

وزارة التعليم العالي والبحث العلمي

BADJI-MOKHAR-ANNABA UNIVERCITY
UNIVERCITE BADJI-MOKHAR-ANNABA



جامعة باجي مختار - عنابة

Faculté des Sciences de l'ingéniorat
Département de Génie Mécanique

THESE

Présentée pour l'obtention du diplôme de DOCTORAT

**Modélisation et commande des robots manipulateurs
par les outils de l'intelligence artificielle**

Option :

Génie Mécanique

Par :

Fateh MAKHLOUFI

DIRECTEUR DE THESE :

**Noureddine GUERSI
Annaba**

Pr.

UBM

Devant le jury

Président :

Abdelaziz. M. BOUCHELAGHEM

Pr.

UBM Annaba

Examineurs :

Lakhdar. LAOUAR

Pr.

UBM Annaba

Omar. BOUHALI

Pr.

UMSB Jijel

Hamid. KHERFANE

MC.

UBM Annaba

ANNABA 2015

Remerciements

الحمد لله رب العالمين

Mes remerciements vont tout d'abord à Dieu le tout-puissant pour la force et la volonté qui m'a procuré pour terminer mon travail.

Ce travail a été effectué en collaboration entre le Laboratoire de Mécanique Industrielle (LMI) et le Laboratoire des Systèmes Embarqués (LASE) de l'Université Badji Mokhtar d'ANNABA, sous la direction du Professeur GUERCI Nouredine.

Je tiens à remercier profondément mon directeur de thèse : le professeur **GUERCI Nouredine** pour la confiance qu'il m'a accordée, ses encouragements, et ses précieux conseils, je le remercie de m'avoir encadré, orienté, aidé et conseillé.

Je tiens à remercier aussi mon Co-directeur de thèse Mr. **SAADI Mohamed Nacer** Maître de Conférences au département d'électronique pour son aide, encouragement et conseils intéressants.

J'exprime ma gratitude envers Mr. **FRIHI Mohamed** Maîtres de conférences au département d'électronique pour son aide concernant la partie expérimentale, je le remercie vivement, de ses conseils, ses orientations et le soutien que j'ai trouvé auprès de lui.

Je tiens à remercier les membres de jury d'avoir accepté d'examiner ce modeste travail de doctorat.

Je remercie très sincèrement le professeur **BOUCHELAGHEM Abdelaziz Mahmoud** le directeur de laboratoire (LMI), d'avoir accepté la présidence du jury,

Je remercie le professeur **LAOUAR Lakhdar** chef de formation doctorale, d'avoir acceptés d'être membres du jury.

J'exprime ma gratitude envers le professeur **BOUHALI Omar** vice-recteur chargé des relations extérieures de l'université Mohamed Seddik Benyahia de Jijel d'avoir accepté de juger ce travail et de faire partie du jury.

Je suis très reconnaissant à Monsieur **KHERFANE Hamid**, maître de conférences au département d'électronique, d'avoir accepté d'examiner mon travail avec attention et rigueur et soit associé pour sa compétence à ce jury.

Je souhaite aussi remercier également mes camarades de promotion et l'ensemble du personnel et des enseignants du département de Génie Mécanique de l'université Badji Mokhtar d'ANNABA et tous ceux qui m'ont enseigné durant toutes mes études et sans oublier Mr. **HAIHEM Ali** Ingénieur de laboratoire au département de Génie Mécanique.

Je ne peux pas oublier les membres de ma famille, qui m'ont soutenu durant cette thèse, je remercie mes chers parents pour leur soutien durant toutes mes années d'études. Pour les sacrifices, la tendresse, l'amour, la patience, le soutien, l'aide, l'encouragement. Je remercie aussi ma sœur Soumaya et son mari Chafik et le petite Mohamed Sirage, ma sœur Dalila, mes frères Walid, Hamza et Said.

Aucune expression de remerciements ne peut traduire mes redevances envers Mlle. Ben Jeddou Hanene pour son aide et encouragement et surtout son soutien moral.

Je souhaite également remercier mes amis : mahdi, said, youcef, wahab, akram, tazir, doudou, ilyes, salah, fakhri, mohamed, walid, asma, zinou, farouk, ramdane, rafik.

Enfin, Nous exprimons notre gratitude à toute personne ayant contribué de près ou de loin dans ce travail.

MAKHOULFI Fateh

Résumé

Les travaux présentés dans cette thèse de doctorat traitent la commande des robots manipulateurs avec modèle dynamique incertain, dans l'objectif est de montrer que les algorithmes de l'intelligence artificielle peuvent être utilisés pour modéliser la dynamique des systèmes non linéaires avec modèle dynamique incertain et concevoir des structures de commande stables et robustes. L'apport principal de cette étude réside dans le développement de méthodologies de commande à base des approches neuro-floues. Ces approches combinent les concepts de la logique floue et les réseaux de neurones artificiels pour constituer un contrôleur intelligent hybride appelé ANFIS (Adaptive Neuro-Fuzzy Inference System). On a utilisé l'ANFIS pour la commande d'un bras manipulateur hydraulique à 6 ddl (FEEDBACK HRA934). Nous proposons deux techniques de commande basée sur ANFIS : Commande adaptative Neuro flou basé sur le calcul de couple (type PD), et la commande adaptative Neuro flou direct type PD plus une correction intégrale. La technique la mieux adaptée à la simulation sera appliqué à notre robot expérimental. Les résultats de simulation montrent que la commande adaptative neuro flou direct type PD plus une correction intégrale donne de meilleurs résultats.

Mots-clés : logique floue, réseaux de neurones, ANFIS, contrôle PID, robot manipulateur hydraulique.

Abstract

The work presented in this doctoral thesis are integrated in the control of robot manipulators with uncertain dynamic model, which aims to show that the algorithms of artificial intelligence can be used to model the dynamics of nonlinear systems with uncertain dynamic model and develop stable and robust control structures. The main contribution of this study lies in the development of control methodologies based neuro-fuzzy approaches. These approaches combine the concepts of fuzzy logic and artificial neural networks to form a hybrid intelligent controller called ANFIS (Adaptive Neuro-Fuzzy Inference System). We used ANFIS for the control of a 6 DOF hydraulic manipulator arm (FEEDBACK HRA934). We propose two control techniques based on ANFIS: Adaptive neuro fuzzy inference system based computed torque controller (type PD) and Adaptive neuro fuzzy inference system based PD plus I controller. The most suitable technique from simulation will be applied to our experimental robot. Simulation results show that the Adaptive neuro fuzzy inference system based PD plus I controller gives better results.

Keywords: Fuzzy Logic, Neural networks, ANFIS, PD control, hydraulically actuated robot arm.

ملخص

العمل المقدم في هذه الأطروحة يندمج ضمن إطار التحكم في ذراع آلية ذات نموذج ديناميكي غير دقيق، والهدف من هذا العمل هو تبيان أن الخوارزميات الذكية يمكن استعمالها لمحاكاة الأنظمة الغير خطية ذات النموذج الديناميكي الغير دقيق، وكذا إنشاء أنظمة تحكم مستقرة وقوية. المساهمة الرئيسية لهذه الدراسة تكمن في تطوير أساليب ومناهج تحكم تعتمد على neuro-floues. هذه الأخيرة تمزج بين تقنيات المنطق الضبابي والشبكة العصبية الاصطناعية وهذا المزج من اجل انشاء متحكم ذكي وهجين يسمى ANFIS. هذا المتحكم الذكي والهجين سيعتعمل في التحكم في ذراع آلية هيدروليكية ذات ستة درجات حرية. بالاعتماد على ANFIS نقترح تقنيتي تحكم: التقنية الأولى وهي تقنية التحكم المتكيف neuro-floues المعتمد على حساب العزم (نوع PD)، والتقنية الثانية وهي تقنية التحكم المتكيف neuro-floues المباشر (نوع PD) مضاف اليه تصحيح تكاملي. التقنية التي تعطي أحسن نتيجة في النمذجة سوف تطبق على الروبوت الالي الحقيقي. وفقا لنتائج النمذجة تبين ان تقنية التحكم المتكيف neuro-floues المباشر (نوع PD) مضاف اليه التصحيح التكاملي اعطت أحسن نتيجة.

الكلمات المفتاحية: المنطق الضبابي، الشبكة العصبية الاصطناعية، ANFIS، المتحكم PD، ذراع آلية هيدروليكية.

Table des matières

Remerciements

Résumé

Abstract

ملخص

Liste des figures

Liste des tableaux

Liste des abréviations

Introduction général

1. Introduction	1
2. Aperçu de la thèse	2

Chapitre I Commande des robots manipulateurs

1.1 Introduction	3
1.2 Commande des bras manipulateurs	4
1.2.1 Commande par articulation	5
1.2.2 Commande jacobienne	5
1.2.3 Commande par découplage non linéaire	6
1.2.4 Commande adaptative	6
1.2.5 Commande fondée sur une fonction de Lyapunov	6
1.2.6 Commande passive	7
1.2.7 Commande prédictive	7
1.2.8 Commande robuste	7
1.2.9 Commande optimale	7
1.3 Approche sélectionnée	8

Chapitre II Modélisation des robots manipulateurs

2.1. Introduction	9
2.2. Modélisation des robots manipulateurs	9
2.3. Modèle géométrique	11
2.3.1 Modèle géométrique direct (MGD)	11

2.3.2. Modèle géométrique inverse (MGI)	12
2.4 Calcule des modèles géométriques (direct et inverse)	13
2.4.1 Calcule du modèle géométrique direct	14
2.4.2 Calcule de modèle géométrique inverse	16
2.5 Modèle cinématique	17
2.5.1 Modèle cinématique direct	18
2.5.2 Modèle cinématique inverse	19
2.6 Modèle dynamique	20
2.6.1 Modèle dynamique inverse	20
2.6.1.1 Formalisme de Lagrange	21
2.6.2 Modèle dynamique direct	27
2.7 Modèle hydraulique	29
2.7.1 Modélisation des servovalves électrohydrauliques	29
2.7.2 Vérins hydrauliques linéaires	32
2.7.3 Vérin hydraulique linéaire agissant autour d'un pivot	34
2.7.4 Modèle hydraulique du robot	37
2.8 Conclusion	38

Chapitre III Commandes classiques des robots manipulateurs

3.1 Introduction.....	39
3.2 Quelque stratégie de commande	39
3.3 Commande par les méthodes classique	40
3.4 Commande par calcul de couple	40
3.4.1 Dérivation de la boucle interne (Feed forward)	42
3.4.2 Conception de la boucle externe	44
3.5 Calcul de couple avec PD	45
3.6 Calcul de couple avec PID	50
3.7. Commande adaptative	57
3.7.1 Introduction	57
3.8 Commande adaptative des robots manipulateurs	58
3.8.1 Commande adaptative par calcul de couple	59
3.8.2 Commande adaptative inertiel	69
3.9 Conclusion	73

Chapitre IV Les techniques de l'intelligence artificielle

4.1 Introduction	74
4.2 Commande basée sur l'intelligence artificielle	74
4.3 Neuro-Flou	76
4.3.1 Structure de commande neuro-flou	76
4.3.2 Définition	76
4.3.3 Principe de fonctionnement	76
4.3.4 Architectures neuro-floues	77
4.3.4.1 Première architecture	77
4.3.4.2 Deuxième architecture	80
4.3.4.3 Troisième architecture	81
4.3.5 Les types d'implémentation des réseaux neuro-flous	81
4.3.6 Systèmes d'inférence neuro-flou adaptatifs (ANFIS).....	82
4.3.6.1 Architecture d'ANFIS	82
4.3.6.2 Algorithme d'apprentissage	85
4.3.7 Conception de contrôleur basée sur ANFIS	86
4.3.7.1 ANFIS basée sur le calcul de couple (type PD)	89
4.3.7.2 ANFIS directe type PD plus une correction intégrale	90

Simulation et validation expérimentale

5.1 Simulation	92
5.2 Expérimentale	95

Conclusion générale et Perspectives

Conclusion générale et Perspectives	98
Annexe.....	100

Références bibliographiques	141
--	-----

Liste des figures :

Figure II.1 : Modèle géométrique directe et inverse	13
Figure II.2 : Les paramètres géométriques d'un bras à deux degré de liberté (2ddl)	14
Figure II.3 : Bras à deux degré de liberté (2ddl)	22
Figure II.4 : Schéma d'une servovalve	30
Figure II.5 : Actionnaire hydraulique linéaire	32
Figure II.6 : Vérin hydraulique linéaire agissant autour d'un pivot	34
Figure III.1 : Système de commande par calcul de couple,	41
Figure III.2 : Commande par calcul de couple avec PD	46
Figure III.3 : Erreur et poursuite de position de la première articulation	48
Figure III.4 : Erreur et poursuite de position de la deuxième articulation	48
Figure III.5 : Erreur et poursuite de vitesse de la première articulation	49
Figure III.6 : Erreur et poursuite de vitesse de la deuxième articulation	49
Figure III.7 : Schéma de commande par calcul de couple avec PID	50
Figure III.8 : Erreur de poursuite de position	52
Figure III.9 : Poursuite de position de la première articulation	52
Figure III.10 : Poursuite de position de la deuxième articulation	53
Figure III.11 : Erreur de vitesse	53
Figure III.12 : Poursuite de vitesse de la première articulation	54
Figure III.13 : Poursuite de vitesse de la deuxième articulation	54
Figure III.14 : Schéma général de la commande adaptatif	56
Figure III.15 : Schéma de commande adaptative par calcul de couple	64
Figure III.16 : Erreur de poursuite de position	65
Figure III.17 : Poursuite de position de la première articulation	65
Figure III.18 : Poursuite de position de la deuxième articulation	66
Figure III.19 : Erreur de vitesse	66
Figure III.20 : Poursuite de vitesse de la première articulation	67

Figure III.21 : Poursuite de vitesse de la deuxième articulation	67
Figure III.22 : Schéma fonctionnel du contrôleur inertie-connexe adaptatif	68
Figure III.23 : Erreur de poursuite de position	69
Figure III.24 : Poursuite de position de la première articulation	69
Figure III.25 : Poursuite de position de la deuxième articulation	70
Figure III.26 : Erreur de vitesse	70
Figure III.27 : Poursuite de vitesse de la première articulation	71
Figure III.28 : Poursuite de vitesse de la deuxième articulation	71
Figure III.29 : Estimation des masses	72
Figure IV.1 : Principe du système neuro-flou	76
Figure IV.2 : Première architecture neuro-floue	77
Figure IV.3 : Forme d'un neurone	78
Figure IV.4 : Forme d'opérateur flou ET	79
Figure IV.5 : Deuxième architecture des réseaux neuro-flou réalisation en série	80
Figure IV.6 : Réseau de neurones fonctionnant en aval d'un système flou	81
Figure IV.7 : Les différentes couches d'ANFIS	82
Figure IV.8 : Architecture ANFIS	83
Figure IV.9 : Marche à suivre d'ANFIS	85
Figure IV.10 : Robot manipulateur hydraulique FEEDBACK HRA934	87
Figure IV.11 : Schéma fonctionnel du bras expérimental	87
Figure IV.12 : Vue de plan de la configuration HRA934 restreinte	88
Figure IV.13 : Le schéma fonctionnel global du système de commande	89
Figure IV.14 : Le schéma fonctionnel global du système de commande	91
Figure V.1 : Poursuite de position de la première articulation	92
Figure V.2 : Erreur de position de la première articulation	93
Figure V.3 : Poursuite de position de la deuxième articulation	93
Figure V.4 : Erreur de position de la deuxième articulation	94

Figure V.5 : Poursuite de position de la première articulation	95
Figure V.6 : Erreur de position de la première articulation	95
Figure V.7 : Poursuite de position de la deuxième articulation	96
Figure V.8 : Erreur de position de la deuxième articulation	96

Liste des tableaux :

Tableau II.1 : Configuration de Denavit-Hartenberg du robot à 2ddl	14
Tableau II.2 : Variation de module de compressibilité avec la température de fluide	33
Tableau A.1 : Tableau ou matrice des règles	113

Liste des abréviations :

ANFIS : Adaptive Neuro Fuzzy Inference System : système d'inférence adaptative neuro-floue.

2ddl : Deux degré de liberté.

MGD : Modèle géométrique direct.

MGI : Modèle cinématique inverse.

DH : Convention de Denavit-Hartenberg.

ATAN2 : En trigonométrie, atan2 à deux arguments est une variation d'ARC tangente.

W : watt.

CTC : The Computed Torque Control : Commande par calcul de couple.

PID : Proportionnel Intégral et Dérivé.

PD : Proportionnel et Dérivé.

diag : diagonale.

kg : kilogram.

m : mètre.

s : seconde.

E/S : Entré/Sortie.

NG : négative grande.

NM : négative moyenne.

EZ : environ de zéro (zéro).

PM : positive moyenne.

PG : positive grande.

RNA : Les réseaux de neurones artificiels.

LF : La logique floue.

mV : milli volte.

Introduction générale

1. Introduction

De nos jours, les robots sont devenus des outils nécessaires dans l'environnement de fabrication, permettant des tâches à automatiser s'étendant du plus répétitif au plus complexe. Pendant que la technologie avance, la robotique évolue pour être à la fois plus précise et pratique. Le problème de la commande est en quelque sorte un problème de prise de décision : étant donné l'observation de l'état du processus et une base connaissance on doit décider de l'action à prendre. Des systèmes à base de connaissance (Knowledge based systems) et en particulier les approches à base de règles sont idéalement approprié à une telle tâche de prise de décision. Le contrôleur flou est un simple système de commande à base de règles. La connaissance utilisée dans ce cas ne vient pas seulement d'un opérateur expert. Elle peut également venir du concepteur du ou des systèmes. L'un des avantages principaux de l'utilisation d'une approche floue est que la logique floue fournit la meilleure technique pour la représentation de la connaissance. D'autre part les réseaux de neurones artificiels avec leurs parallélismes massifs et leurs capacités d'apprendre n'importe quel genre de non linéarité sont aujourd'hui utilisés pour résoudre des problèmes de commande très pratiques. Un neuro-contrôleur (système de commande à base de réseaux de neurones) en général, exécute une forme spécifique de commande adaptative, avec le contrôleur sous forme d'un réseau multicouche et les paramètres d'adaptation étant définis comme des poids ajustables. En général, les réseaux de neurones sont des structures parallèles distribuées de traitement de l'information, ce qui fait d'eux l'un des meilleurs candidats pour la commande des systèmes multivariables.

Les concepts régnant sur de la commande intelligente sont :

1. Commande par logique floue.
2. Commande basée sur les réseaux de neurones.
3. Commande Hybride (ANFIS).

2. Aperçu de la thèse

Cette thèse commence par une introduction à la commande des robots, et plus précisément les robots manipulateurs, dans le but de présenter un état de l'art dans ce domaine et de discuter les différentes approches proposées dans la littérature.

Le deuxième chapitre est dédié à une présentation des différents modèles utilisés pour décrire le mouvement des articulations d'un bras manipulateur, le modèle couvre des aspects des modèles géométriques, modèles cinématique et les modèles dynamiques. On présentera quelques définitions concernant ces modèles ainsi que la façon de leurs obtentions.

Le troisième chapitre présente la commande des bras manipulateurs par quelques techniques de commandes conventionnelles employées fréquemment dans la littérature et ainsi que les résultats de simulations obtenues pour chaque méthode.

Le quatrième chapitre est consacré à l'étude des méthodes de commandes intelligente hybride basée sur les réseaux de neurones et la logique floue (ANFIS). Nous présentons tout d'abord un historique des réseaux de neurones et des différentes structures proposées. Nous abordons ensuite, les principaux types de réseaux de neurones, avec des rappels sur leur principale propriété. Par la suite, nous présentons les principales notions théoriques des systèmes d'inférence floue et les méthodes d'optimisation utilisées. Après avoir introduit les concepts de base concernant la structure générale, les différents types de contrôleurs et modèles flous et les caractéristiques structurelles et paramétriques sont présentés.

Le chapitre cinq constitue notre contribution à la commande d'un robot à paramètres incertains où nous montrerons l'efficacité de l'approche proposée qui combine les concepts de la logique floue et les réseaux de neurones pour constituer un contrôleur intelligent hybride appelé (ANFIS). Une validation expérimentale sur un bras manipulateur hydraulique (FEEDBACK HRA934) est présentée.

Enfin, les conclusions et perspectives des travaux (théorique et pratique) de recherche menés tout au long cette thèse clôturent ce présent travail.

CHAPITRE 1

Commande des robots manipulateurs

1.1 Introduction

Les robots manipulateurs sont actuellement d'une très large utilisation dans les applications industrielles et spatiales. Ils sont d'une importance majeure, surtout dans les travaux dangereux, fastidieux et monotones [Hassan. K et Khalil, 2002]. Ils se substituent à l'homme ou prolongent son action en apportant précision, rapidité ou capacité à appliquer d'importants efforts, afin de lui faciliter les tâches difficiles et répétitives, telles que le soulèvement de pièces lourdes, serrage, découpage, assemblage, etc... Le nombre des applications de ces manipulateurs a augmenté chaque jours, des millions de robots effectuent chaque jour, des tâches complexes de nature différente dans des usines, station de service et même à domicile (tondeuse à gazon, robot aspirateur, etc...). Avec les développements scientifiques, spécifiquement de l'électronique et de l'informatique, mais aussi : automatique, mathématique, mécanique et matériaux, que la technologie robotique a progressé, les robots actuels sont dotés d'une « intelligence » qui leur donne une certaine autonomie qui va leur permettre de se diffuser dans de nouveaux domaines. Nous parlons maintenant de robots d'exploration, de robots d'intervention, de robots ludiques, de robots de service, des robots chirurgicaux, etc. A cette époque les robots étaient conçus en respectant les contraintes imposées par le milieu industriel, comme la répétabilité, la précision dans la réalisation des tâches, le respect des cadences de production, etc... Prenons l'exemple des bras manipulateurs industriels des années 80, très semblables aux bras industriels actuels, mais ils se déplacent à très grandes vitesses sur des trajectoires calculées hors ligne dans des environnements protégés et structurés, sans interaction avec les opérateurs humains, pour pouvoir exploiter les capacités de ces robots, il faut élaborer des commandes qui peuvent assurer les performances voulues et pour atteindre cet objectif.

L'élaboration d'une loi de commande pour un procédé physique tel que les robots nécessitent la prise en compte de certains paramètres tels que le suivi de la consigne, le rejet de la perturbation, une marge de robustesse vis-à-vis de certains paramètres du procédé à piloter. La littérature propose une multitude de structure de commande, chacune d'elle s'applique dans un domaine particulier avec des propriétés particulières (cas linéaire, cas non linéaire, procédé stable, consigne d'un type donné,...etc.).

Il est impossible, de faire une synthèse (même partielle) des techniques de commande des bras manipulateurs, tant ce domaine recouvre de problématiques différentes. Outre les travaux liés à la modélisation et l'identification [Khalil, 1978] [Gorla et Renaud, 1984] [Khalil et Dombre,

1999], et à la planification de mouvements [Brooks, 1983] [Faverjon, 1984] [Craig, 1986], le traitement de la redondance [Liégeois, 1977] [Vukobratovic et Kircanski, 1984] [Bailleul et al., 1984] [Yoshikawa, 1985] [Sciavicco et Siciliano, 1988], des singularités géométriques, la commande dynamique [Khalil et al., 1979] [Luh et al., 1980]. La commande par retour d'efforts [Withney, 1977] [Withney, 1985] [Boissonnat et al., 1988], et plus généralement la commande référencée capteurs [Espiau et al., 1992] [Samson et al., 1991], sont autant de domaines de recherche qui mériteraient à eux seuls une synthèse bibliographique. En dehors des quelques références précitées, nous renvoyons le lecteur aux ouvrages et articles de synthèse suivants : [Dombre et Khalil, 1988] pour la modélisation, la commande dynamique et adaptative, [Samson et al., 1991]. Pour le cadre méthodologique de synthèse de retours d'état par l'approche par fonctions de tâches, [Canudas de Wit et al., 1996] pour un état de l'art sur la modélisation et les techniques de commande des bras manipulateurs, [Siciliano, 1990] [Yoshikawa, 1990] pour une synthèse des techniques de commande de manipulateurs redondants, [Asada et Slotine, 1986] pour des synthèses de commandes dynamiques.

