

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

Université Mouloud Mammeri De Tizi-Ouzou



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

**Mémoire de Fin d'Etude
De MASTER ACADEMIQUE**
Spécialité : **Commande des systèmes**

Présenté par
**ASSAHLI Massinissa
SADOUNI Tarik**

Mémoire dirigé par **KHERRAZ Khedoudja**

Thème

**Commande neuro-floue d'un bras
manipulateur**

Mémoire soutenu publiquement le 26 Juin 2014 devant le jury composé de :

M Prénom NOM

Grade, Lieu d'exercice, Président

M Prénom NOM

Grade, Lieu d'exercice, Rapporteur

M Prénom NOM

Grade, Lieu d'exercice, Examineur

M Prénom NOM

Grade, Lieu d'exercice, Examineur

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

Université Mouloud Mammeri De Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

**Mémoire de Fin d'Etude
de MASTER ACADEMIQUE
Spécialité : Automatique**

Présenté par
**Kaddour MOHAND KACI
Mohand ADDELLALI**

Mémoire dirigé par Xavier ROBERT et co-dirigé par Prénoms NOM

Thème

**Ecrire le thème sans modifier ni la police
ni la taille des caractères. Eviter de
modifier l'ensemble de cette page.
N'ajouter ni logo ni rubrique ni couleurs**

Mémoire soutenu publiquement le 24 juin 2014 devant le jury composé de :

M. Nacer BENNAMROUCHE

Pr, UMMTO, Président

M. Abdellah SID ELHAOURI

MA CArade, Lieu d'exercice, Rapporteur

M Prénom NOM

Grade, Lieu d'exercice, Examineur

M Prénom NOM

Grade, Lieu d'exercice, Examineur

Dédicace

Je tiens à dédier ce mémoire :

*A la plus chère pour moi ma mère, pour l'amour quelle
M'a apporté depuis ma naissance.*

*A mon cher père : en connaissance de tous ce qu'il a fait
Pour moi.*

*A mes frères et ma sœur qui mont toujours soutenu et m'ont
encouragé dans ce travail.
Et toute ma famille.*

*A tous mes amis(es) et mes camarades de la faculté de Génie
électrique et d'informatique et tous mes enseignants de
l'université Mouloud MAMMERI de Tizi-Ouzou.*

Je dédie ce travail également à mon binôme et à toute sa famille.

A.MASSINISSA

Dédicace

Je tiens à dédier ce mémoire :

A la mémoire de ma mère. Que ton âme repose en paix.

*A mon cher père, ma chère grand mère : en connaissance de tous
ce qu'ils ont fait pour moi.*

*A mes frères et mes sœurs qui m'ont toujours soutenu et m'ont
encouragé dans ce travail.*

Et toute ma famille.

*A tous mes amis(es) et mes camarades de la faculté de Génie
électrique et d'informatique et tous mes enseignants de
l'université Mouloud MAMMERY de Tizi-Ouzou.*

Je dédie ce travail également à mon binôme et à toute sa famille.

S.TARIK

Remerciements

Nous remercions avant tous le bon Dieu de nous avoir permis de réaliser ce travail.

Nous tenons à exprimer nos vifs remerciements à notre promotrice Mme KHERRAZ pour avoir dirigé ce travail pour son suivi, ses conseils et sa disponibilité à chaque instant.

Notre gratitude et reconnaissance vont s'adresser à tous les enseignants qui ont contribué à notre formation pendant notre cursus universitaire.

Nos remerciements vont s'adresser également au président et membres du jury qui nous feront l'honneur d'évaluer notre travail.

Sans oublier tous les amis(es) et camarades qui ont contribué de près ou de loin à la réalisation de ce modeste projet.

Sommaire

Introduction générale.....	1
----------------------------	---

CHAPITRE I

Modélisation d'un bras manipulateur

I.1 Introduction.....	3
-----------------------	---

I.2 Généralités sur les robots.....	4
-------------------------------------	---

I.2.1 Définition générale.....	4
--------------------------------	---

I.2.2 Historique.....	4
-----------------------	---

I.2.3 Constituants mécaniques de robot.....	5
---	---

I.2.3.1 Unité informationnelle.....	5
-------------------------------------	---

I.2.3.2 Unité opérationnelle.....	5
-----------------------------------	---

I.2.3.3 La structure mécanique articule.....	6
--	---

I.2.3.4 Systèmes actionneurs.....	7
-----------------------------------	---

I.2.3.5 Capteur.....	8
----------------------	---

I.2.3.6 Organe terminal.....	9
------------------------------	---

I.2.3.7 Système de commande et de traitement de l'information.....	9
--	---

I.3 Mouvement de robots.....	9
------------------------------	---

I.3.1 Le degré de liberté.....	9
--------------------------------	---

I.3.2 Espace articulaire.....	9
-------------------------------	---

I.3.3 Espace opérationnel.....	9
--------------------------------	---

I.4 Modélisation d'un système mécanique articule.....	9
---	---

I.4.1 Modélisation géométrique.....	10
-------------------------------------	----

I.4.1.1 Modélisation géométrique directe.....	10
---	----

I.4.1.2 Modélisation géométrique inverse.....	11
---	----

I.4.1.3 Relation entre l'espace articulaire et l'espace de données	
--	--

Cartésiennes de l'effecteur.....	11
----------------------------------	----

I.4.2 Modélisation cinématique.....	12
-------------------------------------	----

I.4.2.1 Modèle cinématique directe.....	12
---	----

Sommaire

I.4.2.2 Modèle cinématique inverse.....	13
I.4.3 Modélisation dynamique.....	13
I.4.3.1 Les formalisme de Lagrange-Euler.....	14
I.4.3.2 Modélisation dynamique d'un bras manipulateur à 2 ddl.....	14
I.5 Conclusion.....	17

CHAPITRE II

Synthèse de la commande des robots manipulateurs

II.1 Introduction.....	18
II .2. Commande par mode glissant.....	18
II.2.1. Synthèse de la loi de commande.....	19
II.2.1.1 Choix de la surface de glissement.....	19
II.2.1.2 Condition d'existence du mode de glissement.....	20
II.2.1.3 Calcul de la commande.....	20
II.2.2 Le broutement (chattering).....	21
II.3 La commande adaptative.....	22
II.3.1 Concept de la commande adaptative.....	22
II.3.2 Commande adaptative à modèle de référence	23
II.3.3 Approche du gradient.....	24
II.3.4 Fonction de Lyapunov.....	25
II.4 Commandes classiques.....	26
II.4.1 Régulateur proportionnel P.....	26
II.4.2 Régulateur proportionnel-intégrateur PI.....	27
II.4.3 Régulateur I pur.....	28
II.4.4 Régulateur proportionnel derivateur PD.....	29
II.4.5 Régulateur PID.....	30
II-5. Conclusion.....	32

Sommaire

CHAPITRE III

Les réseaux de neurones et les systèmes flous

III.1 Introduction.....	33
III.2 Les réseaux de neurones.....	33
III.2.1 Le neurone biologique.....	33
III.2.2 Le neurone formel.....	34
III.2.3 Modélisation d'un neurone formel.....	34
III.2.4 Fonctions d'activation.....	36
III.2.5 Types de réseaux de neurones.....	37
III.2.5.1 Réseaux de neurones non bouclés.....	37
III.2.5.2 Réseaux de neurones bouclés.....	37
III.2.6. Apprentissage dans les réseaux de neurones.....	38
III.2.6.1. Notions d'apprentissage.....	38
III.2.6.2. Types d'apprentissage.....	39
III.2.6.2.1. Apprentissage non supervisé.....	39
III.2.6.2.2. Apprentissage supervisé.....	39
III.2.7. Algorithme de retro-propagation du gradient.....	40
III.3 Les systèmes flous.....	41
III.3.1 Notions de base et connaissances préliminaires.....	42
III.3.1.1 Les variables linguistiques.....	42
III.3.1.2 Les fonctions d'appartenance.....	42
III.3.1.3 Règles d'inférences.....	43
III.3.1.4 Les opérateurs en logique floue.....	43
III.3.2 Structure interne d'un contrôleur flou.....	44
III.3.2.1 Base de connaissance.....	45
III.3.2.1.1. La base de données.....	45
III.3.2.1.2. La base des règles floues.....	45
III.3.2.2 La fuzzification.....	45

Sommaire

III.3.2.3L'inférence.....	46
III.3.2.4La défuzzification.....	46
III.3.3 Modèles d'inférence floue.....	47
III.3.3.1 Système d'inférence floue de type Mamdani.....	47
III.3.3.2 Système d'inférence flou de type Takagi-Sugeno.....	47
III.4 Systèmes Neuro-flous.....	47
III.4.1Définition.....	48
III.4.2Méthodes des combinaisons neuro-floues.....	49
III.4.2.1. Réseau flou neuronal.....	49
III.4.2.2. Système neuronal/flou simultanément.....	49
III.4.2.3. Modèles neuro-flous coopératifs.....	49
III.4.2.4. Modèles neuro-flous hybrides.....	49
III.4.3Architecture del' ANFIS.....	51
III.5 Conclusion.....	55

CHAPITRE VI

Applications et résultats de simulation

IV.1 Introduction.....	56
IV.2 Stratégie de la commande PD.....	56
IV.3 Résultat de simulation du contrôleur PD.....	57
IV.3.1 Sans perturbation.....	57
IV.3.2 Avec perturbation paramétrique.....	60
IV.3.3 Avec perturbation externe et paramétrique.....	62
IV.4 Interprétations des résultats obtenus par le régulateur PD.....	64
IV.5 Stratégie de la commande neuro-floue.....	64
IV.6Résultat de simulation de la commande combinée.....	67
IV.6.1 Sans perturbation.....	67
IV.6.2 Avec perturbation paramétrique.....	69
IV.6.3 Avec perturbation externe et paramétrique.....	71

Sommaire

IV.7 Interprétations des résultats obtenus par commande combinée.....	73
IV.8 Conclusion.....	73
Conclusion générale.....	74

Tables des figures

Table des figures

Figure I.1 Structure fonctionnelle d'un robot.....	5
Figure I.2 Symbole de l'articulation rotoïde.....	6
Figure I.3 Symbole de l'articulation prismatique.....	7
Figure I.4 Exemple d'image réelle d'un robot manipulateur du type SCARA.....	8
Figure I.5 Relation entre l'espace articulaire et l'espace des coordonnées cartésiennes de l'effecteur.....	12
Figure I.6 Robot manipulateur à 2 ddl.....	14
Figure II.1 Différents modes de convergence pour la trajectoire d'état.....	17
Figure II.2 Principe de la commande en mode glissant avec commande équivalente.....	21
Figure II.3 Le phénomène de broutement	22
Figure II.4 commande adaptative à modèle de référence.....	23
Figure II.5 Structure d'une commande adaptative indirecte.....	25
Figure II.6 Schéma fonctionnel du régulateur p.....	27
Figure II.7 Asservissement par régulateur PI.....	27
Figure II.8 Schéma fonctionnel du régulateur PI.....	28
Figure II.9 Schéma fonctionnel du régulateur PD.....	29
Figure II.10 Schéma fonctionnel du régulateur PID.....	31
Figure III.1 Un neurone avec son arborisation dendritique.....	35
Figure III.2 Modèle mathématique d'un neurone formel.....	36
Figure III.3 Les fonctions d'activation les plus utilisées.....	37
Figure III.4 Réseaux de neurones non bouclés.....	38
Figure III.5 Réseaux de neurones bouclés.....	39
Figure III.6 Apprentissage non supervisé.....	40
Figure III.7 Apprentissage supervisé.....	40
Figure III.8 La retro-propagation.....	42
Figure III.9 Variable linguistique.....	43
Figure III.10 Formes usuelles des fonctions d'appartenance.....	44

Tables des figures

Figure III.11 : Structure de base d'un régulateur flou.....	45
Figure III.12 : Principe du système neuro-flou.....	49
Figure III.13 : Principe de fonctionnement d'un Réseau neuro-flou.....	49
Figure III.14 : Différents types des systèmes neuro-flous	51
Figure III.15 : L' Architecture de l' ANFIS.....	53
Figure III.16 : Exemple ANFIS à 2 entrées avec 9 règles.....	55
Figure IV.1 commande PD d'un robot manipulateur.....	56
Figure IV.2 Trajectoire réelle et désirée de l'articulation 1.....	58
Figure IV.3 Trajectoire réelle et désirée de l'articulation 2.....	58
Figure IV.4 Erreur de position et de vitesse de l'articulation 1 et 2.....	59
Figure IV.5 les commandes PD des deux articulations.....	59
Figure IV.6 trajectoire désirée et réelle de l'articulation 1.....	60
Figure IV.7 trajectoire désirée et réelle de l'articulation 2.....	60
Figure IV.8 Erreurs de position et de vitesse de l'articulation 1 et 2.....	61
Figure IV.9 les commandes PD des articulations 1 et 2.....	61
Figure IV.10 trajectoire désirée et réelle de l'articulation 1.....	62
Figure IV.11 trajectoire désirée et réelle de l'articulation 2.....	62
Figure IV.12 Erreurs de positions et de vitesse des deux articulations.....	63
Figure IV.13 commande PD des deux articulations.....	63
Figure IV.15 structure de régulateur neuro flou (2 ^{ème} articulation).....	64
Figure IV.16 Schéma de contrôleur neuro flou + contrôleur PD.....	66
Figure IV.17 la consigne et la commande hybride de la 1 ^{er} articulation.....	67
Figure IV.18 la consigne et la commande hybride d a 2 ^{ème} articulation.....	67
Figure IV.19 les erreurs de position et de vitesse.....	68
Figure IV.20 les commande de chaque articulation.....	68
Figure IV.21 la consigne et la commande avec perturbation paramétrique.....	69

Tables des figures

Figure IV.22 commande Hybride et sa consigne de la 2 ^{ème} articulation.....	69
Figure IV.23 les erreurs de position et de vitesse de chaque articulation.....	70
Figure IV.24 la commande chaque articulation.....	70
Figure IV.25 la Commande Hybride et sa consigne de la 1 ^{er} articulation.....	71
Figure IV.26 la Commande Hybride et sa consigne pour la 2 ^{ème} articulation.....	71
Figure IV.27 les erreurs de position et de vitesse de chaque articulation.....	72
Figure IV.28 les commande de chaque articulation.....	72

Liste des tableaux

Tableau II .1 Résumé des effets respectifs des actions P, I, et D.....	32
Tableau III.1 : Analogie entre le neurone biologique et le neurone formel.....	36
Tableau III.2 : les différentes couches d'un système ANFIS.....	55

*Introduction
générale*

Introduction générale

Introduction générale

Depuis la révolution industrielle, une discipline a marqué l'évolution du monde technologique : la Robotique. L'avènement des robots dans l'industrie a permis de soulager l'homme des travaux répétitifs et difficiles tels que : le déplacement d'objets lourds, les tâches d'assemblages, les microsoudures... etc. Ceci avec plus d'efficacité et de précision.

La compétition incessante dans l'industrie conduit à une nécessaire d'augmentation de la productivité en préservant la qualité et en diminuant le coût de revient des produits. Cependant, les robots manipulateurs existants souffrent encore de faiblesses qui les empêchent de mener à bien certaines tâches et limitent leurs champs d'action.

Ces robots sont en effet rigides, lourds et encombrants ce qui se traduit par une grande énergie consommée et une vitesse d'exécution lente. La consommation peut constituer un point crucial quand l'énergie est limitée, comme c'est le cas dans les applications spatiales.

A cet effet, des recherches universitaires et laboratoires de recherche tentent de développer de nouvelles méthodes et algorithmes de manière à libérer les résultats pour des usages industriels après validation. Les recherches sont appliquées à différents domaines tels que la planification de mouvement, la manipulation, et le contrôle de la position, etc. Mais, les robots réels sont inaccessibles à cause des prix élevés, donc des modèles simulés sont largement demandés par les laboratoires de recherche.

Les simulations et les modélisations sont plus faciles à installer, moins chères, plus rapides et plus pratiques à utiliser. Une simulation de configuration robotique est moins chère que de vrais robots (configurations du monde réel), permettant ainsi une meilleure exploration de la conception. La conception finale peut être vérifiée avant que l'on se lance dans le processus coûteux de la construction d'un prototype.

Être capable de simuler ouvre une large gamme d'options, en aidant à résoudre de nombreux problèmes créatifs. On peut étudier, concevoir, visualiser et tester un objet avant de faire une réalité.

Lorsque la complexité d'un système augmente, notre capacité à faire une description précise et significative de son comportement diminue, le traitement des systèmes complexes nécessite souvent la manipulation d'information vague et imprécise.

Tous ces critères à satisfaire montrent la nécessité d'une bonne loi de commande plus performante et plus robuste qui répond aux exigences.

Introduction générale

Depuis de nombreuses années, les modèles neuro-flous suscitent beaucoup d'intérêt, et la littérature relative à ce sujet est importante. Ces modèles ont eu un succès remarquable dans la description et la commande des systèmes complexes.

L'objectif de notre travail consiste à commander un bras manipulateur rigide à deux axes, en position et en vitesse, en garantissant la stabilité et la robustesse. La loi de commande que nous proposons est une combinaison de deux contrôleurs. Le premier est le contrôleur classique PD, choisi pour sa caractéristique de stabilité, et le deuxième est le contrôleur adaptatif neuro-flou, choisi pour tirer profit des avantages de la logique floue et des réseaux de neurones.

Le travail réalisé et présenté dans ce mémoire s'articule de la façon suivante :

L'objet du premier chapitre est d'apporter quelques définitions de base, décrire les constituants technologiques d'un robot, définir les principaux termes du domaine, d'établir les modèles géométriques et cinématiques et à la fin de donner le modèle dynamique du robot.

Dans le deuxième chapitre, nous allons décrire quelques commandes conventionnelles et modernes à savoir la commande classique, la commande adaptative et la commande par mode glissant.

Dans le troisième chapitre, nous allons présenter et analyser quelques structures intelligentes basées sur les concepts de la logique floue, des réseaux de neurones et des systèmes neuro-flous. Cette étude va nous permettre de choisir le modèle le plus fidèle possible pour atteindre de meilleures performances et une meilleure robustesse pour notre système.

Le quatrième chapitre est consacré à la présentation des lois de commandes choisies, suivies d'une série de simulations pour montrer l'efficacité de la commande proposée et sa robustesse vis-à-vis des perturbations structurées et non structurées.

Enfin, nous terminerons notre thèse par une conclusion générale résumant les différents résultats obtenus ainsi que les principales perspectives de ce travail.

Chapitre I

I.1 Introduction

Depuis quelques décennies, la recherche dans la robotique s'est concentrée presque entièrement sur la commande des robots manipulateurs. Pour commander ou simuler le comportement d'un système mécanique articulé (robot), on doit disposer d'un modèle. Plusieurs types de modélisation sont possibles selon les objectifs, les contraintes de la tâche et les performances recherchées, les modèles mathématiques nécessaires sont :

- les modèles cinématiques directs et inverses qui expriment les vitesses de l'organe terminal en fonction des variables articulaires et inversement.

- les modèles géométriques directes et inverses qui expriment les positions de l'organe terminal en fonction des variables articulaires et inversement.

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

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

Ce premier chapitre introduit tous les termes et concepts de base qu'on utilisera dans la suite de notre travail.

Nous présentons en premier lieu, les notions générales, théoriques et technologiques nécessaires pour une meilleure compréhension du fonctionnement des robots.

En second lieu, nous abordons les modèles mathématiques à savoir les modèles géométriques direct et inverse, ensuite le problème des modèles cinématiques direct et inverse est présenté, il permet de donner la relation entre les vitesses de l'organe terminal et celles des actionneurs. Enfin, nous présentons le modèle dynamique pour permettre d'obtenir théoriquement la loi de commande du mouvement en fonction de la variation des états du mécanisme.

I.2 Généralités sur les robots

I.2.1 définitions générales

La définition que l'on donne actuellement du robot industriel diffère quelque peu selon les pays :

I.2.1.1 Robot Institute of America

« Un robot est un manipulateur reprogrammable à fonctions multiples. Il est conçu pour déplacer des matériaux, des pièces, des outils ou des instruments spécialisés suivant des trajectoires variables programmées, en vue d'accomplir des tâches très diverses » [1].

I.2.1.2 Association Japonaise de Robotique Industrielle

« Tout mécanisme permettant d'effectuer, en tout ou en partie, une tâche normalement réalisée par l'homme » [1].

I.2.1.3 L'association Française de Robotique Industrielle (AFRI)

« Un robot industriel est une machine formée de divers mécanismes comportant divers degrés de liberté, ayant souvent l'apparence d'un ou de plusieurs bras se terminant par un poignet capable de maintenir un outil, une pièce ou un instrument de contrôle. En particulier, son unité de contrôle doit contenir un système de mémorisation, et il peut parfois utiliser des accessoires sensitifs et adaptables qui tiennent compte de l'environnement et des circonstances. Ces machines, ayant un rôle pluridisciplinaire, sont généralement conçues pour effectuer des fonctions répétitives, mais sont adaptables à d'autres fonctions » [1].

I.2.1.4 International Standard Organization (ISO)

« Une machine formée par un mécanisme incluant plusieurs degrés de libertés, ayant souvent l'apparence d'un ou plusieurs bras se terminant par un poignet capable de tenir des outils, des pièces ou un dispositif d'inspection » [2].

I.2.2 Historique

1947 : premier manipulateur électrique télé opéré.

1954 : premier robot programmable.

1961 : apparition d'un robot sur une chaîne de montage de General Motors, premier robot avec contrôle d'effort.

1963 : utilisation de la vision pour commander un robot.

1973 : premier robot mobiles à roues.

I.2.3 Constituants mécaniques d'un robot

« De même qu'un trou n'est défini que par ce qui l'entoure, un robot n'a de sens que dans un environnement qu'il modifie » [4]. C'est pourquoi on peut distinguer les ensembles interactifs dans un robot en fonctionnement [5].

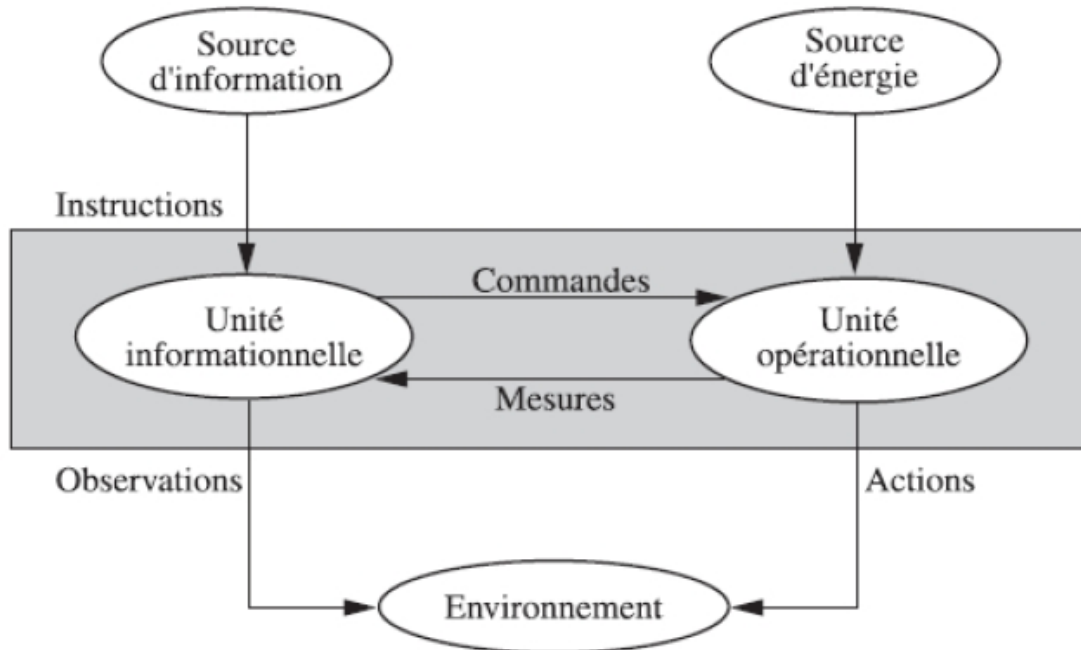


Figure I.1 structure fonctionnelle d'un robot [6]

I.2.3.1 Unité informationnelle

Reçoit les instructions décrivant la tâche à accomplir, les mesures relatives à l'état interne de la structure mécanique qui constitue le bras manipulateur et les observations concernant son environnement. Elle élabore en conséquence les commandes de ses différentes articulations en vue de l'exécution de ses tâches. Les systèmes actuels fonctionnent en interaction permanentent selon le cycle information-décision-action [6].

I.2.3.2 Unité opérationnelle

Exerce les actions commandées en empruntant la puissance nécessaire à la source d'énergie. Cette partie, qui constitue le robot physique, intègre la structure mécanique (segments, articulations, architecture,...), les modules d'énergie (amplificateurs, variateurs, servovalves....), les convertisseurs d'énergie (moteurs, vérins....), les chaînes cinématiques de transmission mécanique (réducteurs, vis à billes, courroies crantées), les capteurs de

proprioceptifs placés sur chaque axe pour mesurer en permanence leur position et leur vitesse, et enfin l'effecteur, ou organe terminal, qui est en interaction avec l'environnement [6].

I.2.3.3 La structure mécanique articulée

Un robot manipulateur est constitué généralement par deux sous-ensembles distincts : un organe terminal qui est le dispositif destiné à manipuler des objets, et une structure mécanique articulée (SMA), constituée d'un ensemble de solides reliés entre eux, généralement les uns à la suite des autres où chaque solide est mobile par rapport au précédent. Cette mobilité s'exprime en termes de degrés de liberté (d.d.l) qui est par définition le nombre de mouvements indépendants possibles d'un solide C_1 par rapport au solide qui lui est directement relié C_2 .

Une structure mécanique articulée peut être représentée par une architecture composée de plusieurs chaînes de corps rigides assemblés par des liaisons appelées articulations. Les chaînes peuvent être dites soit ouvertes ou en série dans les quelles tous les corps ont au plus deux liaisons, ou bien arborescentes où au moins l'un des corps a plus de deux liaisons. Les chaînes peuvent aussi être fermées dans les quelles l'organe terminal est relié à la base du mécanisme par l'intermédiaire de plusieurs chaînes [7].

Notion d'articulation:

Une articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre. Soit m le nombre de degré de liberté résultant, encore appelé mobilité de l'articulation. La mobilité d'une articulation est telle que :

$$0 \leq m \leq 6$$

Lorsque $m = 1$; ce qui est fréquemment le cas en robotique, l'articulation est dite simple: soit trochoïde, soit prismatique. Donc les articulations peuvent être divisées en deux types [3]:

- Articulation rotoïde : Il s'agit d'une articulation de type pivot, notée R , réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe (voir la figure suivante).



Figure I.2 Symbole de l'articulation rotoïde [3].

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



Figure I.3 Symbole de l'articulation prismatique [3].

I.2.3.4 Systèmes actionneurs

Pour être animé, la structure mécanique articulée comporte des moteurs le plus souvent associés à des transmissions (courroies crantées), l'ensemble constitue les actionneurs. Les actionneurs utilisent fréquemment des moteurs électriques à aimant permanent, à courant continu, à commande par l'induit. On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas.

Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique).

Les actionneurs pneumatiques sont d'un usage général pour les manipulateurs à cycles (robots tout ou rien). Un manipulateur à cycles est une structure mécanique articulée avec un nombre limité de degrés de liberté permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air [3]).

I.2.3.4 Capteurs [8]

Lorsque l'on traite des capteurs en robotique, il convient de distinguer deux catégories bien distinctes :

I.2.3.4.1 Capteurs internes (capteurs proprioceptifs)

Il s'agit des capteurs qui délivrent les informations sur l'état (interne) du robot. Pour un robot manipulateur, il s'agit des capteurs de position, vitesse, couple au niveau de chaque articulation. Les capteurs sont utilisés dans la chaîne d'asservissement des actionneurs. Ces capteurs relèvent de technologies classiques : potentiomètres, codeurs, génératrice. Pour un robot mobile, il peut s'agir également des capteurs de cap, vitesse, attitude, les capteurs internes mesurent alors les déplacements du véhicule ; ils jouent un rôle important dans le contrôle de la locomotion et de la navigation relative.

