approximateur universel énoncé par Wang [5] : Pour toute fonction réelle φ , continue sur un ensemble compact $U_c \in R^n$ et $\varepsilon \geq 0$, il existe un système flou $\hat{\varphi}$ tel que $\sup_{x \in U_c} |\hat{\varphi}(x) - \varphi(x)| < \varepsilon$. Un système flou, avec le singleton pour la « fuzzyfication », le produit comme moteur d'inférence et le centre de gravité pour la « defuzzyfication », peut être écrit sous la forme suivante:

$$\hat{\varphi}_i(X, \theta) = \frac{\sum_{k=1}^M \theta_{ik} (\prod_{j=1}^{p_n} \mu_{I_j}^k)}{\sum_{k=1}^M (\prod_{j=1}^{p_n} \mu_{I_j}^k)} \text{ où } \theta_{ik} \text{ est le centre de } O^k \text{ (le centre de l'ensemble de la partie}$$

conséquence de la $k^{\text{ème}}$ règle floue) et $\mu_{I_j}^k$ degré d'appartenance de la $J^{\text{ème}}$ entrée. Sous forme

matricielle, on peut écrire : $\hat{\varphi}_i(X, \theta) = \zeta_i^T(x) \cdot \theta_i, 1 \leq i \leq 2$ ou $\zeta_i(x) = [\zeta_{i1}(x), \dots, \zeta_{Mi}(x)]^T$

$$\text{Et } \zeta_{ki}(x) = \frac{(\prod_{j=1}^{p_n} \mu_{I_j}^k)}{(\prod_{j=1}^{p_n} \mu_{I_j}^k)} \text{ et } \theta_i = [\theta_{i1}, \dots, \theta_{Mi}] \text{ On peut ainsi écrire : } \hat{\varphi}(x, \theta) = \xi^T \theta,$$

$$\text{avec } \xi = \begin{bmatrix} \xi_1 & 0 & \dots & 0 \\ 0 & \dots & 0 & \dots \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & \xi_p \end{bmatrix} \text{ et } \theta = [\theta_1 \dots \theta_p]^T.$$

D'où l'expression :

$$U_f = \xi^T(X|\theta) \cdot \theta. \quad (3.7)$$

Pour que le système flou U_f joue son rôle de façon optimale, il faudrait considérer un nombre infini de règles floues. Or, la complexité de calcul et d'implémentation ne le permet pas. Pour remédier à ce problème, on va fixer une partie du système floue (ξ) et ajuster le vecteur θ en ligne suivant une loi d'adaptation qu'on va présenter dans ce qui suit. Un système flou adaptatif est un système à base de logique flou équipé d'un algorithme d'apprentissage. L'avantage principal de celui-ci est la possibilité d'intégrer les informations floues provenant de l'expert humain.

3.4. Mise en œuvre de la commande

Dans cette partie, nous développons de façon analytique, une loi d'adaptation pour ajuster le vecteur des paramètres θ ainsi que la loi de commande U_h permettant d'atténuer l'effet des

résidus d'élimination et des perturbations externes. En utilisant (3.7), l'équation (3.5) peut être réécrite sous la forme :

$$e^{(n)} = -K^T E - [U_f - (\Delta F + \Delta G U)] + U_h - d \quad (3.8)$$

qu'on peut écrire sous forme matricielle :

$$\dot{E} = AE + B[U_f - (\Delta F + \Delta G U) + U_h - d] \quad (3.9)$$

$$\text{Ou } A = \begin{bmatrix} 0 & I & \dots & 0 \\ \vdots & \vdots & \dots & 0 \\ 0 & & 0 & \dots & I \\ -K_n & -K_{n-1} & \dots & -K_1 \end{bmatrix} \text{ et } B = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}$$

On définit l'erreur minimale d'approximation par :

$$w_a = \xi^T \theta^* - (\Delta F + \Delta G) \quad (3.10)$$

où θ^* représente la valeur optimale de l'approximation de θ .

Ainsi, l'équation (3.9) devient

$$\dot{E} = AE + B[\xi^T \tilde{\theta} + w_a + U_h - d] \quad (3.11)$$

avec $\tilde{\theta} = \theta - \theta^*$

et A étant une matrice stable. On peut donc lui associer une équation algébrique de Riccati

$$A^T + PA + Q - 2PBR^{-1}B^T P + \frac{1}{2}PBB^T P = 0 \quad (3.12)$$

ayant une solution symétrique définie positive P, où ρ est le facteur d'atténuation.