1.2 Commande des bras manipulateurs

Dans [Sciavicco et al, 2000], le problème de la commande d'un robot manipulateur peut être formulé comme la détermination de l'évolution des forces généralisées (forces ou couples) que les actionneurs doivent exercer pour garantir l'exécution de la tâche tout en satisfaisant certains critères de performance. Différentes techniques sont utilisées pour la commande des bras manipulateurs. La conception mécanique du bras manipulateur a une influence sur le choix du schéma de commande [Ignacio H. A, 2007]. Un robot manipulateur est une structure mécanique complexe dont les inerties par rapport aux axes des articulations varient non seulement en fonction de la charge mais aussi en fonction de la configuration, des vitesses et des accélérations.

Deux types de mouvements apparaissent quand on parle de commande du bras manipulateur. Un premier type considère que les mouvements nécessaires pour la réalisation de la tâche sont exécutés dans l'espace libre. Le deuxième type considère des mouvements spécifiques avec des forces de contact pour l'organe terminal qui se déplace dans un espace contraint. Toute tâche de robotique est réalisée par une combinaison de ces deux types de mouvement. Pour simplifier la commande, les deux types de mouvement sont abordés séparément.

Nous ne pouvons pas dans le cadre de cette thèse traiter en détail l'ensemble des techniques de commande des bras manipulateurs. Pour une étude plus détaillée, le lecteur pourra se

référer à [Paul, 1979], [Gorla et al., 1984], [Khalil et al., 1999], [Sciavicco et al., 2000] et [Canudas et al., 1997]. Nous présentons juste une vue générale de différentes techniques rapportées dans la littérature. Dans les paragraphes suivants nous nous limiterons à une description simple des différentes techniques.

1.2.1 Commande par articulation

Cette technique est utilisée par des robots manipulateurs qui utilisent des servomoteurs avec de forts rapports de réduction. Lorsque le système présente un comportement linéaire, l'asservissement du mouvement peut être réalisé par des techniques classiques de commande. Nous parlons alors d'une commande décentralisée de type PID. Dans [Gorla et al., 1984] et [Canudas et al., 1997], le schéma classique est amélioré avec des signaux d'anticipation pour corriger les effets de la force de gravité et de couplage.

1.2.2 Commande jacobienne

Cette technique est utilisée depuis les travaux de Whitney [Whitney, 1969] et elle est appelée de cette façon lorsqu'elle utilise la matrice jacobienne inverse du bras manipulateur pour calculer les vitesses de consigne aux articulations. Elle est aussi connue sous le nom de commande à mouvement résolu. Les approches les plus courantes sont de type : mouvement à vitesse résolu, mouvement à accélération résolu [Luh et al., 1980] et mouvement à force résolu [Wu et al., 1982]. La technique de mouvement résolu commande la position de l'organe terminal du manipulateur dans l'espace cartésien, par combinaison des mouvements de plusieurs articulations.

Ce type de schémas peut se présenter sous la forme d'un schéma cinématique quand les vitesses calculées aux articulations sont utilisées directement comme des consignes pour les boucles de commande de chaque articulation, ou sous la forme d'un schéma dynamique quand le schéma utilise le modèle dynamique pour découpler les articulations.

Son principal inconvénient est l'utilisation de l'inverse de la matrice jacobienne, qui peut devenir singulière dans certaines conditions.

1.2.3 Commande par découplage non linéaire

Cette technique est aussi connue sous les noms de commande dynamique ou commande par couple calculé. Lorsque l'application exige des évolutions rapides avec des contraintes dynamiques, la commande doit prendre en compte les forces d'interaction. Ce type de technique considère l'ensemble des articulations et, pour les découpler, utilise la théorie du découplage non linéaire. Cette théorie utilise le modèle dynamique du robot pour le calcul de la loi de commande, ce qui conduit à des lois de commande centralisées non linéaires. Des signaux d'anticipation peuvent être utilisés pour minimiser des effets non linéaires [Er, 1993]. Ce type de technique permet la commande dans l'espace des articulations ou dans l'espace cartésien, avec l'avantage que les articulations sont découplées et peuvent évoluer à grandes vitesses avec de fortes inerties. Cette méthode dépend fortement du modèle du système, elle est très sensible aux imprécisions du modèle qui entraînent un découplage imparfait. Ceci constitue son principal inconvénient.

1.2.4 Commande adaptative

Ce type de techniques vise à corriger les déficiences de la commande par découplage non linéaire, comme la connaissance approximative des paramètres du modèle du robot ou pour s'adapter aux différentes conditions opératoires. Ce type de schémas cherche à estimer ou ajuster en ligne les valeurs des paramètres utilisées dans le calcul de la loi de commande. Un travail plus intéressants sur ce sujet est la commande proposé par [Slotine et al., 1987] appelée commande de Slotine-Li ou commande adaptative passive. Plusieurs travaux sur la commande adaptative sont présentés dans [Lozano et al., 2001].

Les avantages de ce type de techniques sont évidents, malheureusement la puissance de calcul demandée au système constitue un inconvénient important.

1.2.5 Commande fondée sur une fonction de Lyapunov

Des méthodes basées sur une fonction de Lyapunov ont été utilisées pour la commande des bras manipulateurs de façon satisfaisante pour des tâches de suivi. Particulièrement lorsqu'on cherche à garantir la convergence asymptotique et non à linéariser le système ou à obtenir le découplage [Canudas et al., 1997].

1.2.6 Commande passive

Cette technique considère le robot comme un système passif, c'est à dire un système qui dissipe de l'énergie. De telles lois de commande permettent de modifier l'énergie naturelle du robot pour qu'il réalise la tâche. En utilisant le formalisme de Hamilton, la commande cherche à minimiser l'énergie du système en utilisant un bloc non linéaire passif dans la boucle de retour. La commande passive tend à être plus robuste que le découplage non linéaire, lorsque la technique ne recherche pas à annuler les non linéarités [Sciavicco et al., 2000].

1.2.7 Commande prédictive

Ce type de commande, en utilisant le modèle du système et les consignes, est capable de prédire son évolution, de telle manière qu'il est capable d'agir en fonction de l'erreur de prédiction. Trois schémas différents sont proposés dans [Hedjar et al., 2002] : point final fixe, horizon fini et une combinaison des deux précédents. Un grand avantage de ce type d'approche est lié au fait que l'erreur de prédiction n'est pas contaminé par les bruits de mesure mais la dépendance au modèle reste forte.

1.2.8 Commande robuste

Dans le cas de paramètres fixes, il est connu que la technique de découplage non linéaire peut devenir instable en présence d'incertitudes. Si les paramètres du modèle ne sont pas connus de façon précise et si l'incertitude sur les paramètres admet des bornes connus, alors les techniques de commande robuste peuvent être utilisées. Par exemple, les travaux de Slotine [Slotine, 1985] considèrent la technique de mode glissant appelé aussi commande de structure variable. Cette technique utilise une surface de glissement où la stabilité du système est garantie.

1.2.9 Commande optimale

Pour réaliser une tâche, il peut exister un grand nombre de solutions. Dans ce cas, il peut être souhaitable de choisir une solution qui satisfasse un certain critère. La littérature présente différents types de critères pour la commande optimale : la commande en temps minimal [Kahn et al., 1971], [Bobrow et al., 1985], [Chen et al., 1989], du domaine des neurosciences la minimisation du jerk pour maximiser la souplesse du mouvement [Flash et al., 1985],

[Kyriakopoulos et al., 1988], [Seki et al., 2004], [Amirabdollahian et al., 2002] et la minimisation du couple [Egeland et al., 1988], [Martin et al., 1999], entre autres. La complexité du problème de commande optimale a motivé les chercheurs pour diviser la tâche en deux étapes : la première étape est la planification de trajectoire suivi d'une étape d'asservissement de la trajectoire [Lin et al., 1983], [Kim et al., 1985], [Shin et al., 1985].

1.3 Approche sélectionnée

Le travail effectué durant cette thèse a essentiellement contribué à l'extension des approches pour la modélisation et la commande d'un bras manipulateur à 2ddl, En premier lieu, nous avons utilisé la simulation pour exploiter l'approche par les différentes techniques classiques fréquemment utilisées dans la littérature telle que calcul de couple avec PD, commande adaptative par calcul de couple et commande adaptative par une approche inertielle. A côté de ces commandes conventionnelles, on a introduit la commande intelligente, qui essaye de faire une émulation des capacités de connaissances humaines ainsi que la prise de décision, dans l'objectif est de proposer des algorithmes de commande hybride (dans notre cas l'ANFIS) permettant de conserver la précision du suivi de la trajectoire, quelles que soient les conditions d'adhérence et la trajectoire de référence. Ensuite, la technique qui donne le meilleur résultat est appliquée à un vieux bras manipulateur hydraulique (FEEDBACK HRA934) avec modèle dynamique incertain abandonné pendant un certain temps dans notre laboratoire. Les résultats simulés et expérimentaux montrent l'efficacité et l'efficience de chaque technique.

CHAPITRE 2

Modélisation des robots manipulateurs

2.1. Introduction

Pour développer une stratégie de commande performante pour un robot (manipulateur), il est impératif de connaître la cinématique et la dynamique du robot considéré. Et comme les robots manipulateurs sont des systèmes mécaniques poly articulés [Mark W et al, 2004], fortement non linéaires, d'une dynamique qui peut être mal définie, ils ont des paramètres variables et sont soumis à des perturbations externes. Pour pouvoir faire exécuter une action ou une tâche au robot manipulateur, il est nécessaire de modéliser ce système multi variable pour positionner les différents axes. La modélisation de ces systèmes relève très souvent d'une approximation des phénomènes physiques mis en jeu [Craig, 1989], c'est à partir de cette représentation approximative que l'on souhaite construire une commande pour le système réel, cette commande doit être alors robuste dans le sens où elle devra assurer une faible sensibilité aux incertitudes des paramètres, à leurs variations et aux perturbations externes.

Les erreurs de modélisation, les incertitudes sur l'estimation des paramètres physiques ainsi que les différentes perturbations externes influent beaucoup sur la qualité du contrôle. Tous ces facteurs doivent être pris en compte lors de la modélisation et de l'élaboration de la loi de commande pour préserver la qualité et la précision des tâches auxquelles est destiné le robot. En effet la majorité des tâches confiées aux robots sont délicates et exigent une très grande précision sous des trajectoires précisées et rapides. Dans ce cas le type de commande nécessaire est la commande par découplage non linéaire, cette méthode qui est considérée comme la solution théorique idéale pour la commande de ce type de robot [Luh et al, 1980].

Dans ce chapitre, nous examinerons divers concepts de la théorie de contrôle utilisés dans la commande des robots. On élaborera le modèle d'un bras manipulateur à deux degrés de liberté (2ddl) à travers la mise en forme des équations mathématiques régissant le mouvement de ce bras. Nous montrerons comment inclure la dynamique des actionneurs du bras, qui peut être électriques ou hydrauliques.

2.2. Modélisation des robots manipulateurs

Pour commander un robot ou pour simuler son comportement on doit disposer de modèles. Plusieurs niveaux de modélisation sont possibles selon les objectifs, les contraintes de la tâche et les performances recherchées : modèles géométriques, cinématiques et dynamiques. L'obtention de ses différents modèles n'est pas aisée, la difficulté variant selon la complexité de la cinématique de la chaîne articulée. Entrent 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, arborescente ou fermée. Les outils mathématiques que nous présentons dans ce travail utilisent une description des mécanismes fondée sur des notations originales qui permettent d'avoir la même approche quelle que soit la structure.

Ces modèles, qu'ils soient utilisés en simulation ou en commande, doivent représenter le comportement géométrique, cinématique ou dynamique du robot de façon réaliste. Il est donc nécessaire de mettre en œuvre des procédures efficaces d'identification de leurs paramètres constitutifs. Pour qu'une commande puisse effectivement être implantée sur un contrôleur de robot, les modèles doivent être calculés en ligne et donc, le nombre d'opérations à effectuer doit être minimum. Les techniques proposées ont été développées dans ce souci et conduisent aux modèles les plus performants.

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

Le calcul symbolique de ces modèles par ordinateur a fait l'objet d'un grand nombre de travaux [Dillon, 73] [Khalil, 76] [Zabala, 78] [Kreuzer, 79] [Aldon, 82] [Cesareo, 84] [Megahed, 84] [Murray, 84] [Kircanski, 85] [Burdick, 88] [Izaguirre, 86] [Khalil, 89a]. Le logiciel SYMORO+ [Khalil, 97a], dans lequel on retrouve implantés tous les algorithmes présentes dans cet ouvrage, est certainement le plus performant et le seul à pouvoir traiter tous les modèles cités ci-dessus.

2.3. Modèle géométrique

Il permet de déterminer la configuration (position, orientation) de l'effecteur d'un robot en fonction de la configuration de ses liaisons, l'effecteur peut être une pince, une caméra, une pompe de peinture. La modélisation des robots de façon systématique et automatique exige une méthode adéquate pour la description de leur morphologie. Plusieurs méthodes et stations ont été proposées [Denavit, 55] [Sheth, 71] [Renaud, 75] [Khalil, 76] [Borrel, 79] [Craig, 86a].

2.3.1 Modèle géométrique direct (MGD)

Le modèle géométrique d'un mécanisme regroupe les contraintes géométriques qui doivent être respectées par les variables articulaires θ_i (ou bien q_i) afin d'établir la relation entre la configuration du mécanisme définie dans l'espace des coordonnées généralisées et la configuration du mécanisme décrite dans le repère cartésien [Cyril, 2009]. Le modèle géométrique direct est unique et est donné sous forme d'équations explicites.

$$X_i = f(\theta_i) \quad (\text{II.1})$$

Ou:

$\theta = [\theta_1, \theta_2, \dots, \theta_n]^T \in R^n$: Vecteur des variables articulaires.

$X = [x_1, x_2, \dots, x_n]^T \in R^n$: Vecteur des variables opérationnelles.

- **Calcul du MGD**

Pour calculer le MGD, on suppose que les segments du robot sont parfaitement rigides et les articulations mécaniquement parfaites, il existe plusieurs méthodes pour déterminer le MGD, la plus répandue est celle de Denavit-Hartenberg [Denavit, 55]. Mais cette méthode, développée pour des structures ouvertes simples, présente des ambiguïtés lorsqu'elle est appliquée sur des robots ayant des structures fermées ou arborescentes, il existe aussi la notation de Khalil et Kleinfinger qui permet la description homogène, et avec un nombre minimum de paramètres, des architectures ouvertes simple et complexe [Khalil, 86a].

2.3.2 Modèle géométrique inverse (MGI)

Le modèle géométrique direct d'un robot permet de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires, le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Le modèle s'écrit :

$$\theta = f^{-1}(x) \quad (\text{II.2})$$

Lorsqu'elle existe la forme explicite qui donne toutes les solutions possibles. (il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse (MGI). La détermination du modèle géométrique inverse (MGI) est un problème complexe. On doit inverser un système d'équations non linéaires ce qui n'est pas trivial. Toutefois et selon la structure du manipulateur étudié, il existe des méthodes permettant de résoudre le MGI sous une forme explicite. Nous présentons trois méthodes de calcul du MGI :

- La méthode de Paul [Paul, 81] qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels.
- La méthode de Pieper [Pieper, 68] qui permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïde d'axes concourants ou trois articulations prismatiques.
- La méthode générale de Raghavan et Roth [Raghavan, 90] donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

Pour la plupart des manipulateurs utilisés dans l'industrie, la méthode de Paul [Paul, 81]. Permet de donner les solutions du MGI sous forme explicite. Lorsqu'il n'est pas possible de trouver une forme explicite du modèle géométrique inverse, on peut calculer une solution particulière par des procédures numériques. La solution obtenue est locale en ce sens qu'elle dépend des conditions initiales. Citons par exemple :

- Les méthodes de type Newton-Raphson procédant par linéarisation de la matrice de passage du mécanisme [Pieper, 68].

- Les méthodes fondées sur le modèle différentiel inverse [Whitney, 69], [Fournier, 80], [Featherstone, 83a], [Goldenberg, 85].
- Les méthodes fondées sur la transposée de la matrice jacobienne [Wolovich, 84], [Sciavicco, 86].

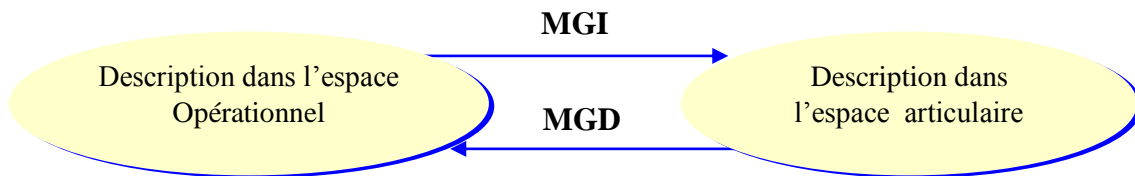


Figure II.1 : Modèle géométrique directe et inverse

2.4. Calcule des modèles géométriques (direct et inverse)

Dans ce travail de thèse, nous nous sommes intéressés au développe ultérieur du modèle géométrique et dynamique du robot, que nous allons utiliser pour la commande.

Nous avons considéré d'abord un robot manipulateur à 2ddl, donné par figure (II.2), dont le mouvement des articulations est rotoïde, nous avons donc $q_i \equiv \theta_i$

l_i : Longueur du segment i .

l_{ci} : Longueur du centre de gravité du segment i .

θ_i : La position angulaire de l'articulation i .

$$\phi = \theta_1 + \theta_2$$

2.4.1 Calcul du modèle géométrique direct

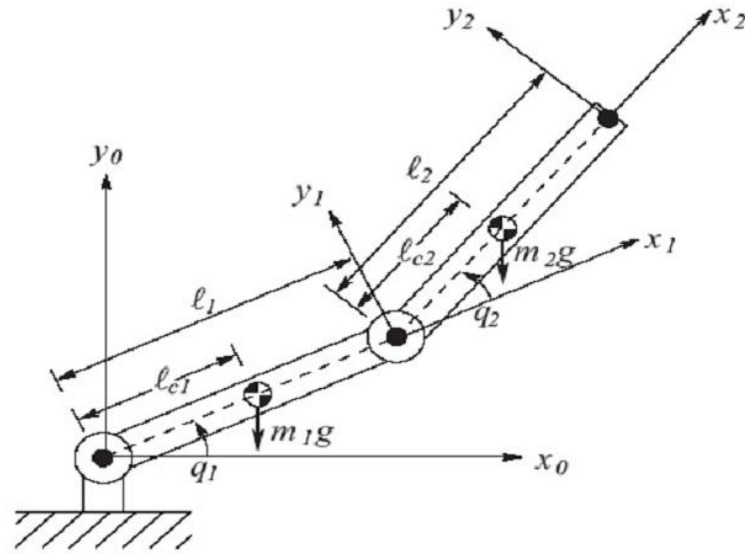


Figure II.2 : Les paramètres géométriques d'un bras à deux degré de liberté (2ddl)

A partir de la convention de Denavit-Hartenberg on obtient le tableau:

Tableau II.1 : Configuration de Denavit-Hartenberg du robot à 2ddl

Articulation	Paramètre d'articulation				
	σ_i	l_i	α_i	θ_i	d_i
1	0	l_1	0	θ_1	0
2	0	l_2	0	θ_2	0

Les matrices de transformation homogène des deux segments sont données par:

$${}^0T_1 = \begin{bmatrix} C_1 & -S_1 & 0 & l_1 C_1 \\ S_1 & C_1 & 0 & l_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.3})$$

$${}^1T_2 = \begin{bmatrix} C_2 & -S_2 & 0 & l_2 C_2 \\ S_2 & C_2 & 0 & l_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.4})$$

Après multiplication des matrices des transformations élémentaires, nous obtenons la matrice finale 0T_2 qui sert à déterminer le modèle géométrique direct :

$${}^0T_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_2 C_{12} + l_1 C_1 \\ S_{12} & C_{12} & 0 & l_2 S_{12} + l_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.5})$$

Avec : $\cos \theta_i = C_i$, $\sin \theta_i = S_i$

$$\cos(\theta_i + \theta_j) = C_{ij}, \quad \sin(\theta_i + \theta_j) = S_{ij}$$

$${}^0T_2 = \begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.6})$$

D'après les deux expressions de 0T_2 précédentes, nous obtenons les équations suivantes :

$$\begin{cases} x_x = C_{12} \dots\dots\dots (1) \\ x_y = S_{12} \dots\dots\dots (2) \\ x_z = 0 \dots\dots\dots (3) \end{cases} \quad (\text{II.7})$$

$$\begin{cases} y_x = -S_{12} \dots\dots\dots (4) \\ y_y = C_1 \dots\dots\dots (5) \\ y_z = 0 \dots\dots\dots (6) \end{cases} \quad (\text{II.8})$$

$$\begin{cases} Z_x = 0 \dots\dots\dots (7) \\ Z_y = 0 \dots\dots\dots (8) \\ Z_z = 1 \dots\dots\dots (9) \end{cases} \quad (\text{II.9})$$

Les coordonnées de position :

$$\begin{cases} p_x = l_2 C_{12} + l_1 C_1 \dots\dots\dots (10) \\ p_y = l_2 S_{12} + l_1 S_1 \dots\dots\dots (11) \\ p_z = 0 \dots\dots\dots (12) \end{cases} \quad (\text{II.10})$$

2.4.2 Calcule du modèle géométrique inverse :

➤ **Calcul de θ_2 :**

D'après les équations tire de l'analyse cinématique directe basée sur la géométrie [Bidaud et al, 2014] nous obtenons:

$$p_x^2 + p_y^2 = l_2^2 C_{12}^2 + l_1^2 C_1^2 + 2l_2 C_{12} l_1 C_1 + l_2^2 S_{12}^2 + l_1^2 S_1^2 + 2l_2 l_1 S_{12} S_1 \quad (\text{II.11})$$

Nous obtenons :

$$p_x^2 + p_y^2 = l_1^2 + l_2^2 + 2l_2 l_1 (C_{12} C_1 + S_{12} S_1) \quad (\text{II.12})$$

$$\Rightarrow C_{12} C_1 + S_{12} S_1 = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1^2 l_2} \quad (\text{II.13})$$

En utilisant les relations trigonométriques suivantes:

$$\sin(\alpha + \beta) = \sin \alpha \cdot \cos \beta + \cos \alpha \cdot \sin \beta \quad \text{et} \quad \cos(\alpha + \beta) = \cos \alpha \cdot \cos \beta - \sin \alpha \cdot \sin \beta$$

Et on posant : $D = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1^2 l_2}$ nous obtenons : $C_2 = D$ et $S_2 = \pm \sqrt{1 - D^2}$

Posons : $\xi = \pm 1$.

Ce qui implique que :

$$\theta_2 = ATAN2(\xi \sqrt{1 - D^2}, D) \quad (\text{II.14})$$

La fonction $ATAN2(\sin \alpha, \cos \alpha)$ est presque similaire au $ArcTg$, elle donne l'angle α en fonction des signes des sinus et cosinus.