I.2.3.4.2 Capteurs d'environnement (capteurs extéroceptifs)

Les capteurs d'environnement sont encore appelés senseurs ou capteurs externes (par opposition aux capteurs internes). Ils délivrent une information relative à l'environnement (reconnaissance, modèle) ou aux interactions entre le robot et son environnement (position, force). La notion de capteur d'environnement se révèle très spécifique à la robotique, en raison :

- des besoins en perception de l'environnement (de niveau élevé en qualité et performances) ;
- des nouvelles technologies de senseurs, à mettre en œuvre ;
- des algorithmes de modélisation et de contrôle-commande qui sont associés.

I.2.3.5 Organe terminal

On regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, ...), ou à les transformer (outils, torche de soudage, pistolet de peinture, ...). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. Un robot, enfin, peut être multi-bras, chacun des bras portant un organe terminal différent. On utilisera indifféremment le terme organe terminal, préhenseur, outil ou effecteur pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique [3].

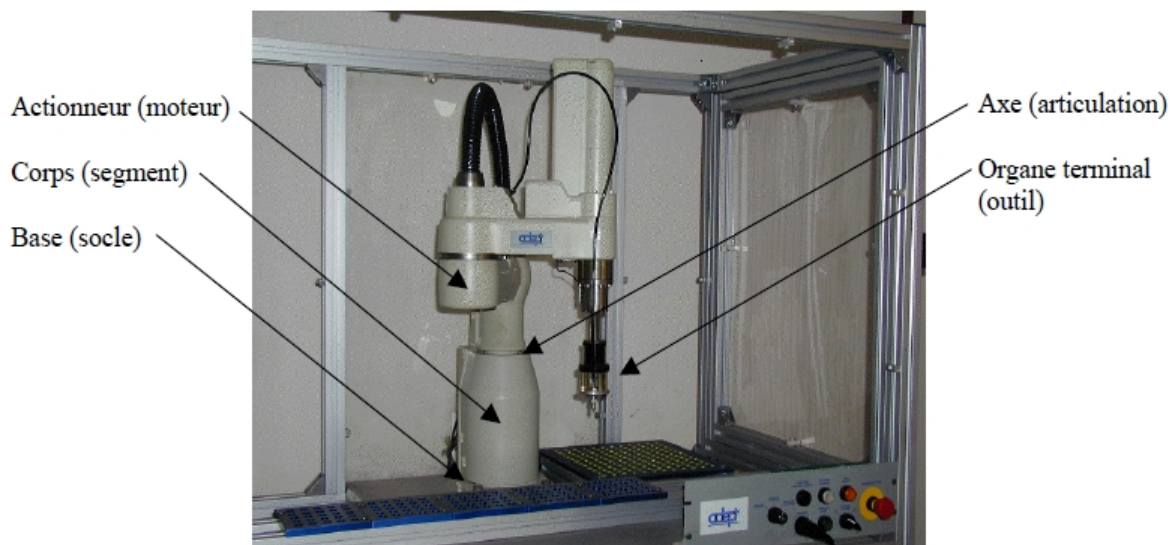


Figure I.4 exemple d'image réelle d'un robot manipulateur du type SCARA[3].

I.2.3.6 Systèmes de commande et de traitement de l'information (cerveau)

Elle assure la gestion et la distribution aux divers actionneurs (génération des signaux de commandes). Le choix des systèmes de commande est guidé par [3]:

- Complexité de la structure mécanique.
- L'ampleur des tâches à réaliser.
- Performances souhaitées.

I.3 Mouvement de robot

I.3.1 les degrés de liberté (d.d.l)

Le nombre de degrés de liberté d'un mécanisme est le nombre de paramètres indépendants qui permettent de définir la position du mécanisme à un instant donné du mouvement [3].

I. 3.2 espaces articulaires (coordonnées généralisées)

On appelle coordonnées généralisées la quantité définissant le mouvement relatif du corps (segment) C_i par rapport au corps C_{i-1} , on note la variable articulaire q_i , elle peut être un angle autour d'un axe (liaison de rotation). L'ensemble q_i , permet de définir les différentes configurations d'un robot manipulateur.

I.3.3 espace opérationnel (coordonnées géométriques)

Elles permettent de caractériser et de mesurer les composantes de l'état (position et orientation) d'un repère orthonormé direct lié à l'organe terminal, par rapport à un repère fixe lié à la base de robot.

I.4 Modélisation d'un système mécanique articulé

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tels que les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot). On distingue :

- les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement.
- les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement.
- les modèles dynamiques qui expriment la relation entre les variables d'état du mécanisme et les forces extérieures agissant sur chaque corps.

Afin de contrôler et commander correctement les actionneurs, il est impératif de faire une bonne modélisation qui consiste à représenter le comportement de la structure mécanique articulée par des équations algébriques.

I.4.1 Modélisation géométrique

Le modèle géométrique d'un robot constitue une représentation mathématique en le considérant comme une structure arborescente comportant n corps rigides, sans masse, articulés entre eux. Chaque corps est réduit à sa plus simple expressions c'est-à-dire à son orientation et sa position par rapport au précédent.

$$C_i = (\text{position}_i, \text{orientation}_i).$$

I.4.1.1 Modèle géométrique direct (MGD)

Dans le modèle géométrique direct, la position et l'orientation de l'organe terminal est donnée en fonction des variable articulaires, nous l'exprimons par la fonction suivante :

$$x_i = F(q_1, q_2, \dots, q_n) \quad (I.1)$$

$$i = 1, 2, \dots, p.$$

Avec

p : le nombre de variables géométriques.

n : c'est le nombre de degrés de liberté.

$X = \begin{pmatrix} X \\ Y \end{pmatrix}$: Vecteur de coordonnées opérationnelles donnant la position de l'organe terminal.

$q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$: Vecteur des coordonnées opérationnelles donnant la position articulaire.

Le modèle géométrique direct d'un bras manipulateur à deux axes avec des liaisons rotoïdes est donné par l'expression suivante :

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \end{pmatrix} \quad (I.2)$$

I.4.1.2 modélisation géométrique inverse (MGI)

Le modèle géométrique inverse nous permet le calcul des variables articulaires en fonction des variables géométriques, il est constitué par la fonction inverse ou réciproque de F .

$$q = F^{-1}(X) \quad (I.3)$$

Avec :

$$q = (q_1, q_2, \dots, \dots, q_n).$$

$$X = (x_1, x_2, \dots, \dots, x_p)$$

I.4.1.3 Relation entre l'espace articulaire et l'espace des coordonnées cartésiennes de l'effecteur [9]

Désignons par $q = (q_1, \dots, \dots, q_m)$, Le vecteur des coordonnées articulaires (CA) d'un manipulateur à n degrés de liberté, ainsi que, $X = (x_1, \dots, \dots, x_n)$, le vecteur des coordonnées cartésiennes (CC) d'un manipulateur à n degrés de liberté

Le modèle géométrique inverse (MGI) consiste à déterminer les (CA) connaissant les (CC).
Le modèle géométrique direct (MGD) consiste à déterminer les (CC) pour des (CA) connus, comme le montre la figure suivante :

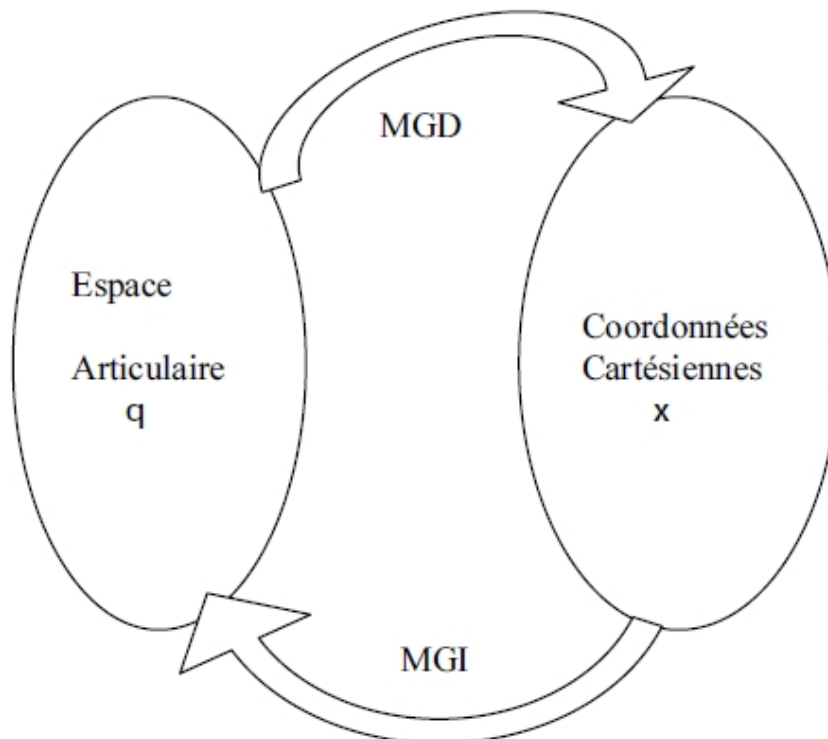


Figure I.5 Relation entre l'espace articulaire et l'espace des coordonnées cartésiennes de l'effecteur

I.4.2 modélisations cinématique

Dans le modèle géométrique, on prend en considération seulement la position et l'orientation des corps et de l'organe terminal, dans le modèle cinématique, il y a apparition des vitesses, celles de l'organe terminal et des actionneurs.

I.4.2.1 modèle cinématique direct

Le modèle cinématique direct d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires et en décrivant les équations de modèle géométrique direct par rapport au temps, on obtient le modèle cinématique suivant :

$$\frac{dX}{dt} = \frac{\partial F}{\partial q} \frac{dq}{dt} \quad (I.4)$$

$\frac{dX}{dt}$: Vecteur de position et d'orientation de l'organe terminal.

$\frac{\partial F}{\partial q}$: $J(q)$, Matrice Jacobéenne de dimension $(m \times n)$.

$\frac{dq}{dt}$: Vecteur de la vitesse généralisée (vitesse des actionneurs).

La matrice jacobéenne intervient dans le calcul du modèle différentielle direct qui donne les variations élémentaires dX des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires.

I.4.2.2 Modèle cinématique inverse

Le modèle cinématique inverse nous permet l'obtention des vitesses qu'il faut appliquer aux actionneurs en fonction des vitesses désirées pour l'organe terminal dans l'espace de la tâche. Il est donné sous la forme suivante :

$$\frac{dq}{dt} = J^{-1}(q) \times \frac{dX}{dt} \quad (I.5)$$

I.4.3 modélisation dynamique

L'utilisation des robots est devenue de plus en plus importante dans le domaine de l'automatisme. Dans les modèles précédents nous avons supposé que les corps du système mécanique articulé sont rigides et l'équilibre statique est réalisé pour toute configuration prise par le robot au cours de son déplacement. Au fur et à mesure que la vitesse augmente, les modèles géométriques et cinématiques vont s'éloigner du modèle réel.

Cependant l'ultra rapidité et la haute précision de la poursuite de trajectoire sont de plus en plus exigées pour atteindre les performances et la perfection désirée du robot.

Pour satisfaire les exigences des utilisateurs, il est nécessaire de tenir compte des phénomènes dynamiques (forces inertielles, centrifuges, Coriolis et de gravité, frottement et perturbations). Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires.

Le modèle dynamique d'un mécanisme articulé est obtenu à l'aide d'une suite d'équations mathématiques.

Cette modélisation est utile dans le cas d'une simulation, le but étant l'obtention de l'évolution des différents angles, vitesses et accélérations angulaires en fonction des couples appliqués aux différentes articulations. Deux approches sont possibles : l'une partant des équations de conservation des énergies cinétique et potentielle de Lagrange-Euler et l'autre développant l'équation fondamentale de la mécanique, la formule de Newton ($F = m.a$). Cette méthode est celle de Newton-Euler [10].

I.4.3.1 Le formalisme d'Euler- Lagrange

Le formalisme d'Euler-Lagrange amène à un algorithme qui consiste à présenter les équations dynamiques de mouvement. L'approche d'Euler-Lagrange sert à modéliser et à présenter la dynamique du robot à travers les équations du mouvement. L'approche d'Euler-Lagrange est donnée par l'équation (I.6) :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} + \frac{\partial E_d}{\partial \dot{q}} = \tau_i \quad (\text{I.6})$$

Avec E_d est l'énergie de dissipation en cas de présence de frottement visqueux, τ_i est la force ou le couple généralisé à la $i^{\text{ème}}$ articulation, n est le nombre de degrés de liberté, q_i est la coordonnée généralisée de l'articulation i , \dot{q}_i est la dérivée de la coordonnée généralisée et L est le Lagrangien exprimé par l'équation suivante :

$$L = E_c - E_p \quad (\text{I.7})$$

I.4.3.1 modélisation dynamique d'un bras manipulateur à 2 ddl

Le robot manipulateur à 2ddl utilisé dans ce travail est schématisé dans la figure suivante :

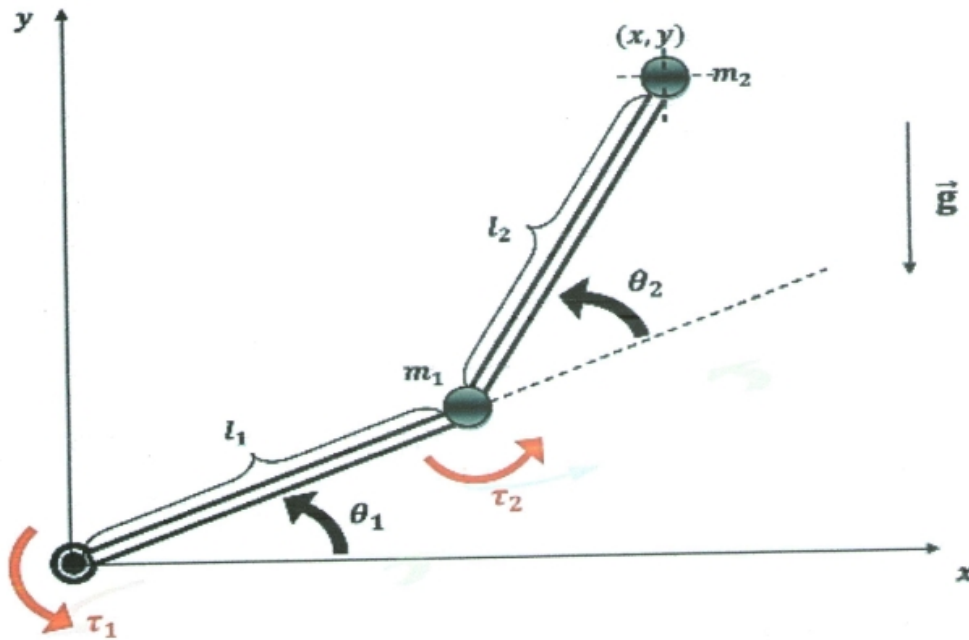


Figure I.6 robot manipulateur à 2 ddl

θ_i : l'angle adjoint à chaque bras avec l'axe des abscisses

m_i : la masse située au niveau de l'extrémité de chaque axe.

l_i : la longueur de chaque axe

τ_i : Le couple

\vec{g} : La force de gravité

On admettra que les masses des tiges composant le robot sont concentrées en leurs extrémités.

Le robot manipulateur est commandé en couple τ_1 et τ_2 fournis par les actionneurs dont on négligera la dynamique.

Les coordonnées cartésiennes sont :

A la 1^{ère} extrémité : $\begin{cases} x_1 = l_1 \cos \theta_1 \\ y_1 = l_1 \sin \theta_1 \end{cases}$, à la 2^{ème} extrémité $\begin{cases} x_2 = x_1 + l_2 \cos(\theta_1 + \theta_2) \\ y_2 = y_1 + l_2 \sin(\theta_1 + \theta_2) \end{cases}$

L'utilisation de la méthode d'Euler-Lagrange conduit à la forme standard de l'équation du mouvement d'un robot manipulateur qui se présente sous la forme suivante :

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (\text{I.8})$$

Avec :

$M(q)$ Matrice d'inertie symétrique définie positive.

$V(q, \dot{q})$ vecteur de force centripète/Coriolis, quadratique en \dot{q} .

Sachant que :

$$V(q, \dot{q}) = V_m(q, \dot{q}) \dot{q}$$

Et dans $V_m(q, \dot{q}) \dot{q}$, les termes impliquant un produit \dot{q}_i^2 sont appelés centrifuges impliquant un produit $\dot{q}_i \dot{q}_j$, avec $i \neq j$ sont les termes de Coriolis.

$G(q)$ vecteur de gravité avec $\|G(q)\| \leq g_b$, g_b étant une fonction scalaire constante.

- Energie cinétique

L'énergie cinétique d'un bras manipulateur à 2 ddl est donnée comme suit :

$$E_c = \frac{1}{2} (m_1 + m_2) l_2^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2 \quad (I.9)$$

- Energie potentiel

$$E_p = (m_1 + m_2) g l_1 \sin \theta_1 + m_2 g l_2 \sin(\theta_1 + \theta_2) \quad (I.10)$$

- Lagrangien du système

$$L = \left(\frac{1}{2} (m_1 + m_2) l_2^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2 \right) - (m_1 + m_2) g l_1 \sin \theta_1 + m_2 g l_2 \sin(\theta_1 + \theta_2) \quad (I.11)$$

L'application de formalisme d'Euler-Lagrange de l'équation (I.6), nous amène à l'équation suivante :

$$\begin{cases} \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_1} \right] - \frac{\partial L}{\partial q_1} = \tau_1 \\ \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_2} \right] - \frac{\partial L}{\partial q_2} = \tau_2 \end{cases} \quad (I.12)$$

Le modèle dynamique du robot manipulateur est donné sous la forme d'équations différentielles non linéaires suivantes :

$$\tau_1 = [(m_1 + m_2) l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 \cos \theta_2] \ddot{\theta}_1 + [m_2 l_2^2 + m_1 l_2 l_1 \cos \theta_2] \ddot{\theta}_2 - m_2 l_1 l_2 \theta_2 (2 \dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 + (m_1 + m_2) g l_1 \cos \theta_1 + m_2 g l_2 \cos(\theta_1 + \theta_2) \quad (I.13)$$

$$\tau_2 = [m_2 l_2^2 + m_2 l_1 l_2 \cos \theta_2] \ddot{\theta}_1 + [m_2 l_2^2] \ddot{\theta}_2 + m_2 l_1 l_2 \dot{\theta}_1^2 \sin \theta_2 + m_2 g l_2 \cos(\theta_1 + \theta_2) \quad (I.14)$$

En réécrivant les équations du système sous forme matricielle comme suit :

$$\begin{bmatrix} (m_1 + m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2 \cos \theta_2 & m_2l_2^2 + m_2l_1l_2 \cos \theta_2 \\ m_2l_2^2 + m_1l_2l_1 \cos \theta_2 & m_2l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -m_2l_1l_2\theta_2(2\dot{\theta}_1 + \dot{\theta}_2)\sin\theta_2 \\ m_2l_1l_2\dot{\theta}_1^2\sin\theta_2 \end{bmatrix} + \begin{bmatrix} (m_1 + m_2)gl_1\cos\theta_1 + m_2gl_2\cos(\theta_1 + \theta_2) \\ m_2gl_2\cos(\theta_1 + \theta_2) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (\text{I.15})$$

On identifie les différents termes de l'équation(I.8) comme suit :

$$M(q) = \begin{bmatrix} (m_1 + m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2 \cos \theta_2 & m_2l_2^2 + m_2l_1l_2 \cos \theta_2 \\ m_2l_2^2 + m_1l_2l_1 \cos \theta_2 & m_2l_2^2 \end{bmatrix} \quad (\text{I.16})$$

$$V(q, \dot{q}) = \begin{bmatrix} -m_2l_1l_2\theta_2(2\dot{\theta}_1 + \dot{\theta}_2)\sin\theta_2 \\ m_2l_1l_2\dot{\theta}_1^2\sin\theta_2 \end{bmatrix} \quad (\text{I.17})$$

$$V_m(q, \dot{q}) = \begin{bmatrix} -2m_2l_1l_2\dot{\theta}_1 \sin \theta_2 & -m_2l_1l_2\dot{\theta}_2 \sin \theta_2 \\ m_2l_1l_2\dot{\theta}_1 \sin \theta_2 & 0 \end{bmatrix} \quad (\text{I.18})$$

$$G(q) = \begin{bmatrix} (m_1 + m_2)gl_1\cos\theta_1 + m_2gl_2\cos(\theta_1 + \theta_2) \\ m_2gl_2\cos(\theta_1 + \theta_2) \end{bmatrix} \quad (\text{I.19})$$

I.5 Conclusion

Pour modéliser un manipulateur à n articulations, il faut représenter le comportement du ce robot sous la forme d'un modèle, une telle démarche s'appelle la modélisation, d'une manière générale, on recherche toujours le modèle le plus simple qui permet d'expliquer, de manière satisfaisante, le comportement du processus dans son domaine d'application ; donc les notions de base de la modélisation ont été données au début de ce chapitre. La modélisation d'un robot manipulateur à 2 ddl qui est le sujet de ce mémoire a été détaillé et à la fin, le modèle dynamique a été donné.

Chapitre II

II.1 Introduction

La commande de robots est une science qui fait son chemin depuis l'existence de la notion même du robot. Cette science a permis d'appriivoiser les robots et de les piloter afin que nous puissions les utiliser pour des tâches aussi bien diverses que complexes.

Le contrôle des robots manipulateurs, dans l'intention d'atteindre et de maintenir avec précision une position désirée, constitue un défi. De nombreuses méthodes de contrôle ont été étudiées. Elles peuvent être regroupées en deux catégories: les méthodes classiques et les méthodes modernes. Les techniques de contrôle classiques sont: le contrôle proportionnel-dérivé (PD), le contrôle proportionnel-dérivé-intégral (PID), le contrôle par retour d'état, la commande linéaire-quadratique (LQR), ... etc.

Les techniques de contrôle modernes sont multiples parmi lesquelles nous pouvons citer la commande par mode de glissement, la commande H l'infini, la commande neuro-floue, la commande adaptative, ... etc. Dans ce chapitre, nous allons présenter quelques techniques de commandes classiques et modernes.

II.2 Commande par mode glissant

La commande par mode glissant a connu un essor considérable durant les dernières décennies. Ceci est dû principalement à la propriété de convergence rapide et en temps fini des erreurs, ainsi, que la grande robustesse par rapport aux erreurs de modélisation et certains types de perturbations extérieures [18].

La commande par mode glissant d'ordre 1 (SMC) est une commande à structure variable pouvant changer de structure et commutant entre deux valeurs suivant une logique de commutation bien spécifique $s(x)$.

Le principe de la commande par mode glissant est de contraindre le système à atteindre une surface donnée appelée surface de glissement et d'y demeurer jusqu'à l'équilibre. Cette commande se fait en deux étapes : la convergence vers la surface et ensuite le glissement le long de celle-ci (Figure II.1)

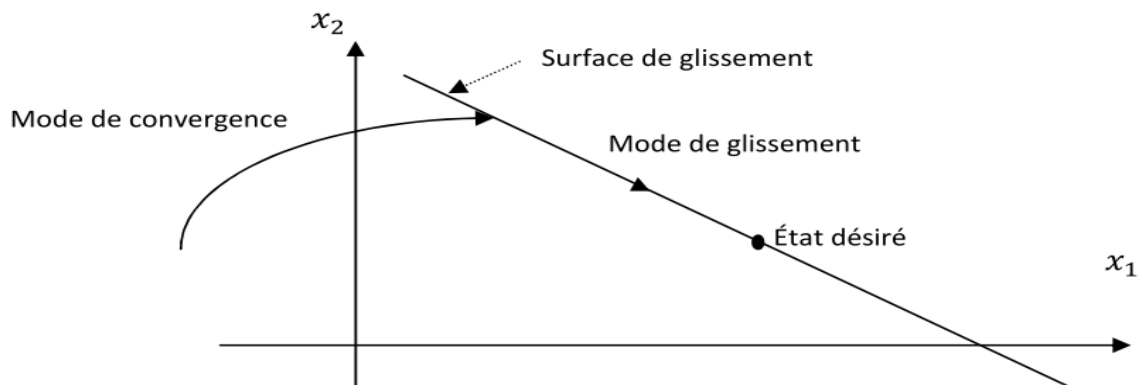


Figure II.1 différents modes de convergence pour la trajectoire d'état

II.2.1 Synthèse de la loi de commande

La synthèse de la commande par mode glissant se fait en trois étapes :

- Choix de la surface de glissement
- Etablir la condition de convergence
- Déterminer la loi de commande qui permet d'atteindre la surface et d'y demeurer.

II.2.1.1 Choix de la surface de glissement

Soit le système décrit par l'équation différentielle suivante :

$$\dot{x}^n = f(x, t) + g(x, t)u \quad (\text{II.1})$$

Où f et g sont des fonctions non linéaires, g est supposée inversible.

u : L'entrée du système.

x : état du système.

Soit x_d la consigne désirée et e l'erreur de poursuite définie par :

$$e = x - x_d \quad (\text{II.2})$$

La formule générale de la surface de glissement est définie en fonction de l'ordre du système comme suit :

$$s(x) = \left(\frac{\partial}{\partial t} + \lambda I \right)^{n-1} e(x) \quad (\text{II.3})$$

Où n : le degré relatif du système par rapport à la sortie $y(t)$, Il représente le nombre minimum de fois qu'il faut dériver la sortie $y(t)$. Par rapport au temps, pour y voir apparaître l'entrée [19].

II.2.1.2 condition d'existence du mode de glissement

Le choix de la fonction de glissement étant fait, la deuxième étape consiste à concevoir une loi de commande qui puisse amener le vecteur d'état à converger vers la surface et y demeurer ($S=0$).

Pour cela, il faut que la loi de commande soit conçue de telle manière à ce que S soit attractive. Pour déterminer la condition d'attractivité, considérons la fonction de Lyapounov suivante :

$$v(s) = \frac{1}{2}S^2 \quad (\text{II.4})$$

Une condition nécessaire et suffisante, appelée condition d'attractivité, pour qu'une variable de glissement $S(x, t)$ tende vers 0 est que la dérivée temporelle de v soit définie négative :

$$S\dot{S} < 0 \quad (\text{II.5})$$

Si la condition (II.5) est vérifiée, alors la variable de glissement et sa dérivée sont de signe contraire quel que soit le temps et que 0 est un centre attracteur pour S .

Le temps de convergence, ou le temps d'atteinte « reaching time » t_r dépend directement du choix de S .

Pour une convergence en temps fini, la condition (II.5) qui ne garantit qu'une convergence asymptotique vers la surface de glissement est remplacée par une condition plus restrictive dite de η -attractivité et donnée par : [19]

$$S \cdot \dot{S} \leq -\eta|S| \quad (\text{II.6})$$

II.2.1.3 calcul de la commande

On peut calculer la commande par la méthode de la commande équivalente, schématisée sur la figure (II.2).

La commande équivalente est une fonction continue qui sert à maintenir la variable à contrôler sur la surface de glissement $\{s = 0\}$ Elle est obtenue grâce aux conditions d'invariance de la surface :

$$s = 0$$

$$\dot{s} = 0$$

Où U_{eq} est déduite de la relation $\dot{s} = 0$

Physiquement la commande équivalente présente la valeur moyenne de la commande u . Cependant, cette commande ne force pas les trajectoires du système à converger vers la surface de glissement. Ainsi, la commande u est la somme de la commande équivalente et d'une composante discontinue assurant une convergence et un régime glissant.

$$u = u_{eq} + u_d \quad \text{Avec} \quad u_d = -\alpha \text{sign}(s)$$

α est une constante positive, $\text{sign}(s)$ est la fonction signe, et u_d est la commande discontinue.