Afin d'étudier la stabilité, on utilise le premier théorème de Lyapunov. Pour cela on considère la fonction

$$V = \frac{1}{2}E^T P E + \frac{1}{2\gamma} \tilde{\theta}^T \tilde{\theta} \quad (3.13)$$

avec γ est un facteur positif de pondération. La dérivée temporelle de V le long de la trajectoire de (3.9) est donnée par :

$$\dot{V} = \frac{1}{2}E^T P E + \frac{1}{2\gamma} (\tilde{\theta}^T \dot{\tilde{\theta}} + \dot{\tilde{\theta}}^T \tilde{\theta}) \quad (3.14)$$

En substituant \dot{E} dans (3.14), on trouve

$$\begin{aligned} 2\dot{V} &= (AE + B[\xi^T \tilde{\theta} + w_a + U_h - d])^T P E + E^T P (AE + B(\xi^T \tilde{\theta} + w_a + U_h - d)) + \frac{2}{\gamma} \dot{\tilde{\theta}} \tilde{\theta} \\ 2\dot{V} &= E^T (A^T P + PA)E + 2E^T P B(\xi^T \tilde{\theta} + w_a + U_h - d) + \frac{2}{\gamma} \dot{\tilde{\theta}} \tilde{\theta} \end{aligned} \quad (3.15)$$

En utilisant l'équation (3.12), cette dernière devient

$$2\dot{V} = -E^T QE + E^T PB(R^{-1}B^T PE + U_h) - \frac{1}{\rho^2} E^T PBB^T PE + E^T PB - \rho(w_a - d) \quad (3.16)$$

Si l'on choisit la commande U_h et la loi d'adaptation

$$\begin{cases} U_h = -BPR^{-1}E \\ \dot{\theta} = -\gamma\xi B^T PE \end{cases} \quad (3.17)$$

L'équation (3.16) devient après simplification

$$\dot{V} = -E^T QE + \rho^2(w_a - d)^T (w_a - d) - \left[\frac{1}{\rho} E^T PB - \rho(w_a - d) \right]^T \left[\frac{1}{\rho} E^T PB - \rho(w_a - d) \right] \quad (3.18)$$

Ainsi

$$\dot{V} \leq -\frac{1}{2} E^T QE + \frac{\rho^2}{2} (w_a - d)^T (w_a - d) \quad (3.19)$$

Si on intègre cette équation entre $t = 0$ et tr , le temps de réponse du système, on obtient le critère H_∞

$$\int_0^{tr} QE dt \leq E^T(0)PE(0) + \frac{1}{\gamma} (\tilde{\theta}^T(0)\tilde{\theta}(0)) + \rho^2 \int_0^{tr} (w_a - d)^T (w_a - d) dt \quad (3.20)$$

Remarques

- Pour garantir l'existence de la solution de l'équation de Riccati (3.12), il faut que la condition $2\rho^2 I \geq R$ soit satisfaite.
- Dans le cas où l'on prendrait en compte les variations sur la commande U_h , le critère (3.20) est modifié et sera

$$\int_0^{tr} (E^T QE + U_h^t R U_h) dt \leq E^T(0)PE(0) + \frac{1}{\gamma} (\tilde{\theta}^T(0)\tilde{\theta}(0)) + \rho^2 \int_0^{tr} (w_a - d)^T (w_a - d) dt$$

avec l'équation de Riccati correspondante

$$A^T + PA + Q - 2PBR^{-1}B^T P + \frac{1}{2} PBB^T P = 0$$

La condition de l'existence de la solution de cette équation dépend de la condition : $\rho^2 I \geq R$.

Le schéma de commande de l'approche proposée est donné par la figure 3.1.