➤ **Calcul de θ_1 :**

En reprenant les équations tire de l'analyse cinématique directe basée sur la géométrie nous pouvons encore écrire en développant C_{12}, S_{12} :

$$\begin{aligned} p_x &= l_1 C_1 + l_2 (C_1 C_2 - S_1 S_2) \\ p_y &= l_1 S_1 + l_2 (S_1 C_2 + S_2 C_1) \end{aligned} \quad (\text{II.15})$$

En regroupant suivant les inconnues C_1 et S_1 , nous trouvons

$$\begin{aligned} p_x &= (l_1 + l_2 C_2) C_1 - l_2 S_2 S_1 \\ p_y &= l_2 S_2 C_1 + (l_1 + l_2 C_2) S_1 \end{aligned} \quad (\text{II.16})$$

On résout le système (II.16) par rapport aux deux inconnues C_1 et S_1 , le déterminant vaut

$p_x^2 + p_y^2$, C_1 et S_1 sont donnés par:

$$\begin{aligned} C_1 &= \frac{p_x (l_1 + l_2 C_2) + p_y l_2 S_2}{p_x^2 + p_y^2} \\ S_1 &= \frac{p_y (l_1 + l_2 C_2) + p_x l_2 S_2}{p_x^2 + p_y^2} \end{aligned}$$

D'où

$$\theta_1 = \text{ATAN2}(S_1, C_1)$$

2.5 Modèle cinématique

Le modèle cinématique permet de trouver une solution au problème géométrique inverse quelle que soit la structure du robot. Il permet aussi d'introduire la notion de vitesse de déplacement du mécanisme, pour calculer les vitesses de translation et de rotation du corps terminal en fonction des vitesses articulaires. Il est donc à la base de leur commande en vitesse. L'outil nécessaire à mettre en œuvre pour ce faire est le modèle cinématique inverse. Dans ce modèle en utilisant la matrice jacobienne de base. Cette matrice peut être décomposée en deux ou trois matrices contenant des termes plus simples. Nous avons ensuite montré comment utiliser la matrice jacobienne pour analyser l'espace de travail et l'espace des vitesses d'un robot. Nous avons aussi vu comment on pouvait utiliser la matrice jacobienne pour formuler le modèle statique et nous avons mis en évidence la dualité de ce modèle avec

le modèle cinématique. Enfin, les modèles cinématiques associés aux différentes représentations des coordonnées opérationnelles ont été établis.

2.5.1 Modèle cinématique direct

Le modèle cinématique complète le modèle géométrique, il permet de déterminer la vitesse de l'organe terminal dans l'espace opérationnel en fonction de la vitesse des variables articulaires. La propriété intéressante du modèle cinématique est sa linéarité par rapport aux vitesses. Ainsi il est plus simple à manipuler que le modèle géométrique et permet d'utiliser des propriétés qui découlent de la résolution des systèmes linéaires. Le modèle est décrit par l'équation :

$$\dot{X} = J(q) \dot{q} \quad (\text{II.17})$$

Où : $J(q) \in R^{m \times n}$ est la matrice jacobienne de dimension $(m \times n)$ du mécanisme, égale à : $\frac{\partial X}{\partial q}$

et fonction de la configuration articulaire q . La même matrice jacobienne intervient dans le calcul du modèle différentiel direct qui donne les variations élémentaires dX des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires dq , soit : $dX = J(q) dq$.

L'intérêt de la matrice jacobienne est multiple [Whitney, 69], [Paul, 81] :

- Elle est à la base du modèle différentiel inverse, permettant de calculer une solution locale des variables articulaires q connaissant les coordonnées opérationnelles X .
- En statique, on utilise la jacobienne pour établir la relation liant les efforts exercés par l'organe terminal sur l'environnement aux forces et couples des actionneurs.
- Elle facilite le calcul des singularités et de la dimension de l'espace opérationnel accessible du robot [Borrel, 86], [Wenger, 89].

L'une des méthodes utilisées pour le calcul de la matrice jacobienne est la dérivation du modèle géométrique direct :

$$J(q) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \quad (\text{II.18})$$

2.5.2 Modèle cinématique inverse

L'objectif du modèle cinématique inverse est de calculer, à partir d'une configuration q donnée, les vitesses articulaires \dot{q} qui assurent au repère terminal une vitesse opérationnelle \dot{X} imposée. Cette définition est analogue à celle du modèle différentiel inverse : ce dernier permet de déterminer la différentielle articulaire dq correspondant à une différentielle des coordonnées opérationnelles dX spécifiée. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires. Il existe plusieurs méthodes pour la détermination du modèle cinématique inverse en considérant trois cas, selon que le manipulateur est régulier, en configuration singulière ou redondant. Ces méthodes sont soit analytiques, soit numériques.

- ✚ La solution analytique: les méthodes analytiques conviennent pour résoudre le problème inverse lorsque la matrice jacobienne associée au mécanisme est régulière. Leur mise en œuvre est simplifiée lorsque le poignet du mécanisme possède trois articulations rotoïdes d'axes concourants : le problème se réduit alors à l'inversion de deux matrices régulières d'ordre trois. Elle a l'avantage de diminuer considérablement le nombre d'opérations, mais on doit traiter séparément tous les cas singuliers [Chevallereau, 87].
- ✚ Les méthodes numériques : sont plus générales, la plus répandue étant fondée sur la notion de pseudo-inverse : les algorithmes traitent de façon unifiée les cas réguliers, singuliers et redondants. Elles nécessitent un temps de calcul relativement important.

2.6 Modèle dynamique

Le modèle dynamique est la relation entre les couples et/ou forces applique aux actionnaires et les positions, vitesses et accélérations. On représenter le modèle dynamique par une relation de la forme :

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad (\text{II.19})$$

Avec :

Γ : Vecteurs des couples/forces des actionnaires, selon que l'articulation est rotoïde ou prismatique. (dans la suite en écrira tout simplement couple).

q : Vecteur des positions articulaires.

\dot{q} : Vecteur des vitesses articulaires.

\ddot{q} : Vecteur des accélérations articulaires.

f_e : Vecteur représentant les efforts extérieur (fores et moment) qu'exerce le manipulateur sur l'environnement.

2.6.1 Modèle dynamique inverse

Le modèle dynamique inverse exprime les couples exercés par les actionneurs en fonction des positions, vitesses et accélérations des articulations. Le modèle dynamique inverse est représenté par une relation de la forme :

$$\Gamma = f(\theta, \dot{\theta}, \ddot{\theta}) \quad (\text{II.20})$$

Γ : Vecteur des couple/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique.

θ : Vecteur des positions articulaires.

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

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

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

- Le formalisme de Lagrange-Euler : cette méthode est mieux adaptée pour le calcul du modèle dynamique que l'on utilise en simulation.

- Le formalisme de Newton-Euler: cette méthode est mieux adaptée pour le calcul du modèle dynamique que l'on utilise en commande.

Nous allons voir dans ce qui suit la modélisation d'un bras manipulateur dynamique rigide à chaîne ouverte simple à deux degrés de liberté et en tenant compte des effets de la vitesse, on va utiliser le formalisme de Lagrange pour déterminer les équations du mouvement pour un bras manipulateur à deux degrés de liberté à chaîne ouverte simple.

2.6.1.1 Formalisme de Lagrange :

Le formalisme de Lagrange décrit les équations du mouvement en termes de travail et d'énergie du robot. La forme générale représentant le mouvement d'un robot manipulateur constitué de n degrés de liberté et de n articulations, s'écrit suivant Lagrange :

$$\Gamma_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} \quad (\text{II.21})$$

Avec : $i = 1, 2, \dots, n$.

L : Fonction lagrangienne de robot égale à : $L = E_c - E_p$.

E_c : Energie cinétique totale du robot manipulateur.

E_p : Energie potentielle totale du robot manipulateur.

$q \in R^n$: Vecteur des positions articulaires.

$\dot{q} \in R^n$: Vecteur des vitesses articulaires.

Les forces généralisées produites par cette méthode ne représentent pas nécessairement des forces physiques réelles qui peuvent être identifiées et ne sont pas constituées de forces contraintes qui n'effectuent aucun travail, car ces forces sont éliminées. L'équation pour les forces généralisées s'écrit :

$$\Gamma_{q_i} = \sum \bar{F} \cdot \frac{\partial \bar{p}}{\partial \bar{q}} = \sum \left(F_x \cdot \frac{\partial x}{\partial q_i} + F_y \cdot \frac{\partial y}{\partial q_i} + F_z \cdot \frac{\partial z}{\partial q_i} \right) \quad (\text{II.22})$$

Où :

F_x, F_y, F_z : sont les composantes des forces et x, y, z sont les coordonnées des points d'application de ces forces définies en fonction des coordonnées généralisées pour permettre leurs dérivées.

Dans le cas d'une articulation rotoïde $q_i \equiv \theta_i$, tandis que pour une articulation prismatique $q_i \equiv d_i$.

Dans notre travail, nous avons considéré d'abord un robot manipulateur à 2ddl, donné par figure (II-3), dont le mouvement des articulations est rotoïde, nous avons donc $q_i \equiv \theta_i$

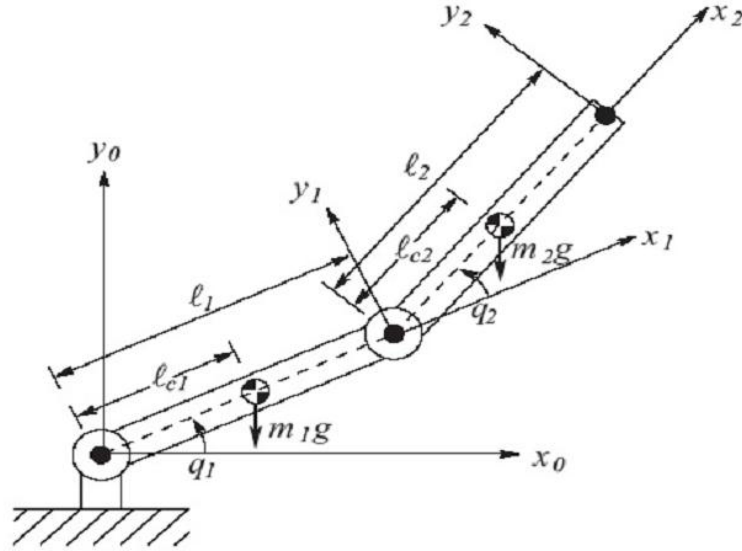


Figure II.3 : Bras à deux degré de liberté (2ddl)

Tel que :

$$q = [\theta_1 \quad \theta_2]^T \text{ et } \tau = [\tau_1 \quad \tau_2]^T$$

A partir de la figure (II.3), nous déterminons les coordonnées du bras manipulateur par rapport au repère xy, ce qui donne :

$$\begin{cases} x_1 = l_{c1} \cos \theta_1 \\ y_1 = l_{c1} \sin \theta_1 \end{cases} \quad (\text{II.23})$$

$$\begin{cases} x_2 = l_1 \cos \theta_1 + l_{c2} \cos \phi \\ y_2 = l_1 \sin \theta_1 + l_{c2} \sin \phi \end{cases} \quad (\text{II.24})$$

Ce qui donne :

$$\begin{cases} \dot{x}_1 = -l_{c1} \dot{\theta}_1 \sin \theta_1 \\ \dot{y}_1 = l_{c1} \dot{\theta}_1 \cos \theta_1 \end{cases} \quad \text{et} \quad \begin{cases} \dot{x}_1^2 = l_{c1}^2 \dot{\theta}_1^2 \sin^2 \theta_1 \\ \dot{y}_1^2 = l_{c1}^2 \dot{\theta}_1^2 \cos^2 \theta_1 \end{cases}$$

$$\begin{cases} \dot{x}_2 = -(l_1 \dot{\theta}_1 \sin \theta_1 + l_{c2} \dot{\phi} \sin \phi) \\ \dot{y}_2 = l_1 \dot{\theta}_1 \cos \theta_1 + l_{c2} \dot{\phi} \cos \phi \end{cases} \quad \text{et}$$

$$\begin{cases} \dot{x}_2^2 = l_1^2 \dot{\theta}_1^2 \sin^2 \theta_1 + l_{c2}^2 \dot{\phi}^2 \sin^2 \phi + 2l_1 l_{c2} \dot{\theta}_1 \dot{\phi} \sin \phi \sin \theta_1 \\ \dot{y}_2^2 = l_1^2 \dot{\theta}_1^2 \cos^2 \theta_1 + l_{c2}^2 \dot{\phi}^2 \cos^2 \phi + 2l_1 l_{c2} \dot{\theta}_1 \dot{\phi} \cos \phi \cos \theta_1 \end{cases}$$

• **Détermination de l'énergie cinétique E_c**

$$E_c = \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\phi}^2 + \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \quad (\text{II.25})$$

Ce qui donne :

$$E_c = \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + I_2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \quad (\text{II.26})$$

Avec :

$$\begin{cases} v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 = l_{c1}^2 \dot{\theta}_1^2 \\ v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = l_1^2 \dot{\theta}_1^2 + l_{c2}^2 \dot{\theta}_2^2 + l_{c2}^2 \dot{\theta}_2^2 + 2l_{c2}^2 \dot{\theta}_1 \dot{\theta}_2 + 2l_1 l_{c2} \dot{\theta}_1^2 \cos \theta_2 + 2l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2 \end{cases}$$

L'équation (II.26) devient :

$$\begin{aligned} E_c = & \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + I_2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} m_1 l_{c1}^2 \dot{\theta}_1^2 \\ & + \frac{1}{2} m_2 (l_1^2 \dot{\theta}_1^2 + l_{c2}^2 \dot{\theta}_2^2 + l_{c2}^2 \dot{\theta}_1^2 + 2l_{c2}^2 \dot{\theta}_1 \dot{\theta}_2 + 2l_1 l_{c2} \dot{\theta}_1^2 \cos \theta_2 + 2l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2) \end{aligned} \quad (\text{II.27})$$

• **Détermination de l'énergie potentielle E_p**

$$E_p = m_1 g h_1 + m_2 g h_2 \quad (\text{II.28})$$

Avec :

$$\begin{cases} h_1 = y_1 = l_{c1} \sin \theta_1 \\ h_2 = y_2 = l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2) \end{cases}$$

L'équation (II.28) devient :

$$E_p = m_1 g l_{c1} \sin \theta_1 + m_2 g (l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2)) \quad (\text{II.29})$$

- **Détermination du Lagrangien** $L = E_c - E_p$

$$\begin{aligned} L = & \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + I_2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} m_1 l_{c1}^2 \dot{\theta}_1^2 - m_1 g l_{c1} \sin \theta_1 \\ & - m_2 g (l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2)) \\ & + \frac{1}{2} m_2 (l_1^2 \dot{\theta}_1^2 + l_{c2}^2 \dot{\theta}_2^2 + l_{c2}^2 \dot{\theta}_1^2 + 2l_{c2}^2 \dot{\theta}_1 \dot{\theta}_2 + 2l_1 l_{c2} \dot{\theta}_1^2 \cos \theta_2 + 2l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2) \end{aligned} \quad (\text{II.30})$$

- **Détermination du modèle dynamique du robot**

Nous appliquons maintenant l'équation d'Euler-Lagrange donnée par équation (II.21) pour déterminer le modèle dynamique correspondant au robot manipulateur à 2ddl, soit :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} = \Gamma_i, \quad i=1,2 \quad (\text{II.31})$$

Nous avons :

$$\frac{\partial L}{\partial \theta_1} = -m_1 g (l_{c1} \cos \theta_1) - m_2 g (l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2)) \quad (\text{II.32})$$

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_1} = & I_1 \dot{\theta}_1 + I_2 \dot{\theta}_1 + I_2 \dot{\theta}_2 + m_1 l_{c1}^2 \dot{\theta}_1 + m_2 (l_1^2 \dot{\theta}_1 + l_{c2}^2 \dot{\theta}_1 + l_{c2}^2 \dot{\theta}_2 \\ & + 2l_1 l_{c2} \dot{\theta}_1 \cos \theta_2 + l_1 l_{c2} \dot{\theta}_2 \cos \theta_2) \end{aligned} \quad (\text{II.33})$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) = & I_1 \ddot{\theta}_1 + I_2 \ddot{\theta}_1 + I_2 \ddot{\theta}_2 + m_1 l_{c1}^2 \ddot{\theta}_1 + m_2 (l_1^2 \ddot{\theta}_1 + l_{c2}^2 \ddot{\theta}_1 + l_{c2}^2 \ddot{\theta}_2 + 2l_1 l_{c2} \dot{\theta}_1 \cos \theta_2 \\ & - 2l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2 + l_1 l_{c2} \ddot{\theta}_2 \cos \theta_2 - l_1 l_{c2} \dot{\theta}_2^2 \sin \theta_2) \end{aligned} \quad (\text{II.34})$$

Ce qui donne :

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} &= (I_1 + I_2 + m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2)) \ddot{\theta}_1 \\ &\quad + (I_2 + m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2)) \ddot{\theta}_2 + (-m_2 l_1 l_{c2} \sin \theta_2) \dot{\theta}_2^2 \\ &\quad + 2(-m_2 l_1 l_{c2} \sin \theta_2) \dot{\theta}_1 \dot{\theta}_2 + m_1 g l_{c1} \cos \theta_1 + m_2 g (l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2)) \end{aligned} \quad (\text{II.35})$$

Nous avons aussi :

$$\frac{\partial L}{\partial \theta_2} = -m_2 (l_1 l_{c2} \dot{\theta}_1^2 \sin \theta_2 + l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2) - m_2 g l_{c2} \cos(\theta_1 + \theta_2) \quad (\text{II.36})$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = I_2 \dot{\theta}_2 + I_2 \dot{\theta}_1 + m_2 (l_{c2}^2 \dot{\theta}_2 + l_{c2}^2 \dot{\theta}_1 + l_1 l_{c2} \dot{\theta}_1 \cos \theta_2) \quad (\text{II.37})$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) = I_2 \ddot{\theta}_2 + I_2 \ddot{\theta}_1 + m_2 (l_{c2}^2 \ddot{\theta}_2 + l_{c2}^2 \ddot{\theta}_1 + l_1 l_{c2} \ddot{\theta}_1 \cos \theta_2 - l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2) \quad (\text{II.38})$$

Ce qui donne :

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} &= (I_2 + m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2)) \ddot{\theta}_1 + (I_2 + m_2 l_{c2}^2) \ddot{\theta}_2 + (m_2 l_1 l_{c2} \sin \theta_2) \dot{\theta}_1^2 \\ &\quad + m_2 g l_{c2} \cos(\theta_1 + \theta_2) \end{aligned} \quad (\text{II.39})$$

A partir de l'équation (II.21), nous avons :

$$\begin{cases} \tau_1 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} \\ \tau_2 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} \end{cases} \quad (\text{II.40})$$

En robotique, si le mouvement des articulations constituant le robot manipulateur est rotoïde, l'expression du couple peut être s'exprimer sous la forme matricielle suivante, ce qui correspondant au modèle dynamique du robot :

$$\Gamma_i = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta), \quad (i = 1, 2) \quad (\text{II.41})$$

Avec :

Γ : Vecteur des couples ou des forces généralisés.

θ : Vecteur des variables articulaires du bras manipulateur.

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

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

$M(\theta)$: Matrice d'inertie de dimension $(n \times n)$.

$C(\theta, \dot{\theta})\dot{\theta}$: Vecteur des forces centrifuges et de Coriolis.

$G(\theta)$: Vecteur de force de gravité de la charge.

Soit $\Gamma_i = [\tau_1 \quad \tau_2]^T$. Le développement de l'équation (II.40) donne :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 2h_{122}\dot{\theta}_2 & h_{211}\dot{\theta}_2 \\ h_{211}\dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \quad (\text{II.42})$$

Ce qui donne :

$$\begin{cases} \tau_1 = M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + h_{122}\dot{\theta}_2^2 + 2h_{112}\dot{\theta}_1\dot{\theta}_2 + g_1 \\ \tau_2 = M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + h_{211}\dot{\theta}_1^2 + g_2 \end{cases} \quad (\text{II.43})$$

L'identification entre équation (II.43) et équation (II.40) permet de définir les éléments suivants:

$$M_{11} = I_1 + I_2 + m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2)$$

$$M_{12} = M_{21} = I_2 + m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2)$$

$$M_{22} = I_2 + m_2 l_{c2}^2$$

$$h_{122} = h_{112} = -h_{211} = -m_2 l_1 l_{c2} \sin \theta_2$$

$$g_1 = m_1 g (l_{c1} \cos \theta_1) + m_2 g (l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2))$$

$$g_2 = m_2 g l_{c2} \cos(\theta_1 + \theta_2)$$

La matrice d'inertie M est égale à :

$$M(\theta) = \begin{bmatrix} I_1 + I_2 + m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2) & I_2 + m_2(l_{c2}^2 + l_1 l_{c2} \cos \theta_2) \\ I_2 + m_2(l_{c2}^2 + l_1 l_{c2} \cos \theta_2) & I_2 + m_2 l_{c2}^2 \end{bmatrix} \quad (\text{II.44})$$

Le vecteur de termes Coriolis et centrifuge est donnée par :

$$c(\theta, \dot{\theta})\dot{\theta} = \begin{bmatrix} -2m_2 l_1 l_{c2} \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 l_1 l_{c2} \sin \theta_2 \dot{\theta}_2^2 \\ m_2 l_1 l_{c2} \sin \theta_2 \dot{\theta}_1^2 \end{bmatrix} \quad (\text{II.45})$$

Le vecteur des termes de forces de gravité G est donné par l'équation suivante:

$$\begin{bmatrix} m_1 g l_{c1} \cos \theta_1 + m_2 g l_1 \cos \theta_1 + m_2 g l_{c2} \cos(\theta_1 + \theta_2) \\ m_2 g l_{c2} \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (\text{II.46})$$

2.6.2 Modèle dynamique direct

Pour simuler le comportement du robot et de sa boucle de commande, on utilise le modèle dynamique direct. Même si les contraintes de temps de calcul sont moins critiques que pour la commande, il est cependant intéressant de disposer d'un modèle performant, ou le modèle dynamique direct exprime la position, la vitesse et l'accélération des articulations en fonction du couple appliqué [Y. Baazi, 2000]. Il est obtenu par inversion du modèle précédent, il s'écrit :

$$\ddot{q} = M^{-1}(q)(\tau - N(q, \dot{q}) - G(q) - \tau_f) \quad (\text{II.47})$$

Nous présentons ici deux méthodes, l'une qui procède par inversion de la matrice d'inertie A . l'autre qui calcule les accélérations articulaires sans inversion.

✚ Calculer par inversion de la matrice d'inertie A du robot :

A partir de l'équation :

$$\Gamma = A(q)\ddot{q} + C(q, \dot{q})\dot{q} + Q(q) \quad (\text{II.48})$$

Et en supposant l'existence d'un effort extérieur sur l'organe terminal.

Ou \mathbf{A} est la matrice ($n \times n$) de l'énergie cinétique, d'élément générique A_{ij} , appelée aussi matrice d'inertie du robot, qui est symétrique est définie positive [Wisama et Dombre. 99]. Ses éléments sont fonction des variables articulaires \mathbf{q} .

Le modèle dynamique direct s'écrit :

$$\ddot{\mathbf{q}} = \mathbf{A}^{-1} [\Gamma - H(\mathbf{q}, \dot{\mathbf{q}})] \quad (\text{II.49})$$

Avec :

$$H(\mathbf{q}, \dot{\mathbf{q}}) = C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + Q + J^T f_{en}$$

✚ Calcule du modèle dynamique direct sans inversion de la matrice d'inertie :

Cette méthode utilise les équations de Newton-Euler et permet de calculer les accélérations articulaires sans avoir à inverse la matrice d'inertie \mathbf{A} du robot [Amstrong, 79], [Featherstone, 83b], [brandl, 86].