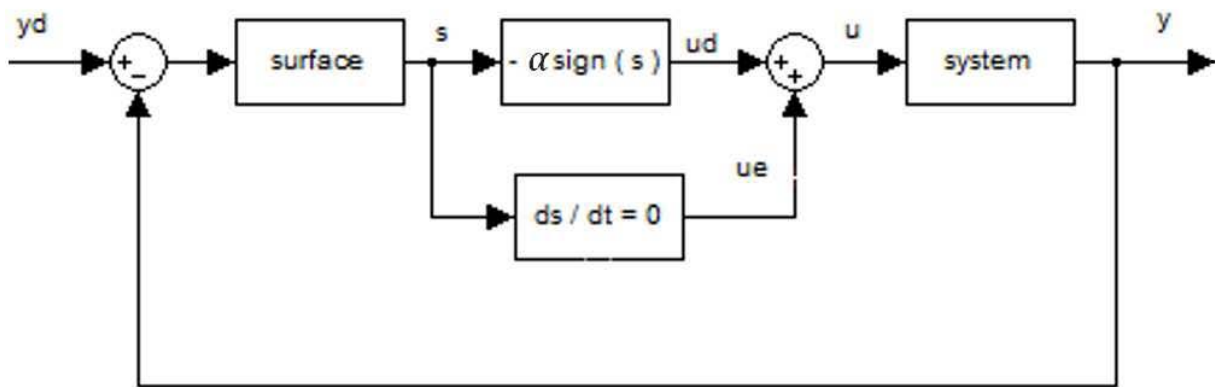


Figure (II.2) Principe de la commande en mode glissant avec commande équivalente

II.2.2 Le broutement (chattering) :

Un régime glissant idéal requiert une commande pouvant commuter à une fréquence infinie. Ainsi, durant le régime glissant, les discontinuités appliquées à la commande peuvent entraîner un phénomène de broutement, appelé réticence ou "chattering" en anglais. Celui-ci se caractérise par de fortes oscillations des trajectoires du système autour de la surface de glissement (figure (II.3)). Les principales raisons à l'origine de ce phénomène sont les limitations des actionneurs ou les retards de commutation au niveau de la commande. Ces commutations détériorent la précision de la commande et peuvent s'avérer néfastes pour l'organe de commande en provoquant une détérioration prématurée des systèmes mécaniques et une élévation de température dans les systèmes électriques (perte d'énergie non négligeable).

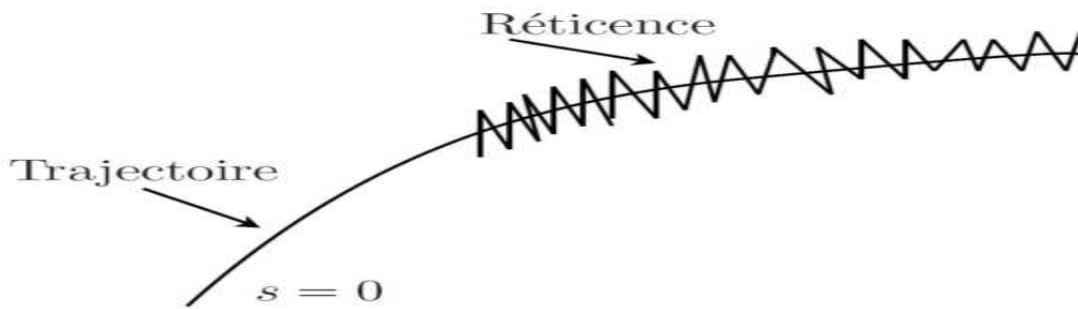


Figure (II.3) : Le phénomène de broutement

II.3. la commande adaptative

La commande adaptative a débuté principalement dans les années 50s, comme solution pour contrôler les processus fonctionnant sous des conditions et environnements variables dans le temps. Dans les années 60s, plusieurs contributions de la théorie de commande ont été introduites dans le développement de la commande adaptative, comme par exemple l'analyse dans l'espace d'état, la théorie de stabilité, la théorie de la commande stochastique et programmation dynamique. Au début des années 70s, les différentes méthodes d'estimation ont été introduites dans la commande adaptative.

La théorie de stabilité de la commande adaptative a commencé au début des années 80s en parallèle avec une rapide évolution en micro électronique qui a permis d'implémenter des régulateurs adaptatifs sur des systèmes à microprocesseurs.

Dans ce chapitre nous allons essayer de donner les principes généraux de la commande adaptative. Nous allons parler du gradient et de la règle MIT et en fin la stabilité par la méthode de Lyapunov.

II.3.1 Concept de la commande adaptative

Dans plusieurs cas de la théorie de commande, le système à contrôler possède des paramètres incertains au début de l'opération de commande. Malgré que cette incertitude paramétrique disparaisse en temps réel par un certain mécanisme d'ajustement, elle peut causer une incertitude ou une instabilité du système de commande. Dans d'autres cas, un phénomène inverse se produit, où on démarre avec des paramètres connus et certains, mais au cours de fonctionnement, ces paramètres perdent leurs valeurs et deviennent incertains. Donc sans une préconception continue du régulateur, le régulateur initial s'avère inefficace.

Généralement, l'objectif principal de la commande adaptative est de maintenir des performances désirées d'un système en présence d'incertitudes ou variation paramétrique du procédé.

Un régulateur adaptatif diffère d'un régulateur ordinaire dans le fait que les paramètres du premier sont variables et qu'il existe un certain mécanisme d'ajustement de ces paramètres [20].

II.3.2 Commande adaptative à modèle de référence [20][21][22][23]

Généralement, une commande adaptative à modèle de référence est donnée comme le montre la (Fig.II.4). Elle est composée de 4 parties :

- Le procédé contenant les paramètres inconnus.
- Un modèle de référence spécifiant les performances désirées de sortie.
- Une loi de commande contenant les paramètres ajustables.
- Un mécanisme d'ajustement pour l'adaptation des paramètres ajustables.

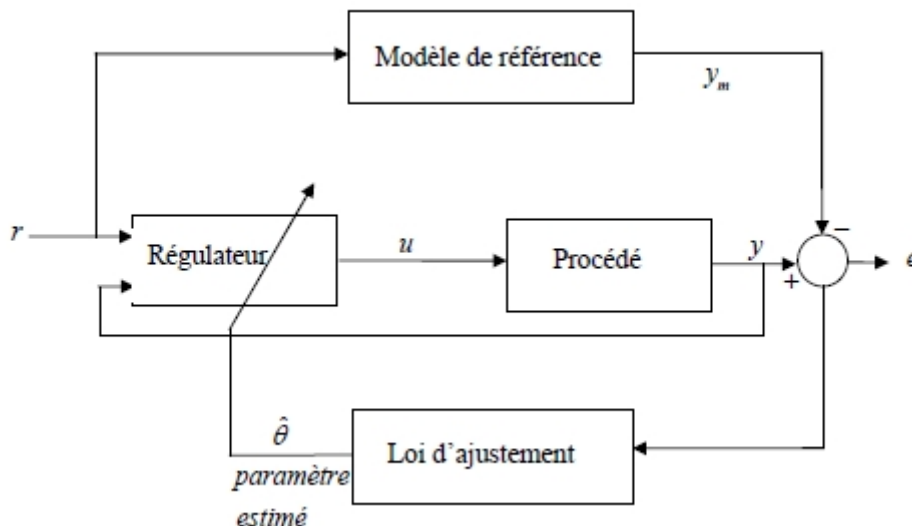


Figure II.4 commande adaptative à modèle de référence

- Le procédé est supposé avoir une structure à paramètres inconnus
- Le modèle de référence est utilisé pour spécifier la réponse idéale de notre système.
- Le régulateur est décrit généralement par un certain nombre de paramètres ajustables, qui seront ajustés par le mécanisme d'ajustement.
- Le mécanisme d'ajustement est utilisé pour ajuster les paramètres dans la loi de commande. La loi d'ajustement va chercher les valeurs des paramètres pour lesquelles la réponse du système sous la commande adaptative devient la même que la réponse du modèle de référence c.-à-d., imposer à l'erreur de poursuite de converger vers zéro.

Dans la commande adaptative à modèle de référence, on rencontre deux structures : la commande adaptative directe et la commande adaptative indirecte :

Dans la commande adaptative indirecte (voir Fig.II.5), on estime d'abord les paramètres du procédé, en suite on calcule les paramètres du régulateur. L'appellation indirecte vient du fait que les paramètres estimés du procédé sont envoyés vers les paramètres du régulateur.

Dans la commande adaptative directe (voir Fig.II.4), les paramètres du procédé et ceux du régulateur seront reliés les uns aux autres pour former une seule structure paramétrique qui sera ajustée par le mécanisme d'ajustement. Donc, dans ce type de commande, la loi de commande est générée directement sans estimation propre aux paramètres du procédé.

Pour analyser et concevoir un système à commande adaptative, on trouve essentiellement deux approches :

- Approche du gradient
- Fonctions de Lyapunov

II.3.3 Approche du gradient

Cette approche est basée sur l'hypothèse que les paramètres du procédé changent d'une manière très lente par rapport aux autres variables dans le système. L'approche du gradient ne fournit pas nécessairement un système stable en boucle fermée. Le mécanisme d'ajustement des paramètres est appelé généralement 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, θ le paramètre à ajuster et J soit le critère d'optimisation suivant :

$$J(\theta) = \frac{1}{2} e^2 \tag{II.7}$$

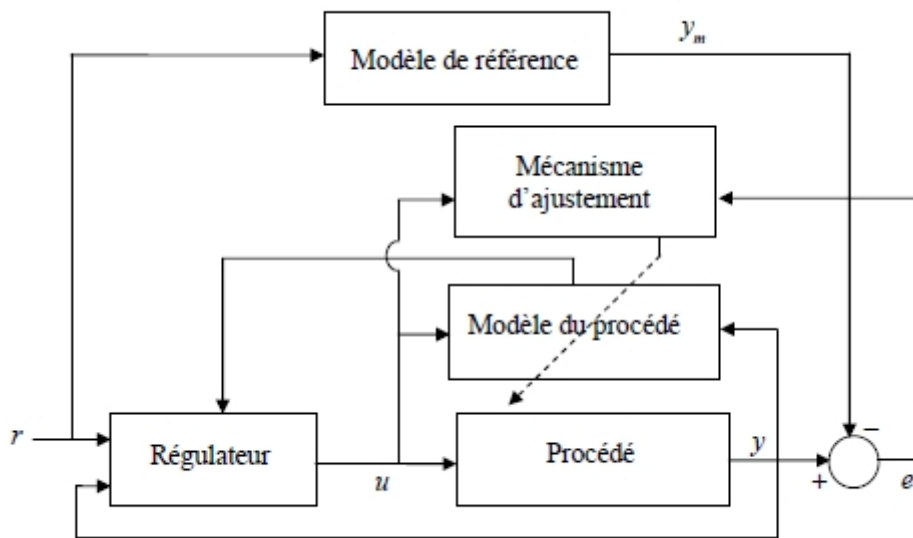


Figure II. 5: Structure d'une commande adaptative indirecte

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

Cette règle MIT fonctionnera bien si le gain d'adaptation γ est faible. La grandeur permise de γ dépendra alors de l'amplitude du signal de référence et du gain du procédé. Par conséquent, il n'est pas possible de donner des limites fixées qui peuvent garantir la stabilité, donc la règle MIT peut donner un système instable en boucle fermée. Une loi d'ajustement modifiée utilisant la théorie de stabilité peut alors être introduite.

II.3.5 Fonction de Lyapunov [24][25]

Dans cette approche nous allons essayer de trouver des lois d'ajustements de telle sorte que la convergence de l'erreur vers zéro soit garantie.

Lyapunov a introduit sa méthode directe pour étudier la stabilité d'une solution à une équation différentielle non linéaire. La philosophie de base de cette méthode est l'extension d'une observation physique fondamentale : si l'énergie totale d'un système mécanique (ou électrique) est continuellement dissipée, alors le système convergera vers un état d'équilibre. Donc, on peut conclure la stabilité d'un système donné par simple examen d'une certaine fonction scalaire.

Soit l'équation différentielle :

$$\dot{\underline{x}} = f(\underline{x}, t) \quad \text{Avec} \quad f(0, t) = 0 \quad (\text{II.8})$$

Ou $\underline{\dot{x}}$ est le vecteur d'état de dimension n , le point d'équilibre est supposé être l'origine.

II.4 Commandes classiques

Corriger(ou réguler) un système, c'est améliorer ses performances intrinsèques (précision, rapidité, régulation, limitation de la commande tout en garantissant sa stabilité). En effet, le processus possède son fonctionnement propre et (souvent) quelques défauts par rapport aux objectifs souhaités, Les correcteurs, usuellement utilisés en pratique sont les régulateurs à effet proportionnel, intégral et dérivé.

Le but de cette partie est d'étudier les différentes actions de ces régulateurs et de voir les avantages et les inconvénients de chaque action puis de voir la façon de choisir chaque action pour assurer une bonne régulation.

II.4.1. Régulateur proportionnelle P

Le rôle de l'action proportionnelle est de minimiser l'écart ε entre la consigne et la mesure, et elle réduit le temps de montée et le temps de réponse. On constate qu'une augmentation du gain K_p du régulateur entraîne une diminution de l'erreur statique et permet d'accélérer le comportement global de la boucle fermée.

On tente de prendre des valeurs du gain élevées pour accélérer la réponse du procédé mais on est limité par la stabilité de la boucle fermée. En effet, une valeur trop élevée du gain, augmente l'instabilité du système et donne lieu à des oscillations

La loi de commande du régulateur P est donnée par:

$$U(t) = K_p \cdot E(t) \quad (\text{II.9})$$

La fonction de transfert du régulateur Pest décrite par:

$$G(s) = \frac{U(s)}{E(s)} = K_p \quad (\text{II.10})$$

Le schéma fonctionnel du régulateur P est donné par:

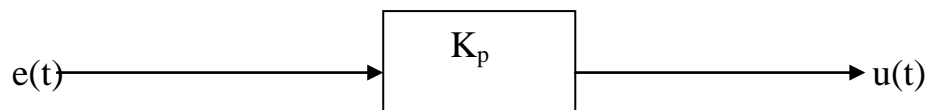


Figure II.6: Schéma fonctionnel du régulateur p

En conclusion, un correcteur proportionnel augmente la rapidité du système (effet souhaitable) mais il augmente aussi son instabilité (effet non souhaitable). C'est le dilemme stabilité-précision. Il faut déterminer la valeur de compromis entre stabilité et précision.

II.4.2 Régulateur proportionnel-intégrateur PI

La commande proposée est formée des deux contributions u_p et u_i , contributions proportionnelle (P) et intégrale (I). Le régulateur est donc à actions proportionnelle et intégrale: c'est un régulateur **PI**

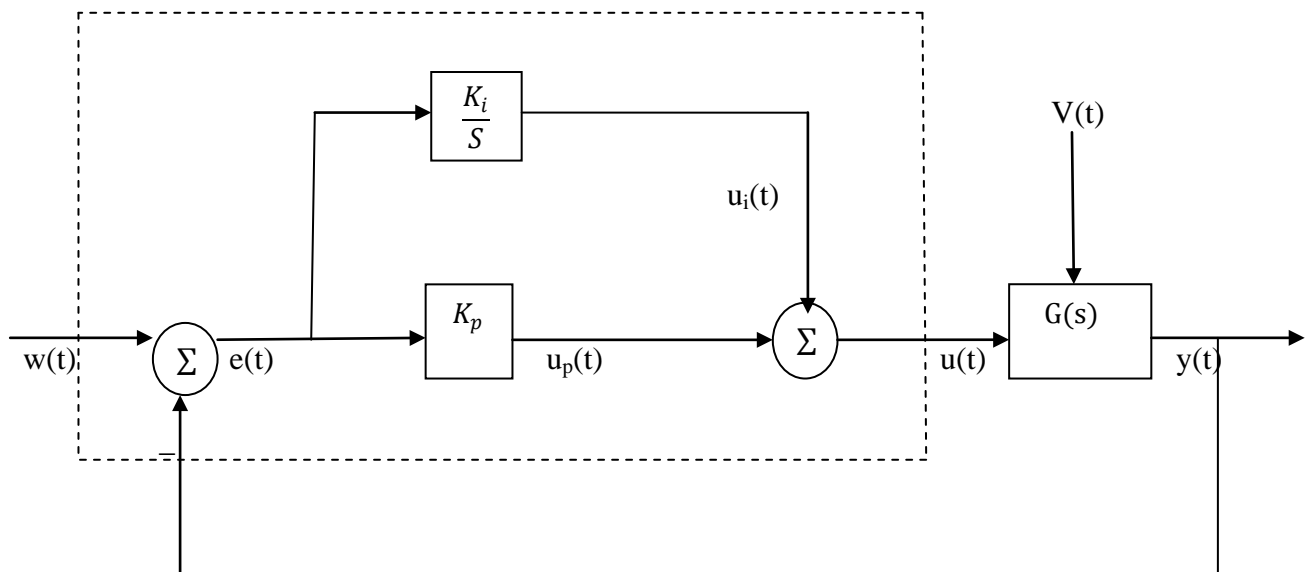


Figure II.7 asservissement par régulateur PI

Le rôle de l'action intégrale est d'annuler l'écart entre la mesure et la consigne. Le signal de sortie du régulateur en intégrateur seul est proportionnel à l'intégrale de l'écart mesure-consigne. L'action intégrale est généralement associée à l'action proportionnelle. Comme dans le cas de l'action proportionnelle, une augmentation excessive de l'action intégrale (diminution de T_i) peut être source d'instabilité.

La loi de commande du régulateur PI est fournie par l'équation suivante :

$$U(t) = e(t) + \frac{1}{T_i} \int_{-\infty}^t e(\tau) \cdot d\tau \quad (\text{II.11})$$

Sa fonction de transfert est donnée comme suit:

$$G(s) = \frac{U(s)}{E(s)} = K_p \frac{1+sT_i}{sT_i} \quad (\text{II.12})$$

Le schéma fonctionnel du régulateur PI est le suivant:

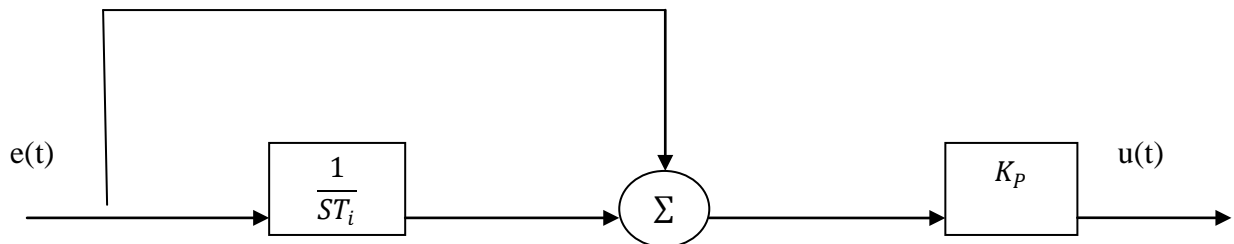


Figure II.8 Schéma fonctionnel du régulateur PI.

Pour un régulateur intégral pur, le régime dynamique est relativement long. D'un autre côté, le régulateur proportionnel réagit immédiatement aux écarts de réglage mais il n'est pas en mesure de supprimer totalement l'erreur statique. La combinaison des actions proportionnelle et intégrale permettent d'associer l'avantage du régulateur P, c'est-à-dire la réaction rapide à un écart de réglage, à l'avantage du régulateur I qui est la compensation exacte de la grandeur pilote.

II.4.3 Régulateur I pur

L'action P du régulateur PI n'est pas utile du point de vue de la précision en régime permanent; cependant, le fait que l'action P permette la transmission instantanée du signal d'erreur rend le régulateur PI plus dynamique que le régulateur I pur, mis en œuvre dans quelques cas particuliers où la vitesse n'est pas importante et où l'on souhaite avoir une action relativement "molle" sur le système à régler.

La loi de commande du régulateur I est formulée par :

$$U(t) = \frac{K_p}{T_i} \int_{-\infty}^t e(\tau) d(\tau) \quad (\text{II.13})$$

Sa fonction de transfert est donnée par :

$$G(s) = \frac{U(s)}{E(s)} = \frac{K_p}{sT_i} = \frac{K_i}{s} \quad (\text{II.14})$$

Le régulateur PI est le régulateur le plus utilisé en pratique où ses contributions à la précision mais aussi à la robustesse du système asservi sont particulièrement appréciées.

II.4.4 Régulateur Proportionnel dérivateur PD

Une commande dérivée aura l'effet d'augmenter la stabilité du système, de réduire le dépassement, et d'améliorer la réponse transitoire. Le correcteur est un correcteur à avance de phase.

Le rôle de l'action dérivée est de compenser les effets du temps mort (retard) du procédé. Elle a un effet stabilisateur mais une valeur excessive peut entraîner l'instabilité. Son rôle est identique quelle que soit la nature du procédé. La sortie du dérivateur est proportionnelle à la vitesse de variation de l'écart. Notons que l'action dérivée ne peut pas être utilisée seule.

La loi de commande du régulateur PD est exprimée comme suit :

$$U(t) = K_p \left(e(t) + T_d \frac{de(t)}{dt} \right) \quad (\text{II.15})$$

Sa fonction de transfert est la suivante :

$$G(s) = \frac{U(s)}{E(s)} = K_p (1 + sT_d) \quad (\text{II.16})$$

Le schéma fonctionnel du régulateur PD est illustré dans la figure suivante:

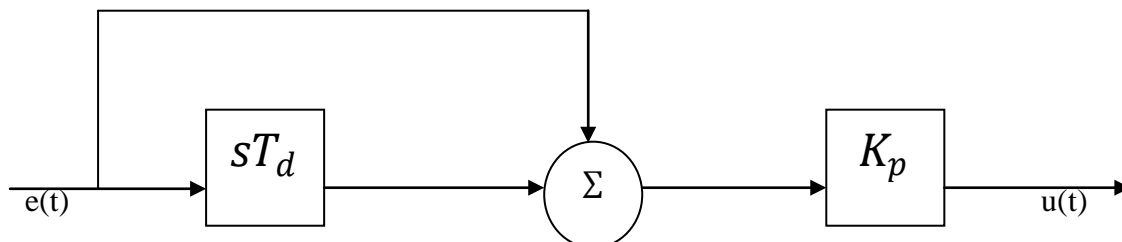


Figure II.9: Schéma fonctionnel du régulateur PD.

L'action D apporte une amélioration notable du comportement dynamique, accélérant la vitesse de réaction du régulateur aux moindres variations de l'erreur. Ainsi, un signal d'erreur, si faible que soit son amplitude, pourra générer une réaction très énergique du régulateur si son taux de croissance de/dt est élevé. Il est donc important de réaliser que l'action D est plutôt stabilisante et améliore la rapidité des systèmes.

Un inconvénient majeur de l'action D est à rechercher au niveau de l'effet des bruits intervenant sur la mesure. Le dérivateur amplifie l'effet des bruits et ceci d'autant plus que ceux-ci se situent par nature dans une gamme de fréquences relativement élevées.

II.4.5 Régulateur PID

Le régulateur standard le plus utilisé dans l'industrie, est le régulateur PID (proportionnel intégral dérivé), car il permet de régler à l'aide de ses trois paramètres, les performances (amortissement, temps de réponse, ...) d'un processus modélisé par un deuxième ordre. Nombreux sont les systèmes physiques qui, même en étant complexes, ont un comportement voisin de celui d'un deuxième ordre. Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel, et il est relativement robuste par rapport aux variations des paramètres du procédé, quand on n'est pas trop exigeant sur les performances de la boucle fermée par rapport à celles de la boucle ouverte (par exemple, accélération très importante de la réponse ou augmentation très importante de l'amortissement en boucle fermée).

L'action conjuguée PID permet une régulation optimale en associant les avantages de chaque action : la composante P réagit à l'apparition d'un écart de réglage, la composante D s'oppose aux variations de la grandeur réglée et stabilise la boucle de régulation ; et la composante I élimine l'erreur statique. Et c'est pour cela que ce type de correcteur est le plus utilisé en milieu industriel.

Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les trois fonctions suivantes :

- Il fournit un signal de commande en tenant compte de l'évolution du signal de sortie par rapport à la consigne.
- Il élimine l'erreur statique grâce au terme intégrateur.
- Il anticipe les variations de la sortie grâce au terme dérivateur.

La loi de commande du régulateur PID est donnée par l'équation suivante:

$$U(t) = K_p(e(t) + \frac{1}{T_i} \int_{-\infty}^t e(\tau) d(\tau) + T_d \frac{de(t)}{dt}) \quad (\text{II.17})$$

- K_p est le gain proportionnel. On le trouve fréquemment sous la dénomination "bande proportionnelle" égale à $100/K$.
- T_i est la constante de temps de l'action intégrale, l'intégrale, en secondes ou en minutes. T_i peut être réglée par son inverse (répétitions par seconde ou par minute).
- T_d est la constante de temps de la partie dérivée, la dérivée, et N permet de filtrer cette dérivée.

La fonction de transfert du régulateur PID s'écrit comme suit:

$$G(s) = \frac{U(s)}{E(s)} = K_p \frac{1+sT_i+s^2T_iT_d}{sT_i} \quad (\text{II.18})$$

Son schéma fonctionnel est montré dans la figure suivante:

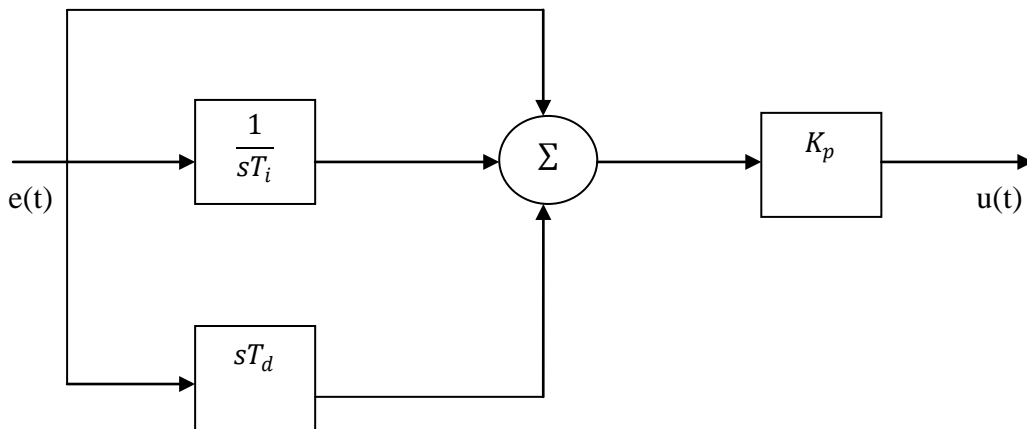


Figure II.10 : Schéma fonctionnel du régulateur PID.

Dans le tableau suivant, nous montrons les avantages et les inconvénients de chaque action.

Action	Avantage	Désavantage
P	dynamique	ne permet pas d'annuler une erreur statique
I	annulation d'erreur statique, amélioration de la robustesse	action lente, ralentit le système (effet déstabilisant)
D	action très dynamique, améliore la rapidité (effet stabilisant)	sensibilité aux bruits forte sollicitation de l'organe de commande

Tableau II .1 Résumé des effets respectifs des actions P, I, et D.

II.5 Conclusion

Une fois le modèle dynamique du robot établi, le plus intéressant reste la commande du manipulateur. Cette matière a fait l'objet de nombreuses études. Dans ce chapitre, nous avons étudié quelques techniques de commandes classiques et modernes, à savoir la commande adaptative, la commande par mode glissant et les contrôleurs classiques. Nous avons parlé de leurs avantages et inconvénients.

La commande adaptative, se heurte très souvent à des problèmes de stabilité quand il s'agit d'avoir une très bonne précision, éternel dilemme de l'automatique.

La commande par mode glissant est très robuste vis-à-vis les perturbations et les incertitudes paramétriques, mais elle présente l'inconvénient de chattering qui est son plus grand défaut, car il peut causer une perte de précision, et surtout une usure prématurée des actionneurs ou du système, en plus de générer un bruit sur des systèmes mécaniques.

Chapitre II Synthèse de la commande de robot manipulateur

Les commandes classiques sont caractérisées par leur facilité d'implantation et leur faible coût. Cependant, elles ne sont plus acceptables lorsque le robot est appelé à effectuer des grands déplacements à des vitesses importantes et requérant une bonne précision. C'est pourquoi nous préconisons l'emploi d'un contrôleur PD associé à un contrôleur neuro-flou pour commander un bras manipulateur à deux axes.

Dans le chapitre suivant, nous allons étudier les réseaux de neurones et les systèmes flous ainsi que les systèmes neuro-flous.

Chapitre III

III.1 Introduction

L'utilisation des réseaux de neurones et des systèmes flous pour la commande des systèmes a connu un essor important au cours de ces dernières années.

Dans ce chapitre, nous allons présenter ces outils ainsi que leur combinaison qui donne naissance aux systèmes neuro-flous, afin de tirer profits des avantages des deux approches.