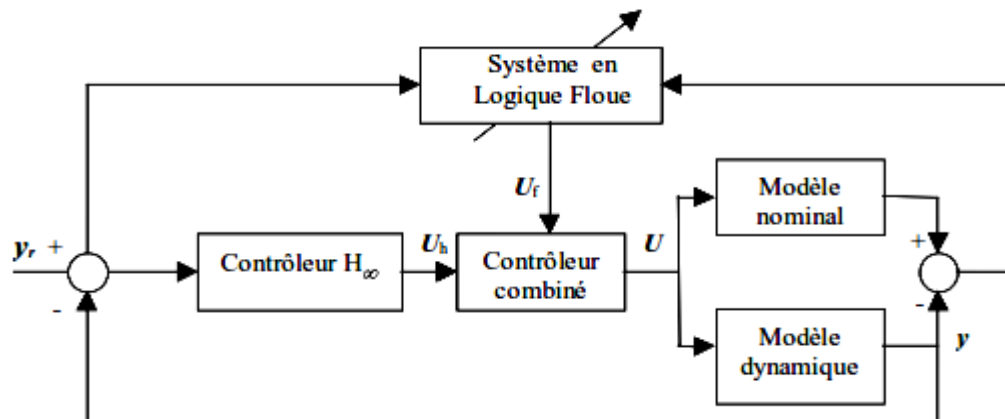


Figure 3.1 : Le schéma de commande de l'approche proposée.

Ainsi, la mise en œuvre de l'approche proposée se fait en deux phases. 1ère phase (hors ligne)

- Fixer le niveau d'atténuation désiré ρ ,
- donner K et Q ,
- déduire A et calculer P ,
- définir les fonctions d'appartenance du système flou adaptatif,
- fixer le facteur de pondération γ et
- choisir la valeur initiale du vecteur des paramètres θ . 2ème phase (en ligne)
- calculer la matrice de régression ξ ,
- mettre à jour le vecteur θ en intégrant) l'équation (3.17),
- calculer U_f et U_h et générer la commande globale U .

3.5. Simulation et résultats

Pour valider notre approche, nous considérons un bras de robot, à deux articulations, de masses m_1 et m_2 et de longueurs l_1 et l_2 , actionnées par deux moteurs à courant continu, comme indiqué sur la figure 3.2.

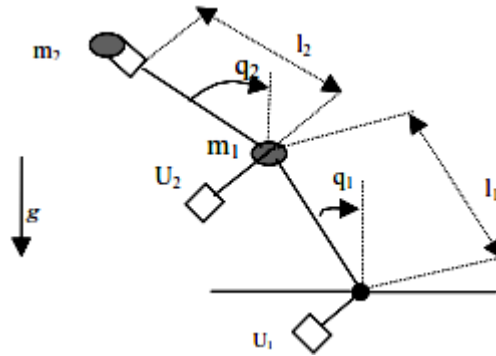


Figure 3.2 : Bras de robot à 2 degrés de liberté.

soit $q = [q_1 \ q_2]^T$ le vecteur des angles de rotation des articulations par rapport à la verticale. En partant de l'équation de Lagrange [3.10], l'équation dynamique du système peut s'écrire sous la forme matricielle suivante

$$M'(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = r + d \quad (3.21)$$

où \dot{q} \ddot{q} sont, respectivement, les vecteurs vitesse et accélération angulaires, $M'(q) \in \mathfrak{R}^{2 \times 2}$ est la matrice des moments d'inertie, symétrique et définie positive

$$M'(q) = \begin{pmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{pmatrix}$$

Avec $M_{11} = (m_1 + m_2)l_1^2$

$$M_{12} = M_2 l_1 l_2 (\sin(q_1) \sin(q_2) + \cos(q_1) \cos(q_2))$$

$$M_{22} = M_2 l_2^2$$

$C(q, \dot{q}) \in \mathfrak{R}^{2 \times 2}$ est la matrice de Coriolis, avec

$$C(q, \dot{q}) = M_2 l_1 l_2 (\cos(q_1) \sin(q_2) + \sin(q_1) \cos(q_2)) \times \begin{bmatrix} 0 & -\dot{q}_2 \\ -\dot{q}_1 & 0 \end{bmatrix}$$

$G(q) \in \mathfrak{R}^2$ est la matrice gravité, avec $G(q) = \begin{bmatrix} -(m_1 + m_2)l_1 g \sin(q_1) \\ -m_2 l_2 g \sin(q_2) \end{bmatrix}$

$\Gamma \in \mathfrak{R}^2$ est le vecteur des couples moteur, et $d \in \mathfrak{R}^2$ celui des perturbations externes.