Le calculer de la matrice \mathbf{A} par une méthode lagrangienne peut être moins couteux en temps [Megahed, 82], [Renaud, 85]. On peut cependant utiliser l'algorithme de Newton-Euler pour calculer ses éléments [Walker, 82] en remarquant, à partir de la relation (II-48), que la $i^{\text{ème}}$ colonne est égal à Γ si :

$$\dot{\mathbf{q}} = 0, \quad \mathbf{g} = 0, \quad \ddot{\mathbf{q}} = \mathbf{u}_i, \quad f_{en} = 0, \quad m_{en} = 0$$

\mathbf{u}_i Étant le vecteur ($n \times 1$) dont tous les éléments sont nuls sauf le $i^{\text{ème}}$ composant qui est égal à 1.

2.7 Modèle hydraulique

Un système hydraulique est constitué des vérins et pompes hydrauliques, des flexibles entre les différentes unités, et des valves pour commander l'écoulement de fluide hydraulique. Le modèle hydraulique d'un robot inclut la dynamique des corps rigide des links, et la dynamique des actionnaires hydrauliques. Ce qui nous donne un système fortement non linéaire. Les équations dynamiques du manipulateur qui sont une extension de la cinématique impliquent les forces qui provoquent le mouvement des links de robot sont ensuite présentées. Le modèle dynamique complet de robot est présenté dans une forme globale avec le modèle des actionnaires hydrauliques étant formulé de telle sorte qu'il est applicable à un manipulateur avec un nombre arbitraire de joints.

Le système hydraulique peut être séparé en deux parties : une servovalve électrohydraulique qui régule l'écoulement de fluide hydraulique, et un actionneur (actionnaire) qui génère le mouvement de l'articulation. La dynamique de ces composants hydrauliques est d'abord formulée et leurs caractéristiques statiques sont examinées, permettant à certaines des non-linéarités d'être visualisées. Ce modèle d'actionneur est alors inclus dans le modèle de manipulateur ordinaire avec les corps rigides. Le modèle global est étendu de telle sorte qu'il est applicable aux robots avec un nombre arbitraire d'articulations à commande hydraulique. Ce travail ne considérera pas l'hydraulique de la pompe.

2.7.1 Modélisation des servovalves électrohydrauliques

Les servovalves électrohydrauliques sont utilisées pour régler le débit de liquide hydraulique, permettre des signaux électriques de faible puissance (moins de 1 W) de commander avec précision des éléments hydrauliques de grande puissance (jusqu'à 7000 W). Dans les systèmes robotiques ils règlent l'écoulement de fluide dans les vérins hydrauliques rotatifs ou linéaires, ce qui fournit le mouvement aux joints du manipulateur.

Il y a beaucoup de différents types de servovalve [Stringer 82], généralement les plus utilisés sont les servovalves à deux étages à quatre voies. Ceux-ci règlent le flux de liquide hydraulique utilisant le déplacement d'un tiroir central, qui ouvre partiellement chaque port par lequel le liquide peut circuler, comme indiqué dans la figure II.4. Le type de la valve représenté est appelé une servovalve à quatre voies, car il dispose de quatre connexions hydrauliques ; une pour la pression d'alimentation P_s , l'autre pour l'échappement P_e et les deux

orifices qui restent, l'une pour l'alimentation du circuit hydraulique P_1 et l'autre étant le retour correspondant P_2 .

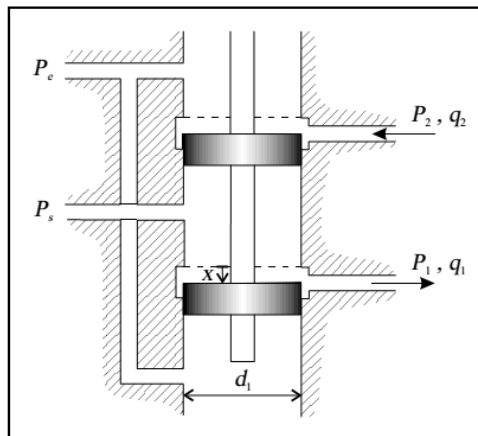


Figure II.4 : Schéma d'une servovalve

Le mouvement de la bobine est assuré au moyen d'un moteur à couple qui convertit une entrée électrique dans une petite déviation angulaire, cette déviation entraîne une valve buse-palette (nozzle-flapper valve) (non représentée sur la figure II.4), qui à son tour actionnait la bobine centrale de la valve principale à quatre voix. Généralement, les moteurs à couple sont actionnés par deux bobines et lorsqu'il est entraîné par une source de courant appropriée, la constante de temps du circuit est insignifiante par rapport au système complet.

L'analyse dynamique générale de ce type de servovalve donne une expression du sixième ordre [Thayer, 65]. Cependant, la fonction de transfert du premier ordre suivante peut être utilisée avec une faible perte de précision pour des fréquences allant jusqu'à 200 Hz. [Clegg, 94] :

$$\frac{x(s)}{i(s)} = \frac{k_i}{\tau_i s + 1} \quad (\text{II.50})$$

Où x est le déplacement de la bobine (m), et i est le courant d'entrée de la bobine (mA). Pour l'analyse statique ou les faibles fréquences et la dynamique de la servovalve peut être complètement ignorée, et le gain de la bobine de servovalve est simplement K_i .

Lorsque la bobine, de diamètre interne d_1 , est déplacé d'une distance x , deux orifices annulaires sont formés, chacun de diamètre d_1 et une largeur x , donner une surface d'orifice de $\pi d_1 x$. Supposons que l'ouverture des soupapes peut être traitée comme orifices est valable puisque x est petit comparé à d_1 ce qui est impliqué que l'alimentation électrique est disponible dans la chute de pression à travers l'orifice est convertie entièrement en énergie

cinétique du liquide. Donc, utilisant l'analyse de dynamique du fluide standard [Stringer, 82], l'expression pour débit volumique par ces orifices, q_1 et q_2 peut être constaté que :

$$q_1 = q_2 = \frac{C_d \pi d_1}{\sqrt{\rho}} x \sqrt{P_s - \text{sgn}(x) P_m} \quad (\text{II.51})$$

Où ρ est la densité du fluide hydraulique on suppose qu'il est constant, et $\text{sgn}(x)$ est la fonction signe ou :

$$\text{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

C_d constant, représente le fait que la réaction formée de l'écoulement par l'orifice est plus petit que l'orifice réel, en raison de l'écoulement turbulent. Typiquement, cela a une valeur dans la plage de 0.5 à 0.6, ici il est pris pour être 0.5 de manière à correspondre au débit nominal de ces servovalves spécifiques. Supposons que la servovalve est symétrique, ce qui permet la chute de pression à travers chaque orifice d'être égalisé $p_s - p_1 = p_2 - p_e$. Aussi, la pression d'échappement est considéré négligeable ($p_e = 0$). Et la pression de charge est définie comme $p_m = p_1 - p_2$. En utilisant cette définition de p_m et les hypothèses de la servovalve symétrie et la pression d'échappement négligeable, il peut être montré que :

$$\begin{aligned} \dot{P}_1 &= \frac{1}{2} \dot{P}_m \\ \dot{P}_2 &= -\frac{1}{2} \dot{P}_m \end{aligned} \quad (\text{II.52})$$

D'autres facteurs qui présentent des nouvelles variations non-linéaires dans le débit sont la température, degré de tour [Stringer 82], fuite interne, résolution mécanique, retard et dérive du signal d'entrée. Ces caractéristiques ne sont pas prises en compte dans le développement de ce modèle.

2.7.2 Vérins hydrauliques linéaires

Dans les robots hydrauliques il existe deux grandes catégories d'actionneur, les actionneurs linéaires et les actionneurs rotatifs. Seulement les actionneurs linéaires sont étudiés ici, Un schéma de principe d'un actionneur hydraulique linéaire est représenté sur la figure II.5.

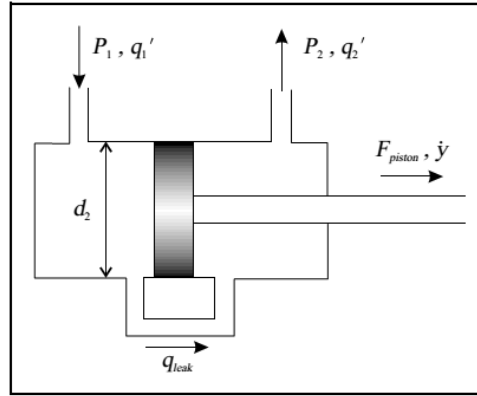


Figure II.5 : Actionnaire hydraulique linéaire

Comme le fluide entre dans le côté gauche du vérin, la pression à travers le piston augmente, qui crée une force, F_{piston} qui à son tour produit un mouvement linéaire du piston \dot{y} . Ce mouvement fait partir le liquide de la chambre droite jusqu'à ce que l'équilibre soit atteint. L'expression de la force, F_{piston} est tout simplement :

$$F_{piston} = A_2 P_m \quad (II.53)$$

Où A_2 est la surface du piston. Souvent, la surface de section transversale de la tige de raccordement à l'intérieur du cylindre est importante, et ensuite la surface du piston doit être considérée comme étant la zone moyenne sur les deux côtés du piston. Il est possible qu'une petite fuite coule à travers le piston, et apparaît sous la forme du trajet d'écoulement séparé comme indiqué la figure II.5. Ceci souvent est délibérément introduit par des créateurs pour améliorer la stabilité du système globale. Ce flux est considéré comme étant proportionnelle à la chute de pression à travers le piston :

$$q_{leak} = k_{leak} P_m \quad (II.54)$$

Comme il n'y a pas de fuite externe, les débits entrant et sortant du vérin sont égaux, et peut être égalisé au volume déplacé selon le mouvement de piston et la fuite à travers cela, ainsi :

$$q_1' = q_2' = A_2 \dot{y} + q_{leak} \quad (II.55)$$

Ces flux sont réglés par une servovalve, tel que décrit dans la section précédente, par conséquent le signal électrique d'entrée à la servovalve dirige le mouvement du piston.

Souvent, les longueurs de flexibles relient l'actionneur à la servovalve et le fluide hydraulique circule entre eux, la vitesse d'écoulement est diminuée en raison de la compression du fluide. Cette diminution est proportionnelle au volume dans lequel le liquide circule et le taux de variation de pression qui donne.

$$\begin{aligned} q_1' &= q_1 - \frac{V_1}{\beta} \dot{p}_1 \\ q_2 &= q_2' - \frac{V_2}{\beta} \dot{p}_2 \end{aligned} \quad (\text{II.56})$$

Où V_1 et V_2 sont les volumes de fluide sur les côtés respectifs du piston, notamment que, dans les tuyaux de raccordement. Le module de compression β_0 , d'un fluide est une mesure de sa compressibilité et pour un fluide hydraulique typique est pris pour être 17.10^8 Nm^{-2} . Le module de compression effective β_0 utilisé dans l'équation II.56, inclut des effets supplémentaires comme la flexibilité de mur de tuyau, la présence d'air emprisonné dans le fluide et les variations de température du fluide, et est par conséquent inférieur que β_0 . La dilatation des parois de la cuve renfermant peut réduire le module de compressibilité par 20%, pour cylindres avec parois en aluminium minces, et peut être jusqu'à 50% pour les tuyaux flexibles, qui sont souvent utilisés dans les applications robotiques. La présence d'air a un effet plus remarquable sur le module de masse car les gaz sont de l'ordre de 100 fois plus compressible que des liquides. Typiquement, si le fluide contient 0,1% d'air en volume, alors le module de masse est réduit de 10%, avec 1% d'air piégé cela augmente à une réduction de 50%. La réduction de la température du fluide est indiquée dans le tableau II.2

Tableau II.2 : Variation de module de compressibilité avec la température de fluide

Température (C°)	Module de compressibilité ($\beta \cdot 10^8 \text{ N m}^{-2}$)
-50	22.0
0	19.0
50	15.5
10	12.5
150	9.5

Puisque tous ces effets se manifestent comme des changements dans le module de compression effective, Il est facile vérifier comment la performance est dégradée dans telles conditions de fonctionnement extrêmes.

Les équations II.52 et II.54 dans II.56 peuvent alors être combinées pour donner :

$$q_1 = q_2 = A_2 \dot{y} + k_{leak} p_m + \frac{V_t}{4\beta} \dot{p}_m \quad (\text{II.57})$$

Où V_t est le volume total de liquide dans le piston et les tuyaux de liaison, et sont utilisés pour représenter les valeurs moyennes de V_1 et V_2 . Cette expression montre que le flux sortant de la servovalve forme trois éléments constitutifs; le flux qui fournit le mouvement du vérin, le débit de fuite et le débit dus à la compressibilité liquide. Égaliser ceci à l'Équation II.51, l'expression de débit à travers la servovalve, et réarranger les rendements.

$$\dot{P}_m = \frac{4\beta}{V_t} \left[\frac{C_d \pi d_1}{\sqrt{\rho}} x \sqrt{P_s - \text{sgn}(x) P_m} - A_2 \dot{y} - k_{leak} P_m \right] \quad (\text{II.58})$$

2.7.3 Vérin hydraulique linéaire agissant autour d'un pivot

Les vérins hydrauliques linéaires utilisés dans des applications robotiques, ils peuvent piloter soit des articulations prismatiques (mouvement linéaire) ou des articulations rotoïdes (rotation autour d'un pivot). L'analyse qui suit concerne les vérins linéaires agissant autour d'un pivot, comme représenté la figure II.6.

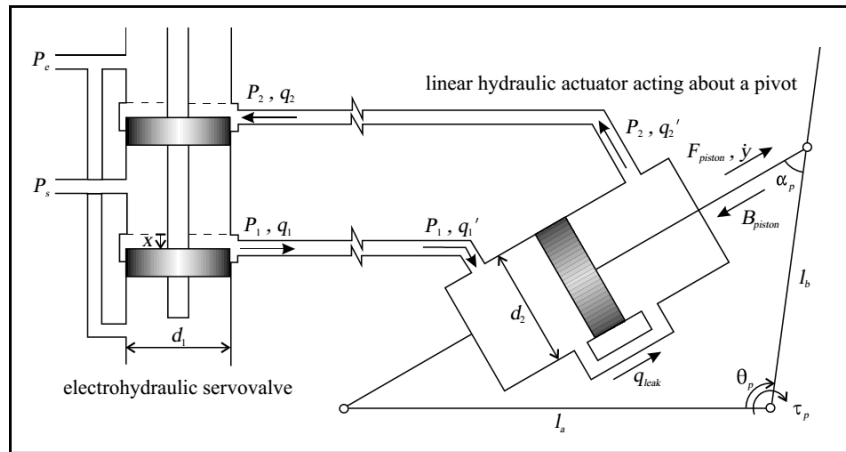


Figure II.6 : Vérin hydraulique linéaire agissant autour d'un pivot

En prend la longueur totale de piston y , les expressions suivantes peuvent être obtenues en appliquant les règles de cosinus et de sinus respectivement à la géométrie de l'actionneur représenté à la figure II.6:

$$y = \sqrt{l_a^2 + l_b^2 - 2l_a l_b \cos(\theta_p)} \quad (\text{II.59})$$

$$y = l_a \frac{\sin(\theta_p)}{\sin(\alpha_p)} \quad (\text{II.60})$$

Une expression pour \dot{y} peut-être ensuite dériver en dérivant l'équation II.59 par rapport au temps, ce qui donne:

$$\dot{y} = l_a l_b \sin(\theta_p) \sqrt{l_a^2 + l_b^2 - 2l_a l_b \cos(\theta_p)} \dot{\theta}_p \quad (\text{II.61})$$

Cette expression associe la vitesse de piston à l'angle du piston et à la vitesse angulaire, pour utiliser dans l'équation II.58. Le couple τ_p autour de l'articulation est en relation avec la force produite par le piston et les frottements présente dans le vérin B_{piston} par :

$$\tau_p = (F_{piston} - B_{piston}) \sin(\alpha_p) l_b \quad (\text{II.62})$$

Les équations II.53, II.59 et II.60, peut être récrit comme :

$$\tau_p = (A_2 P_m - B_{piston}) J_p(\theta_p) \quad (\text{II.63})$$

Ou :

$$J_p(\theta_p) = l_a l_b \sin(\theta_p) \sqrt{l_a^2 + l_b^2 - 2l_a l_b \cos(\theta_p)} \quad (\text{II.64})$$

L'angle de l'articulation du manipulateur θ , comme défini selon les conventions Denavit-Hartenberg modifiées, ne peut pas coïncider avec l'angle fait par le piston et les pivots θ_p . Aussi, en raison de l'arrangement physique de l'actionneur, l'angle de l'articulation peut être dans le sens opposé à θ_p par conséquent, l'équation II.65 est utilisée pour transformer d'un angle par rapport à l'autre:

$$\theta = k_{\theta_{dir}} \theta_p + \theta_{poffset} \quad (\text{II.65})$$

Où $k_{\theta_{dir}}$ est défini pour être +1 ou -1 selon le sens relatif des angles, et $\theta_{poffset}$ est déterminée à partir de la configuration physique de l'actionneur et le link pour aligner les deux angles. Puisque le couple d'articulation est défini dans la même direction que l'angle d'articulation, la relation suivante s'applique aussi :

$$\tau = k_{\theta_{dir}} \tau_p \quad (\text{II.66})$$

Et donc :

$$\tau = k_{\theta_{dir}} (A_2 P_m - B_{piston}) J_p(\theta_p) \quad (\text{II.67})$$

Les frottements présentent dans le vérin, B_{piston} sont modélisés possédant deux composants les frottements visqueux et le frottement de Coulomb. Les frottements visqueux sont proportionnels à la vitesse du piston, et le frottement de Coulomb est constant, sauf pour un signe de dépendance de la vitesse de piston [Elhami et Brookfield, 97]. Donc :

$$B_{piston} = c_p \operatorname{sgn}(\dot{y}) + v_p \dot{y} \quad (\text{II.68})$$

Les frottements sont également présents dans le joint lui-même et se composent de nouveau de deux les frottements visqueux et les frottements de Coulomb. Par conséquent:

$$B_{joint} = c_j \operatorname{sgn}(\dot{\theta}) + v_j \dot{\theta} \quad (\text{II.69})$$

Les frottements de l'articulation sont achevés dans le modèle dynamique complet du robot.

Pour voir les expressions non linéaires qui modèlent un vérin hydraulique, pendant l'état d'équilibre, un faible débit à travers la servovalve est maintenu dû à l'écoulement de fuite à travers le piston de vérin. Mettant les termes dépendant du dérivé et de la vitesse, Incluant les frottements, dans les équations II.50, II.58 et II.63 à zéro donne :

$$\frac{C_d \pi d_1}{\sqrt{\rho}} k_i i \sqrt{P_s - \operatorname{sgn}(k_i i) P_m} - k_{leak} P_m = 0 \quad (\text{II.70})$$

$$\tau_p = A_2 P_m J_p(\theta_p) \quad (\text{II.71})$$

Ces relations donnent la dépendance non linéaire, que le couple autour de l'articulation est sur les courants d'entrée i et l'angle de piston θ_p , dans des conditions statiques. Seules les valeurs positives de i seront considérées ici, puisque les résultats sont symétriques par rapport à $i=0$ Am.

2.7.4 Modèle hydraulique du robot

Le modèle dynamique d'un manipulateur avec un nombre arbitraire d'articulations est représenté par la forme suivante [Craig, 89] :

$$\tau = M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) + B(\dot{q}) + {}^c J^T(q) * {}^c F \quad (\text{II.72})$$

$\tau(t)$ est le vecteur $n \times 1$ des couples appliqués par le bras, $M(q)$ est la matrice d'inertie $n \times n$ bien déterminée positive et symétrique, $V(\theta, \dot{\theta})\dot{\theta}$ est le vecteur des forces centrifuges et de Coriolis, $G(q)$ est le vecteur $n \times 1$ des couples de la gravité, B est le vecteur de frottement agissant à chaque joint, ${}^c J$ est le Jacobien du manipulateur, ${}^c F$ est le vecteur des forces et des couples à l'effecteur, $q(t) \in R^n$: variable de joint, q et \dot{q} sont respectivement les vecteurs $n \times 1$ de la position et de la vitesse communes.

Ces derniers étant calculés en utilisant les équations dynamiques récursives conventionnelles de Newton-Euler. Le vecteur des couples et le vecteur de frottement B pour les manipulateurs hydrauliques sont simplement obtenus en utilisant les équations II.67 et II.69 respectivement pour chaque articulation du robot, cela peut être appliqué à des manipulateurs avec un nombre arbitraire de joints. Cependant que chaque link à une masse rectangulaire homogène, avec centre de gravité étant au centre de ses axes principaux. On peut écrire de l'équation II.72 comme suit :

$$\ddot{q} = M(q)^{-1} \left[\tau - V(q, \dot{q}) - G(q) - B(\dot{q}) - {}^c J^T(q) * {}^c F \right] \quad (\text{II.73})$$

Pour simuler le mouvement du manipulateur tout simplement on intègre l'équation II.73 deux fois par rapport au temps. Même chose la pression de charge P_m , pour chaque actionneur il est nécessaire pour déterminer τ est obtenue en intégrant l'équation II.58 pour chaque articulation du robot.

2.8 Conclusion

Dans ce chapitre nous avons exposé les différents modèles utilisés pour décrire les mouvements des articulations d'un bras manipulateur, dans le but est de fournir les équations mathématiques qui permettent de programmer, commander et exécuter les tâches désirées. Ceci est réalisé en fonction du modèle géométrique, cinématique et dynamique, ainsi que le modèle hydraulique afin de pouvoir faire des études de simulations concernant quelques commandes appliquées à ces types de robots manipulateurs. Dans le chapitre suivant, fera l'objet d'une étude sur la simulation des différentes commandes appliquées au robot.

CHAPITRE 3

Commandes classiques des robots manipulateurs

3.1 Introduction

Le domaine de la commande des robots manipulateur est très vaste, il est encore en pleine évolution. Plusieurs algorithmes de commande ont été développés pour satisfaire les objectifs exigés par les cahiers des charges [Spong, 89], [Samson, 91], [Lewis, 93], [Zodiac, 96]. Un système non linéaire (telle que les robots) peut être commandé de plusieurs manières pour le faire se comporter comme un système linéaire dans sa performance globale. La diversité des comportements des phénomènes dynamiques qui peuvent être observés pour cette classe de systèmes explique la grande variété d'approches développées. Nous constatons cependant, une certaine réticence à adopter des méthodes de commande non linéaire, souvent jugées difficiles à comprendre, compliquées à mettre en œuvre et dont l'analyse systématique des performances se révèle complexe [Benalia. A, 2003] [Benalia. A, 2004]. Confronté à un tel défi, il est clair que nous ne puissions pas nous attendre à ce qu'une procédure particulière s'applique à tous les systèmes non linéaires [Hassan K. Khali. 2002]. Il est également peu probable que la conception d'un contrôleur puisse être basée sur un outil particulier. D'autre part, des techniques d'analyse et de commande des systèmes linéaires, dont le principe de fonctionnement est bien maîtrisé depuis plusieurs années, sont efficacement utilisées pour traiter des problèmes concrets de l'automatique. Ainsi, il est intéressant de prédire si les outils de commandes linéaires pourraient être utilisés pour traiter des problèmes non linéaires. Nous trouvons des méthodes locales basées sur des approximations du premier ordre mais dont la validité et les performances ne seront garanties que localement. D'autres méthodes sont basées sur une transformation globale du système initial vers un système linéaire. Ces transformations peuvent concerner l'état et/ou la commande, et peuvent être dynamiques ou statiques. Ces différentes transformations définissent des notions d'équivalence entre les systèmes. Plusieurs travaux de recherche visaient à définir des classes d'équivalence par bouclage afin d'établir des formes normales pour les systèmes non linéaires [Jean P. C, 96] [Fossard. A. J et al, 93].