III.2 Les réseaux de neurones

Les réseaux de neurones formels sont des systèmes de traitement de l'information dont la structure s'inspire de celle du système nerveux. Ils ont d'abord été développés pour résoudre des problèmes de contrôle, de reconnaissance de formes ou de mots, de décision, de mémorisation comme une alternative à l'intelligence artificielle, et en relation plus ou moins étroite avec la modélisation de processus cognitifs (capable de connaître ou faire connaître) réels et des réseaux de neurones biologiques.

III.2.1 Le neurone biologique

Le neurone biologique est une cellule vivante spécialisée dans le traitement des signaux électriques. Les neurones sont reliés entre eux par des liaisons appelées axones. Ces axones vont eux-mêmes jouer un rôle important dans le comportement logique de l'ensemble. Ces axones conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone. Les neurones font une sommation des signaux reçus en entrée et en fonction du résultat obtenu vont fournir un courant en sortie, (Figure 2.1).

La structure d'un neurone se compose de trois parties :

- La somma : ou cellule d'activité nerveuse, au centre du neurone.
- L'axone : attaché au somma qui est électriquement actif, ce dernier conduit l'impulsion conduite par le neurone.
- Dendrites : électriquement passives, elles reçoivent les impulsions d'autres neurones.

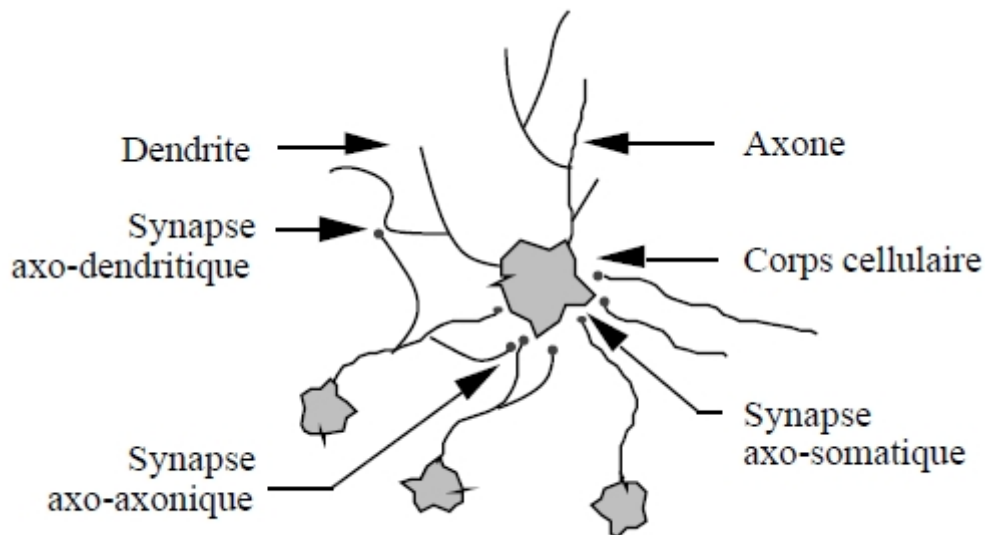


Figure III.1 Un neurone avec son arborisation dendritique.

III.2.2 Le neurone formel

Le neurone artificiel (ou cellule) est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones appartenant à un niveau situé en amont (on parlera de neurones "amonts"). A chacune des entrées est associé un poids w représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones appartenant à un niveau situé en aval (on parlera de neurones "avals"). A chaque connexion est associé un poids.

III.2.3 Modélisation d'un neurone formel

Les réseaux de neurones formels sont à l'origine d'une tentative de modélisation mathématique du cerveau humain. Les premiers travaux datent de 1943 et sont l'œuvre de MM. Mac Culloch et Pitts. Ils présentent un modèle assez simple pour les neurones et explorent les possibilités de ce modèle. La modélisation consiste à mettre en œuvre un système de réseau neuronal sous un aspect non pas biologique mais artificiel, cela suppose que d'après le principe biologique on aura une correspondance pour chaque élément composant le neurone biologique, donc une modélisation pour chacun d'entre eux.

Le modèle mathématique d'un neurone artificiel est illustré dans la figure suivante :

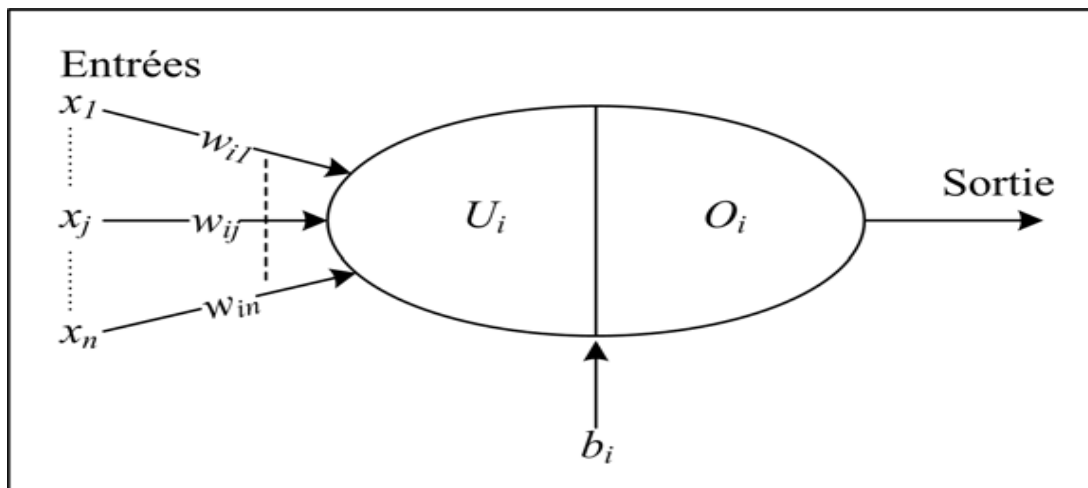


Figure III.2 : Modèle mathématique d'un neurone formel

U_i représente la somme pondérée des entrées du neurone, elle est donnée par :

$$U_i = \sum_j w_{ij} x_j + b_i$$

Où : x_j représente l'entrée j connectée au neurone i , b_i représente le seuil interne du neurone et w_{ij} désigne le poids de la connexion reliant l'entrée j au neurone.

$O_i = f(U_i)$ est la sortie du neurone et f sa fonction de transfert.

On pourra résumer cette modélisation par le tableau 1, qui nous permettra de voir clairement la transition entre le neurone biologique et le neurone formel [14].

Neurone biologique	Neurone formel
Synapses	Poids de connexions
Axones	Signal de sortie
Dendrite	Signal d'entrée
Somma	Fonction d'activation

Tableau III.1 : Analogie entre le neurone biologique et le neurone formel

En utilisant le tableau ci-dessus, Nous allons vous expliquer un peu plus en détail leur fonctionnement :

- Il y a d'abord le signal d'entrée : c'est un nombre qui, suivant le programme, peut être binaire (soit 1, soit 0) ou alors un réel (n'importe quel nombre) ;
- Ensuite, la valeur passe dans un poids de connexion : ce poids va être utilisé pour multiplier le signal d'entrée.
- Une fois multipliée, cette valeur va passer dans la fonction d'activation : Là, la valeur va être traitée par une fonction du type $f(x)=a$ où x est le signal d'entrée multiplié, et a le résultat de la fonction.

- Enfin, la fonction d'activation va générer une valeur finale, qui sera le résultat, le signal de sortie. Cela peut être soit un signal binaire, soit un nombre réel.

III.2.4 Fonctions d'activation

Cette fonction permet de définir l'état interne du neurone en fonction de son entrée totale, citons à titre d'exemple quelques fonctions souvent utilisées :

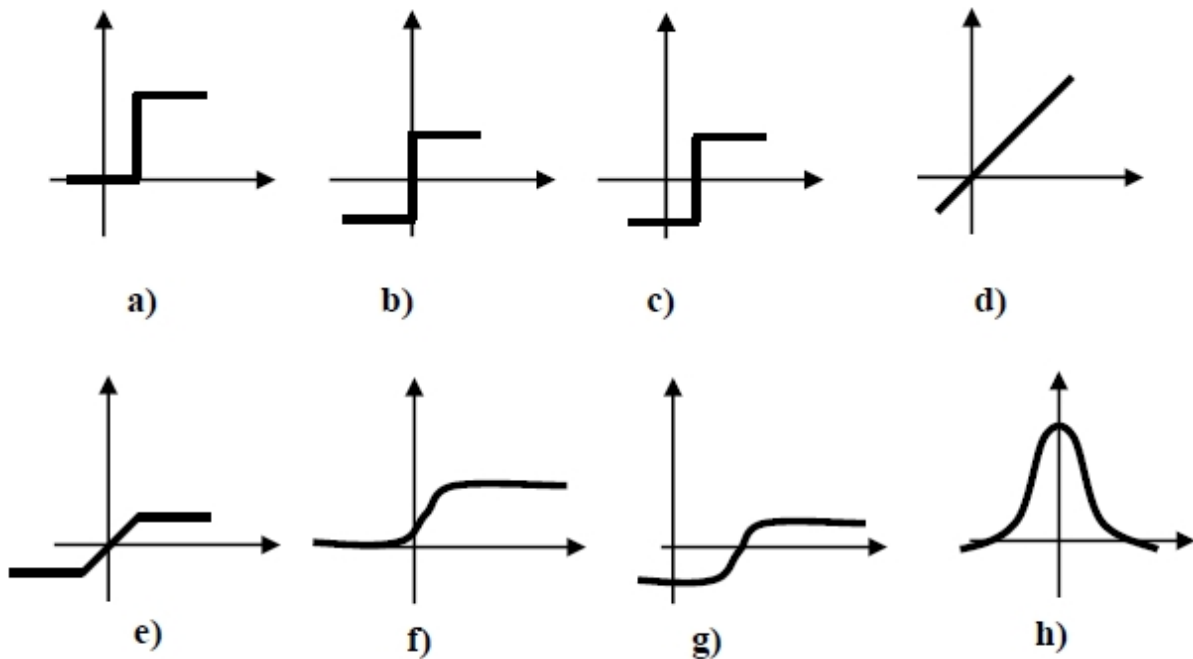


Figure III.3 les fonctions d'activation les plus utilisées [15]

Avec

- | | |
|--------------------------|--------------------------|
| a) Tout ou rien | e) Saturation |
| b) Fonction signe | f) Sigmoïde |
| c) Plus au moins à seuil | g) Fonction arc tangente |
| d) Fonction affine | h) Fonction gaussienne |

Nous constatons que les équations décrivant le comportement des neurones n'introduisent pas la notion de temps. En effet c'est le cas pour la plupart des modèles actuels des réseaux de neurones, nous avons affaire à des modèles discrets, synchrones, dont le comportement des composants ne varie pas dans le temps.

III.2.5 Types de réseaux de neurones [15]

III.2.5.1 Réseaux de neurones non bouclés

Dans ce type de structure dite feedforward, la propagation de l'information se fait uniquement de l'entrée vers la sortie. Les neurones de la même couche peuvent se connecter uniquement avec les neurones de la couche suivante. L'architecture la plus utilisée est le Perceptron multicouches. Les neurones composant ce réseau s'organisent en N couches successives ($N \leq 3$).

Dans l'exemple suivant (figure III.4), nous présentons un perceptron à trois couches. Les neurones de la première couche, nommée couche d'entrée, voient leur activation forcée à la valeur d'entrée. La dernière couche est appelée couche de sortie. Elle regroupe les neurones dont les fonctions d'activation sont généralement de type linéaire. Les couches intermédiaires sont appelées couches cachées. Elles constituent le cœur du réseau. Les fonctions d'activation utilisées sont de type sigmoïde.

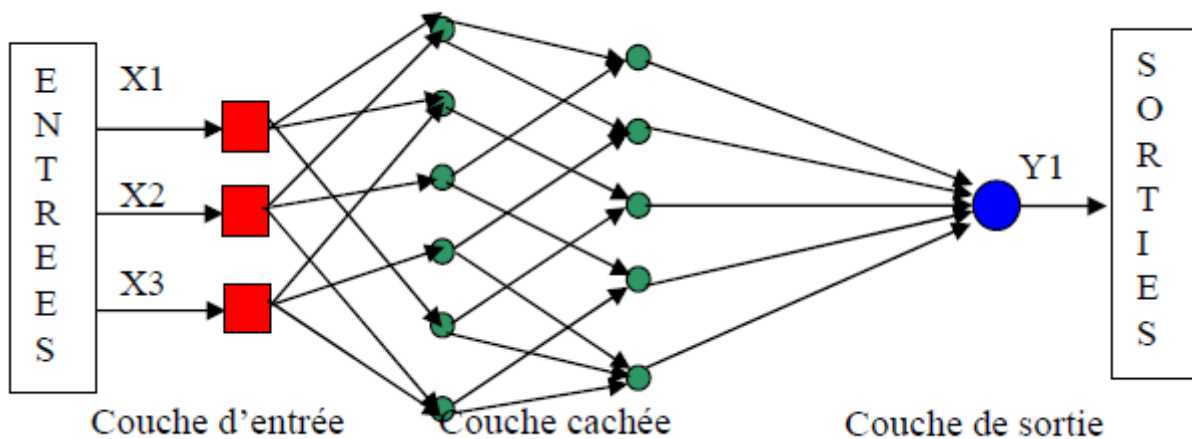


Figure III.4 : Réseaux de neurones non bouclés

III.2.5.2 Réseaux de neurones bouclés

Les réseaux récurrents (recurrent network) possèdent une structure similaire à celle des réseaux unidirectionnels mais complétée par des connexions entre les éléments de la même couche ou vers des couches amont. Ces réseaux sont assez puissants, car leur fonctionnement est séquentiel et adopte un comportement dynamique.

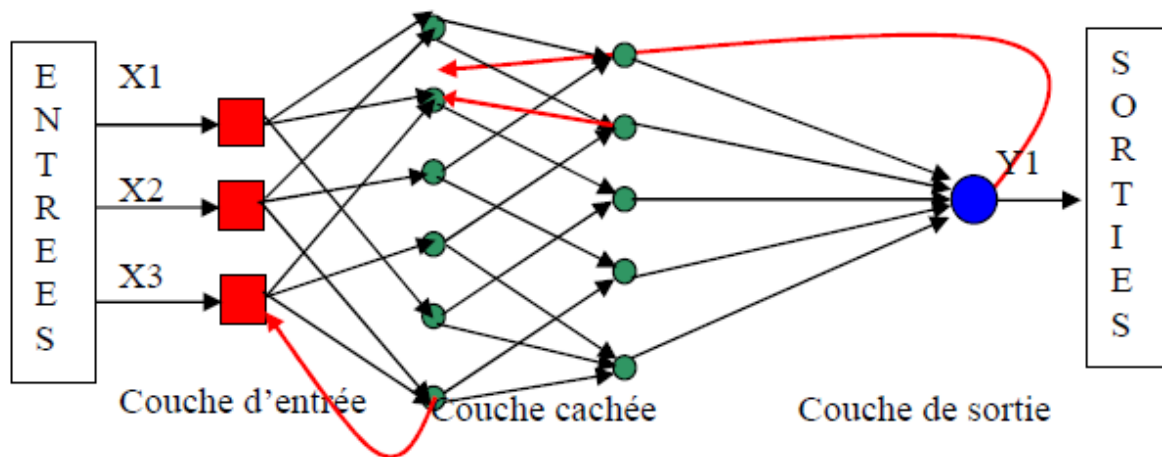


Figure III.5 : Réseaux de neurones bouclés

III.2.6 Apprentissage dans les réseaux de neurones

III.2.6.1 Notions d'apprentissage

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux. Elle ne concerne cependant pas tous les modèles, mais les plus utilisés. L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement.

Durant cette phase de fonctionnement, le réseau adapte sa structure (le plus souvent, les poids des connexions) afin de fournir sur ses neurones de sortie les valeurs désirées. Cet apprentissage nécessite des exemples désignés aussi sous l'appellation d'échantillon d'apprentissage, ainsi qu'un algorithme d'apprentissage. Après initialisation des poids du réseau (en général des valeurs aléatoires), il y a présentation des exemples au réseau et calcul des sorties correspondantes. Une valeur d'erreur ou de correction est calculée et une correction des poids est appliquée.

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissages. Dans le cas de l'apprentissage supervisé, les exemples sont des couples (entrée, sortie associée) alors que l'on ne dispose que des valeurs (entrée) pour l'apprentissage non supervisé. Remarquons cependant que les modèles à apprentissage non

supervisé nécessitent avant la phase d'utilisation une étape de labellisation effectuée par l'opérateur, qui n'est pas autre chose qu'une part de supervision [16].

III.2.6.2 Types d'apprentissage

III.2.6.2.1 Apprentissage non supervisé

L'apprentissage est qualifié de non supervisé lorsque, seules les valeurs d'entrées sont disponibles. Il repose sur un critère interne de conformité du comportement du réseau par rapport à des spécifications générales et non sur des observations externes. Dans ce cas, l'apprentissage est basé sur des probabilités, le réseau va se modifier en fonction des régularités statistiques de l'entrée. L'algorithme d'apprentissage exploite pour se faire une mesure prédéterminée de la qualité de représentation de la connaissance afin d'ajuster les paramètres du réseau [17].

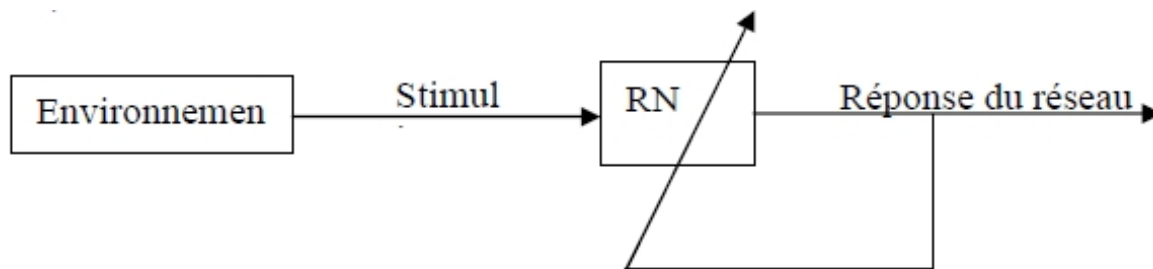


Figure III.6 apprentissage non supervisé

III.2.6.2.2 Apprentissage supervisé

L'apprentissage supervisé consiste à présenter au réseau une série d'exemples sur ces entrées, ainsi le réseau calcule sur ses sorties un vecteur d'état qui est comparé au vecteur des états désirés. Une erreur est calculée entre le vecteur des sorties désirées et le vecteur des états calculés par le réseau. L'algorithme d'apprentissage adapte les poids du réseau de façon à corriger son comportement comme le montre la figure (III.7)

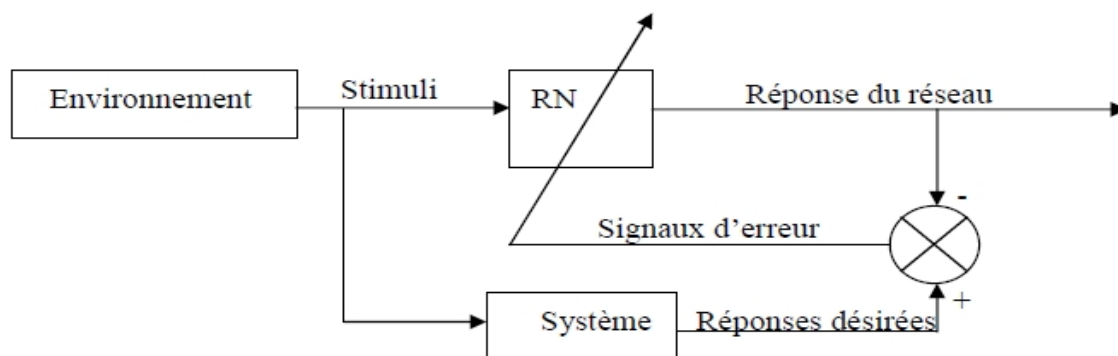


Figure III .7 apprentissage supervisé

III.2.7 Algorithme de retro-propagation du gradient

L'apprentissage consiste à modifier le poids des connections entre les neurones. Il existe plusieurs règles et d'algorithmes de modifications, l'algorithme le plus utilisé est celui de la retro-propagation du gradient d'erreur. Il a été créé en généralisant les règles d'apprentissage de Widrow-Hoff, aux réseaux multicouches à fonction de transfert non linéaire.

La retro-propagation du gradient de l'erreur est utilisée pour ajuster les poids et les biais du réseau afin de minimiser l'erreur quadratique entre la sortie du réseau et la sortie réelle.

A chaque couple entrée / sortie, une erreur est calculée, le gradient, ou pente de l'erreur est déterminé, ensuite les poids et les biais sont modifiés en ligne sur le réseau, on réitère ces calculs jusqu'à l'obtention du critère d'arrêt.

Soit une réponse spécifique désirée D_p à chaque vecteur d'entrée X_p , la mise à jours des coefficients synaptiques W s'effectue progressivement jusqu'à ce que l'erreur entre les sorties du réseau et les résultats désirés sont minimisés. Pour se faire, considérons le problème de minimisation de la fonction coût défini pour n exemples $(X_p; D_p)$ et le Y_p la sortie de réseau, par l'erreur qui peut être quadratique de la forme [16] :

$$E(W) = \sum_p (Y_p - D_p)^2 \quad (\text{III.1})$$

L'apprentissage revient à déterminer $W = \text{ArgMin}(E)$

En général, on utilise des méthodes de gradient partiel, total ou stochastique :

$$W(t+1) = W(t) - \lambda \text{grad}_w(E) \quad (\text{III.2})$$

$$\text{Gradient partiel: } W(t+1) = W(t) + \lambda (D_p - f(WX_p)) f'(WX_p) X_p \quad (\text{III.3})$$

$$\text{Gradient total : } W(t+1) = W(t) - \lambda \cdot \text{grad}_w(E(t)) \quad (\text{III.4})$$

$$\text{Gradient stochastique : } W(t+1) = W(t) - \lambda \cdot \text{grad}_w(E_p(t)) \quad (\text{III.5})$$

Où $E_p = (Y_p - D_p)^2$, erreur calculée sur un seul exemple que l'on tire au hasard à chaque t .

f : fonction d'activation.

$\lambda(t)$: pas du gradient (fixe, décroissant, ou adaptatif)

Pour calculer $\frac{dE_p}{dw_p}$, on exploite la rétro-propagation du gradient, qui est une méthode

astucieuse de calcul de gradient de la fonction coût en utilisant la dérivation composée pour retro-propager l'erreur, comme le montre la figure suivante :

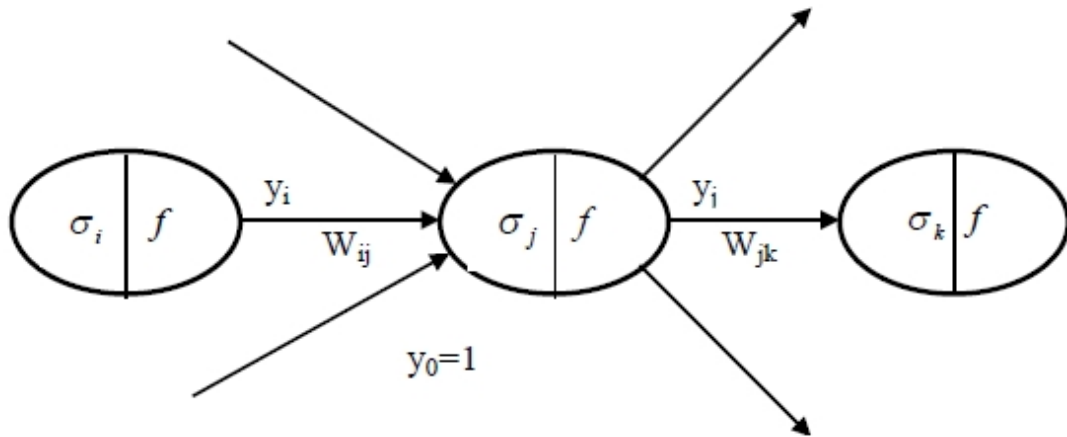


Figure III.8 la retro-propagation

$$\frac{dE_p}{dw_{ij}} = \frac{dE_p}{d\sigma_j} \frac{d\sigma_j}{dw_{ij}} = \left(\frac{dE_p}{d\sigma_j} \right) y_i \quad (\text{III.6})$$

On pose $\delta_j = \frac{dE_p}{d\sigma_j}$,

D'où $w_{ij}(t+1) = w_{ij}(t) - \lambda y_i \delta_j$ (III.7)

Mais

$$\delta_j = \frac{dE_p}{d\sigma_j} = \sum_k \left(\frac{dE_p}{d\sigma_k} \right) \left(\frac{d\sigma_k}{d\sigma_j} \right) = \sum_k \delta_k \left(\frac{d\sigma_k}{d\sigma_j} \right) = \sum_k \delta_k W_{jk} \left(\frac{dy_j}{d\sigma_j} \right) \quad (\text{III.8})$$

D'où $\delta_j = \left(\sum_k \delta_k W_{jk} \right) \dot{f}(\sigma_j)$ si neurone j est caché (III.9)

Et $\delta_j = \left(\frac{dE_p}{d\sigma_j} \right) = \left(\frac{dE_p}{dy_j} \right) \left(\frac{dy_j}{d\sigma_j} \right)$ (III.10)

D'où $\delta_j = 2(y_j - D_j) \dot{f}(\sigma_j)$ (III.11)

Cependant δ_j se calcule de proche en proche par rétro-propagation de l'erreur.

III.3 Les systèmes flous

Les systèmes flous peuvent être considérés comme des systèmes logiques qui utilisent des règles linguistiques pour établir des relations entre leurs variables d'entrée et de sortie. Ils sont apparus pour la première fois dans les années soixante-dix avec des applications dans le domaine du contrôle des processus.

Aujourd'hui, les applications des systèmes flous sont très nombreuses outre la commande, ils sont utilisés aussi pour la modélisation. Nous présentons brièvement quelques notions de base de ces systèmes.

III.3.1 Notions de base et connaissances préliminaires

III.3.1.1 Les variables linguistiques

La notion de variable linguistique permet de modéliser les connaissances imprécises ou vagues sur une variable dont la valeur précise est inconnue. Une variable linguistique, ou variable floue, est donc une variable dont les valeurs floues appartiennent à des ensembles flous pouvant représenter des mots du langage naturel. Ainsi une variable floue peut prendre simultanément plusieurs valeurs linguistiques. Le domaine sur lequel ces termes et ces variables sont définis, constitue l'univers de discours. Le découpage de cet univers de discours par les termes flous est appelé une partition floue. Lorsque l'univers de discours est totalement recouvert par les termes flous, et que pour toutes valeurs, la somme des degrés d'appartenance est égale à 1, on parle alors de partition floue forte.