L'équation électrique des deux moteurs peut s'écrire sous la forme matricielle suivante

$$RI + L \frac{di}{dt} + K_b \frac{dq_m}{dt} = U \quad (3.22)$$

avec $R = \begin{pmatrix} R1 & 0 \\ 0 & R2 \end{pmatrix}$, $I = \begin{pmatrix} i_1 \\ i_2 \end{pmatrix}$, $L = \begin{pmatrix} L1 & 0 \\ 0 & L2 \end{pmatrix}$, $U = \begin{pmatrix} U1 \\ U2 \end{pmatrix}$, $K_b = \begin{pmatrix} K_{b1} & 0 \\ 0 & K_{b2} \end{pmatrix}$ est la matrice

des coefficients des forces contre-électromotrices ; et $q_m = N q$, le vecteur des angles de

rotation mécanique des moteurs. N la matrice des coefficients de réduction diagonale d'ordre 2 définie positive.

La relation entre les couples développés et les courants est donnée par

$$\Gamma = NK_t I \quad (3.23)$$

où

$$\Gamma = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}, K_t = \begin{pmatrix} K_{t1} & 0 \\ 0 & K_{t2} \end{pmatrix} \text{ la matrice des constantes des couples.}$$

En substituant l'équation (3.23) dans l'équation (3.22), on obtient

$$\underbrace{R(NK_t)}_{R_n} \Gamma + \underbrace{L(NK_t)}_{L_2} \dot{\Gamma} + K_b \frac{dq_m}{dt} = U,$$

Qui peut s'écrire

$$R_n \Gamma + L_n \dot{\Gamma} + k_{bn} \frac{dq}{dt} = U \quad (3.24)$$

, ou $k_{bn} = k_b N$

En substituant l'équation (3.21) dans l'équation (3.24), nous obtenons

$$M(q)q^3 + D(q, \dot{q}, \ddot{q}) = U + R_n d + L_n \dot{d} \quad \text{ou } M(q) = L_N M'(q) \quad (3.25)$$

et

$$D(q, \dot{q}, \ddot{q}) = [R_n M'(q) + L_N (M'(q) + C(q, \dot{q}))] \ddot{q} + [R_n C(q, \dot{q}) + L_N \dot{C}(q, \dot{q}) + K_{bn}] \dot{q} + R_n G(q) + L_n \dot{G}(\dot{q})$$

Notons que l'équation (3.25) est équivalente à l'équation (3.1).

Pour la simulation on considère donc un robot avec les caractéristiques suivantes : $m_1 = m_2 = 4 \text{ Kg}$, $l_1 = l_2 = 0,5 \text{ m}$.

Les rotations sont assurées par deux moteurs à courant continu, de résistances

$R_1 = R_2 = 1.5 \Omega$, d'inductances $L_1 = L_2 = 0.05 \text{ H}$, de coefficients de forces contre électromotrices $k_{b1} = k_{b2} = 0.2$ et de constantes de couples $k_{t1} = k_{t2} = 0.3$.

Les facteurs de transmission sont respectivement $N_1 = 60$ pour le premier moteur et $N_2 = 100$ pour le second.

On suppose que le système présente : des incertitudes sur les masses des bras de la forme

$$d_{m1} = d_{m2} = 0.1 \sin(2t) , \text{ et des perturbations externes } d_1 = 0.1 \sin(2t) \text{ et } d_2 = 0.05 \sin(2t).$$

Le robot est supposé au repos et la position initiale des articulations est $q_1 = q_2 = 1.2 \text{ rad}$.

Suivant les étapes de mise en œuvre définies précédemment, on procède comme suit :

Afin de montrer l'influence du facteur d'atténuation, on a choisi $\rho=0.1$, puis égal à 0.8.

$K_1=\text{diag}(5,8)$, $K_2=\text{diag}(10,14)$, et $K_3=\text{diag}(9,7)$. $Q=\text{diag}(100,100,10,10,1,1)$. Pour simplifier les calculs, on a pris $R=2.\rho I(2 \times 2)$. La résolution de l'équation de Riccati a donné

$$p = \begin{bmatrix} 150.4412 & 0.0000 & 95.8824 & 0.0000 & 10.0000 & 0.0000 \\ 0.0000 & 118.1889 & 0.0000 & 48.7056 & 0.0000 & 6.2500 \\ 95.8824 & 0.0000 & 92.5588 & 0.0000 & 10.0882 & 0.0000 \\ 0.0000 & 48.7056 & 0.0000 & 29.2750 & 0.0000 & 3.8361 \\ 10.0000 & 0.0000 & 10.0882 & 0.0000 & 1.1765 & 0.0000 \\ 0.0000 & 6.2500 & 0.0000 & 3.8361 & 0.0000 & 0.6194 \end{bmatrix}$$