3.2 Quelques stratégies de commande

L'utilisation de la commande par le découplage non linéaire constitue une bonne approche dans ce sens. Un tel contrôle est également connu comme le contrôle dynamique ou calcul de couple parce qu'il est basé sur l'utilisation du modèle dynamique du robot [SPONG, 89]. Pour avoir une idée sur les procédures existantes on va présenter dans cette partie quelques méthodes de commande conventionnelles (dites basiques), et à côté de ces commandes conventionnelles,

on a introduit la commande intelligente ou la commande basée sur l'intelligence artificielle qui est l'objectif du prochain chapitre.

3.3 Commande par les méthodes classique

La résolution du problème de la commande des robots manipulateurs nécessite la détermination d'un ensemble d'entrées articulaires (les couples τ) qui résulte par le suivi de l'organe effecteur d'une trajectoire désirée, spécifiée typiquement par des séquences de positions et de vecteurs d'orientation de l'organe effecteur x ou par une trajectoire continue. Plusieurs types de commandes ont été étudiées pour les robots manipulateurs, la méthode la plus simple et qui reste toujours employée pour les manipulateurs industriels est la commande articulaire indépendante ou chaque articulation du manipulateur est commandée comme un système à une seule entrée et une seule sortie (S.I.S.O) [Arato.S, 95].

La stratégie de cette commande est que chaque actionneur d'une articulation est contrôlé indépendamment : tous les effets de couplage entre les deux articulations sont ignorés ou traités comme des perturbations. Cette commande a donné des résultats satisfaisants pour les simples déplacements, mais n'est pas convenable pour les déplacements rapides et ceux avec une large variation.

La méthode de la commande articulaire indépendante peut être considérée comme une forme simplifiée de la méthode du couple calculé qui est la technique commune la plus utilisée pour les robots manipulateurs, la plupart des méthodes de commandes des manipulateurs peuvent être considérés comme des cas spéciaux de cette technique. La méthode du calcul de couple en elle-même est une application de la technique de linéarisation du (Feed-Back) pour les systèmes non linéaires.

3.4 Commande par calcul de couple

Au fil des ans il a été proposé de nombreuses sortes de systèmes de commande de robot. Comme il arrive, la plupart d'entre eux peuvent être considérés comme des cas particuliers de la classe des contrôleurs type calcul de couple. Cette technique est aussi connue sous les noms de commande dynamique. Lorsque l'application exige des évolutions rapides avec des contraintes dynamiques, la commande doit prendre en compte les forces d'interaction. Ce type de technique considère l'ensemble des articulations et, pour les découpler, utilise la théorie du découplage non linéaire. Cette théorie utilise le modèle dynamique du robot pour le

calcul de la loi de commande, ce qui conduit à des lois de commande centralisées non linéaires. Des signaux d'anticipation peuvent être utilisés pour minimiser des effets non linéaires [Er, 93]. Calcul de couple (CT) en même temps est une application particulière de la rétroaction (feedback) linéarisation des systèmes non linéaires, qui a gagné en popularité dans la théorie des systèmes modernes. [Hunt et al. 83], [Gilbert and Ha 84]. En fait, une façon de classer les systèmes de contrôle du robot est de les diviser comme « calcul de couple » ou « non calcul de couple ». Calcul de couple comme la commande apparaît dans la commande robuste, la commande adaptative, commande par apprentissage, et ainsi de suite. La commande par calcul de couple nous permet de tirer facilement des contrôleurs des robots très efficaces, tout en fournissant un cadre pour rassembler le contrôle commun indépendant classique et quelques techniques de conception modernes.

La stratégie de la commande par calcul de couple peut être considérée comme un contrôleur en deux parties, une section étant basée modèle alors que l'autre étant la portion de la loi de servocommande (Fig. III-1) [Arato.S, 95]. La figure III-1 montrée La stratégie de la commande par calcul de couple, montrant les deux boucles internes et externes.

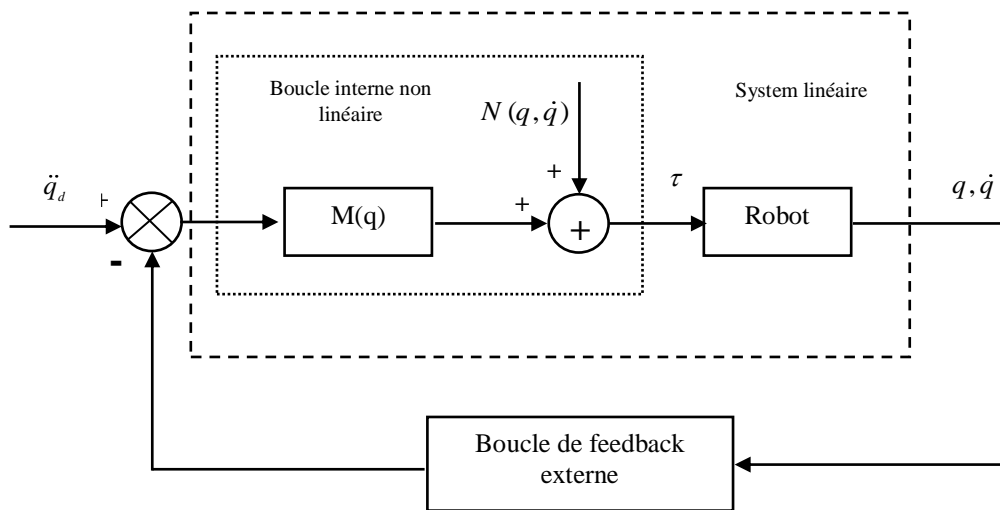


Figure III.1 : Système de commande par calcul de couple, montrant la boucle interne et externe.

3.4.1 Dérivation de la boucle interne (Feed forward)

La dynamique de bras de robot est :

$$\tau = M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d \quad (\text{III.1})$$

Ou simplement :

$$\tau = M(q)\ddot{q} + N(q, \dot{q})\dot{q} + \tau_d \quad (\text{III.2})$$

$N(\theta, \dot{\theta})$: Les limites non linéaires représentées par :

$$N(q, \dot{q}) = V(q, \dot{q})\dot{q} + F_v(\dot{q}) + F_d(\dot{q}) + G(q)$$

$$F(\dot{q}) = F_v(\dot{q}) + F_d(\dot{q})$$

$F_v(\dot{q})$: est le vecteur $n \times 1$ des coefficients de frottement visqueux et $F_d(\dot{q})$: est le vecteur $n \times 1$ des coefficients de frottement dynamique, ou n correspond au nombre de degrés de liberté du robot.

$q(t) \in R^n$: Variable de joint, q et \dot{q} sont respectivement les vecteurs $n \times 1$ de la position et de la vitesse communes.

$\tau(t)$: est le vecteur $n \times 1$ des couples appliqués par le bras. Si cette équation comprend la dynamique de l'actionneur à moteur, alors $\tau(t)$ est une tension d'entrée.

$M(q)$ est la matrice d'inertie $n \times n$ bien déterminée positive et symétrique, $V(\theta, \dot{\theta})\dot{\theta}$ est le vecteur des forces centrifuges et de Coriolis, $G(q)$ est le vecteur $n \times 1$ des couples de la gravité, Des perturbations inconnues bornées (incluant les dynamiques non structurées, et non modélisées) représentées par τ_d .

Pour assurer le suivi de trajectoire par la variable joint, nous choisissons une commande qui stabilise le system, de sorte que l'erreur $e(t)$ tende vers zéro.

$$e(t) = q_d(t) - q(t) \quad (\text{III.3})$$

Pour démontrer l'influence de l'entrée $\tau(t)$ sur l'erreur de poursuite, nous dérivons deux fois pour obtenir :

$$\dot{e}(t) = \dot{q}_d(t) - \dot{q}(t) : \text{Vecteur de l'erreur de vitesse.}$$

$$\ddot{e}(t) = \ddot{q}_d(t) - \ddot{q}(t) : \text{Vecteur de l'erreur d'accélération.}$$

Résoudre maintenant \ddot{q} dans (III.2) et son remplacement dans les derniers rendements de l'équation :

$$\ddot{e} = \ddot{q}_d + M^{-1}(N + \tau_d - \tau) \quad (\text{III.4})$$

Définir la fonction d'entrée de commande :

$$u = \ddot{q}_d + M^{-1}(N - \tau) \quad (\text{III.5})$$

Et la fonction de perturbation :

$$w = M^{-1}\tau_d \quad (\text{III.6})$$

Nous pouvons définir un état $x(t) \in R^{2n}$ par:

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (\text{III.7})$$

Et écrire la dynamique d'erreur de poursuite par :

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} w \quad (\text{III.8})$$

Il s'agit d'un système d'erreur linéaire sous forme canonique de Brunovsky constitué de n paires de doubles intégrateurs $\frac{1}{s^2}$ un par joint. Elle est entraînée par l'entrée de commande $u(t)$ et la perturbation $w(t)$.

La transformation de linéarisation de feedback (III.5) peut être inversée pour donner :

$$\tau = M(\ddot{q}_d - u) + N \quad (\text{III.9})$$

Nous appelons cela la loi de commande par calcul de couple. L'importance de ces manipulations est le suivant. Il y a eu aucune transformation espace-état en passant de (III.1) à (III.8).

Par conséquent, si nous choisissons une commande $u(t)$ qui stabilise (III.8) de sorte que $e(t)$ tend vers zéro, puis l'entrée de commande non linéaire donnée par $\tau(t)$ (III.9) va provoquer la

poursuite de la trajectoire du bras de robot (III.1). En fait, on remplace (III.9) dans (III.2) ça donne :

$$M\ddot{q} + N + \tau_d = M(\ddot{q}_d - u + N)$$

Ou bien :

$$\ddot{e} = u + M^{-1}\tau_d \quad (III.10)$$

Ce qui est exactement (III.8).

La stabilisation de (III.8) n'est pas difficile. En fait, la transformation non linéaire (III.5) a converti un problème de conception des contrôles non linéaires complexes en un problème de conception simple pour un système linéaire constitué de n sous-systèmes découplés, obéissant chacun les lois de Newton. Le système de contrôle résultant apparaît dans la figure III.1. Il est important de noter qu'il est constitué d'une boucle non linéaire interne plus un signal de commande externe $u(t)$. Nous verrons plusieurs façons pour choisir $u(t)$. Comme $u(t)$ dépend de $q(t)$ et $\dot{q}(t)$, la boucle externe sera une boucle de rétroaction (feedback).

Il est important de savoir que le calcul de couple dépend de l'inversion de la dynamique du robot, et est d'ailleurs parfois appelée commande de la dynamique inverse. En fait, (III.9) montre que $\tau(t)$ est calculé en substituant $\ddot{q}_d - u$ pour \ddot{q} dans (III.2), c'est en résolvant l'inverse robot dynamique problème. Une bonne façon de calculer le couple $\tau(t)$ est d'utiliser la formulation dynamique de Newton-Euler de l'inverse efficace [Craig, 85] avec $\ddot{q}_d - u$ à la place de $\ddot{q}(t)$.

PD Conception de la boucle externe :

3.4.2 Conception de la boucle externe

Une façon de sélectionner le signal de commande auxiliaire $u(t)$ est que la proportionnelle et dérivée (PD) rétroaction (feedback).

$$u = -k_v \dot{e} - k_p e \quad (III.11)$$

Ensuite, l'entrée globale de bras de robot devient :

$$\tau = M(q)(\ddot{q}_d + k_v \dot{e} + k_p e) + N(q, \dot{q}) \quad (III.12)$$

La dynamique d'erreur en boucle fermée est :

$$\ddot{e} + k_v \dot{e} + k_p e = w \quad (\text{III.13})$$

Ou sous forme d'espace-état :

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} w \quad (\text{III.14})$$

Choix des gains PD : Il est habituel de prendre la matrices de gain diagonales $n \times n$ de telle sorte que :

$$k_v = \text{diag}\{k_{vi}\}, \quad k_p = \text{diag}\{k_{pi}\} \quad (\text{III.15})$$

Il est important de noter qu'en choisissant la diagonale de matrices de gain de PD dans le contrôle détaché au niveau de boucle extérieure, il n'aboutit pas à une stratégie de contrôle commun indépendant. C'est parce que la multiplication par le $M(q)$ et l'addition des limites non linéaires (feed forward) $N(q, \dot{q})$ dans la boucle intérieure perturbé le signal $u(t)$ Parmi tous les joints (articulations). Ainsi, les informations sur toutes les positions $q(t)$ et des vitesses $\dot{q}(t)$ est généralement nécessaire pour calculer le contrôle $\tau(t)$ pour n'importe quel joint (articulation) donné.

3.5 Calcul de couple avec PD

Le schéma de commande par calcul de couple avec PD donnée par la figure III.2. Le modèle dynamique du robot donné par:

$$\tau = M(q) \ddot{q} + V(q, \dot{q}) \dot{q} + G(q) + F(\dot{q}) + \tau_d \quad (\text{III.16})$$

Il s'agit de la forme standard :

$$\begin{aligned}
 & \begin{bmatrix} (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2 \cos \theta_2 & m_2a_2^2 + m_2a_1a_2 \cos \theta_2 \\ m_2a_2^2 + m_2a_1a_2 \cos \theta_2 & m_2a_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \\
 & \begin{bmatrix} -m_2a_1a_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \sin \theta_2 \\ m_2a_1a_2\dot{\theta}_1^2 \sin \theta_2 \end{bmatrix} + \begin{bmatrix} (m_1 + m_2)ga_1 \cos \theta_1 + m_2ga_2 \cos(\theta_1 + \theta_2) \\ m_2ga_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad \text{(III.17)} \\
 & = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}
 \end{aligned}$$

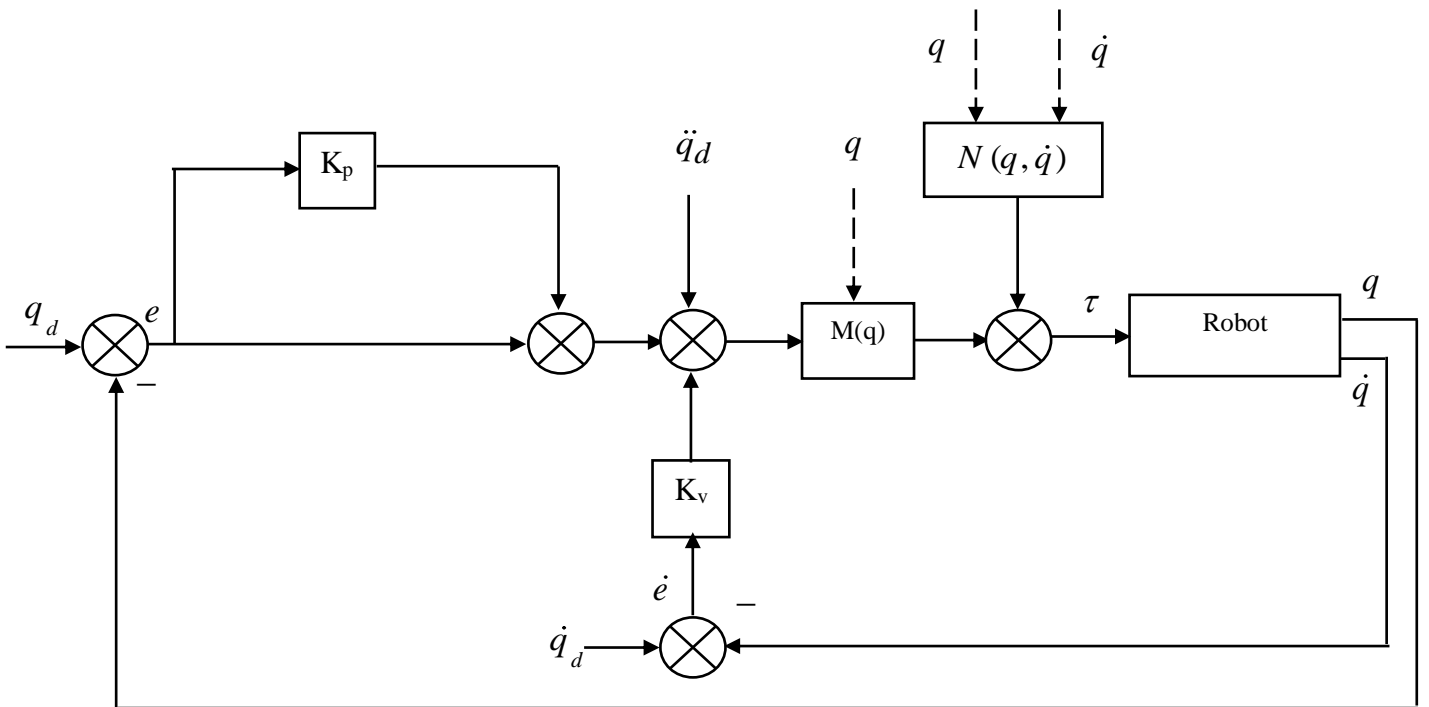


Figure III.2 : Commande par calcul de couple avec PD

La loi de commande par calcul de couple (PD) est donnée par:

$$\tau = M(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + N(q, \dot{q}) \quad \text{(III.18)}$$

Une façon de sélectionner le signal de commande auxiliaire $u(t)$ comme c'est le proportionnelle dérivative (PD).

La fonction d'entrée de commande :

$$\mu = -K_v \dot{e} - K_p e \quad \text{(III.19)}$$

Les limites non linéaires étant représentées par :

$$N(q, \dot{q}) = V(q, \dot{q})\dot{q} + F_v(\dot{q}) + F_d(\dot{q}) + G(q)$$

En choisissant les matrices de gain K_v et K_p comme des matrices diagonales avec des valeurs positives le long de la diagonale, cette erreur du système peut être rendue asymptotiquement stable. Il est important de noter que malgré la sélection des matrices diagonales de gain, il en résulte un découplage de la commande au niveau de la boucle externe mais ceci n'implique pas une stratégie de commande articulaire découplée. Les critères de choix des paramètres du contrôleur tel que les gains sont détaillés par Frank L [Frank L, 2004].

Parce que la multiplication par la matrice d'inertie et l'addition de termes non linéaires dans la loi basée modèle laisse la loi de commande u affecter toutes les articulations. Pour calculer l'entrée couple de n'importe quelle articulation, les positions et vitesses des autres articulations sont nécessaires.

Nous présenterons dans cette section les résultats de simulation de notre système en utilisant MATLAB, on prend les masses de liens de 1 kg et leurs longueurs de 1 m, les gains PD sont habituellement choisis pour l'amortissement critique $\xi = 1$

$$k_p = 100, \quad k_v = 20$$

$$\text{Erreur de poursuite : } e(t) = q_d(t) - q(t)$$

Dans cette simulation nous utiliserons une trajectoire de référence donnée par :

$$\begin{cases} \theta_1 = \sin(t) \\ \theta_2 = \cos(t) \end{cases}$$

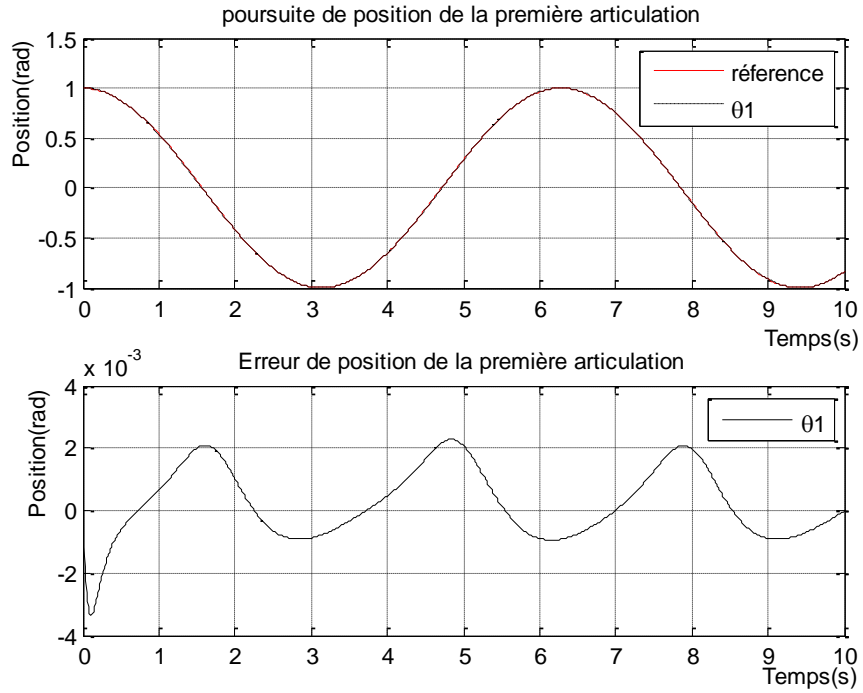


Figure III.3: Erreur et poursuite de position de la première articulation

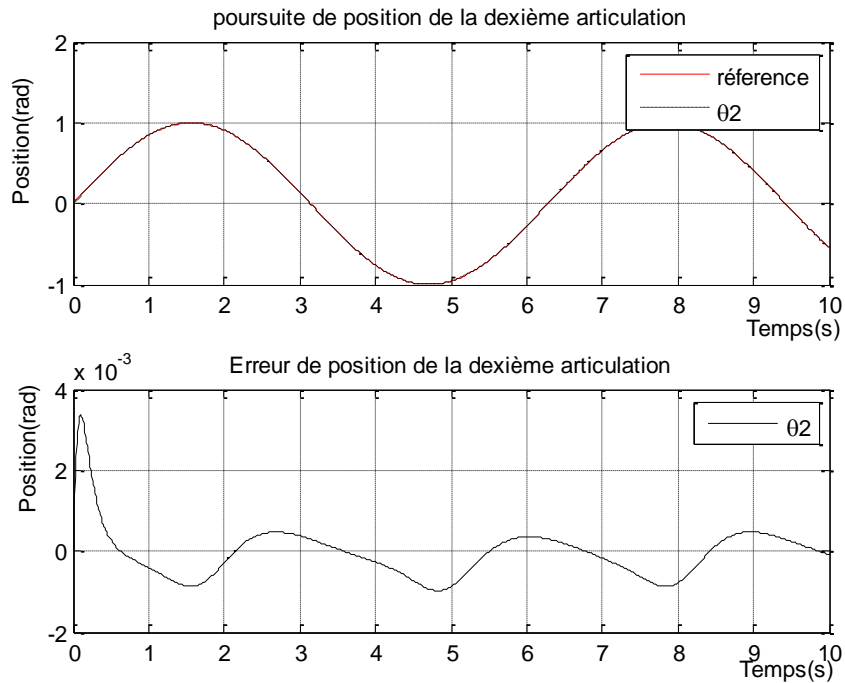


Figure III.4: Erreur et poursuite de position de la deuxième articulation

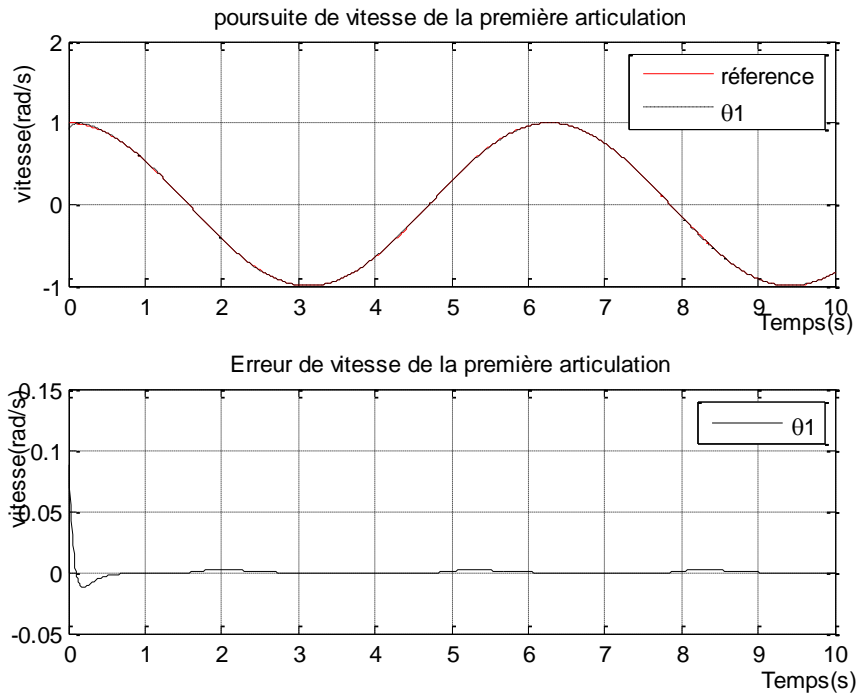


Figure III.5: Erreur et poursuite de vitesse de la première articulation

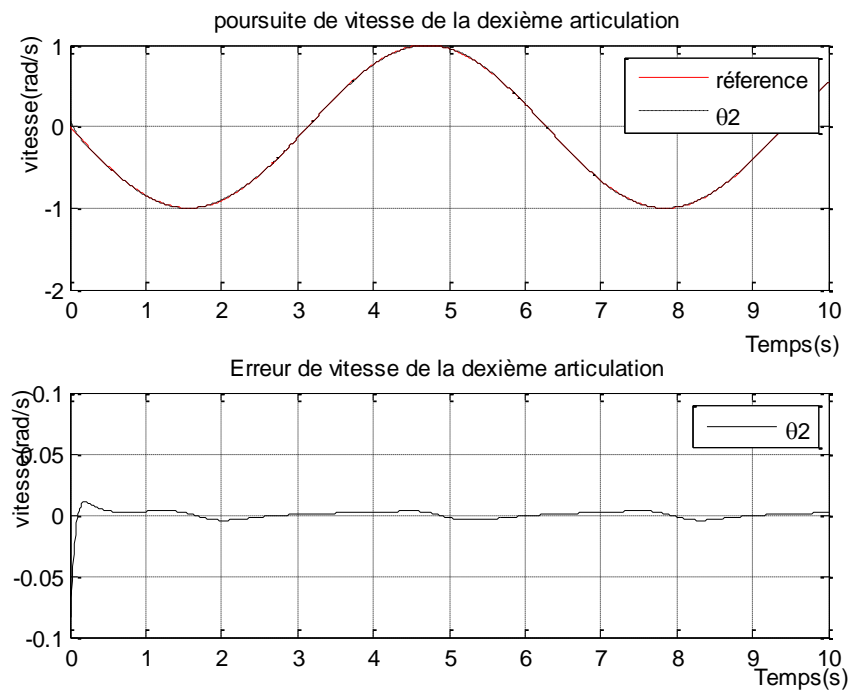


Figure III.6 : Erreur et poursuite de vitesse de la deuxième articulation

D'après les résultats de simulation durant 10s on remarque que dès la première seconde la position suit la référence avec une erreur peu différente de 3.10^{-3} et la vitesse de déplacement suit celle de référence avec une erreur peu différente de 2.10^{-3} . Ainsi la commande par calcul de couple s'avère très performante dans ce cas (elle donne de bons résultats dans le cas où les paramètres du bras sont exactement connus).