La variable linguistique peut être représentée par un triplet $(x, T(x), U)$ dans lequel x est le nom de la variable linguistique, $T(x)$ l'ensemble des valeurs linguistiques de x et U l'univers de discours. La figure III.9 illustre un exemple de variable linguistique 'vitesse' avec trois termes linguistiques : petite, moyenne et grande. [26]

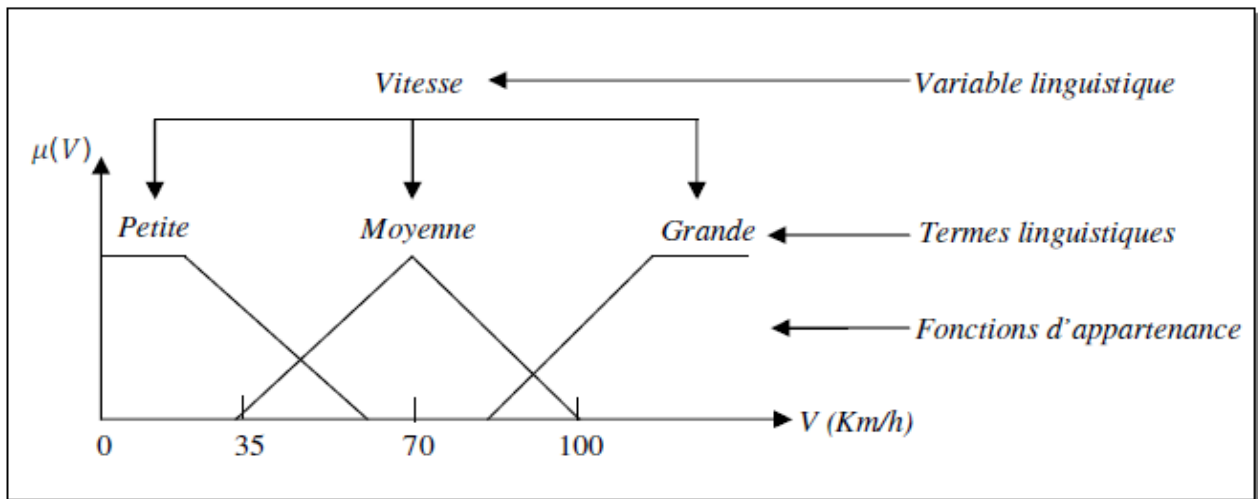


Figure. III.9 : Variable linguistique

III.3.1.2 Les fonctions d'appartenance

Soit un ensemble E et un sous-ensemble A de E ($A \subset E$), et x un élément de E appartenant à A ($x \in A$). Pour illustrer cette caractéristique, on utilise la fonction d'appartenance $\mu_A(x)$ compris entre 0 et 1, qui représente le degré d'appartenance de x à

l'ensemble flou A . Le plus souvent, la fonction d'appartenance est déterminée par l'une des fonctions suivantes (figure III.10) : [26]

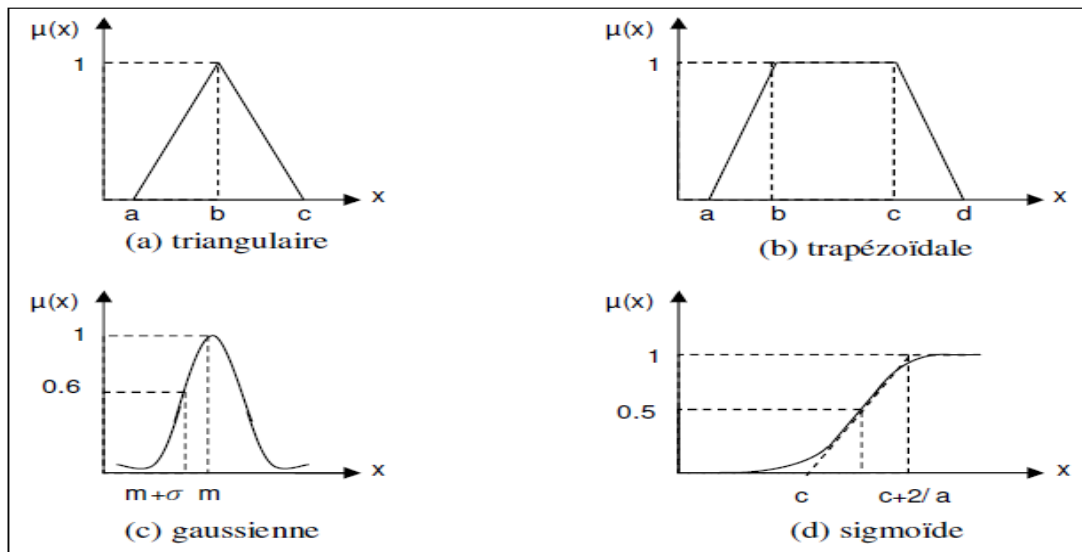


Figure III.10 : Formes usuelles des fonctions d'appartenance

III.3.1.3 Règles d'inférence [27]

On appelle règles d'inférence l'ensemble des différentes règles reliant les variables floues d'entrée d'un système aux variables floues de sortie de ce système à l'aide des différents opérateurs. Ces règles se présentent sous la forme :

Si condition 1 ET/OU condition 2 (ET/OU) alors action sur les sorties OU

Si condition 3 ET/OU condition 4 (ET/OU) alors action sur les sorties OU

⋮

Si condition n ET/OU condition n + 1 (ET/OU) alors action sur les sorties OU

III.3.1.4 Les opérateurs en logique floue [27]

Soient A et B deux ensembles flous dans U ayant respectivement μ_A , μ_B comme fonctions d'appartenance. L'union, l'intersection et la complémentation des ensembles flous sont définis à l'aide de leurs fonctions d'appartenances.

Union : L'union de deux sous-ensembles flous A et B de E est un sous-ensemble flou $A \cup B$ qui est défini par le plus grand sous-ensemble flou qui contient A et qui contient B . Sa fonction d'appartenance est donnée par :

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] ; \forall x \in E \quad (\text{III.12})$$

Intersection : L'intersection de deux sous-ensembles flous A et B de E est un sous ensemble flou $A \cap B$ qui est défini par le plus petit sous-ensemble contenu à la fois dans A et dans B . Sa fonction d'appartenance est donnée par :

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] ; \forall x \in E \quad (\text{III.13})$$

Complément : Le complément d'un sous-ensemble flou A de E est un ensemble flou dénoté par \bar{A} dont la fonction d'appartenance est donnée par :

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) ; \forall x \in E \quad (\text{III.14})$$

Inclusion : Soit les deux sous-ensembles flous A et B de l'ensemble E , on dira que A est inclus dans B si :

$$\forall x \in E : \mu_A(x) \leq \mu_B(x) \quad (\text{III.15})$$

Et on notera alors : $A \subset B$

III.3.2 Structure interne d'un contrôleur flou [26] [28] [29]

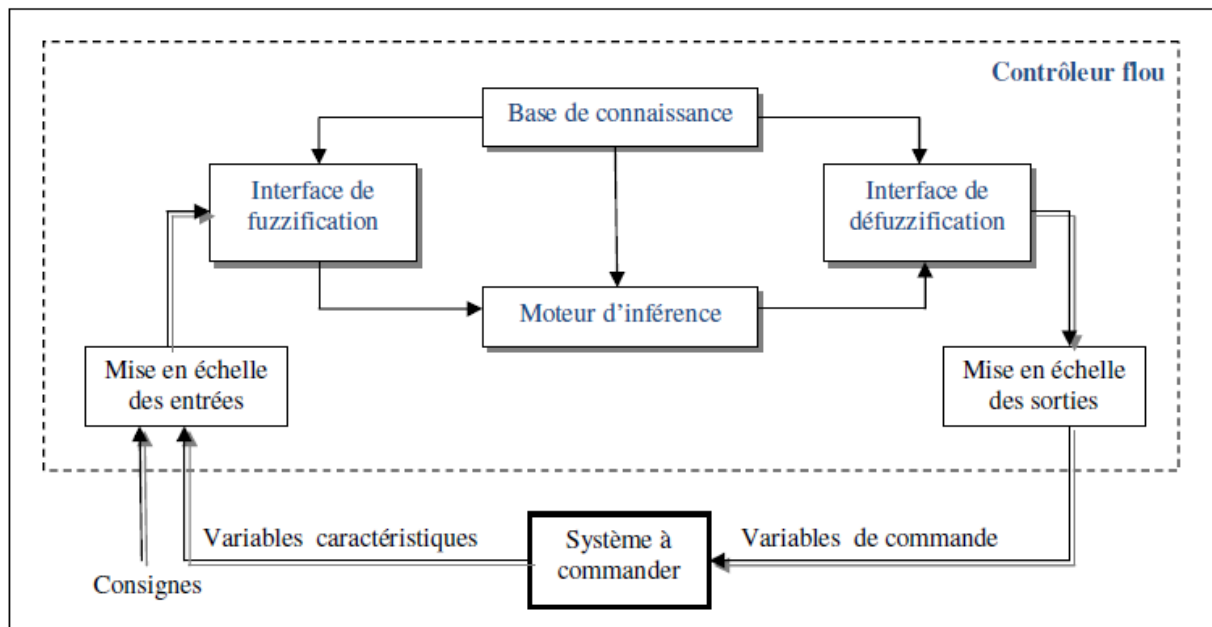


Figure III.11 : Structure de base d'un régulateur flou

Les variables caractéristiques du système à commander et les consignes définissent les variables d'entrées 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 système. Les variables de sortie du contrôleur flou sont les commandes appliquées au processus. La Mise en échelle (normalisation/dénormalisation) des grandeurs d'entrées/sorties permet d'adapter le traitement des signaux d'entrées/sorties. Par convention, la plage de variation des variables d'entrées/sorties est comprise entre -1 et +1.

Les opérations de normalisation et de dénormalisation sont optionnelles.

La configuration de base du contrôleur flou comprend quatre parties :

III.3.2.1 Base de connaissance

La conception d'une base de connaissances représente la phase dans la conception des systèmes experts. Elle comprend la base de données et la base des règles floues.

III.3.2.1.1. La base de données

Elle contient la définition des ensembles flous, les facteurs d'échelle pour la normalisation des ensembles de référence et la partition de l'espace flou d'entrée et sortie.

III.3.2.1.2 La base des règles floues

Elle rassemble l'ensemble des règles floues de type « Si-Alors » décrivant en termes linguistiques basées sur la connaissance d'un expert le comportement dynamique du système :

$R_1 : \text{Si } x_1 \text{ et } A_1^l \text{ et } \dots \text{ et } x_n \text{ et } A_n^l \text{ alors } \mu_1 \text{ et } B^l$

Avec : $[x_1, \dots, x_n]$: les entrées du régulateur.

μ_1 :: la sortie du régulateur.

III.3.2.2 La fuzzification

Le processus de la fuzzification consiste simplement à attribuer aux différentes variables linguistiques d'entrée des fonctions d'appartenance convenables. Le choix des formes des fonctions d'appartenance est arbitraire. La forme la plus fréquemment utilisée en commande floue est la forme triangulaire. Le nombre de fonctions d'appartenance est généralement impair et se répartissent autour de zéro. En général, on introduit pour une variable linguistique trois, cinq ou sept ensembles flous. Le choix du nombre dépend de la précision souhaitée. Les fonctions d'appartenance peuvent être symétriques, non symétriques et équidistantes ou non équidistantes.

Cette opération est nécessaire vu que la mesure de la sortie y et la donnée de la référence ou consigne sont toujours exprimées en des variables qui sont nettes et précises et que le régulateur flou ne peut traiter que des variables floues.

III.3.2.3 L'inférence

C'est un mécanisme de décision. Il permet à partir d'un fait observé de la base des règles floues une décision en exploitant le raisonnement approximatif. Dans les inférences de régulateur par logique floue interviennent les opérateurs ET et OU. L'opérateur ET s'applique aux variables à l'intérieur d'une règle tandis que l'opérateur OU lie les différentes règles.

A cause de l'empiètement des fonctions d'appartenance, en général deux ou plusieurs règles sont activées en même temps. Ce fait doit être pris en considération lors de la réalisation de l'opérateur OU.

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

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

- Méthode d'inférence max-min (Mamdani)
- Méthode d'inférence max-prod (Larsen)
- Méthode d'inférence somme-prod (Sugeno)

III.3.2.4 La défuzzification

Les méthodes d'inférence fournissent une fonction d'appartenance résultante, $\mu_{Res}(x_R)$ pour la variable de sortie x_R , il s'agit donc d'une information floue. Par cette étape, se fait alors le retour aux grandeurs de sortie réelles. Il s'agit à cet effet, de calculer à partir des degrés d'appartenance à tous les ensembles flous de la variable de sortie, l'abscisse qui correspond à la valeur de cette sortie. Cette transformation est appelée défuzzification. Dans la commande en temps réel, un critère de choix de la commande de défuzzification est la simplicité de calcul. Plusieurs stratégies de défuzzification existent, les plus utilisées sont:

- Méthode de centre de gravité.
- Méthode du maximum.
- Méthode de la moyenne des maxima.

III.3.3 Modèles d'inférence floue

III.3.3.1 Système d'inférence floue de type Mamdani [30]

En 1974, E.H Mamdani avait présenté, pour la première fois, la technique de commande par logique floue. Celle-ci consiste à déterminer un ensemble de règles qui maîtrise le comportement dynamique du système à commander. L'obtention de ces règles est facile auprès des experts qui connaissent bien le système. Il avait utilisé des règles à prémisses et conclusions symboliques, l'inférence (max, min), et la défuzzification par centre de gravité. La forme de l'implication floue définie par ce type de contrôleur est de la forme :

Si x_1 est A et x_2 est B alors y est C

La conséquence de ce type de système est une valeur floue.

III.3.3.2 Système d'inférence flou de type Takagi-Sugeno [31]

Dans ces systèmes, les prémisses des règles sont exprimées symboliquement et les conclusions sont exprimées par des fonctions linéaires. Notons par $x = [x_1, x_2, \dots, x_n]^T$ les entrées du contrôleur flou, et par y sa sortie. Pour chaque x_i est associé m_i ensemble flou F_i^j dans X_i tel que $x_i \in X_i$. La base de règles complète du contrôleur flou comporte $N = \prod_{i=1}^n m_i$ règles floues de la forme:

R_K : Si x_1 est F_1^K et x_2 est F_2^K et ... et x_n est F_n^K alors $y = f_K(x)$; $K = 1, 2, \dots, n$

En général $f_K(x)$ est une fonction polynomiale en fonction des variables d'entrées :

$$f_K(x) = a_0^K + \sum_{i=1}^n a_i^K x_i \quad (\text{III.16})$$

III.4 Systèmes Neuro-flous

Les systèmes Neuro-flous permettent de combiner les avantages de deux techniques complémentaires. Les systèmes flous fournissent une bonne représentation des connaissances. L'intégration de réseaux de neurones au sein de ces systèmes flous améliore leurs performances grâce à la capacité d'apprentissage de réseaux de neurones. Inversement, l'injection des règles floues dans les réseaux de neurones, souvent critiques pour leur

manque de lisibilité, clarifie la signification des paramètres du réseau et facilite leur initialisation, ce qui représente un gain de temps de calcul considérable pour leur identification.

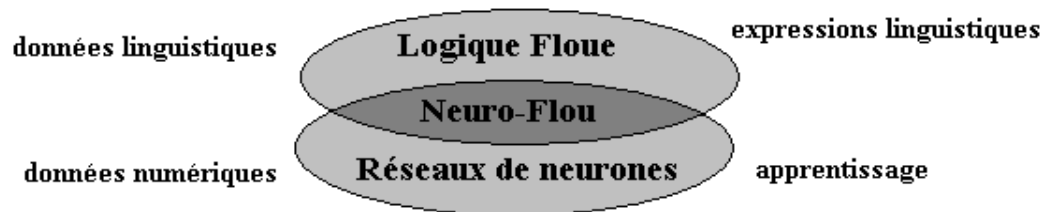


Figure III.12 : Principe du système neuro-flou.

Afin de clarifier les définitions, nous proposons dans ce chapitre une présentation brève des quelques types des systèmes neuro-flous.

III.4.1 Définition

Les systèmes neuro-flous sont des systèmes flous formés par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones. La technique d'apprentissage opère en fonction de l'information locale et produit uniquement des changements locaux dans le système flou d'origine comme il est montré sur la figure suivante.

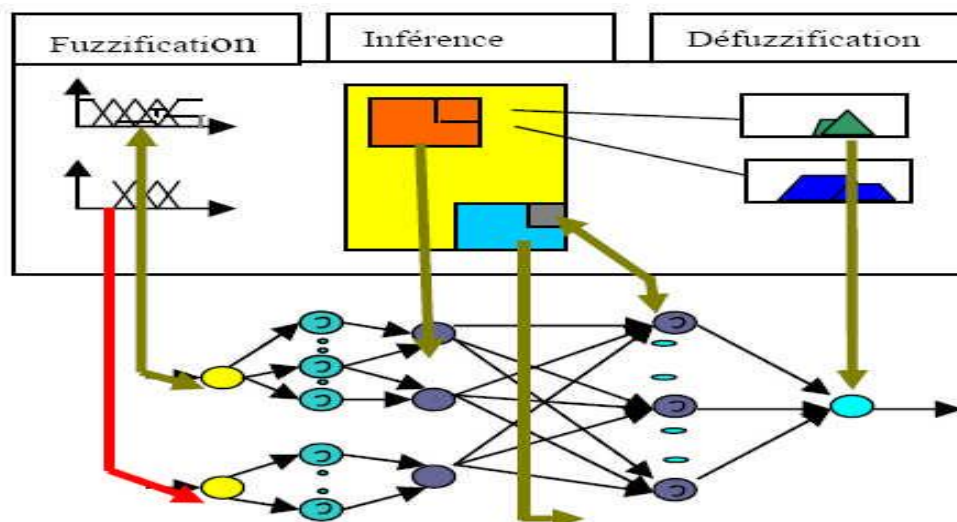


Figure III.13 : Principe de fonctionnement d'un Réseau neuro-flou.

III.4.2 Méthodes des combinaisons neuro-floues [32]

Il existe quatre grandes catégories de combinaisons des réseaux de neurones avec la logique floue :

III.4.2.1. Réseau flou neuronal

Dans ces réseaux, les techniques floues sont employées pour augmenter les possibilités du processus d'apprentissage et d'exécution des réseaux de neurones.

III.4.2.2. Système neuronal/flou simultanément

Le réseau de neurones et le système flou fonctionnent ensemble sur la même tâche, mais sans s'influencer, c.à. d. ni l'un ni l'autre n'est employé pour déterminer les paramètres de l'autre. Habituellement le réseau neuronal traite les entrées, ou post-traite les sorties du système flou.

III.4.2.3 Modèles neuro-flous coopératifs

Le réseau de neurones est employé pour déterminer les paramètres (les règles et les ensembles flous) d'un système flou. Après la phase d'apprentissage, le système flou fonctionne sans le réseau de neurone. C'est une forme simple des systèmes neuro-flous.

III.4.2.4. Modèles neuro-flous hybrides

Les approches neuro-floues modernes sont de cette forme. Un réseau neuronal et un système flou sont combinés dans une architecture homogène. Le système peut être interprété comme un réseau neuronal spécial avec des paramètres flous ou comme un système flou mis en application sous une forme distribuée parallèle.

Diverses associations des méthodes et architectures des systèmes neuro-flous hybrides ont été développées depuis 1988. La figure suivante montre quelques architectures : FALCON (a) et GARIC (b) interprètent la règle floue avec une structure à 5 couches, NEFCLASS (c) interprète la règle floue avec une structure à 3 couches et ANFIS (d) interprète la règle floue avec une structure à 6 couches. [32]

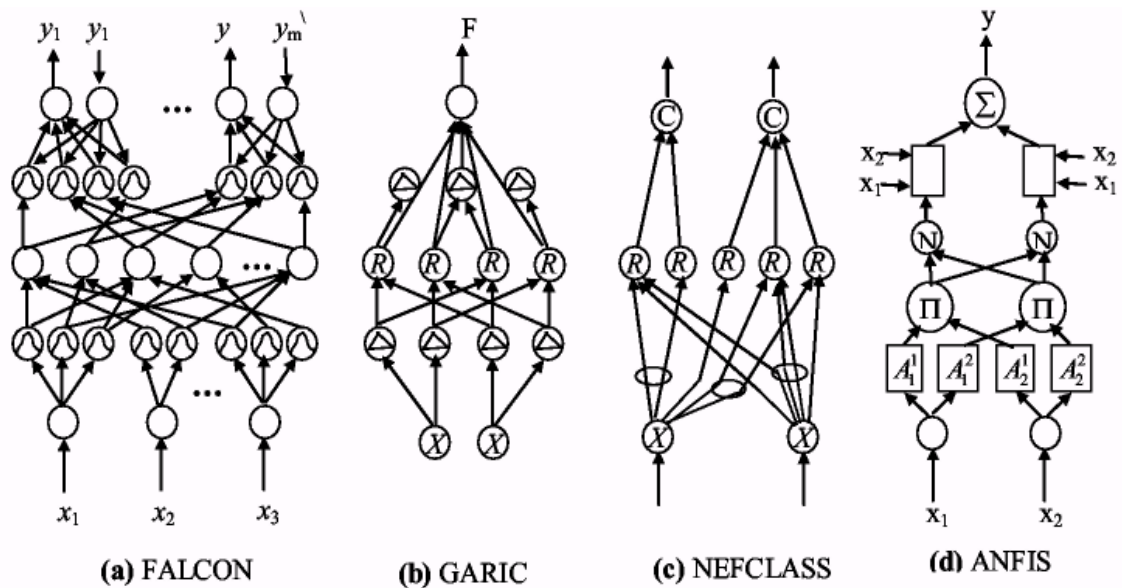


Figure III.14 : Différents types des systèmes neuro-flous

- **FALCON (Fuzzy Adaptive learning Control Network)**

Il s'agit de modèles à 5 couches, utilisant la fuzzification en entrée et la défuzzification en sortie. Ceci correspond à l'interprétation juste de la technique de Mamdani. La précision accrue des résultats provoque une lenteur dans l'exécution du système. Ce modèle est rarement utilisé en pratique mais il reste le meilleur pour la commande.

- **NEFCLASS**

Ce modèle est utilisé généralement en classification, il est constitué de 3 couches : Une couche d'entrée avec les fonctions d'appartenance, une couche cachée représentée par des règles et une couche de sortie définissant les classes. Ce modèle est facile à mettre en application, il évite l'étape de défuzzification, tout en étant précis dans le résultat final, avec une rapidité bien supérieure aux autres architectures.

- **NEFPROX (Neuro Fuzzy function approximator) :**

Modèle obtenu par l'association des deux architectures NEFCLASS et NEFCON, il est utilisé dans différentes applications comme la classification et l'approximation de fonctions. NEFCLASS utilise un algorithme supervisé pour définir les règles floues, le NEFCON

utilise un algorithme d'apprentissage non supervisé avec le calcul de l'erreur de sortie. Les deux modèles emploient la rétro propagation afin de définir les sous-ensembles flous. Comparé au modèle ANFIS, NEFPROX est beaucoup plus rapide, mais ANFIS donne de meilleurs résultats en approximation. Le NEFPROX est, le premier système interprétable et lisible, dédié à l'approximation de fonction. Néanmoins, ses résultats en classification restent moins bons que ceux donnés par le NEFCLASS.

- **ANFIS (Adaptive-Neural-based Fuzzy Inference System):**

ANFIS représente un système à inférence floue mis en application dans le cadre des réseaux adaptatifs. Il utilise la procédure d'apprentissage hybride. Cette architecture affine les règles floues obtenues par des experts humains pour décrire le comportement d'entrée-sortie d'un système complexe. Ce modèle donne de très bons résultats en poursuite de trajectoire, approximation non linéaire, commande dynamique et traitement du signal.

Un système ANFIS réalise une approximation linéaire de la variable de sortie en décomposant l'espace des entrées en différents espaces flous.

III.4.3 Architecture de l'ANFIS [33]

ANFIS (*Adaptive Network Based Fuzzy Inference System*) est un système d'inférence adaptatif neuro-flou qui consiste à utiliser un réseau de neurones à 5 couches pour lequel chaque couche correspond à la réalisation d'une étape d'un système d'inférence floue de type Takagi Sugeno. Pour la simplicité, nous supposons que le système d'inférence flou à deux entrées x et y , et z comme une sortie. Supposons que la base de règle contient deux règles floues de type Takagi-Sugeno.

Règle1 :

$$SI x \text{ est } \mathbf{A1} \text{ et } y \text{ est } \mathbf{B1} \text{ ALORS } z1 = p1 x + q1 y + r1$$

Règle2 :

$$SI x \text{ est } \mathbf{A2} \text{ et } y \text{ est } \mathbf{B2} \text{ ALORS } z2 = p2 x + q2 y + r2$$

L'ANFIS à une architecture définie par cinq couches comme représentée sur la figure suivante :

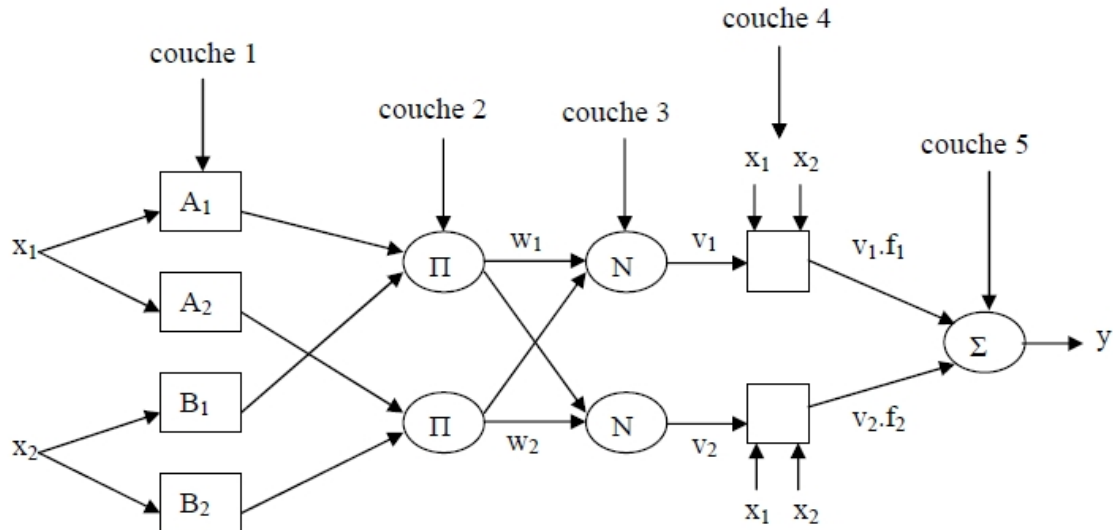


Figure III.15 : L'Architecture de l'ANFIS

- **La première couche (Fuzzification)**

Cette couche (layer 1) permet la "fuzzification" des entrées x et y . Chaque neurone dans cette couche correspond à une variable linguistique. Les entrées x et y sont passées par la fuzzification en utilisant des fonctions d'appartenance des variables linguistiques A_i et B_j , (généralement sont des formes triangulaires, trapézoïdales ou Gaussiennes). Par exemple la fonction d'appartenance Gaussienne est définie par:

$$\mu_{A_i}(x) = \exp \left[-\frac{1}{2} \frac{(x - \bar{x})^2}{\sigma_{x_i}^2} \right] \quad (\text{III.17})$$

$$\mu_{B_j}(y) = \exp \left[-\frac{1}{2} \frac{(y - \bar{y})^2}{\sigma_{y_j}^2} \right] \quad (\text{III.18})$$

Où \bar{x} est le centre et σ la largeur de la fonction d'appartenance.

Les sorties de la première couche sont :

$$x_{1,i} = \mu_{A_i}(x) \quad \text{pour } i = 1,2 \quad (\text{III.19})$$

$$y_{1,j} = \mu_{B_j}(y) \quad \text{pour } j = 1,2 \quad (\text{III.20})$$

Alors, la valeur obtenue $\mu_{A_1}(x)$ représente le degré d'appartenance de la valeur x à l'ensemble A .

- **La deuxième couche (règles)**

Chaque nœud correspond à une T-Norme floue (l'opérateur T-Norme permet de réaliser l'équivalent d'un "ET" booléen). Il reçoit la sortie des nœuds de fuzzification et calcule sa valeur de sortie grâce à l'opérateur produit (cet opérateur est généralement utilisé mais il en existe d'autres : max, min ...).

La fonction d'activation des neurones i de la première couche :

$$w_i = \min \{ \mu_{A_i}(x), \mu_{B_j}(y) \}, i = 1, 2, j = 1, 2 \quad (\text{III.21})$$

Ou

$$w_i = \mu_{A_i}(x) \times \mu_{B_j}(y), i = 1, 2, j = 1, 2 \quad (\text{III.22})$$

- **La troisième couche (normalisation)**

Cette couche normalise les résultats fournis par la couche précédente. Les résultats obtenus représentent le degré d'implication de la valeur dans le résultat final.

$$W_i = \frac{w_i}{\sum_{i=1}^2 w_i} \quad (\text{III.23})$$

L'ensemble des sorties de cette couche est appelé : les poids normalisés.

- **La quatrième couche (conséquents)**

Chaque nœud de cette couche est relié aux entrées initiales. On calcule le résultat en fonction de son entrée et d'une combinaison linéaire du premier ordre des entrées initiales (approche de TAKAGI - SUGENO).

$$f_i^4 = y_i = \bar{w}_i \times (p_i x_1 + q_i x_2 + r_i) \quad (\text{III.24})$$

Où \bar{w}_i est la sortie de la troisième couche, et $\{ p_i, q_i, r_i \}$ sont l'ensemble des paramètres désignés sous le nom : *conséquents*.