Pour, le système flou, on a pris sept fonctions d'appartenance gaussiennes pour chaque entrée e, \dot{e}, \ddot{e}

$$u_{IK}^1(X_K) = \exp \left[- \left(\frac{X_k + 3a_k}{b_k} \right)^2 \right], \quad u_{IK}^2(X_K) = \exp \left[- \left(\frac{X_k + 2a_k}{b_k} \right)^2 \right],$$

$$u_{IK}^3(X_K) = \exp \left[- \left(\frac{X_k + a_k}{b_k} \right)^2 \right], \quad u_{IK}^4(X_K) = \exp \left[- \left(\frac{X_k}{b_k} \right)^2 \right],$$

$$u_{IK}^5(X_K) = \exp \left[- \left(\frac{X_k - a_k}{b_k} \right)^2 \right], \quad u_{IK}^6(X_K) = \exp \left[- \left(\frac{X_k - 2a_k}{b_k} \right)^2 \right],$$

$$u_{IK}^7(X_K) = \exp \left[- \left(\frac{X_k - 3a_k}{b_k} \right)^2 \right],$$

avec $a_k=0.5, 0.5, 1, 1, 2, 2$ et $b_k=10, 10, 20, 20, 30, 30$

pour $k=1, \dots, 6$.

Afin de simplifier l'implémentation, seule les règles correspondant à la diagonale ont été retenues à savoir 14 règles.

$$si(x_1 EST I_1^K) ET (X_3 EST I_2^K) ET (X_5 EST K_3^K) ALORS (U_{f1} EST O^K)$$

$$si(x_2 EST I_2^K) ET (X_4 EST I_4^K) ET (X_6 EST K_6^K) ALORS (U_{f2} EST O^K)$$

pour $k=1, \dots, 7$.

Le facteur de pondération est choisi suivant le nombre de règles et l'efficacité du système flou.

Dans cet exemple, on a $\gamma=200$ et les valeurs initiales des paramètres ajustables ont été initialisées à zéro ($\theta(0)=0$).

Pour la résolution de l'équation de Riccati et l'intégration, avec un pas de 0.01s, on a utilisé les fonctions «Are » et «Ode45 » de Matlab.

Le choix de la position initiale du robot très loin de la trajectoire désirée, ainsi que les grandes valeurs des perturbations ont été choisies à bon escient pour évaluer les performances de notre algorithme.

Les résultats présentés sur les figures 3.3 à 3.8 sont obtenus pour $\rho=0.1$. Les figures 3.3 et 3.4 montrent la poursuite au niveau des deux articulations q_1 et q_2 . Les figures 3.5 et 3.6 montrent les tensions appliquées aux deux moteurs. La convergence des erreurs quadratiques est satisfaisante comme montré dans les figures 3.7 et 3.8.

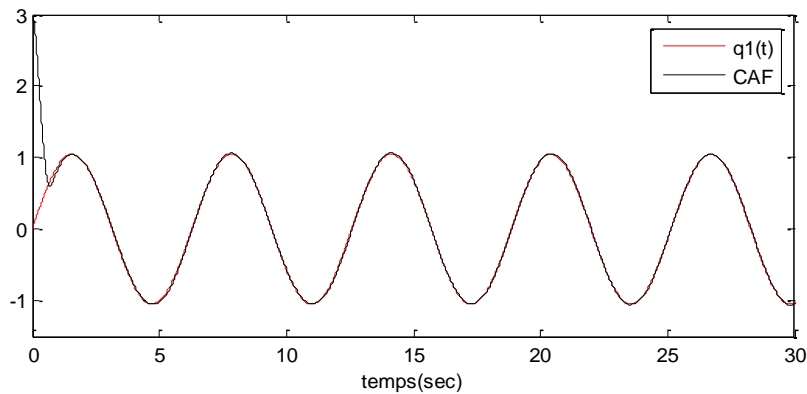


Figure 3.3 : La réponse de $q_1(t)$ (trait plein) et la valeur désirée $q_1(t)=\sin(t)$ (trait pointillé).