3.6 Calcul de couple avec PID

Nous venons de voir que la commande de calcul de couple avec PD est très efficace si tous les paramètres de bras sont connus et qu'il n'y a aucune perturbation τ_d . Cependant, de la théorie de contrôle classique nous savons qu'en présence de perturbations constantes, le contrôle avec PD donne une erreur différente de zéro en régime permanent. Par conséquent, afin d'éliminer cette erreur on inclut un intégrateur dans la boucle directe ceci peut être réalisé en utilisant un calcul de couple avec PID.

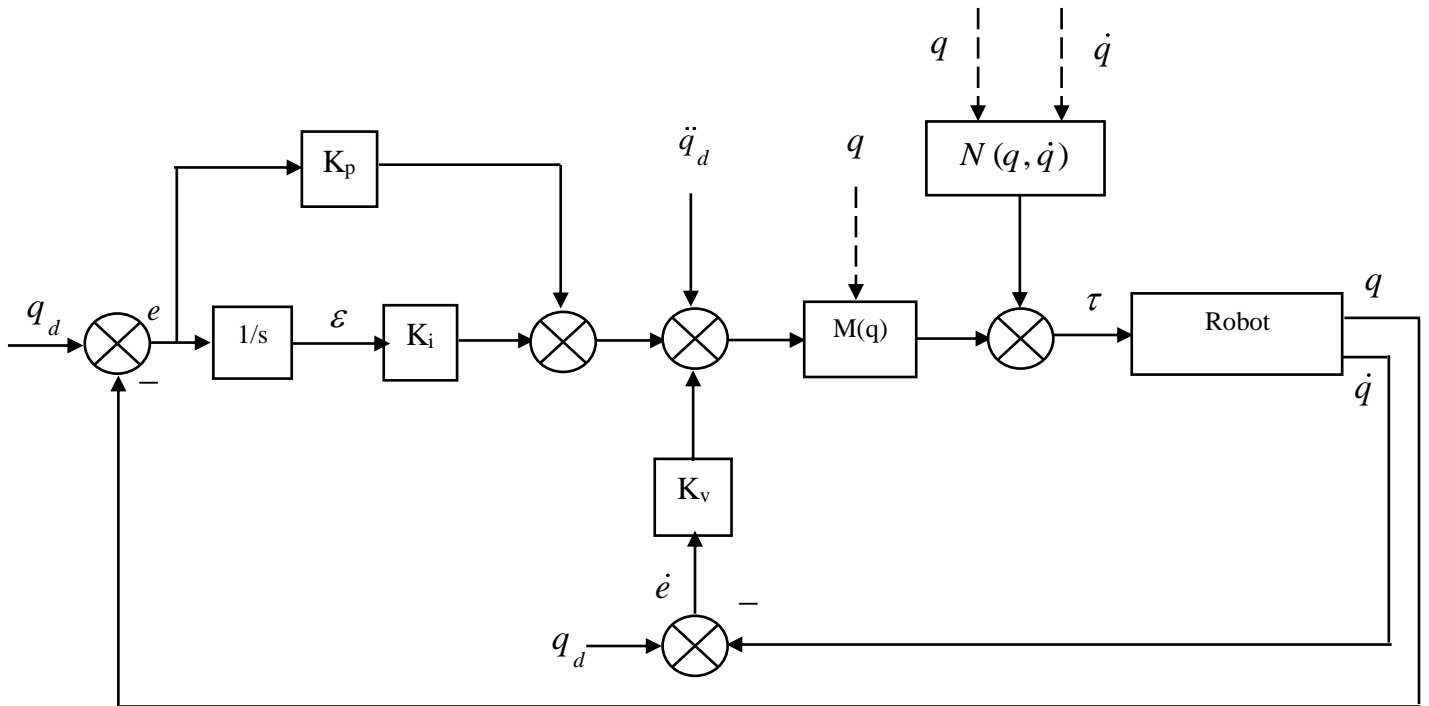


Figure III.7: Schéma de commande par calcul de couple avec PID

Avec :

$$\dot{\varepsilon} = e \quad (\text{III.20})$$

$$u = -k_v \dot{\varepsilon} - k_p e - k_i \varepsilon \quad (\text{III.21})$$

Ce qui donne l'entrée de commande du bras :

$$\tau = M(q) + (\ddot{q}_d + k_v \dot{\varepsilon} + k_p e + k_i \varepsilon) + N(q, \dot{q}) \quad (\text{III.22})$$

Avec $\dot{\varepsilon}(t)$ l'intégrale de l'erreur de trajectoire $e(t)$. Ainsi la dynamique a été ajoutée à la boucle extérieure linéaire. Cette loi de contrôle est commodément décrite en définissant l'état comme : $x = [\varepsilon^T \ e^T \ \dot{\varepsilon}^T]^T \in R^{3n}$ Et augmentation de la dynamique d'erreur (III.8) avec un intégrateur, pour que :

$$\frac{d}{dt} \begin{bmatrix} \varepsilon \\ e \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon \\ e \\ \dot{\varepsilon} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} w \quad (\text{III.23})$$

Sélection des gains de contrôle diagonale [Frank L, 2004]:

$$k_v = \text{diag}\{k_{vi}\} \quad k_p = \text{diag}\{k_{pi}\} \quad k_i = \text{diag}\{k_{ii}\}$$

Avec : $k_{ii} < k_{vi} k_{pi}$

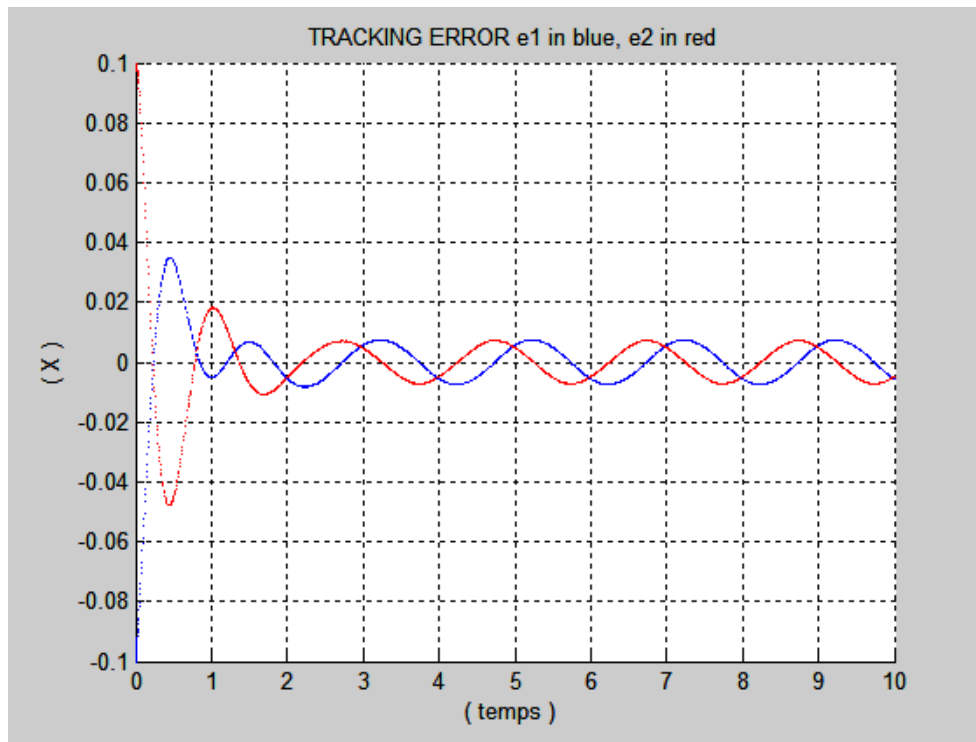


Figure III.8: Erreur de poursuite de position

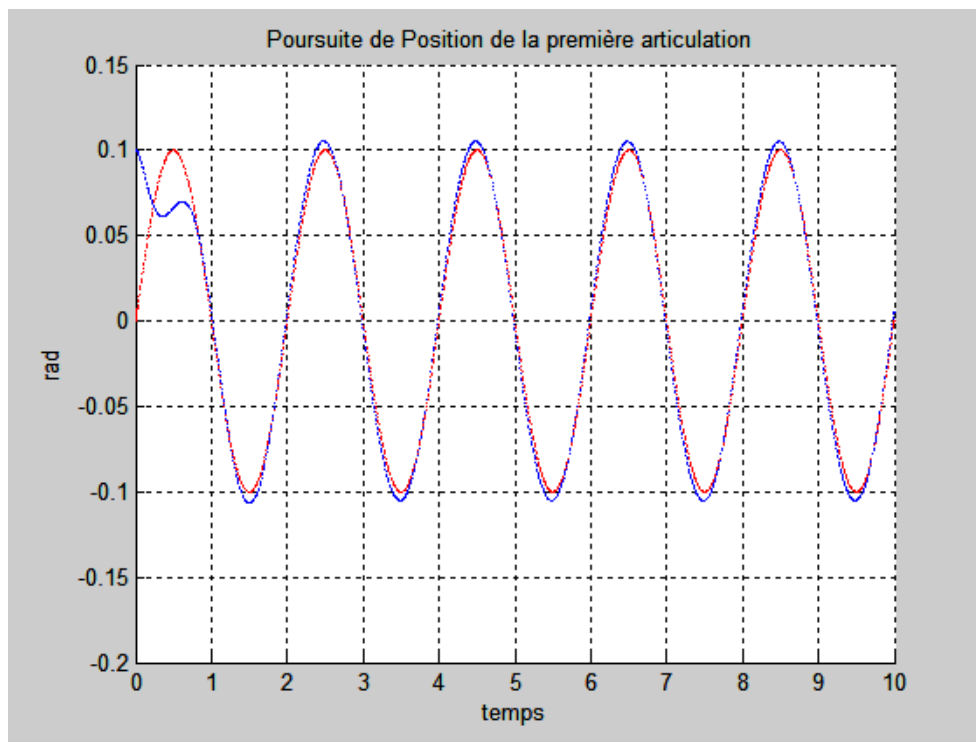


Figure III.9: Poursuite de position de la première articulation

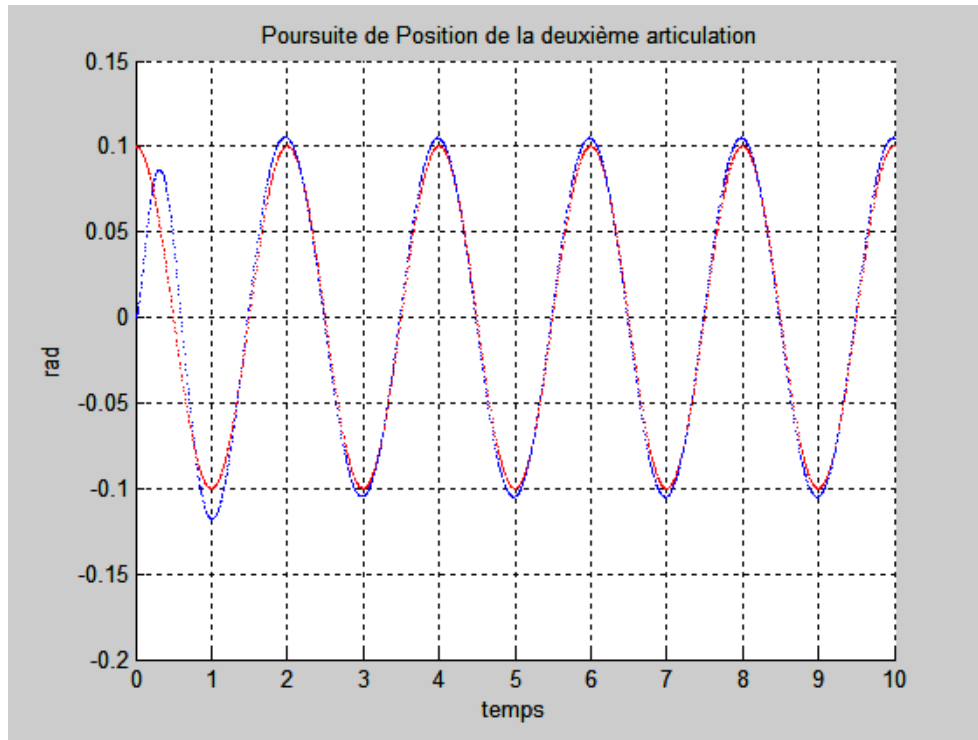


Figure III.10: Poursuite de position de la deuxième articulation

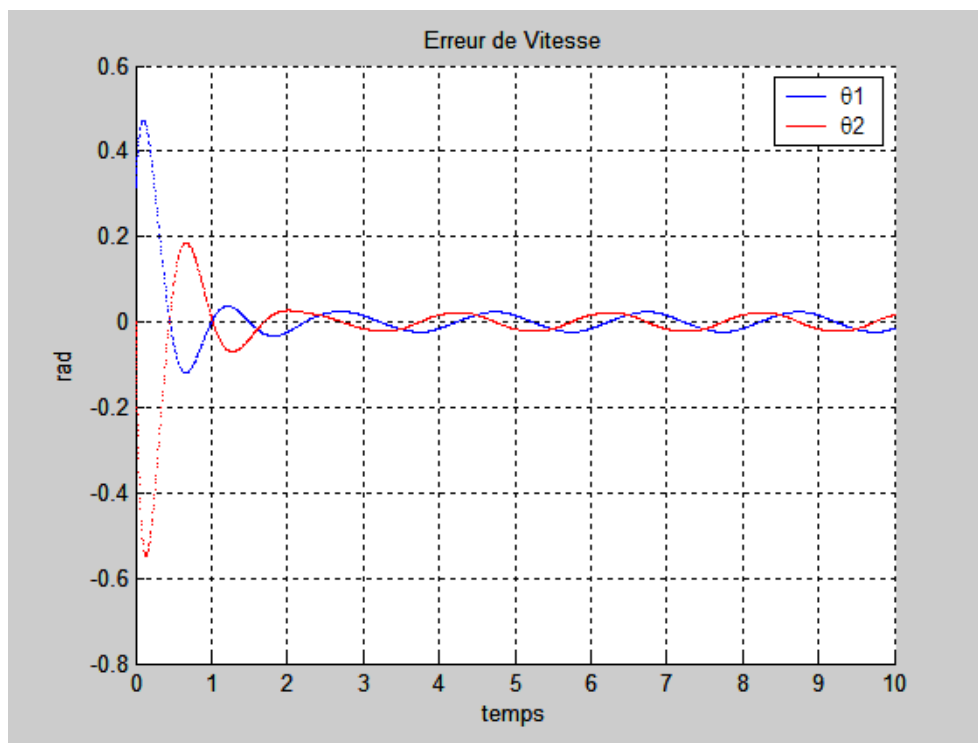


Figure III.11: Erreur de vitesse

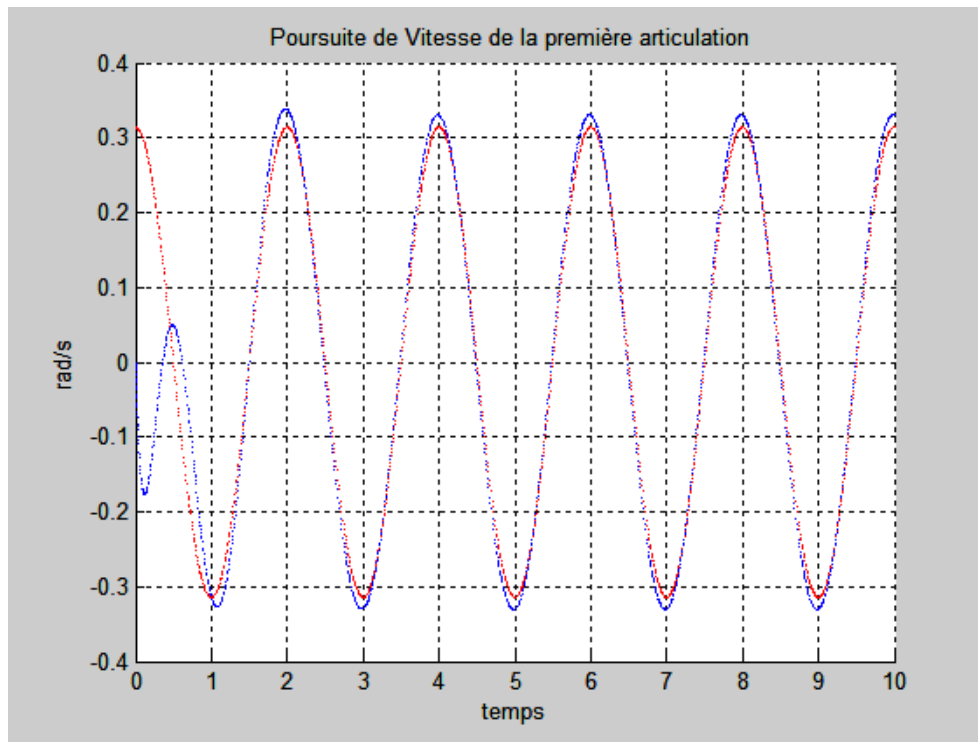


Figure III.12: Poursuite de vitesse de la première articulation

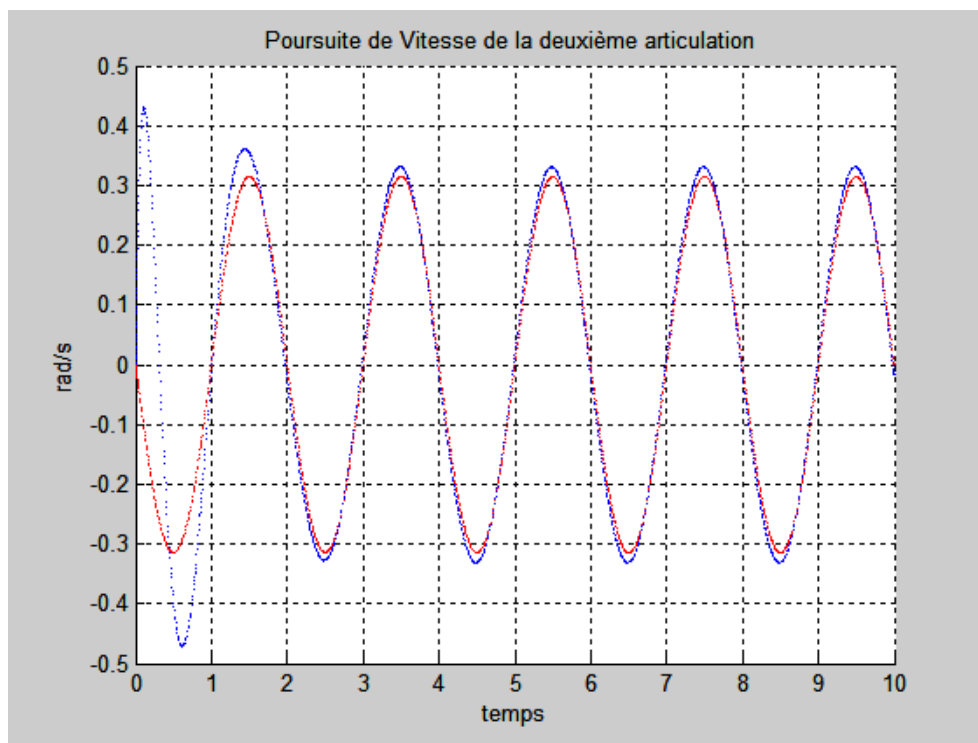


Figure III.13: Poursuite de vitesse de la deuxième articulation

Dès la première seconde la position suit la référence avec des oscillations très remarquables et une erreur peu différente de 0.02. La vitesse de déplacement suit celle de référence avec des oscillations plus importantes et avec une erreur peu différente de 0.05. Il est également intéressant de remarquer que la sortie ne suit pas bien la référence, les erreurs de poursuite (en position et vitesse) de la commande par PID sont un peu plus élevées, il montre aussi que les oscillations sont plus importantes que dans le cas de la commande par PD. Cela est dû à la nature de la commande par PID.