- **La couche de sortie (sommation)**

Elle est constituée d'un seul neurone qui calcule la somme des signaux de la couche précédente, alors:

$$y = \sum_{i=1}^2 y_i \quad (\text{III.25})$$

La figure suivante représente un système ANFIS, à 2 entrées chaque entrée répartie en trois sous ensemble floue et 9 règles.

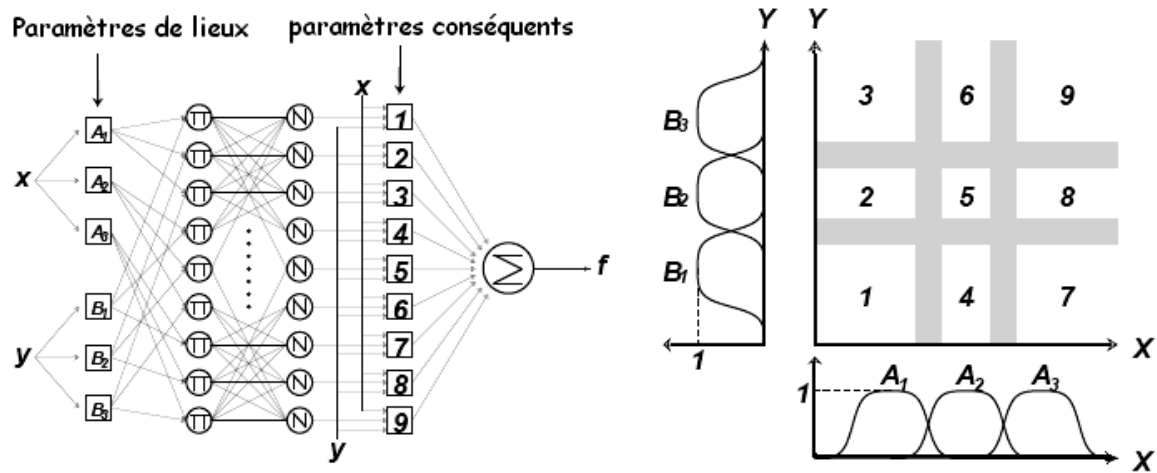


Figure III.16 : Exemple ANFIS à 2 entrées avec 9 règles.

Les différentes couches	Type des couches	Le nombre de neurone dans la couche
Couche 0	Les entrées	n
Couche 1	Fuzzification	$(p \cdot n)$
Couche 2	Les règles	p^n
Couche 3	La normalisation	p^n
Couche 4	Linéarisation des fonctions	p^n
Couche 5	Somme	1

Tableau III.2 : les différentes couches d'un système ANFIS

Tel que :

n : le nombre des entrées.

p : le nombre des sous ensembles flous d'entrée (partition flou).

Noter que les neurones dans ANFIS ont différentes structures:

- Valeurs [fonction d'appartenance définie par différentes formes].
- Règles [habituellement produit].
- Normalisation [division de somme et d'arithmétique].

- Fonctions [régressions linéaires et multiplication avec, tel que W est la normalisation du poids w].
- la sortie [Somme Algébrique].

III.5 Conclusion

Nous avons présenté dans ce chapitre une description générale des réseaux de neurones, la théorie de la logique floue, et les systèmes neuro-flous. Nous nous sommes intéressés principalement aux systèmes ANFIS, alors nous pouvons dire que l'utilisation conjointe des méthodes neuronales et floues permet de tirer l'avantage des qualités de l'une et de l'autre. Principalement, des capacités d'apprentissage des premières, la lisibilité et la souplesse des dernières. L'architecture du système neuro-flou adopté dans ce chapitre montre des bonnes performances dans les simulations des systèmes non linéaires.

Chapitre IV

IV.1 Introduction

Les stratégies de contrôle des robots rigides utilisent comme hypothèse de base le fait qu'un positionnement précis des articulations implique nécessairement un positionnement précis de l'extrémité du robot. Les stratégies utilisées sont donc généralement appliquées localement en implémentant les lois de commandes au niveau de la position et de la vitesse des articulations.

La loi de commande que nous utilisons est une combinaison de deux contrôleurs, le contrôleur PD et le contrôleur neuro flou, afin de suivre la trajectoire de consigne pour les variables articulaires (positions et vitesses) d'un robot manipulateur à deux axes.

IV.2 Stratégie de la commande PD

Soit un robot manipulateur ayant pour équation de mouvement :

$$M\ddot{q}(q, \dot{q}) + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (\text{IV.1})$$

L'objectif de la commande est de déterminer les couples agissant sur les articulations permettant d'effectuer les tâches auxquelles est destiné le robot.

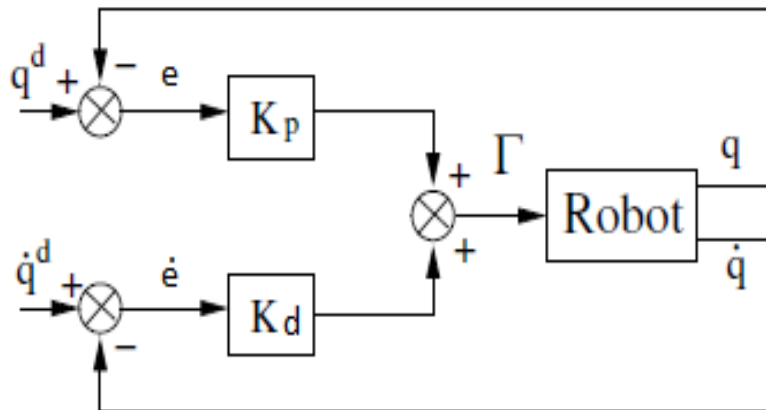


Figure IV.1 commande PD d'un robot manipulateur

La loi de commande est donnée par :

$$\tau_1 = K_{p1}e_1 + K_{d1}\dot{e}_1$$

$$\tau_2 = K_{p2}e_2 + K_{d2}\dot{e}_2$$

On prend $K_p = 180$ et $K_d = 130$

Chapitre IV

Erreur de position

$$e_1 = q_{d1} - q_1$$

$$e_2 = q_{d2} - q_2$$

Erreur de vitesse

$$\dot{e}_1 = \dot{q}_{d1} - \dot{q}_1$$

$$\dot{e}_2 = \dot{q}_{d2} - \dot{q}_2$$

(\dot{q}_d, q_d) Sont la vitesse et la position désirées.

(q, \dot{q}) Sont la position et la vitesse réelles

Et la trajectoire désirée a pour équation :

$$q(t) = q_0 + \left(\frac{q_f - q_0}{2\pi}\right) \left(\frac{2\pi}{t_f} t - \sin \frac{2\pi t}{t_f}\right) \quad (\text{IV.2})$$

Pour l'articulation 1 :

$$q_{01} = -2.35$$

$$q_{f1} = -1.48$$

Pour l'articulation 2 :

$$q_{02} = 2.35$$

$$q_{f2} = 0.52$$

IV.3 résultat de l'application de contrôleur PD

Sans perturbation :

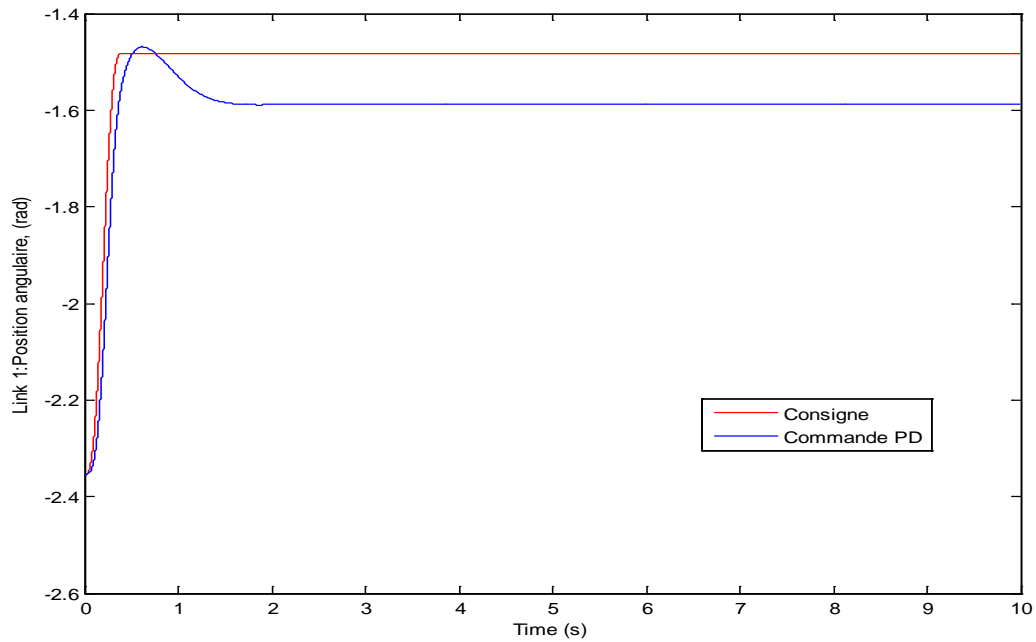


Figure VI.2 Trajectoires réelle et désirée de l'articulation 1

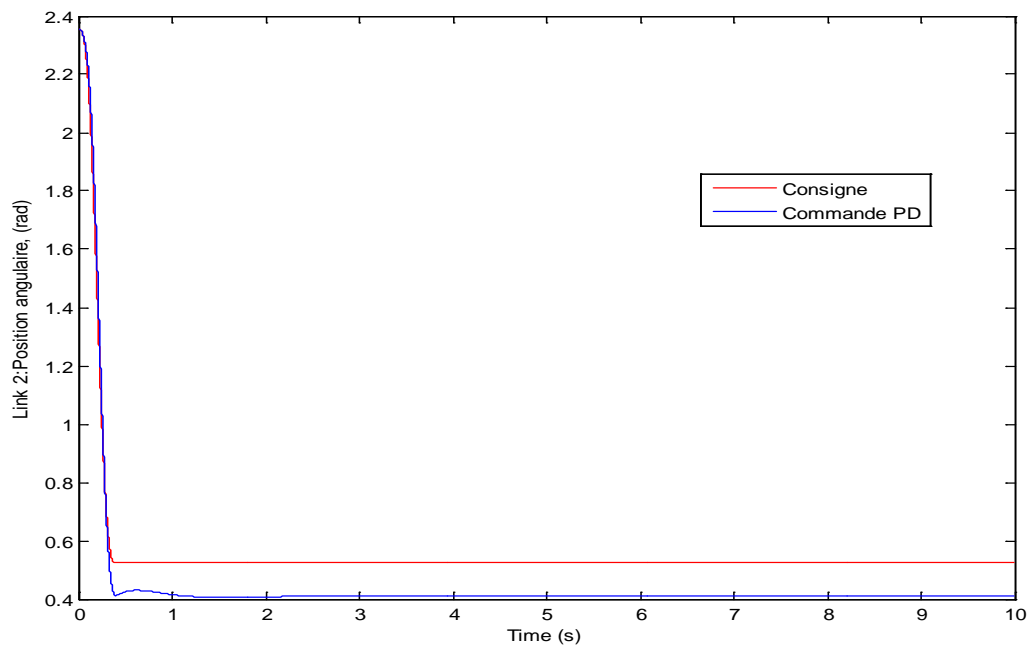


Figure VI.3 Trajectoires réelle et désirée de l'articulation 2

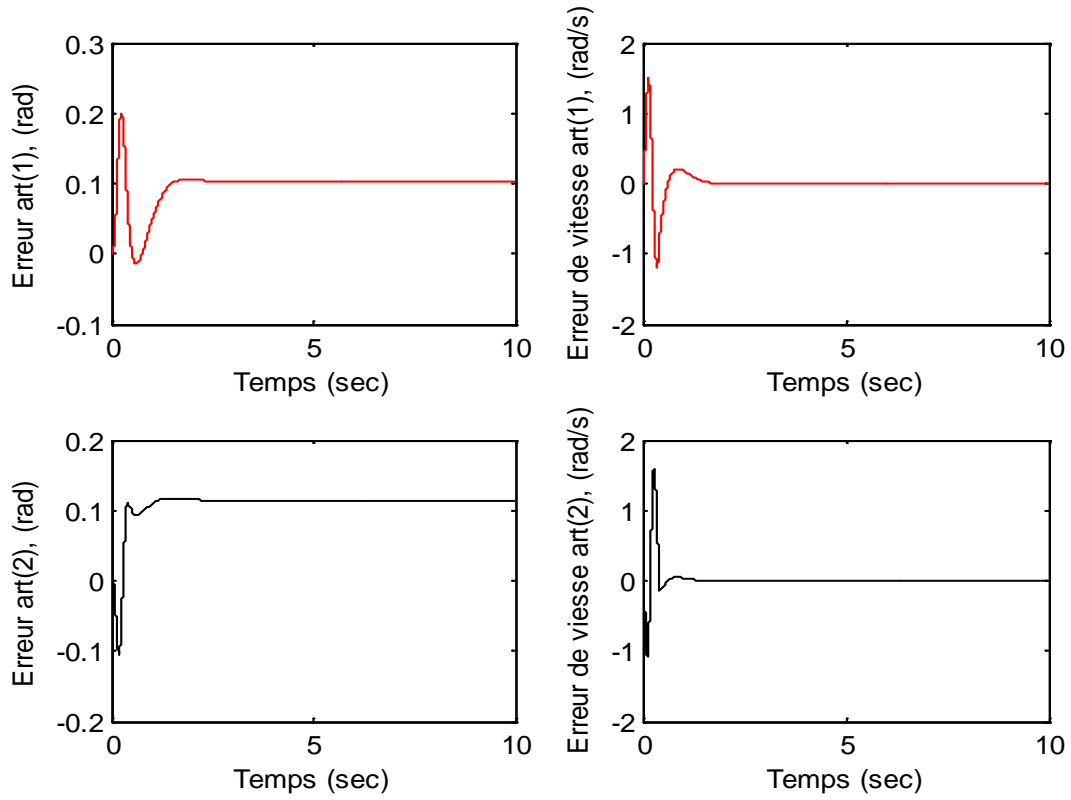


Figure VI.4 Erreurs de positions et de vitesses pour les deux articulations

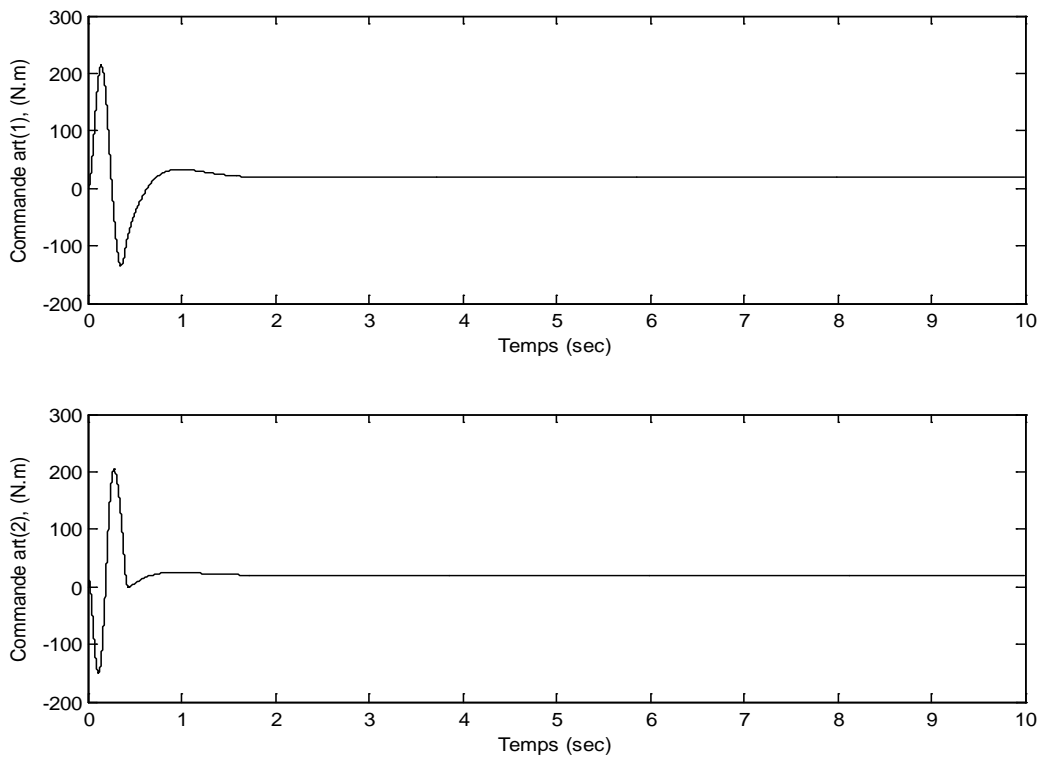


Figure VI.5 Les commandes PD des deux articulations

Avec perturbation paramétrique

Pour tester la robustesse de la commande face aux variations paramétriques, nous avons multiplié la matrice d'inertie par 900, sur l'intervalle [4 s 4.5 s].

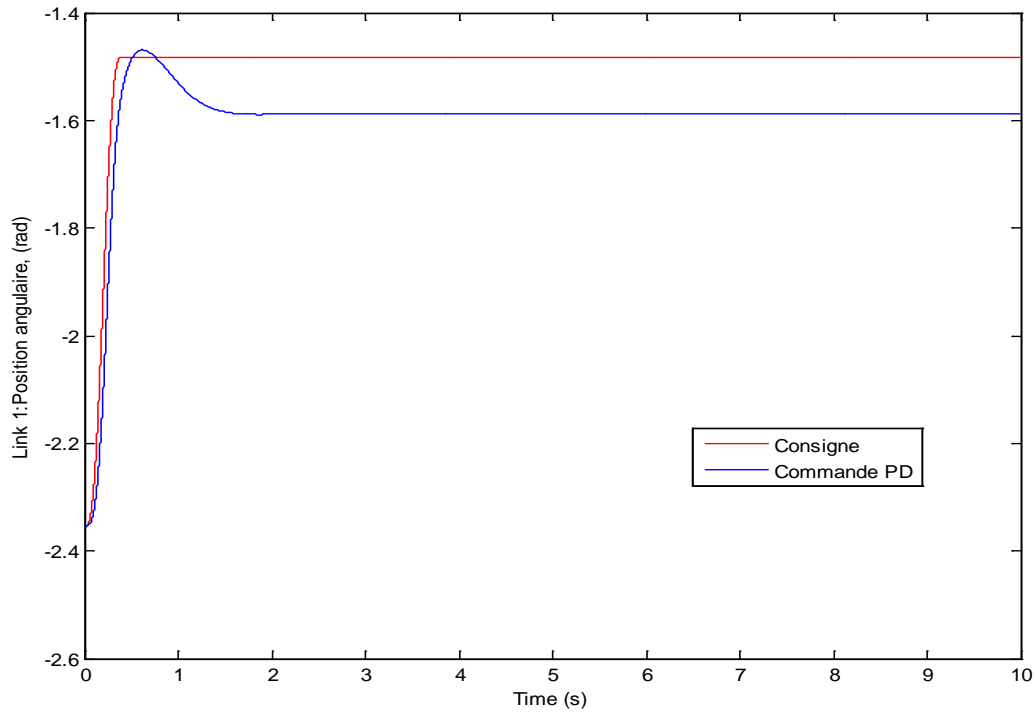


Figure VI.6 trajectoires désirée et réelle de l'articulation 1

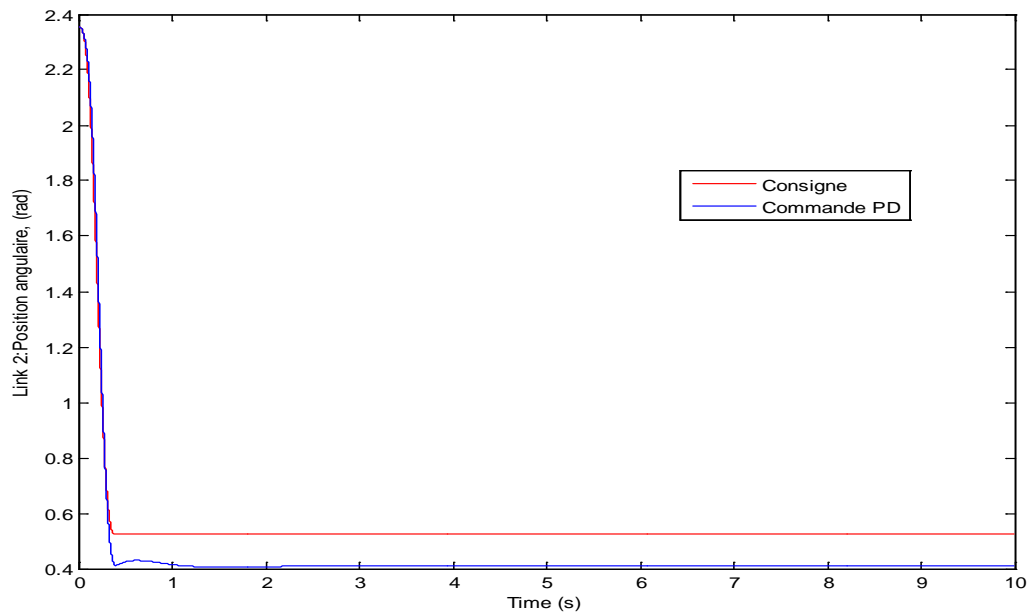


Figure VI.7 trajectoire désirée et réelle de l'articulation 2

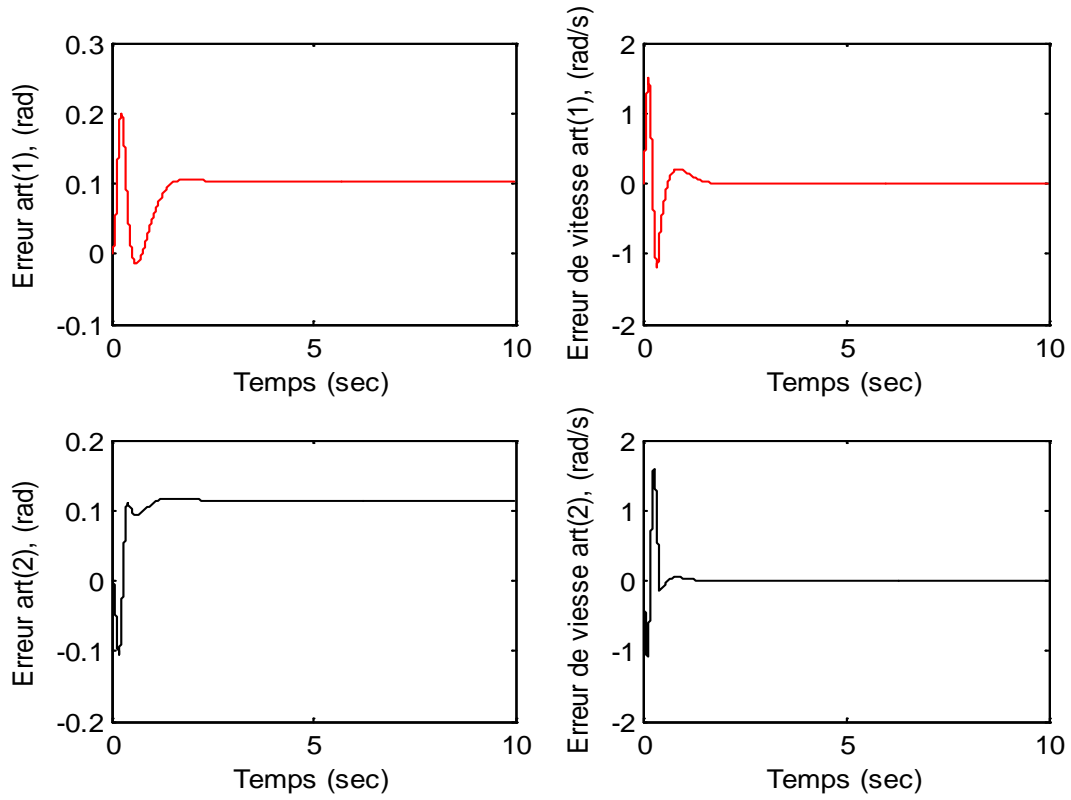


Figure VI.8 Erreurs de position et de vitesse de l'articulation 1 et 2

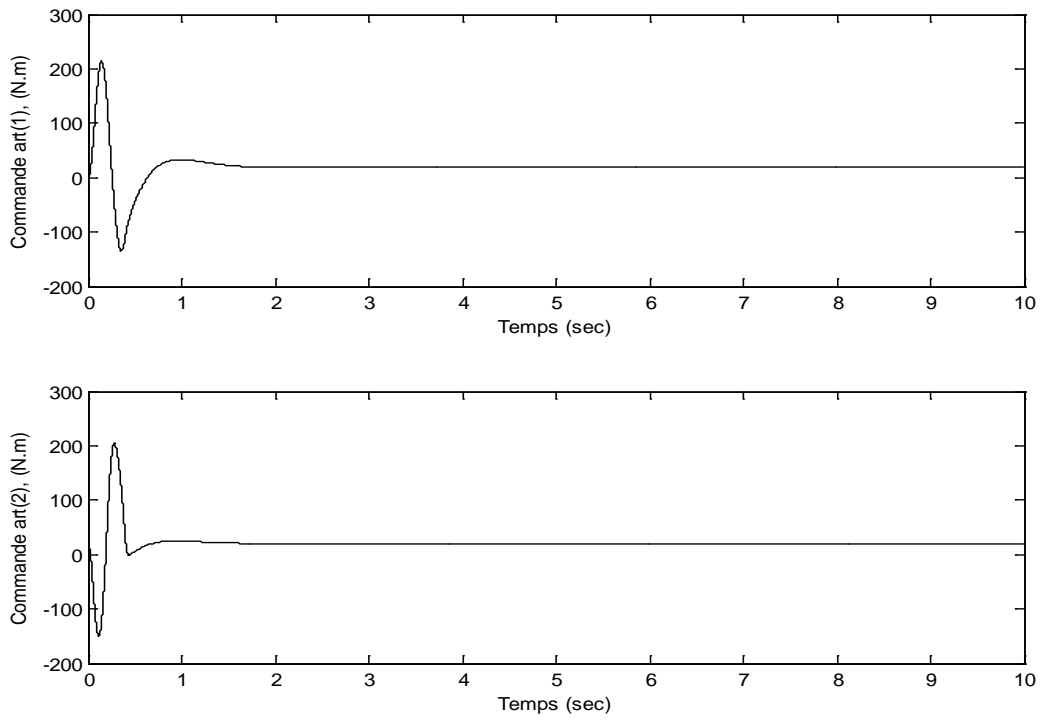


Figure VI.9 les commandes PD des articulations 1 et 2

Avec perturbation externe :

Pour tester la capacité d'adaptation de cette structure de commande nous avons introduit une perturbation externe d'amplitude importante sur l'intervalle [4 s 4.5 s].