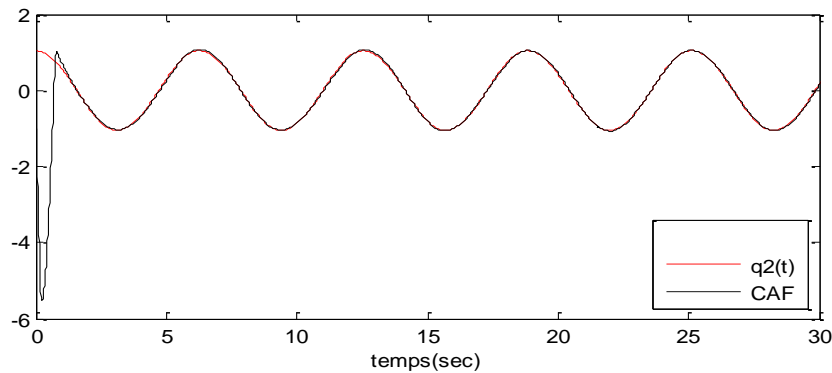


Figure 3.4 : La réponse de $q_2(t)$ (trait plein) et la valeur désirée $q_2(t)=\cos(t)$ (trait pointillé).

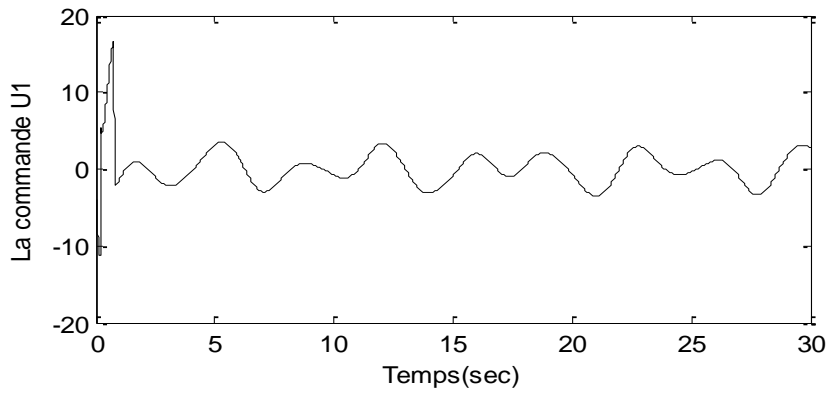


Figure 3.5 : La tension U_1 au niveau de la 1^{ème} articulation).

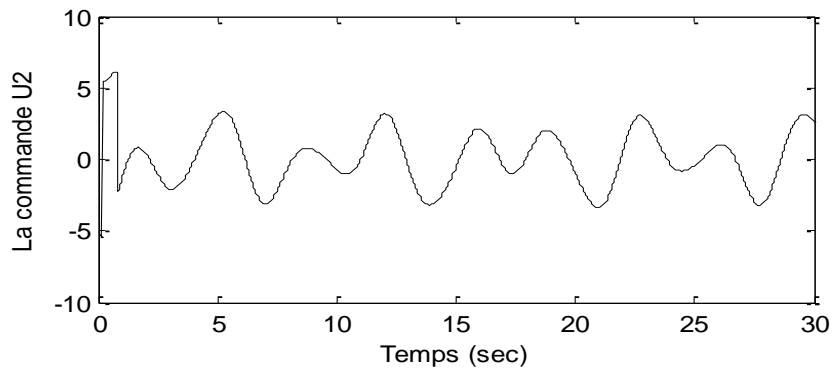


Figure 3.6 : La tension U_2 au niveau de la 2^{ème} articulation).