3.7 Commande adaptative

3.7.1 Introduction

Le modèle dynamique d'un robot ne pouvant être parfait, il était naturel que les chercheurs appliquent les théories de la commande adaptative pour estimer ou ajuster en ligne les valeurs de paramètres dynamiques utilisés dans la commande. Une définition de la commande adaptative est proposée par Y.D. Landau [LAN, 79]: "Un système adaptatif mesure un index de performance (IP) en utilisant les entrées, les d'états et les sorties d'un système ajustable. Cet index est comparé à un ensemble de valeurs fixées. A partir de cette comparaison, le mécanisme d'adaptation modifié les paramètres du système ajustable (en anglais, Parameter Adaptation Method: PAM) et/ou engendre une entrée auxiliaire (en anglais, Signal Synthesis Method: SSM) afin de maintenir la valeur de l'index proche des valeurs de référence. Néanmoins, nous préférons à cette définition celle de S. Tosunoglu et D. Tesar [TOS, 87]: "Un système de commande en boucle fermée est adaptatif si les gains sont choisis à l'aide de l'information, obtenue en ligne, sur les sorties et/ou les variables d'état du processus commandé." On peut à l'heure actuelle considérer que la commande adaptative des robots rigides a atteint une bonne maturité comme en témoigne les nombreuses méthodes publiées sur le sujet [Bayard 88], [Ortega 89]. Il s'en dégage cinq stratégies, significatives des progrès réalisés dans ce domaine :

1. Simplification du modèle dynamique [Dubowsky 79], [Takegaki 81a].
2. Application des techniques adaptatives développées pour les systèmes linéaires au cas des robots [Horowitz 80], [Hsia 86], [Nicosia 84].
3. Proposition d'une commande adaptative non linéaire asymptotiquement stable [Craig 86b].
4. Utilisation des propriétés de la passivité pour développer une loi de commande adaptative non linéaire asymptotiquement stable [Slotine 87], [Sadegh 87], [Landau 88], [Kelly 88].
5. Utilisation pour l'adaptation d'un modèle qui n'est pas fonction des accélérations articulaires tel que le modèle dynamique filtré [Middleton 88], [Li 89] ou le modèle énergétique [El Serafi 91a].

Le schéma général d'un système de commande adaptatif peut être le suivant :

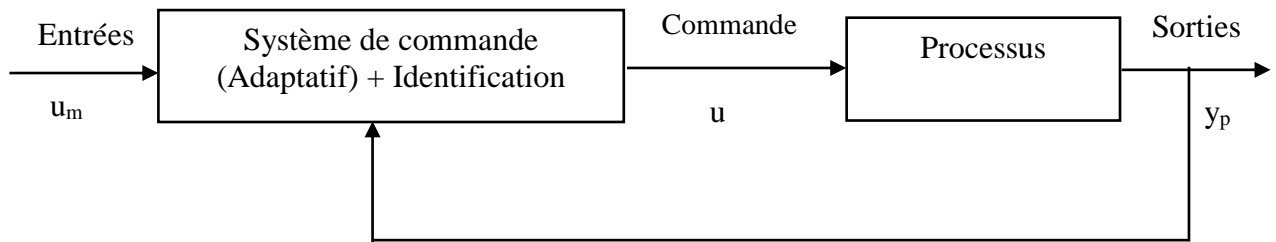


Figure III.14: Schéma général de la commande adaptatif

Les lois de commande proposées dans les deux premières stratégies ne sont valables que pour des mouvements lents et ne prennent pas en compte le modèle complet du robot. La loi de commande adaptative non linéaire de Craig dépend de deux hypothèses restrictives : la mesure des accélérations articulaires et l'existence d'une borne de l'inverse de la matrice d'inertie estimée. Les lois de commande des quatrième et cinquième stratégies n'ont pas besoin de mesures des accélérations articulaires et sont, au moins de point de vue théorique, les plus intéressantes. D'une façon générale il existe diverses commandes adaptatives. Dans le cadre de la commande des robots manipulateurs, on peut citer la commande adaptative directe et indirecte [Patricia, 87].

3.8 Commande adaptative des robots manipulateurs

Le problème de la conception de lois de commande adaptative pour des robots manipulateurs rigide afin assurent le suivi de trajectoire désiré à intéresser des chercheurs depuis plusieurs années. Le développement de contrôleurs adaptatifs efficaces représente une étape importante vers les applications robotiques à grande vitesse / précision, même dans une installation industrielle bien structurée, les robots peuvent faire face à une incertitude concernant les paramètres décrivant les propriétés dynamiques de la charge de portée (par exemple les moments d'inertie inconnus). Puisque ces paramètres sont difficiles de calculer ou mesurer, ils limitent le potentiel pour des robots pour manipuler avec précision des objets de taille et de poids considérable. Pour compenser cette incertitude paramétrique, beaucoup de chercheurs ont proposé des stratégies de commande adaptatives. Un avantage de l'approche adaptative est que l'exactitude des charges inconnues de transport d'un manipulateur s'améliore avec le temps.

L'utilisation des techniques adaptatives est motivée par l'ajustement automatique des paramètres du système quand ils sont inconnus et/ou ont des variations imprévisibles dans le temps.

Les contrôleurs adaptatifs sont formulés en séparant les paramètres constants inconnus des fonctions connues dans l'équation dynamique de robot.

Dans la suite, nous présentons la loi de commande adaptative par calcul de couple et la loi adaptative inertielle.

3.8.1 Commande adaptative par calcul de couple

En réalité, nous n'avons jamais la connaissance exacte du modèle de robot dû à beaucoup des problèmes liés à la formulation modèle. Deux incertitudes communes qui ne permettent pas l'annulation modèle exacte dans des applications robotiques sont les masses inconnues de lien dues aux perturbations de charge utile et aux coefficients inconnus de frottement. La méthode pour traiter ces types d'incertitudes paramétriques serait d'employer le contrôleur de calcul de couple avec une certaine évaluation fixe des paramètres inconnus au lieu des paramètres réels. Les contrôleurs adaptatifs sont formulés en séparant les paramètres constants inconnus des fonctions connues dans l'équation dynamique de robot. Le mécanisme d'adaptation continue d'extraire l'information à partir de l'erreur de trajectoire. La règle adaptative de mise à jour est formulée par l'analyse de stabilité du système d'erreur de trajectoire. C'est-à-dire, nous assurons la stabilité du système d'erreur de trajectoire en formulant la règle adaptative de mise à jour et par l'analyse stabilité du système d'erreur de trajectoire en même temps.

La dynamique de robot peut être écrite sous la forme :

$$W(q, \dot{q}, \ddot{q})\varphi = M(q)\ddot{q} + V_m(q, \dot{q}) + G(q) + F(\dot{q}) \quad (\text{III.24})$$

$W(q, \dot{q}, \ddot{q})$ est une matrice de $n \times r$ des fonctions de temps connues.

φ : est un vecteur $r \times 1$ des paramètres constants inconnus.

Cette séparation des paramètres inconnus et des fonctions de temps connues sera employée dans la formulation de la règle adaptative de mise à jour et également dans l'analyse de stabilité du système d'erreur de trajectoire.

Nous pouvons écrire :

$$\tau = W(q, \dot{q}, \ddot{q}) \quad (\text{III.25})$$

De le contrôleur adaptatif de calcul de couple est donné par :

$$\tau = \hat{M}(q) + (\ddot{q}_d + k_v \dot{e} + k_p e) + \hat{V}_m(q, \dot{q}) + \hat{G}(q) + \hat{F}(\dot{q}) \quad (\text{III.26})$$

Il est facile de voir de notre définition de l'erreur de trajectoire comment (III.26) peut être écrit à :

$$\tau = \hat{M}(q) + (\ddot{e} + k_v \dot{e} + k_p e) + \hat{M}(q) \ddot{q} + \hat{V}_m(q, \dot{q}) + \hat{G}(q) + \hat{F}(\dot{q}) \quad (\text{III.27})$$

Par l'utilisation (III.24), (III.27) peut être écrit à :

$$\tau = \hat{M}(q) + (\ddot{e} + k_v \dot{e} + k_p e) + W(q, \dot{q}, \ddot{q}) \varphi \quad (\text{III.28})$$

Le système d'erreur de trajectoire :

$$\ddot{e} + k_v \dot{e} + k_p e = \hat{M}^{-1}(q) + W(q, \dot{q}, \ddot{q}) \varphi \quad (\text{III.29})$$

Sous la forme d'espace d'état :

$$\dot{e} = A e + B M^{-1}(q) W(q, \dot{q}, \ddot{q}) \varphi \quad (\text{III.30})$$

Là où l'erreur de paramètre est :

$$\tilde{\varphi} = \varphi - \hat{\varphi} \quad (\text{III.31})$$

$$e = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad B = \begin{bmatrix} O_n \\ I_n \end{bmatrix} \quad A = \begin{bmatrix} O_n & I_n \\ -k_p & -k_v \end{bmatrix}$$

I_n : est la matrice d'identité de $n \times n$.

O_n : étant la matrice zéro de $n \times n$.

Le positif-défini Lyapunov-comme la fonction :

$$V = e^T P e + \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} \quad (\text{III.32})$$

P : est un $2^n \times 2^n$ positif-défini, la matrice constante et symétrique,

Γ : est une matrice diagonale et positif-définie de $r \times r$. C'est-à-dire, Γ peut être écrit a :

$$\Gamma = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_r)$$

Là où les γ_i sont des constants scalaires positive.

La différenciation (III.32) en ce qui concerne le temps rapporte :

$$\dot{V} = e^T P \dot{e} + \dot{e}^T P e + 2\tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \quad (\text{III.33})$$

Avec :

$$\left[\tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} \right]^T = \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}}$$

Remplacer e de (III.30) dans (III.33) rapporte :

$$\dot{V} = e^T P (Ae + B\tilde{M}^{-1}(q)W(\cdot)\tilde{\varphi}) + (Ae + B\tilde{M}^{-1}(q)W(\cdot)\tilde{\varphi})^T P e + 2\tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \quad (\text{III.34})$$

La combinaison des limites dans (III.34) et l'utilisation de la propriété scalaire de transport donne :

$$\dot{V} = -e^T Q e + 2\tilde{\varphi}^T (\Gamma^{-1} \dot{\tilde{\varphi}} + W^T(\cdot)\hat{M}^{-1}(q)B^T P e) \quad (\text{III.35})$$

Là où Q est une matrice positif-définie et symétrique qui satisfait l'équation de Lyapunov

$$A^T P + P A = -Q \quad (\text{III.36})$$

Le choix de la règle de mise à jour d'adaptation devient évident. Spécifiquement, par la substitution :

$$\dot{\tilde{\varphi}} = -\Gamma W^T(\cdot)\hat{M}^{-1}(q)B^T P e \quad (\text{III.37})$$

(III.35) devient :

$$\dot{V} = -e^T Q e \quad (\text{III.38})$$

Nous pouvons substituer (III.31) dans (III.37) pour obtenir la règle adaptative de mise à jour :

$$\dot{\tilde{\varphi}} = \Gamma W^T(\cdot)\hat{M}^{-1}(q)B^T P e \quad (\text{III.39})$$

Maintenant, de l'équation de robot (III.24), il est clair que :

$$\ddot{q} = M^{-1}(q)(\tau - V_m(q, \dot{q})\dot{q} - G(q) - F(q)) \quad (\text{III.40})$$

Les gains :

$$k_v = k_v I_n \quad \text{et} \quad k_p = k_p I_n$$

La matrice P soit choisie par :

$$P = \begin{bmatrix} P_1 I_n & P_2 I_n \\ P_2 I_n & P_3 I_n \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (k_p + \frac{1}{2}k_v)I_n & \frac{1}{2}I_n \\ \frac{1}{2}I_n & I_n \end{bmatrix}$$

Pour voir si notre choix de P donne un Q positif-défini, effectuer l'opération de matrice :

$$A^T P + P A = -Q \quad (\text{III.41})$$

$$Q = \begin{bmatrix} \frac{1}{2}k_p I_n & O_n \\ O_n & (k_v + \frac{1}{2})I_n \end{bmatrix}$$

Le vecteur associé d'évaluation de paramètre est :

$$\hat{\phi} = \begin{bmatrix} \hat{m}_1 \\ \hat{m}_2 \end{bmatrix}$$

Avec des règles de mise à jour :

$$\begin{aligned} \dot{m}_1 &= \gamma_1 \left[(W_{11} M I_{11} + W_{21} M I_{21})(P_2 e_1 + P_3 e_1) + (W_{11} M I_{21} + W_{21} M I_{22})(P_2 e_2 + P_3 e_2) \right] \\ \dot{m}_2 &= \gamma_2 \left[(W_{12} M I_{11} + W_{22} M I_{21})(P_2 e_1 + P_3 e_1) + (W_{12} M I_{21} + W_{22} M I_{22})(P_2 e_2 + P_3 e_2) \right] \end{aligned}$$

(III.42)

Là où :

$$M I_{11} = \frac{1}{\Delta} (\hat{m}_2 l_2^2)$$

$$M I_{21} = -\frac{1}{\Delta} (\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2)$$

$$M I_{22} = \frac{1}{\Delta} (2 \hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2) l_1^2)$$

$$\Delta = (2 \hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2) l_1^2 (\hat{m}_2 l_2^2) - (\hat{m}_2 l_2^2 + \hat{m}_2 l_1 l_2 c_2))$$

Nous présenterons dans cette section l'application de cette technique sur notre bras manipulateur avec les résultats de simulation obtenus par cette technique.

Et avec une trajectoire désirée de :

$$q_{d1} = q_{d2} = \sin(t)$$

Et les masses $m_1 = 0.8 \text{ kg}$ et $m_2 = 2.3 \text{ kg}$, le contrôleur calcul de couple a été simulé avec $q(0) = \dot{q}(0) = 0$, avec les gains de contrôleur réglés à :

$$k_p = k_v = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

Là où $l_1 = l_2 = 1 \text{ m}$ et g est la constante de la gravité. Nous choisissons $\hat{m}_1 = 0.85 \text{ kg}$ et $\hat{m}_2 = 2.2 \text{ kg}$ puisqu'on assume que les valeurs réelles sont inconnues. Après avoir substitué la loi de commande ci-dessus dans la dynamique de robot de 2ddl.

Loi de commande:

$$\Gamma = M(q)\ddot{q}_d + (\ddot{q}_d + K_v\dot{e} + K_p e) + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \quad (\text{III.43})$$

Le « ^ » dénote les paramètres estimés.

Ou sous la format :

$$\begin{aligned} \tau_1 &= (2\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2) l_1^2) (\ddot{q}_{d1} + k_{v1}\dot{e}_1 + k_{p1}e_1) \\ &\quad + (\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2) (\ddot{q}_{d2} + k_{v2}\dot{e}_2 + k_{p2}e_2) - \hat{m}_2 l_1 l_2 s_2 \dot{q}_2^2 \\ &\quad - 2\hat{m}_2 l_1 l_2 s_2 \dot{q}_1 \dot{q}_2 + \hat{m}_2 l_2 g c_{12} + (\hat{m}_1 + \hat{m}_2) l_1 g c_1 \\ \tau_2 &= (\hat{m}_2 l_2^2 + \hat{m}_2 l_1 l_2 c_2) (\ddot{q}_{d1} + k_{v1}\dot{e}_1 + k_{p1}e_1) + \hat{m}_2 l_2 g c_{12} \\ &\quad + \hat{m}_2 l_2^2 (\ddot{q}_{d2} + k_{v2}\dot{e}_2 + k_{p2}e_2) + \hat{m}_2 l_1 l_2 s_2 \dot{q}_1^2 \end{aligned}$$

Nous pouvons former le système d'erreur :

$$\ddot{e} + k_v \dot{e} + k_p e = \hat{M}^{-1}(q) W(q, \dot{q}, \ddot{q}) \tilde{\varphi}$$

Là où $\hat{M}^{-1}(q)$ est l'inverse de la matrice d'inertie $M(q)$ avec m_1 et m_2 remplacés par \hat{m}_1 et \hat{m}_2 , respectivement. Et la matrice $W(q, \dot{q}, \ddot{q})$, a parfois appelé la matrice de régression, est une matrice 2×2 donnée par :

$$w(q, \dot{q}, \ddot{q}) = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

$$W_{11} = l_1^2 \ddot{q}_1 + l_1 g c_1$$

$$W_{12} = l_2^2 (\ddot{q}_1 + \ddot{q}_2) + l_1 l_2 c_2 (2 \ddot{q}_1 + \ddot{q}_2) + l_1^2 \ddot{q}_1 - l_1 l_2 s_2 \dot{q}_2^2 - 2 l_1 l_2 s_2 \dot{q}_1 \dot{q}_2 + l_2 g c_{12} + l_1 g c_1$$

$$W_{21} = 0$$

$$W_{22} = l_1 l_2 c_2 \ddot{q}_1 + l_1 l_2 s_2 \dot{q}_1^2 + l_2 g c_{12} + l_2^2 (\ddot{q}_1 + \ddot{q}_2)$$

Le vecteur $\tilde{\varphi}$ appelé le vecteur d'erreur de paramètre, est un vecteur 2×1 donné par :

$$\tilde{\varphi} = \begin{bmatrix} \tilde{\varphi}_1 \\ \tilde{\varphi}_2 \end{bmatrix}$$

Là où :

$$\tilde{\varphi}_1 = m_1 - \hat{m}_1 \quad \text{et} \quad \tilde{\varphi}_2 = m_2 - \hat{m}_2$$

Les matrices associées de vecteur et de gain 2×2 de l'erreur de trajectoire 2×1 en (3) sont données par :

$$e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \quad k_v = \begin{bmatrix} k_{v1} & 0 \\ 0 & k_{v2} \end{bmatrix}, \quad k_p = \begin{bmatrix} k_{p1} & 0 \\ 0 & k_{p2} \end{bmatrix}$$

Nous ne pouvons pas employer le système d'erreur (3) pour choisir les valeurs pour le k_v et k_p avec des méthodes de contrôle linéaires standard puisque le côté droit du système d'erreur se compose de fonctions non linéaires de : q , \dot{q} , et \ddot{q} . Pour le contrôleur approximatif de calcul de couple, la simulation peut être employée pour choisir les valeurs appropriées pour k_v et k_p . Un bon point de départ pour choisir les valeurs de k_p et le k_v pour le contrôleur approximatif de calcul de couple est d'employer les mêmes valeurs, car l'arrangement de calcul de couple précepte, pour un constante d'amortissement désiré donner et une fréquence naturelle.

Règle de la mise à jour :

$$\dot{\varphi} = \Gamma W^T(q, \dot{q}, \ddot{q}) M^{-1}(q) B^T P e \quad (\text{III.44})$$

Où :

$$w(q, \dot{q}, \ddot{q}) = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \quad p = \begin{bmatrix} p_1 I_n & p_2 I_n \\ p_2 I_n & p_3 I_n \end{bmatrix} \quad e = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad B = \begin{bmatrix} O_n \\ I_n \end{bmatrix}$$

$$W(q, \dot{q}, \ddot{q})\varphi = M(q)\ddot{q}_d + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q})$$

$$A^T P + P A = -Q \quad \text{Ou : } A = \begin{bmatrix} O_n & I_n \\ -k_p & -k_v \end{bmatrix}$$

Les matrices P et Q sont définies symétriques positives.

Stabilité : le vecteur de suivi d'erreur e est asymptotiquement stable.