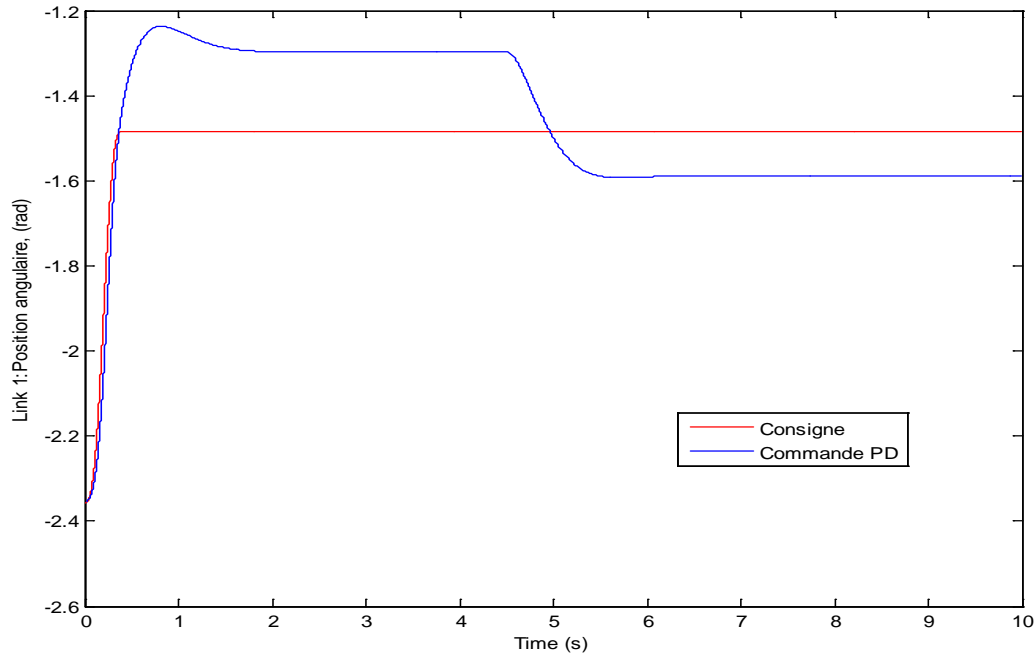


Figure VI.10 trajectoires désirée et réelle de l'articulation 1

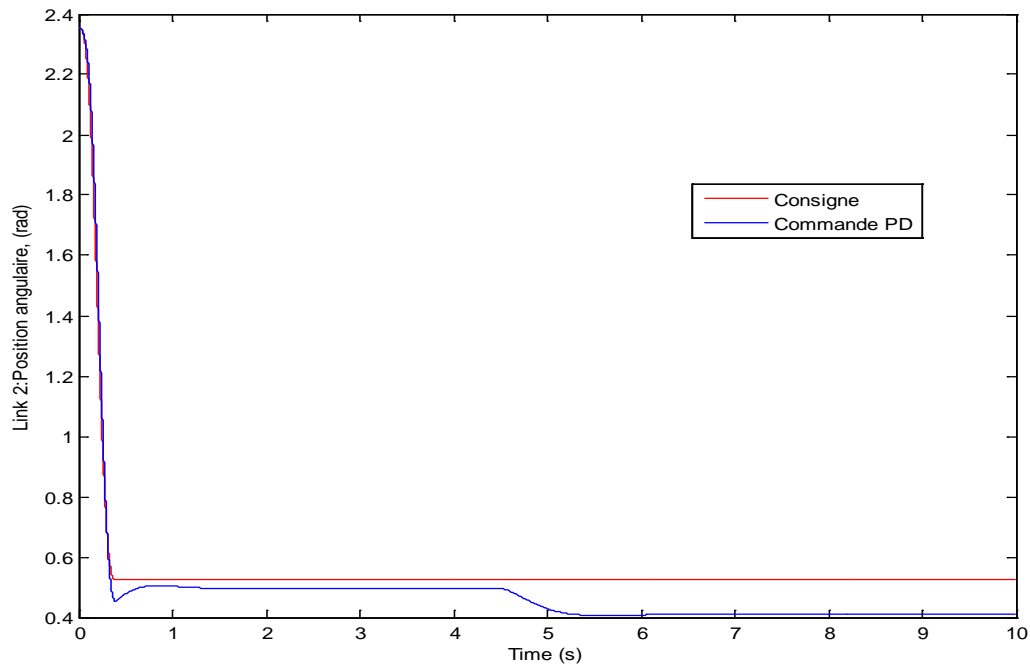


Figure VI.11 trajectoires désirée et réelle de l'articulation 2

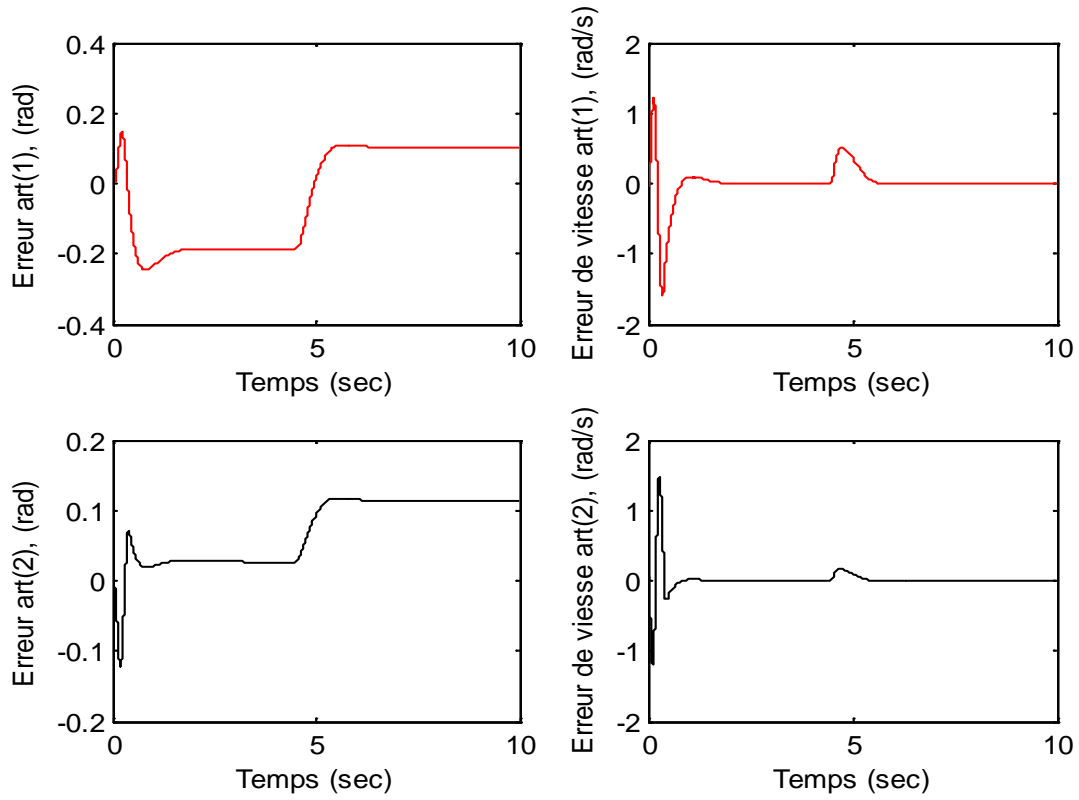


Figure VI.12 Erreurs de positions et de vitesse des deux articulations

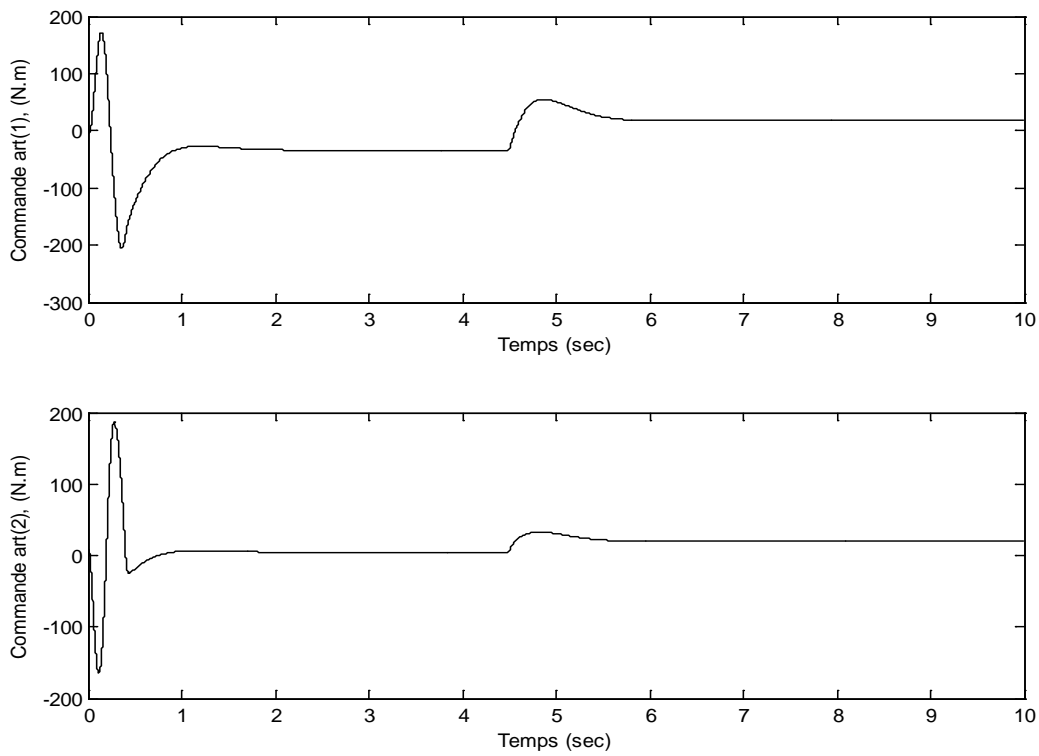


Figure VI.13 commande PD des deux articulations

Interprétations des résultats de simulation obtenus par le régulateur PD

Sans perturbation :

Les résultats de simulation décrivant l'évolution des trajectoires désirées, des erreurs en position, des erreurs en vitesse et des commandes montrent que le suivi de trajectoire (position angulaire) n'est pas vraiment atteint. L'erreur de position atteint 0.1 rad pour la 1^{ère} articulation et légèrement supérieure à 0.1 rad pour la 2^{ème} articulation. Nous remarquons qu'il y a un léger dépassement et le système est un peu lent. L'erreur obtenue sur la vitesse angulaire est très faible pour les deux articulations. Les couples exercés sont d'amplitudes de 18.5 N.m et 20.5 N.m pour l'articulation 1 et l'articulation 2 respectivement.

Avec variation paramétrique :

Nous avons introduit une variation paramétrique sur la matrice d'inertie en la multipliant par 900. Nous constatons que les résultats ne sont pas affectés, ce qui explique que le contrôleur PD est insensible aux variations paramétriques.

Avec perturbation externe :

Avec l'application de la perturbation externe, nous remarquons que les trajectoires réelles se divergent de celles désirées et les erreurs statiques augmentent. Si nous augmentons encore la perturbation, la position angulaire de la deuxième articulation va se dévier dans l'autre sens. Après disparition de la perturbation (à $t=4.5$ s), les positions angulaires reviennent à leurs états initiaux. Ce qui correspond à la caractéristique de stabilité d'un contrôleur PD. Nous notons aussi que les erreurs de vitesses sont caractérisées par des pics pendant l'application de la perturbation, et que les couples appliqués sur les deux articulations réagissent pour compenser la perturbation.

Afin de corriger les limites du contrôleur PD, on le combine avec un contrôleur neuro-flou en parallèle.

IV.4 stratégie de la commande neuro-floue

Nous avons combiné la commande neuro-floue (ANFIS) avec un contrôleur PD, afin d'atteindre et de maintenir avec précision la position et la vitesse désirées. Considérons que le vecteur d'entrée pour chaque articulation i , $V_{ei} = [e_i \ \dot{e}_i]$ à deux composantes représentant l'erreur de position et l'erreur de vitesse.

Nous avons choisi trois ensembles flous N (négatif), Z (zéro) et P (positif)). Les fonctions d'appartenance de l'erreur et sa dérivée sont choisies de forme triangulaire en raison

Chapitre IV

de leur simplicité, la structure de contrôleur neuro-flou est illustrée par la figure suivante :

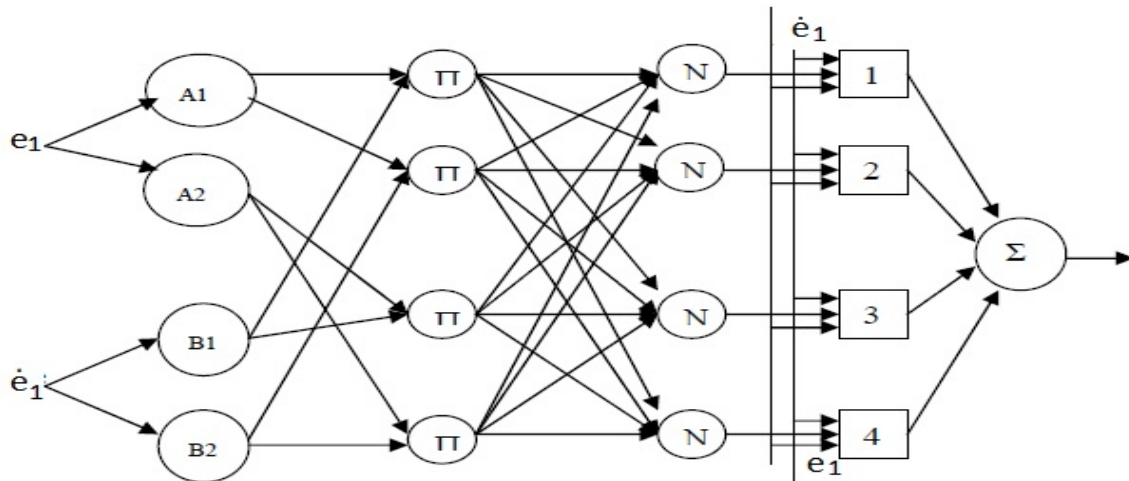


Figure IV.14 structure de régulateur neuro-flou (1^{er} articulation)

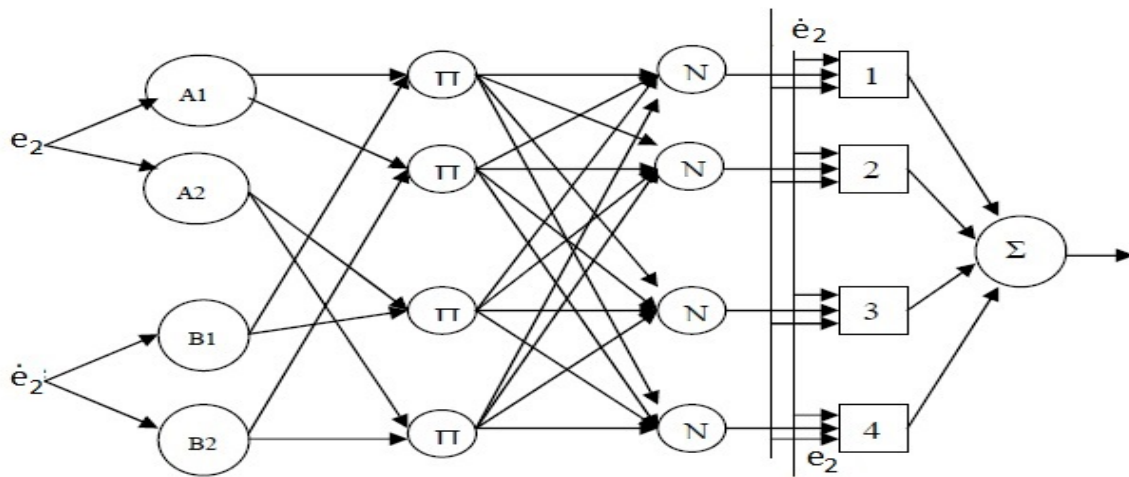


Figure IV.15 structure de régulateur neuro-flou (2^{ème} articulation)

La structure de contrôleur (ANFIS) est décrite dans le chapitre précédent. Nous avons combiné les réseaux de neurones et le système flou de type SUGENO.

Pour réajuster les trois paramètres de la couche 4, nous avons utilisé l'algorithme de rétro propagation avec le filtre de KALMAN.

Apprentissage :

Pour estimer les paramètres du réseau de neurones, nous avons utilisé le filtre de KALMAN étendu comme un algorithme d'apprentissage. Le régulateur est caractérisé par un vecteur de paramètre ϕ . Notre objectif est de trouver les valeurs de vecteur ϕ en minimisant l'erreur suivante :

$$e(k) = \frac{1}{2} (h_d(k) - h(k))^2 \quad (\text{IV.3})$$

$k=1..2$ et

Chapitre IV

L'approche du filtre de KALAMAN étendu consiste à linéariser à tout instant la sortie y autour du vecteur estimé $\hat{\Phi}$. Ceci revient à écrire :

$$h_d(k) = h(k) + \Psi_i^T(k)(\Phi_i - \hat{\Phi}_{i-1}) \quad (\text{IV.4})$$

$$\Psi_i^T(k) = \frac{\partial h(k)}{\partial \Phi_i} \quad (\text{IV.5})$$

Par conséquent, les paramètres sont ajustés selon la relation suivante :

$$\Phi_i = \Phi_{i-1} + P_i \Psi_i^T e_i \quad (\text{IV.6})$$

$$\begin{cases} e_i = h_d - h \\ \Psi_i^T = [O_{3,i} O_{3,i} e_i O_{3,i} e_i] \\ P_i = \frac{\alpha_1}{\alpha_2 + \Psi_i^T \Psi_i} \end{cases} \quad (\text{IV.7})$$

$P(k)$ est le gain de l'algorithme d'estimation, α_1 et α_2 fixés à 1 sont des gains d'adaptation pour modifier la vitesse de convergence.

Nous remarquons que cette méthode nécessite le calcul du gradient qui n'est rien d'autre que la méthode de retro propagation utilisée dans les réseaux de neurones.

$$\Psi = \frac{\partial h}{\partial \Phi}$$

Pour le cas étudié, le vecteur des paramètres est :

$$\Phi_i = [r_i p_i \rho_i]^T \quad (\text{IV.8})$$

Par conséquent nous avons

$$\Psi_i = \frac{\partial y}{\partial \Phi_i} = \left[\frac{\partial h}{\partial r_i} \quad \frac{\partial h}{\partial p_i} \quad \frac{\partial h}{\partial \rho_i} \right]^T \quad (\text{IV.9})$$

Donc la combinaison des deux contrôleurs est illustrée dans la figure suivante :

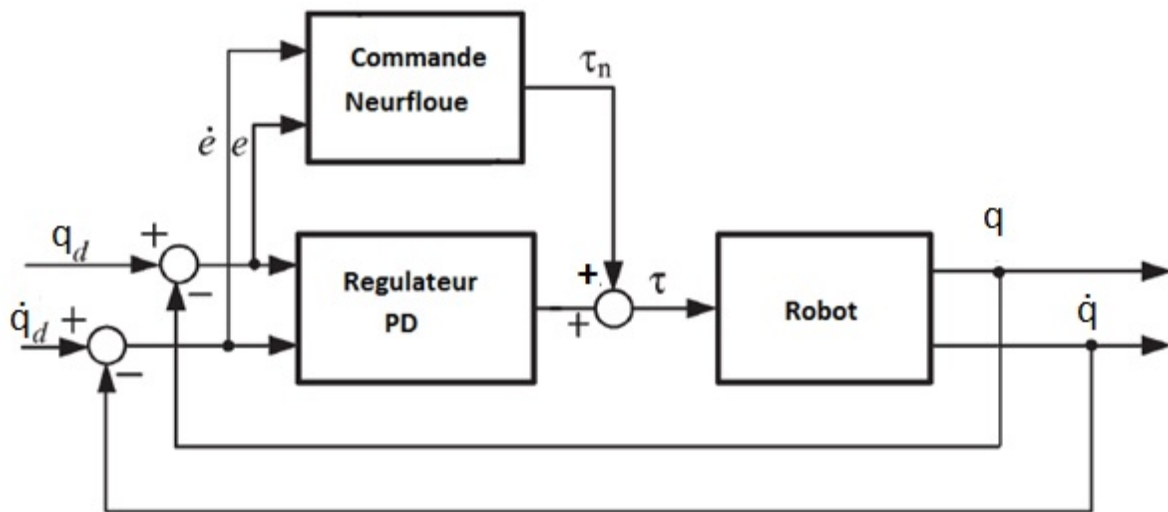


Figure IV.16 Schéma de contrôleur neuro-flou combiné à un contrôleur PD

IV.5 résultats de simulation de la commande Hybride

Sans perturbation :

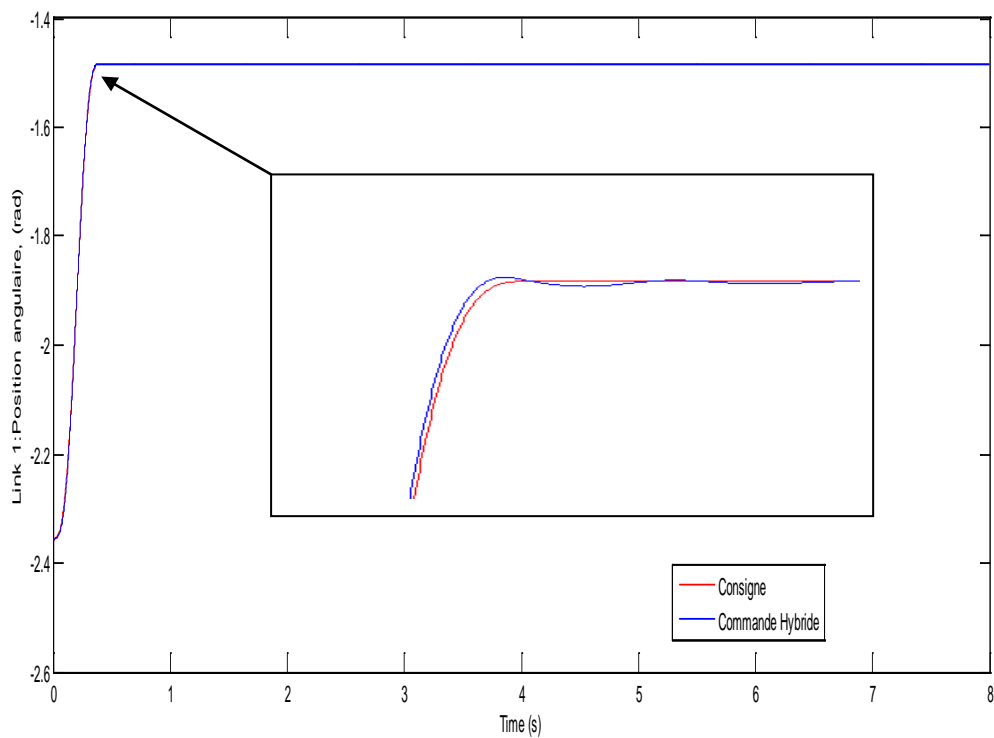


Figure IV.17 Les trajectoires désirée et réelle de la l'articulation 1.

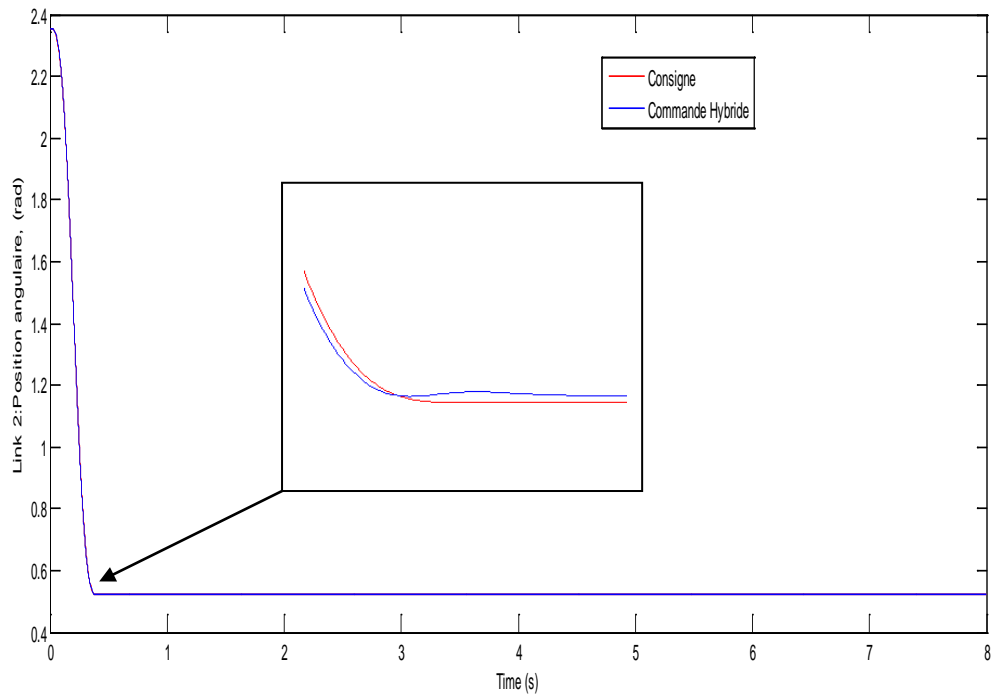


Figure IV.18 Les trajectoires désirée et réelle de la l'articulation 2.

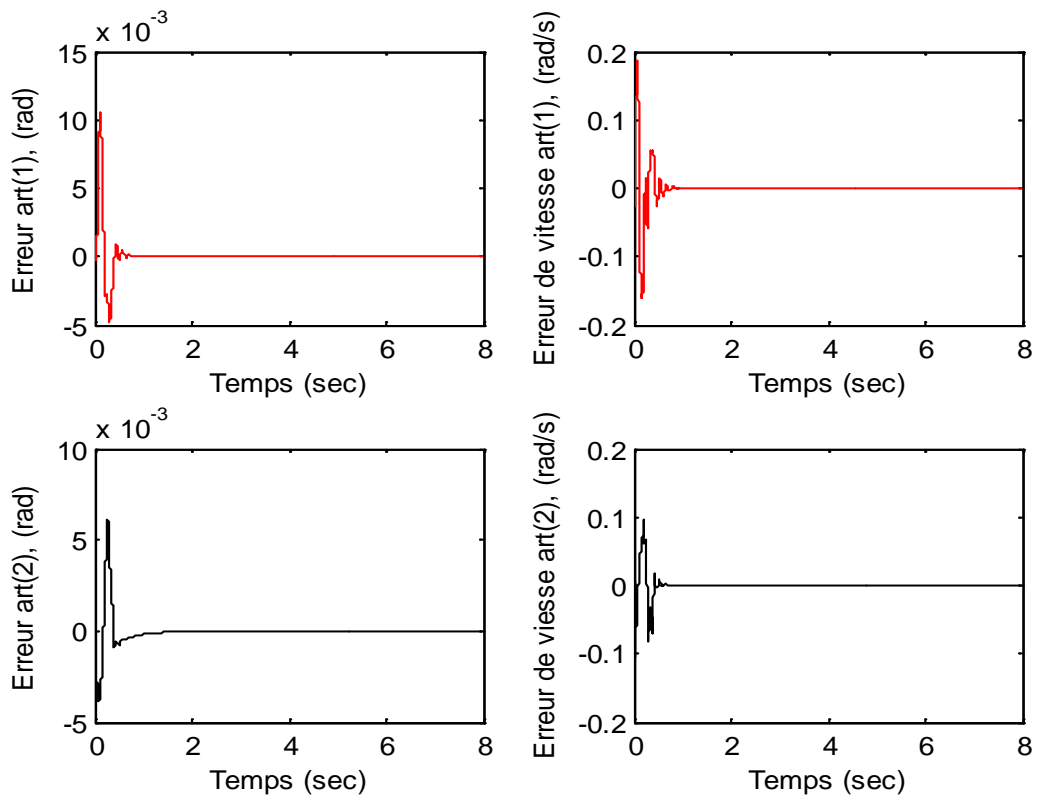


Figure IV.19 Les erreurs de position et de vitesse pour les deux articulations.