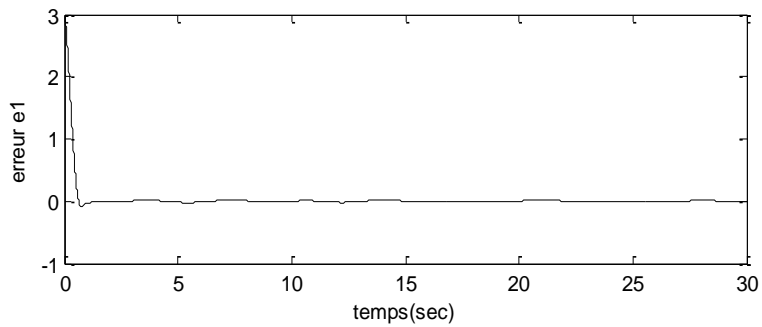


Figure 3.7 : L'erreur quadratique au niveau de la 1^{ère} articulation

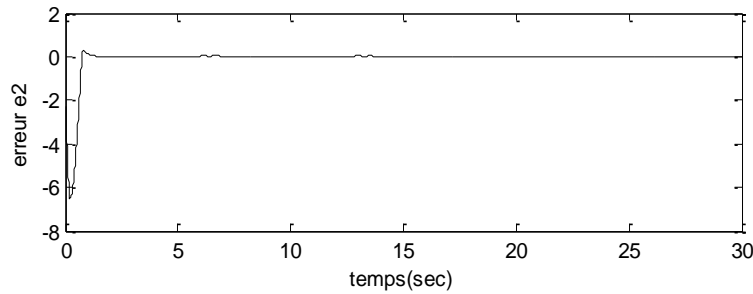


Figure 3.8 : L'erreur quadratique au niveau de la 2^{ème} articulation).

3.6. Conclusion

Dans ce chapitre, nous avons réalisé la poursuite des systèmes non linéaires, incertains soumis aux perturbations externes et traiter le problème de broutement. Nous avons proposée une commande adaptative floue en utilisant un système flou pour approximer la dynamique du système. La loi de commande synthétisée a permis de remédier aux inconvénients de la connaissance de la dynamique de système ainsi que des bornes supérieures des perturbations.

L'approche considérée précédemment est développée pour commander les systèmes multi variables (MIMO). Une application a été présentée pour valider l'approche développée et montrer la rapidité de convergence, la performance de poursuite et la robustesse du système en boucle fermée.

3.7. Conclusion générale

Dans ce mémoire, nous avons présenté une commande adaptative floue robuste utilisant un système flou pour éliminer les incertitudes sur le système dynamique par rapport à son modèle nominal. L'effet des résidus d'élimination et des perturbations externes est atténué à un niveau donné. Ainsi, nous avons combiné l'intelligence des systèmes flous et l'efficacité de la technique H_∞ . Cette commande est facile à mettre en œuvre puisqu'elle ne nécessite qu'un seul système flou. Contrairement à la méthode proposée dans la référence [7] où deux systèmes flous sont utilisés. Le temps de calcul en est ainsi réduit.

Des règles linguistiques émanant de l'expert peuvent être directement incorporées dans le contrôleur. Ceci permettra d'améliorer le temps de convergence de l'algorithme. Néanmoins, certaines hypothèses doivent être satisfaites à savoir une connaissance parfaite d'un modèle de référence ; le choix du paramètre d'atténuation doit être de sorte que l'équation de Riccati existe. Plus ρ est petit, plus la commande en régime transitoire est importante ; il faudrait chercher un compromis entre ρ et les sollicitations requises en entrées.

L'une des informations les plus importantes que nous avons prises était de savoir comment traiter les systèmes à travers la logique floue ainsi que leur contrôle. Nous avons abordé les algorithmes utilisés dans la méthode de contrôle de la logique floue.

Nous avons ensuite parlé de l'histoire des robots depuis leur création jusqu'à nos jours. Si ce mémo ne l'avait pas été, nous aurions peut-être passé notre vie à ne pas connaître cette quantité d'informations, et l'une des meilleures choses que nous avons trouvées est que certains systèmes robotiques sont originaires en arabe et nous en sommes fiers.

Nous avons atteint la dernière partie et la plus importante en termes d'énorme quantité d'informations que nous avons obtenues de lui. Nous avons appris à connaître de nombreuses techniques compris H_∞ , et nous avons prouvé l'efficacité de cette technologie dans le cas de l'existence d'un système.

3.8. Résumé— L'objectif de ce travail est de résoudre le problème de poursuite d'un système non linéaire incertain avec des perturbations externes. Nous proposons une commande adaptative floue robuste, composée d'un bloc flou permettant d'éliminer les incertitudes du système, et d'un bloc superviseur de type H_∞ pour atténuer à un niveau donné l'effet des perturbations externes et les erreurs d'approximation des incertitudes. Il s'agit de combiner deux techniques afin d'allier l'efficacité de la commande de type H_∞ à l'intelligence de la commande adaptative floue. Cette méthode est facilement généralisable à certains systèmes mécaniques non linéaires ayant une grande incertitude et/ou une non connaissance des variations des paramètres ou de sa structure et soumis à des perturbations externes.

Mots clés— Robot ; Commande robuste ; Système flou ; Poursuite ; Commande adaptative floue ; Critère H_∞ .

Abstract— The purpose of this work is to resolve the problem of tracking of uncertain non linear systems with external disturbances. We propose a control based on a fuzzy adaptive system and a (H_∞) algorithm to impose the performances of tracking. This approach combines the efficiency of H_∞ and the intelligence of the fuzzy adaptive control.

We shall show that the effect of both error of cancellation due to the fuzzy system and to the external disturbances is attenuated at a desired level. This method can be applied to some non linear mechanical systems subject to external disturbances, to uncertainties, or to internal structural variation of its parameters.

Keyword— Robot; Robust Control; Fuzzy System; Tracking; Fuzzy Adaptive Control; H_∞ Criteria.

الملخص— الغرض من هذا العمل هو حل مشكلة تتبع الأنظمة غير الخطية غير المؤكدة باستخدام الاضطرابات الخارجية. نقترح عنصر تحكم يعتمد على نظام تكيفي غامض وخوارزمية (H_∞) لفرض أداء التتبع. يجمع هذا النهج بين كفاءة H_∞ وذكاء التحكم التكيفي الضبابي.

يجب أن نبين أن تأثير كل من خطأ الإلغاء بسبب النظام الضبابي والخارجي تضعف عند المستوى المطلوب. يمكن تطبيق هذه الطريقة على بعض الميكانيكية غير الخطية الأنظمة المعرضة للاضطرابات الخارجية، أو للتغير البنيوي الداخلي لها المعلمات.

الكلمات الرئيسية— روبوت؛ تحكم قوي نظام غامض تتبع التحكم التكيفي الضبابي؛ معايير H_∞ .

3.9.References

- [1]. J.-J. E. Slotine, W. Li, "Applied nonlinear control", Prentice Hall, Englewood Cliffs, NJ, USA, (1991).
- [2]. S. W. Kim, J. J. Lee, J. Contr. Automat," Syst Eng", (ICASE), 1, N° 2, (1995), 88-93.
- [3]. G. Lightbody, Q. W. Irwin, "in Proc. Inst. Elec. Eng.-Contr. Theory Appl", 142 (1995) 3143.
- [4]. H. K. Lam, F. H.F. Leung., P. K. S. Tam, IEEE Trans. On Syst., man and Cyber, Part. A, 30, N° 6 (2000) 825-840.
- [5]. L. X. Wang, "adaptive fuzzy systems and control", Englewood Cliff's NJ: Prentice Hall (1994).
- [6]. A. Hamzaoui, J. Zaytoon, A. El kari, Proceeding I.F.A.C. C.A.O., Saint-Petersburg, (2000) 149-152.
- [7]. A. Hamzaoui, A. El kari, J. Zaytoon, "Robust adaptive fuzzy control application to uncertain non-linear systems in robotics", IMECHE, Journal of Systems and Control Engineering (2001).
- [8]. L. X. Wang, IEEE Trans. on Systems, Man and Cybernetics- part B, 26 N° 5 (1996) 677-691.
- [9]. A. Isidori, "Nonlinear control systems", 2nd Ed. Springer-Verlag. Berlin, Germany (1989).
- [10].J. J. Craig, "Adaptive control of mechanical manipulators", A.-W. Publishing company, (1988).
- [11]. <http://www.ferdinandpiette.com/>
- [12]. <http://thesis.univ-biskra.dz/1152/5/Chapitre%2002.pdf>
- [13].Laghouag Ibrahim,"Commande par logique floue type -2-appliquée à la machine asynchrone", Mémoire master, Contrôle industriel, Université mohamed boudiaf M'sila,JUIN 2016 .
- [14].www.memoireonline.com
- [15].Mooeez Bellaamine,ISTIA, Université Angers Jean-Louis Boimond,Université de Tunis,2016/2017.
- [16].Chebbah Abdelouahab,"commande des Robots Manipulateurs", Mémoire Master, Automatique et système, Univ Jijel,2018-2019 .
- [17].Ahmed HAMOUANI,"Etude et réalisation d'un bras robotique", Mémoire Master, Commande des systèmes, univ Tizi Ouzou,2017 .
- [18].M. SELAMI, "Commande adaptative floue robuste par mode glissant pour les systèmes non linéaires incertains via backstepping", Mémoire Master, Automatique, Univ Ouargla, 2017.

[19].A.F. Benchikh, Y. Oumaya, "Commande adaptative floue robuste basée sur un observateur flou pour système non linaires incertains (SNLI) via backstepping", Mémoire master, Automatique, Univ Ouargla, 2017.

[20].<http://roboticus.org/>