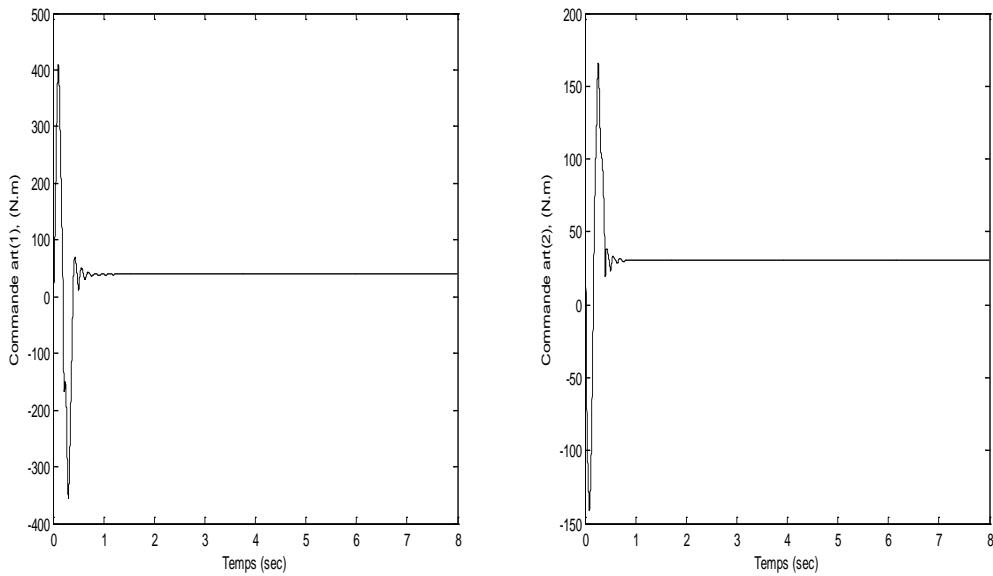


Figure IV.20 Les commande des deux articulations

Avec perturbation paramétrique :

Afin de tester la robustesse de notre commande face aux variations paramétriques, nous avons introduit à $t = [4 \ 4.5]$ une variation sur la matrice d'inertie en la multipliant par 900:

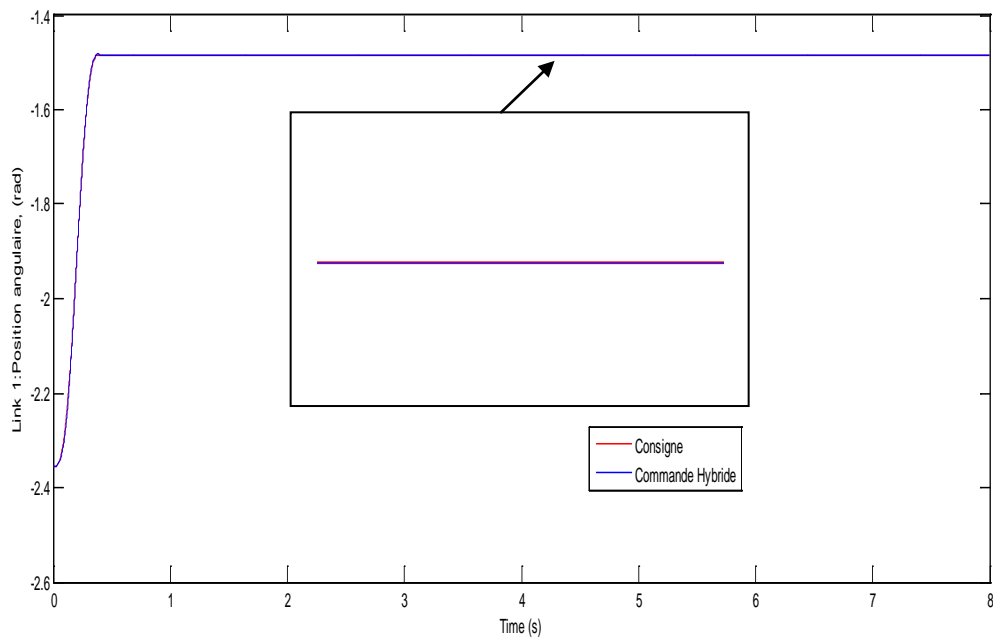


Figure IV.21 Les trajectoires désirée et réelle de la l'articulation 1.

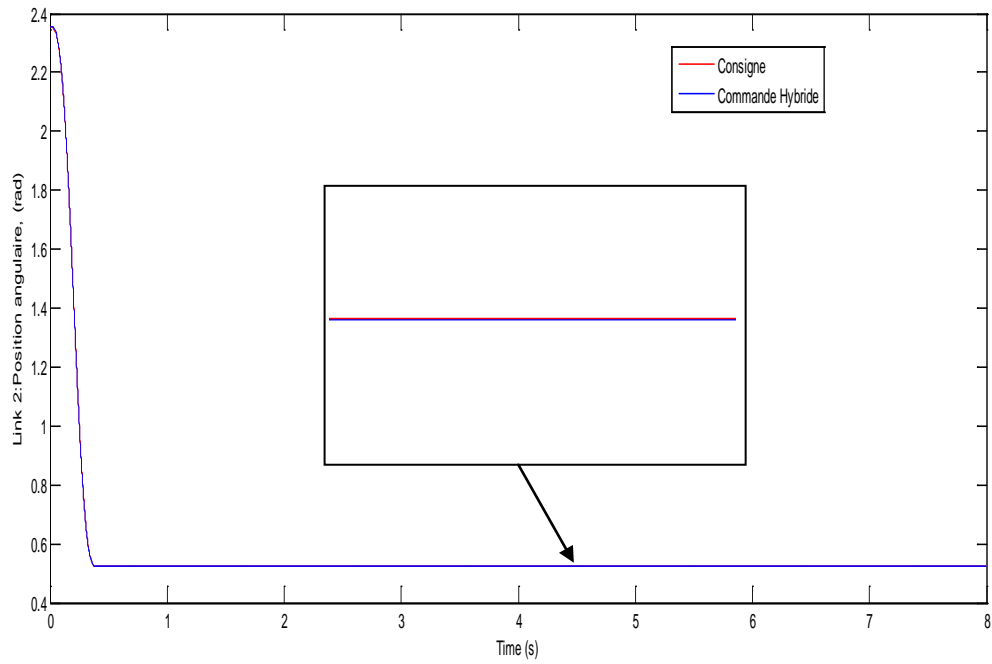


Figure IV. Les trajectoires désirée et réelle de la l'articulation 2.

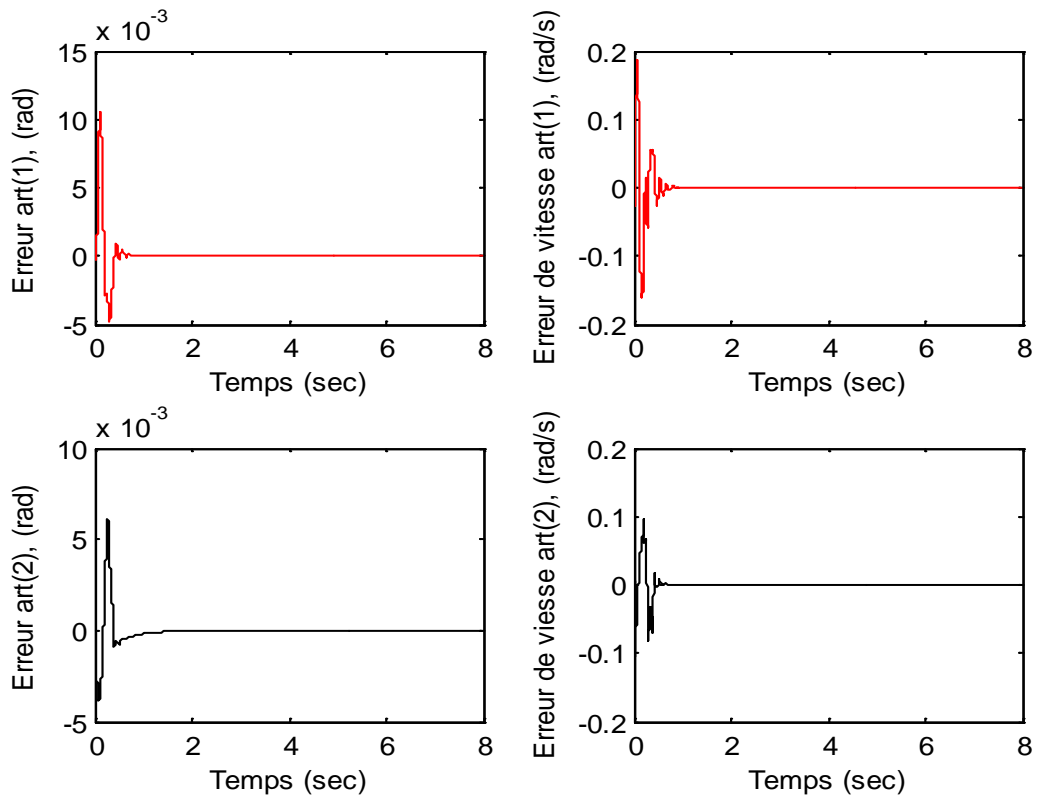


Figure IV.23 les erreurs de position et de vitesse de chaque articulation

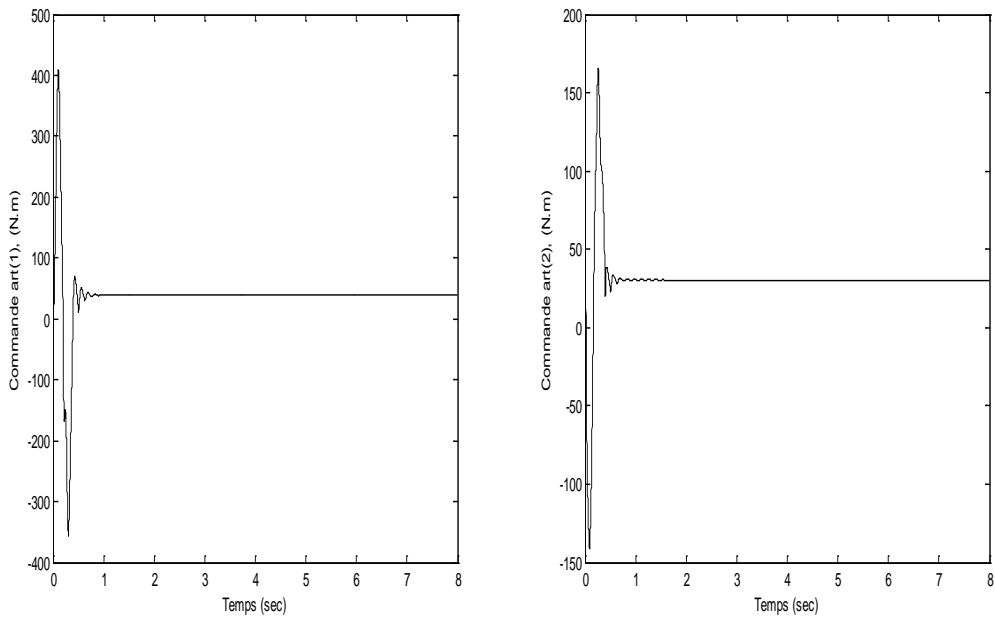


Figure IV.24 Les commandes des deux articulations.

Avec perturbation externe :

Pour confirmer la validité de notre commande et tester sa robustesse vis-à-vis des perturbations structurées ou non structurées, nous avons introduit à $t = [4 \ 4.5]$ la même perturbation externe que celle appliquée pour le contrôleur PD :

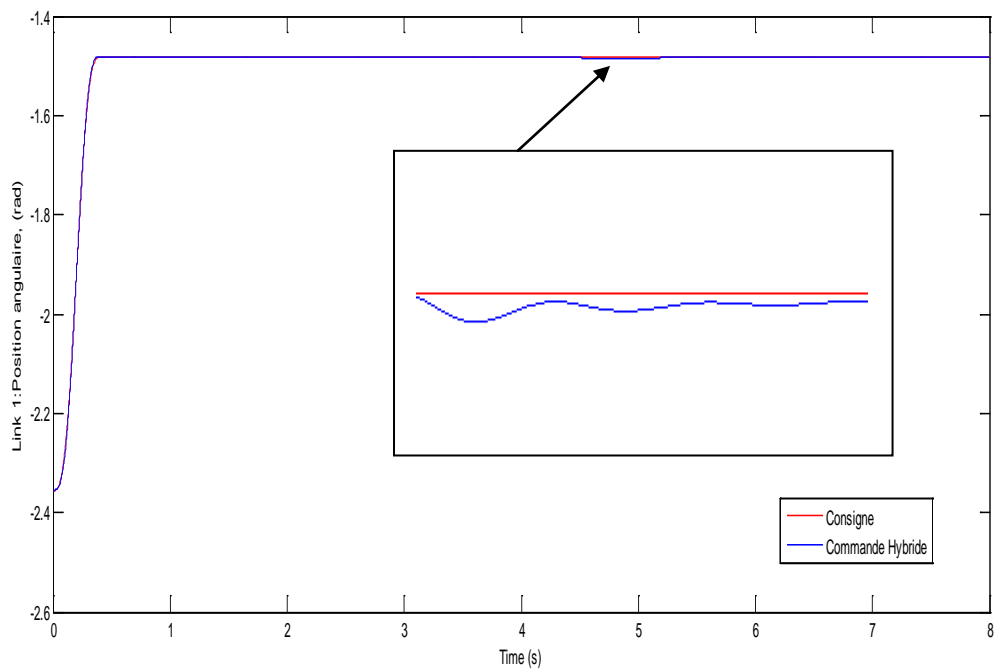


Figure IV.25 Les trajectoires désirée et réelle de l'articulation 1.

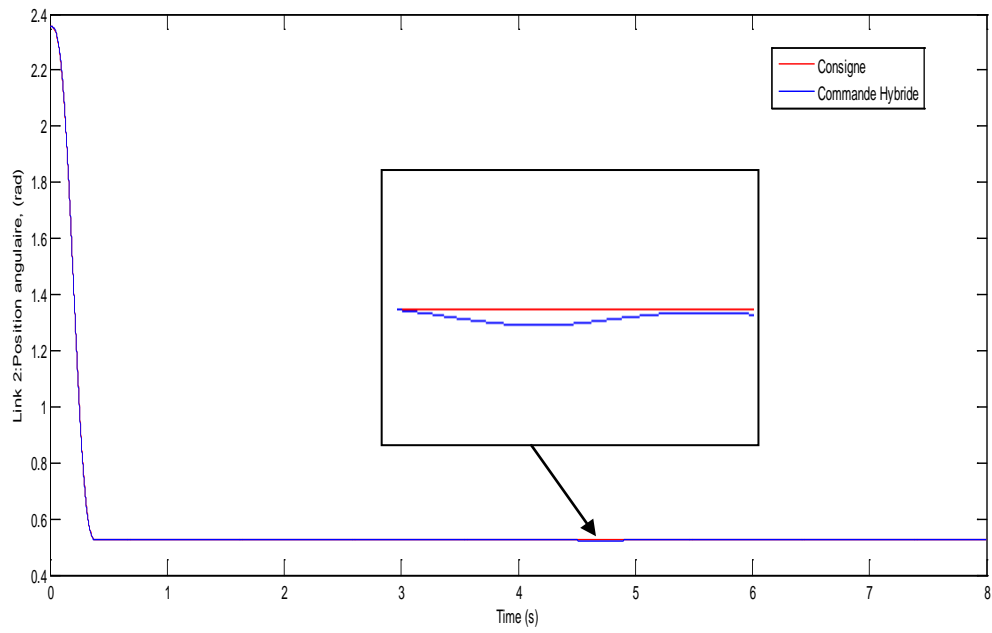


Figure IV.26 Les trajectoires désirée et réelle de l'articulation 2.

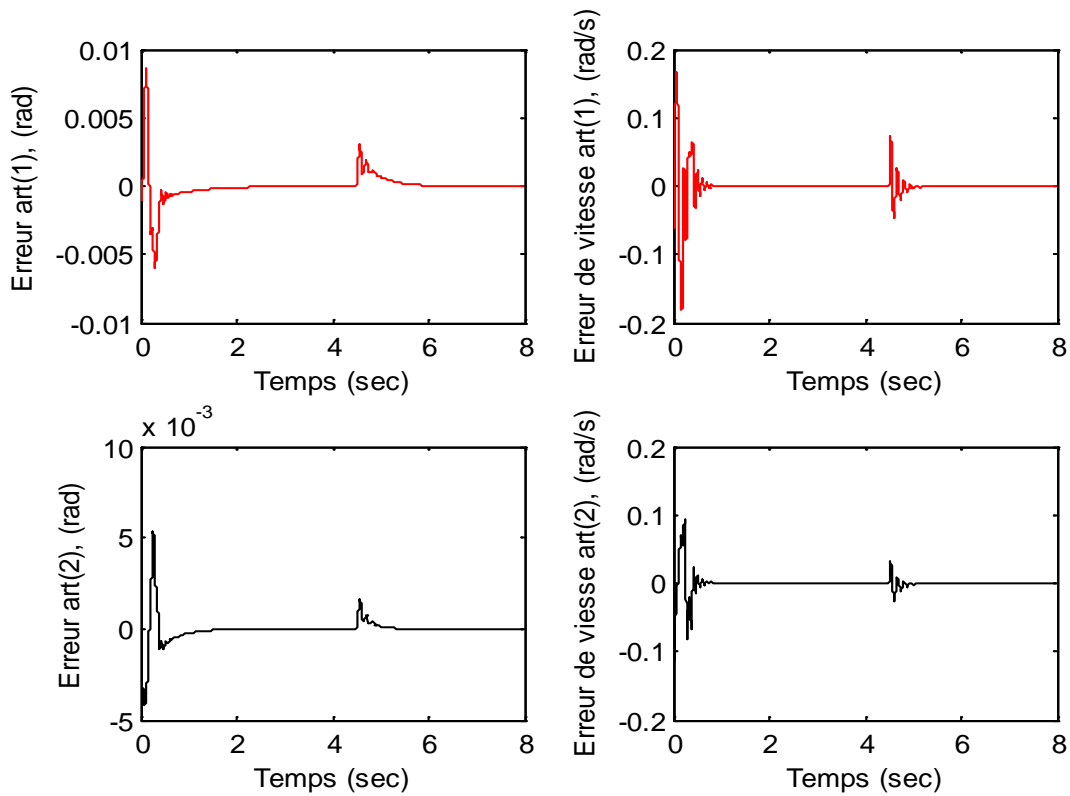


Figure IV.27 les erreurs de position et de vitesse de chaque articulation.

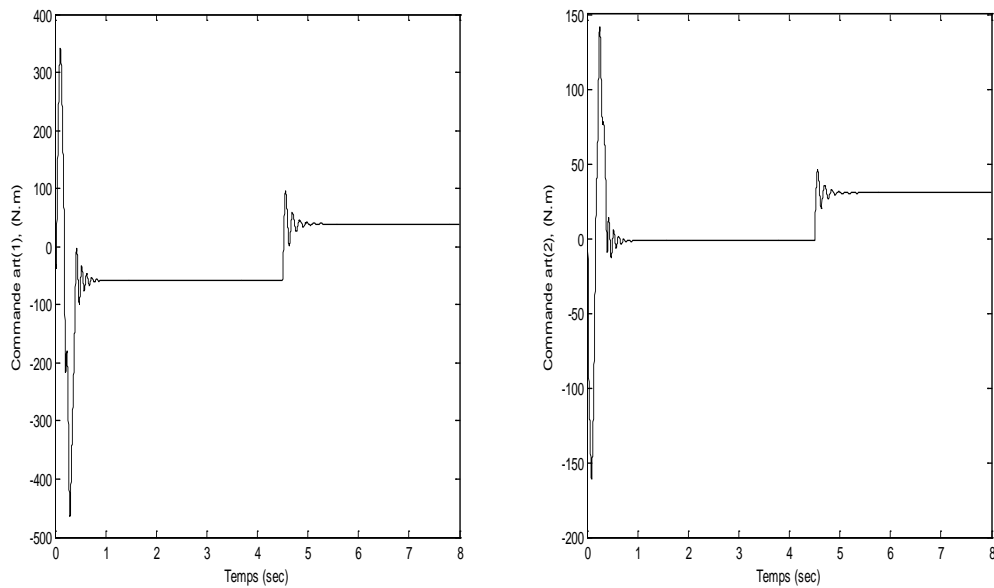


Figure IV.28 les commandes des deux articulations.

IV.6 Interprétation des résultats

Sans perturbation :

Après combinaison des deux contrôleurs à savoir le PD et le contrôleur Neuro Flou la 1^{ère} simulation est faite sans aucune perturbation. Nous constatons que la trajectoire réelle suit bien la trajectoire désirée avec un dépassement d'erreur de 10^{-2} rad pour l'articulation 1 et de 6.10^{-3} rad pour l'articulation 2. Le temps de réponse diminue, ce qui correspond à un système plus précis et plus rapide. Les erreurs de vitesse sont très faibles, et les couples exercés sont légèrement supérieurs à ceux obtenus avec le contrôleur PD.

Avec variation paramétrique :

Nous avons appliqué la même variation paramétrique que précédemment pour tester la robustesse de notre système. Les résultats de simulation montrent que le contrôleur hybride est insensible vis-à-vis de variation paramétrique.

Avec perturbation externe :

Les résultats de la dernière simulation nous montrent que les performances ne sont pas affectées, elles restent très bonnes. Au moment de l'injection de la perturbation externe, le système réagit et rejette rapidement la perturbation afin de restaurer la stabilité et la précision de notre système.

Quand aux erreurs de positions et de vitesses, elles présentent des pics au moment de l'injection de la perturbation qui est rapidement rejetée.

Chapitre IV

Ceci se traduit par le fait que les paramètres du contrôleur sont automatiquement adaptés pour se débrouiller avec les perturbations et les incertitudes pour donner un système plus robuste et plus stable.

IV.7 Conclusion

D'après les résultats de simulation obtenus, nous pouvons conclure que le contrôleur hybride proposé (un contrôleur conventionnel PD qui travaille en parallèle avec le contrôleur neuro- flou) donne des meilleures performances et une haute robustesse vis-à-vis des perturbations structurées et non structurées comparé au contrôleur autonome conventionnel.

*Conclusion
générale*

Conclusion générale

L'objectif principal assigné à ce travail est de concevoir une commande d'un robot manipulateur pour le suivi fidèle de trajectoires en position et en vitesse. Cette commande fonctionne par l'association de deux contrôleurs. Le premier contrôleur est le proportionnel dérivé, son but est d'assurer un control stable des variables articulaires. Le deuxième contrôleur est constitué d'une commande adaptative neuro-floue (ANFIS). Son but est de compenser les erreurs dues aux incertitudes structurées et non structurées.

Au premier chapitre, nous avons décrit les notions générales, théorique et technologiques nécessaires pour une meilleure compréhension du fonctionnement des robots. Le bras manipulateur étudié dans le présent mémoire est un système non linéaire. Pour sa modélisation, nous avons utilisé le formalisme d'Euler-Lagrange.

Dans le deuxième chapitre, nous avons étudié quelques techniques de commandes classiques et modernes, dans l'intention d'atteindre et de maintenir avec précision une position désirée pour le contrôle du robot manipulateur. Nous avons vu les différentes commandes à savoir la commande adaptative, la commande par mode de glissement et les commandes conventionnelles (PI, PD et le PID), Parmi lesquelles nous avons choisi le contrôleur PD.

Au troisième chapitre, nous avons présenté les réseaux de neurones, la théorie de la logique floue, et les systèmes neuro-flous. Nous nous sommes intéressés principalement au système ANFIS, un réseau neuronal constitué de 5 couches, dans lequel chaque couche correspond à la réalisation d'une étape d'un système d'inférence floue de type Takagi Sugeno. Ce modèle donne de très bons résultats en poursuite de trajectoire, approximation non linéaire et commande dynamique.

Au dernier chapitre, deux techniques de commande ont été appliquées pour la commande de ce bras manipulateur:

Une technique de commande classique, à savoir un contrôleur de type PD, dont l'implémentation est relativement simple, mais cette commande n'est pas robuste puisqu'elle est sensible aux perturbations externe.

Une deuxième commande intelligente basée sur la logique floue et les réseaux de neurones a été combinée avec ce contrôleur PD. La commande neuro floue appliquée réunit la théorie puissante de la logique floue et celle des réseaux de neurones, afin d'introduire des capacités d'apprentissage et d'adaptation dans les systèmes flous. Cette combinaison nous a donné de meilleures performances en termes de robustesse, poursuite et stabilité.

Conclusion générale

Selon les résultats obtenus, nous pouvons conclure que le système neuro-flou fait apparaître un bon compromis entre la caractérisation et l'efficacité des calculs. Sa robustesse, sa rapidité et la précision de ses sorties lui permettent de donner des décisions correctes et d'éviter les cas d'indécisions, les réseaux de neurones avec leurs capacités à s'adapter à des situations inconnues par le biais de l'apprentissage, et la logique floue avec sa capacité de modélisation de connaissances imprécises et gestion de l'incertitude.

Comme travaux futurs, nous proposons de poursuivre ce projet en considérant l'implémentation de la commande proposée dans le cas d'un manipulateur flexible.

*Références
bibliographiques*

Références bibliographiques

- [1] **P. Fisette, H. Buyse, J.C. Samin**, MECA 2732 : Introduction à la Robotique, 10 novembre 2004.
- [2] **A. Benali**, Robotique et Automatisation Industrielle.
- [3] **Jean-Louis Boimond** ISTIA, Université Angers : cours Robotique
- [4] **P. Coiffet, Hermes**, La Robotique, principes et applications.
- [5] **H. Hamdi**, Introduction à la Robotique. Les éditions de l'université Mentouri Constantine 2002-2003.
- [6] **E. Dombre**, Analyse et Modélisation des Robots Manipulateurs, 22 janvier 2002.
- [7] **A. Chrifa**, Analyse Cinématique des Robots Parallèles. Thèse Docteur D'état. Université Mentouri Constantine 2007/2008
- [8] **Guy ANDRE**, Capteurs d'environnement en robotique : introduction (INRIA), Techniques de l'Ingénieur, traité Informatique industrielle R 7 748
- [9] **Sayd MAMDOUH** solution du problème géométrique direct des manipulateurs star
- [10] **T. LATOUR**, modélisation cinématique et dynamique du robot industriel IRB2400 de l'ECAM, 2006, ECAM, Bruxelles.
- [11] **A. Hamzaoui, N. Essounbouli, J. Zaytoon**, commande par mode glissant flou d'un système non linéaire incertain, 2003
- [12] **Arato. S. Deo** thèse de PhD "Inverse Kinematics and Dynamic Control Methods for Robotic Systems" Houston .Texas. 1995
- [13] **Claude TOUZET**, les réseaux de neurones artificiels, introduction au connexionnisme, cours, exercices et travaux pratiques, juillet 1992.
- [14] <https://sites.google.com/site/tpeiag5lnbahmv/3-neurones-artificielles/a-leurs-principes-et-leurs-fonctionnements>
- [15] **Borne P. Haggege J.** "les réseaux de neurones présentation et application" , Paris : Editions Technique, 2007.
- [16] **Mellah .R.** « contribution à la commande adaptative neuro floue .Application à la robotique » Thèse de Doctorat, Université des Sciences et de la Technologie Houari Boumediene. Alger. Mai 2006.
- [17] **Breton, S.** « Une approche neuronale du contrôle robotique utilisant la vision binoculaire par reconstruction tridimensionnelle » Thèse de Doctorat, Université de Haute alsace U.F.R .des sciences et Techniques Juin 1999.
- [18] **A Hamzaoui, N Essounbouli and J Zaytoon**, "Fuzzy Sliding Mode Control for Uncertain SISO Systems", Processing of IFAC Conf. on Intelligent Control Systems and Signal ICONS'03, pages 233–238, 2003
- [19] **Slotine, J.-J. E., & Li, W.** "Applied Nonlinear Control". London: Prentice-Hall, Inc. 1991
- [20] **K. J. Aström, et B. Wittenmark**, "Adaptive Control," Addison-Wesley, 1989.
- [21] **I. D. Landau, R. Lozano et M. M'Saad**, "Adaptive Control," Springer, 1998
- [22] **J. Jacques, et E. Slotine**, "Applied Nonlinear Control," Prentice Hall, 1991.

Références bibliographiques

- [23] **Samer RIACHY**, " Contribution à l'estimation et à la commande de systèmes mécaniques sous-actionnés " l'Ecole Centrale de Lille, 2008.
- [24] **H. K. Khalil**, "Nonlinear systems," Prentice Hall, 1996
- [25] **M. Vidyasagar**, "Nonlinear systems analysis," Prentice Hall, 1993.
- [26] **C. C. Lee**, "Fuzzy logic in control systems: fuzzy logic controller. Part1", *IEEE Transactions on systems Man and Cybernetics*, Vol. 20, 1990, pp. 404-418.
- [30] **E. Mamdani**, "Application of fuzzy logic to approximate reasoning using linguistic systems", *Fuzzy Sets and Systems*, vol.26, 1977, pp.1182-1191.
- [31] **T. Takagi, M. Sugeno**, "Fuzzy identification of systems and its applications to modeling and control", *IEEE Transactions on systems Man and cybernetics*, Vol. 15, n° 1, 1985, pp. 116-132.
- [28] **C. C. Lee**, "Fuzzy logic in control systems: fuzzy logic controller. Part2", *IEEE Transactions on systems Man and Cybernetics*, Vol. 20, 1990, pp. 419-435.
- [29] **K. M. Passino, S. Yurkovich**, "Fuzzy control", *Addison Wesley Longman, Inc*, 1998.
- [27] **Nicolas PALLUAT**, "Méthodologie de surveillance dynamique à l'aide des réseaux neuro-fous temporels", 2006.
- [32] **OtiliaElina VASILE**, "Contribution au pronostic de défaillance par réseau neuroflou", 2008.
- [33] **J.S.R. Jang**, ANFIS : " Adaptive-network-based fuzzy inference systems ". *IEEE Transactions on Systems, Man and Cybernetics* 23(3).665-685, 1993.