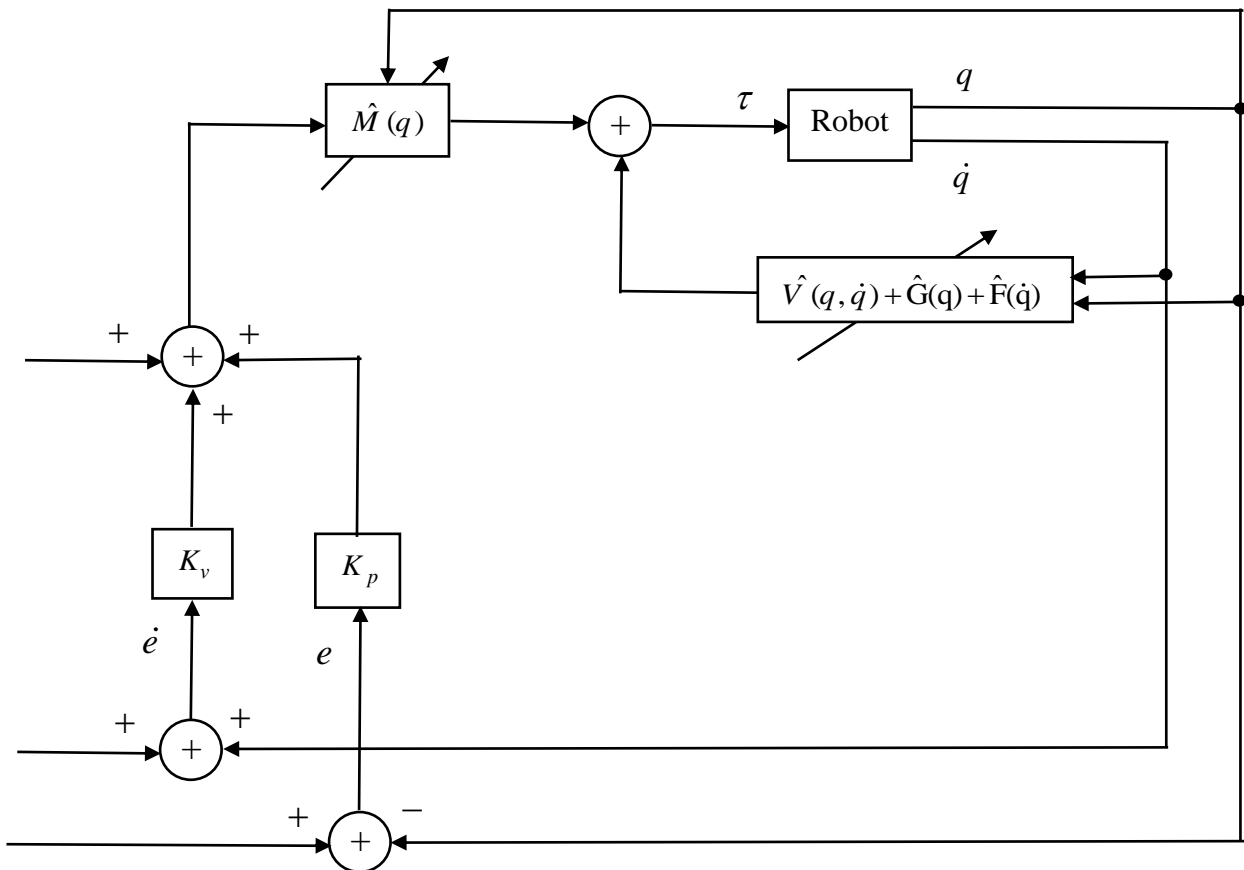


Figure III.15: Schéma de commande adaptative par calcul de couple

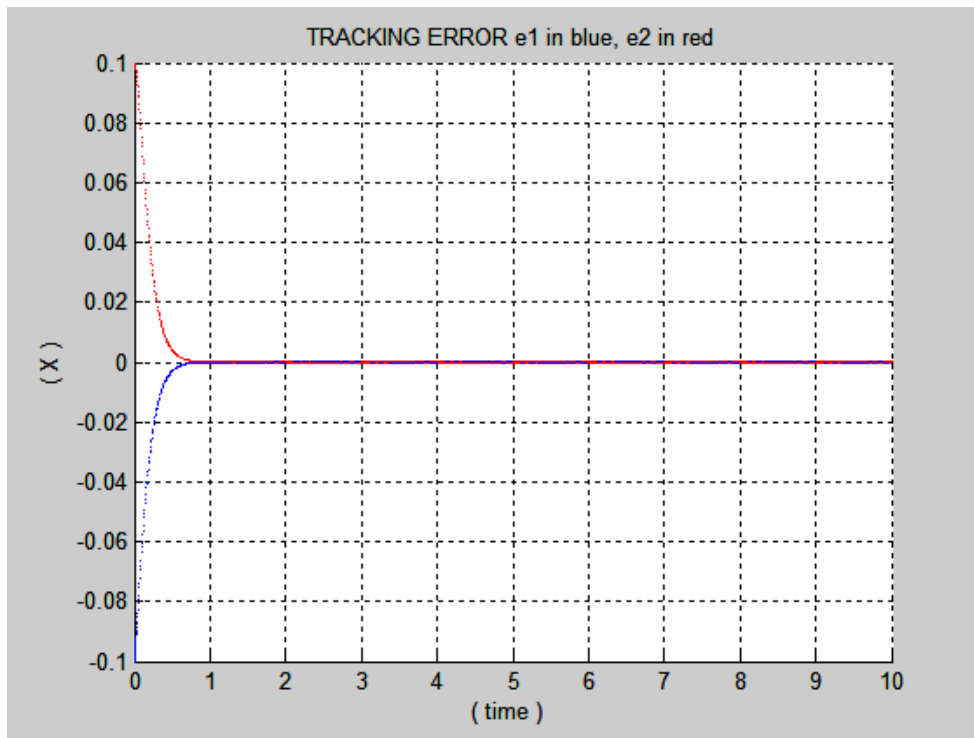


Figure III.16: Erreur de poursuite de position

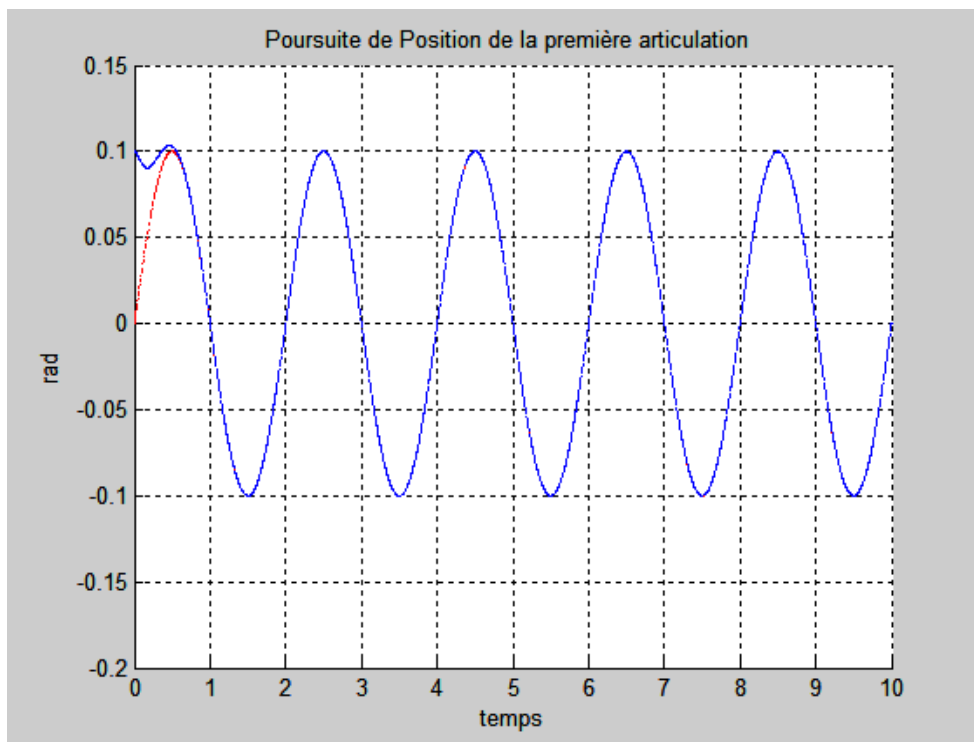


Figure III.17: Poursuite de position de la première articulation

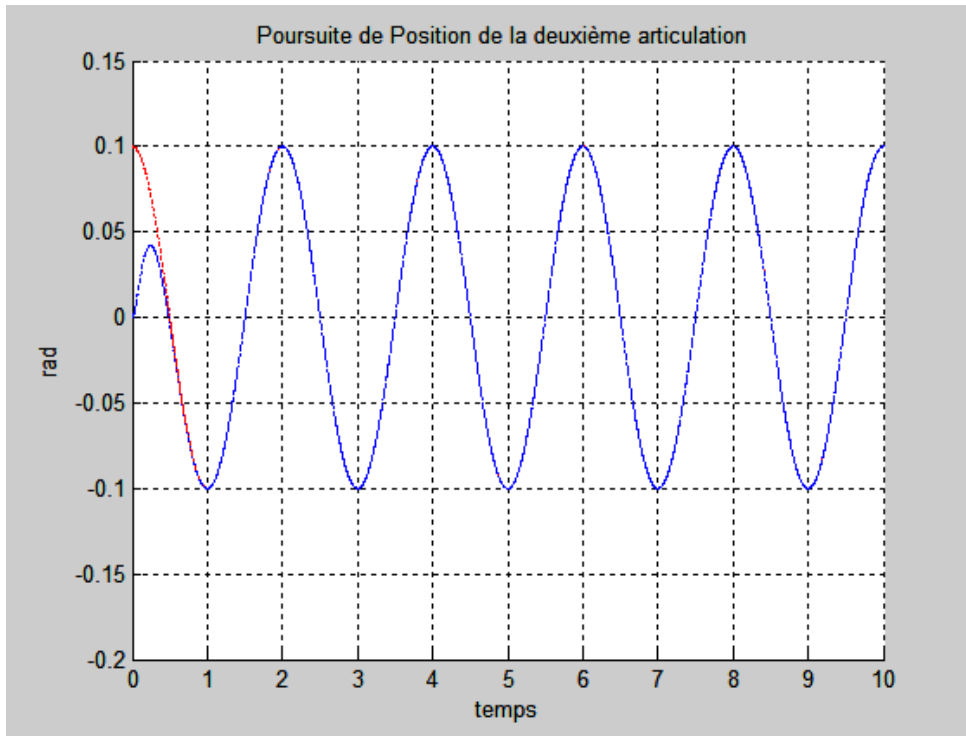


Figure III.18 : Poursuite de position de la deuxième articulation

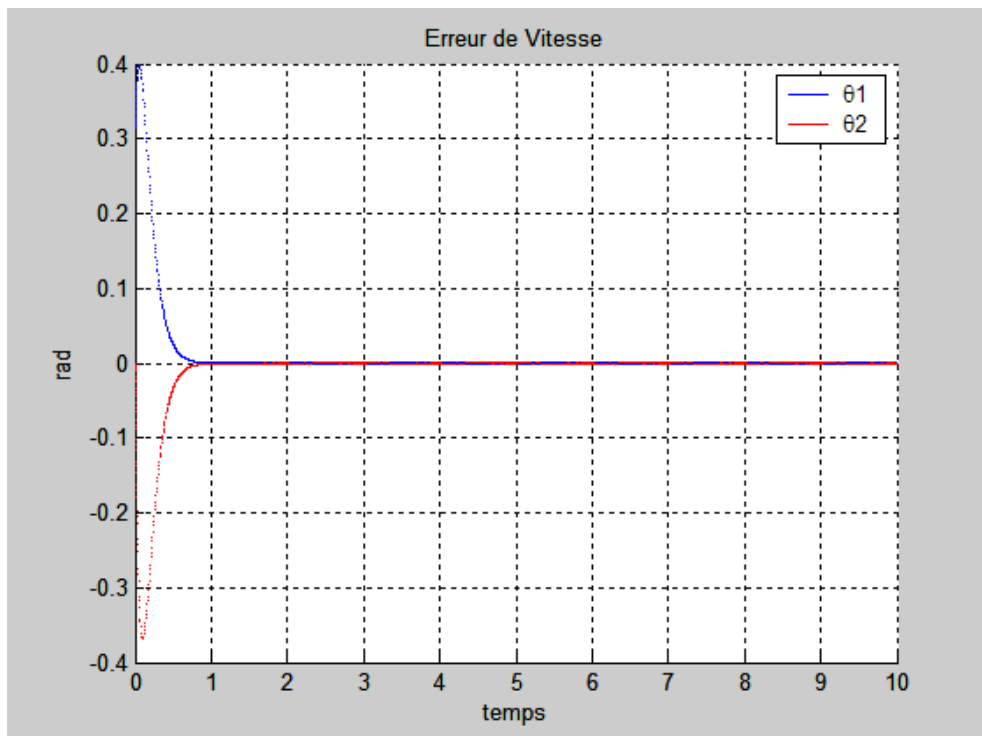


Figure III.19: Erreur de vitesse

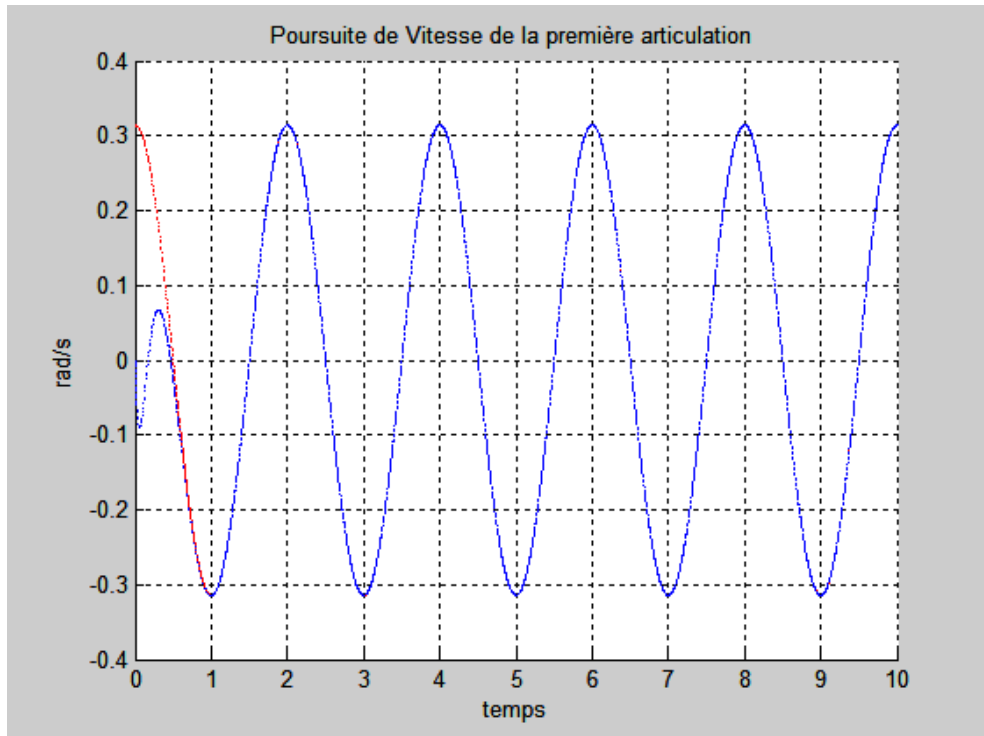


Figure III.20: Poursuite de vitesse de la première articulation

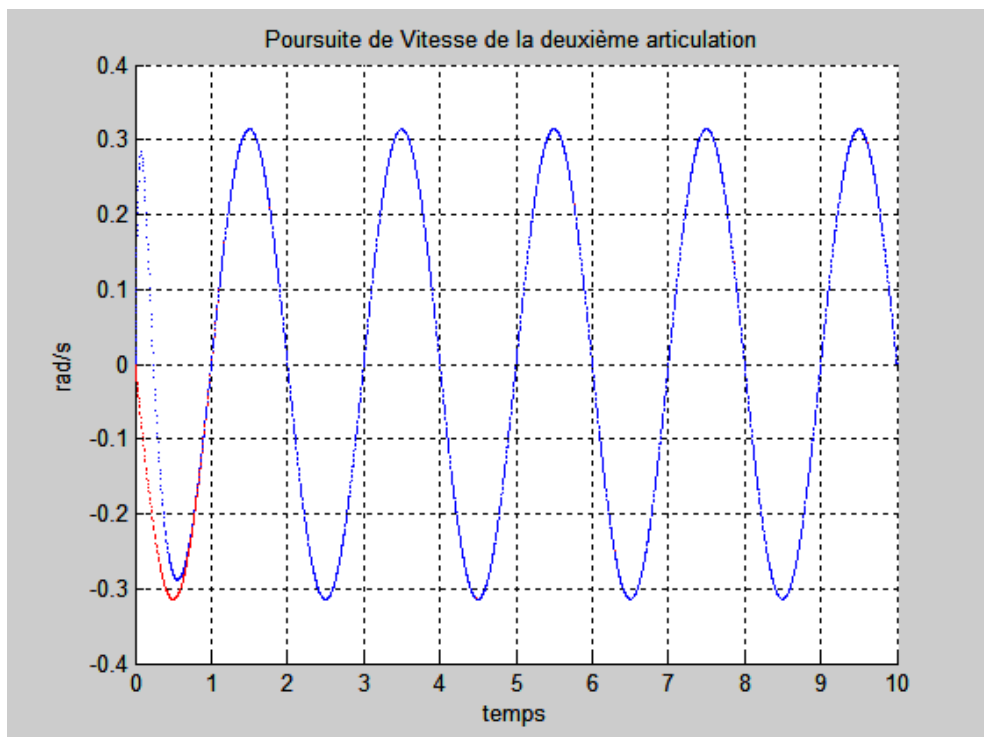


Figure III.21: Poursuite de vitesse de la deuxième articulation

Dans la commande adaptative par calcul de couple les erreurs de poursuite (en position et vitesse) sont graduellement réduites et l'erreur s'annule à 0,6(s), cela est dû aux paramètres qui sont incertains donc il prend un peu plus de temps pour converger vers les paramètres réels. Après les 0.6s on remarque l'absence d'oscillations et les erreurs de poursuite demeurent relativement petites et presque identiques.

3.8.2 Commande adaptative inertiel

La loi de commande est donné par:

$$\tau = Y(q, \dot{q}, \ddot{q})\varphi + K_v \dot{e} + K_v \Lambda e$$

Règle de mise à jour : $\dot{\varphi} = \Gamma Y^T(q, \dot{q}, \ddot{q})(\Lambda e + \dot{e})$

Où : $Y(q, \dot{q}, \ddot{q})\varphi = M(q)(\ddot{q}_d + \Lambda e) + V_m(q, \dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(q)$

Stabilité : le suivi d'erreur e et \dot{e} sont apparemment stable. Y : est une matrice de $n \times r$ des fonctions de temps connues. Le paramètre estimatif φ est délimité.

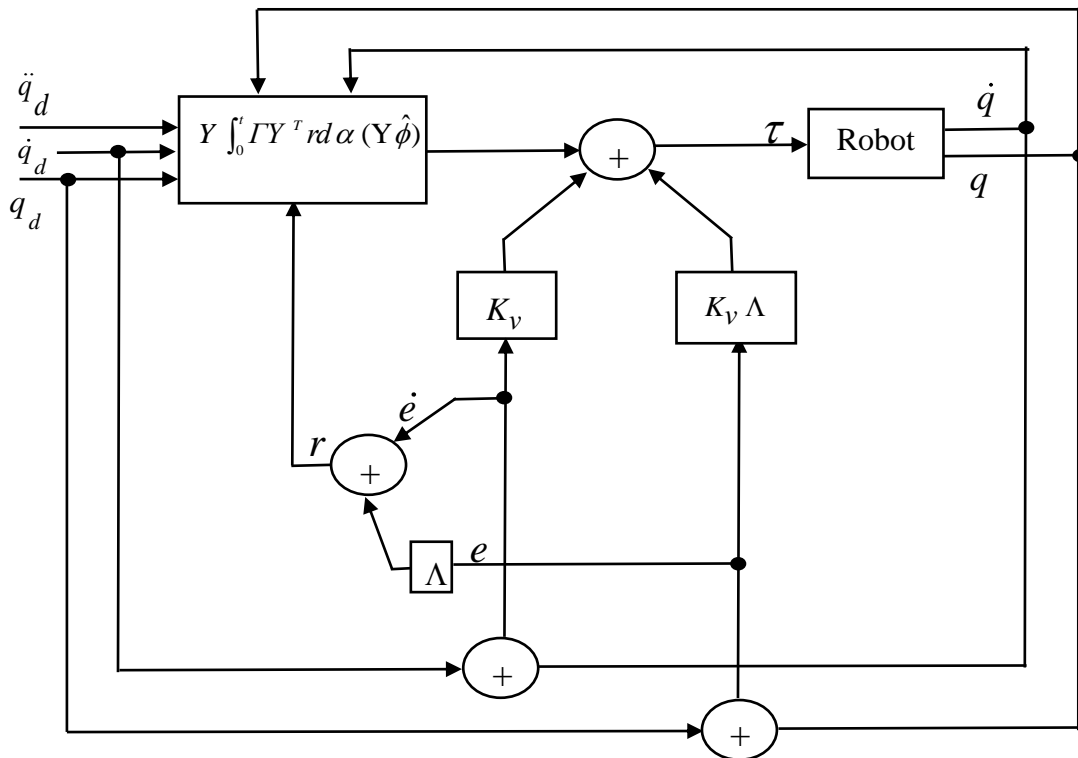


Figure III.22: Schéma fonctionnel du contrôleur inertie-connexe adaptatif

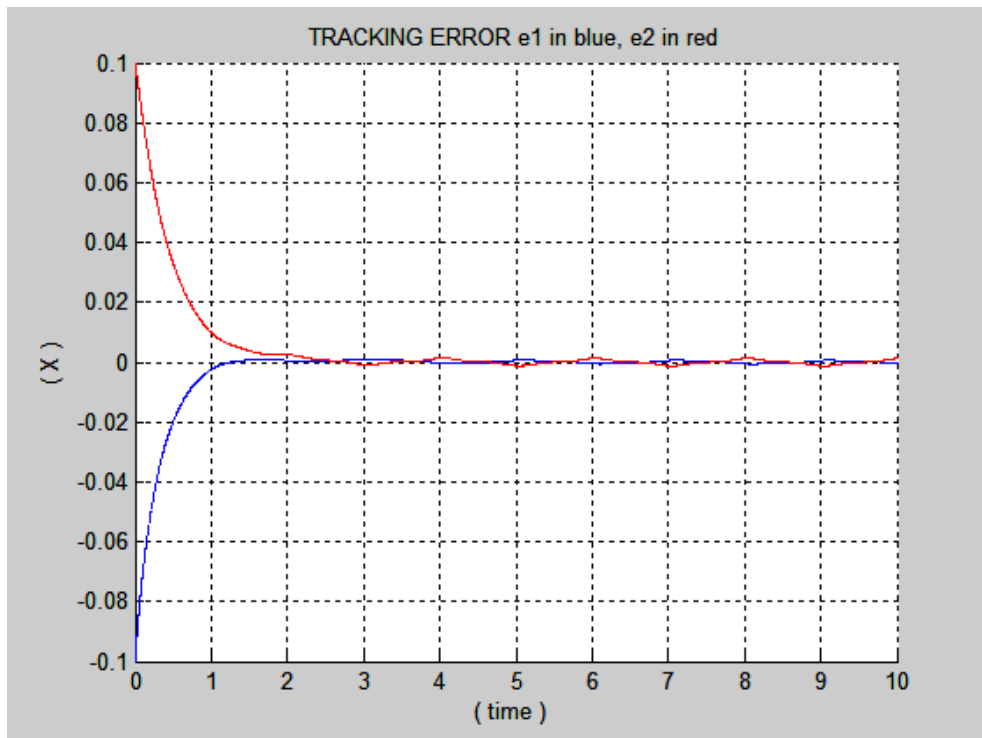


Figure III.23: Erreur de poursuite de position

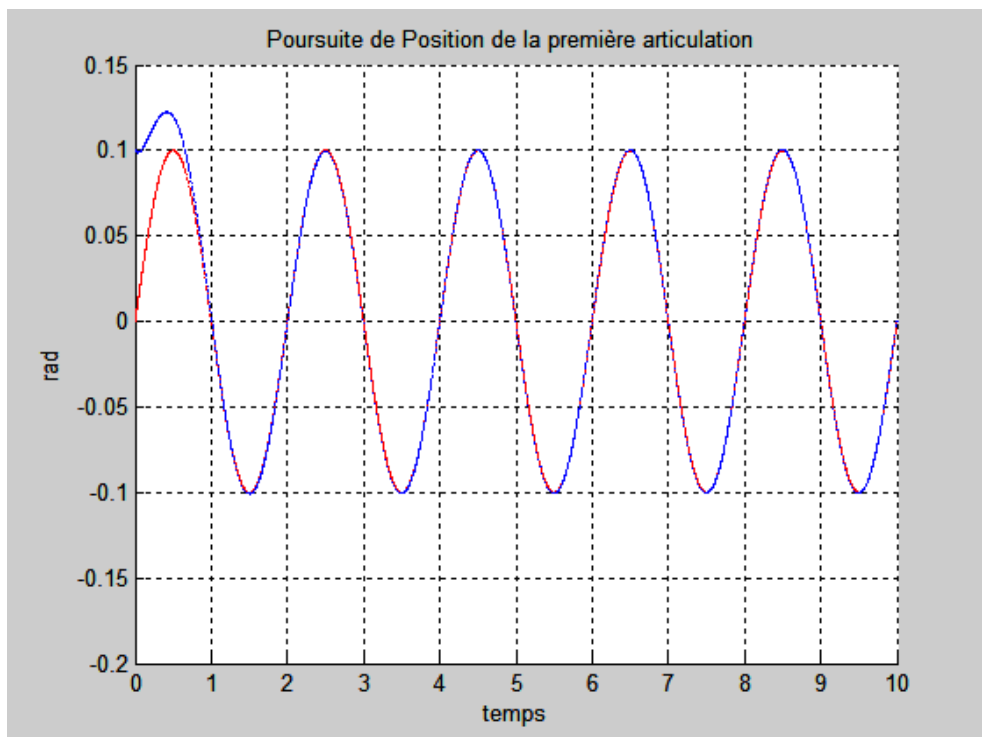


Figure III.24: Poursuite de position de la première articulation

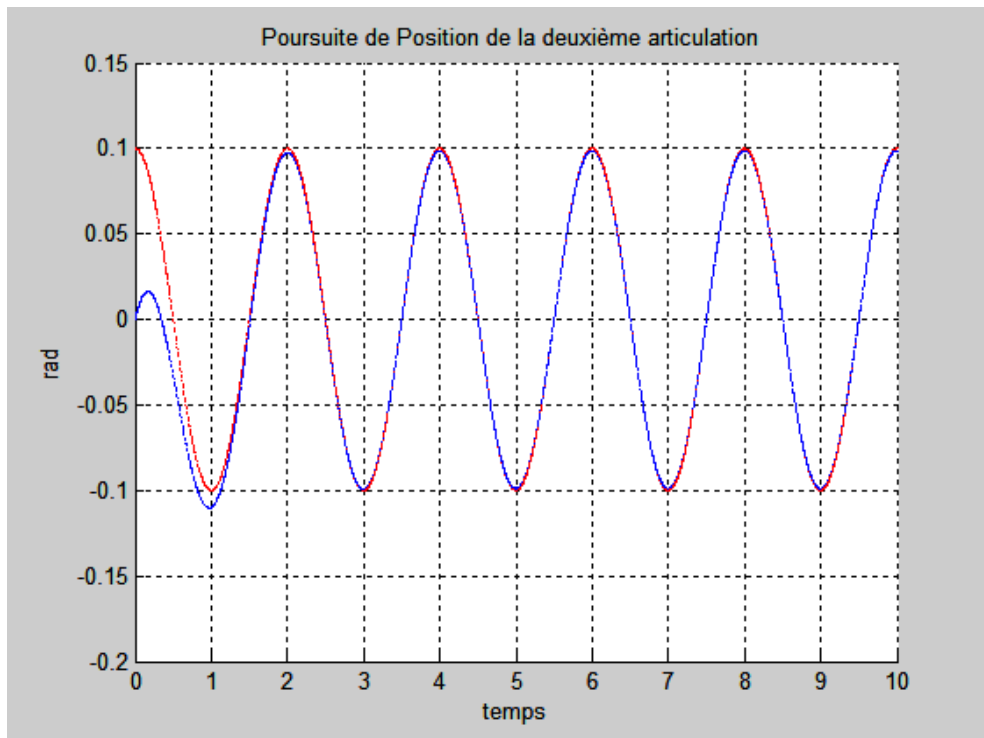


Figure III.25: Poursuite de position de la deuxième articulation

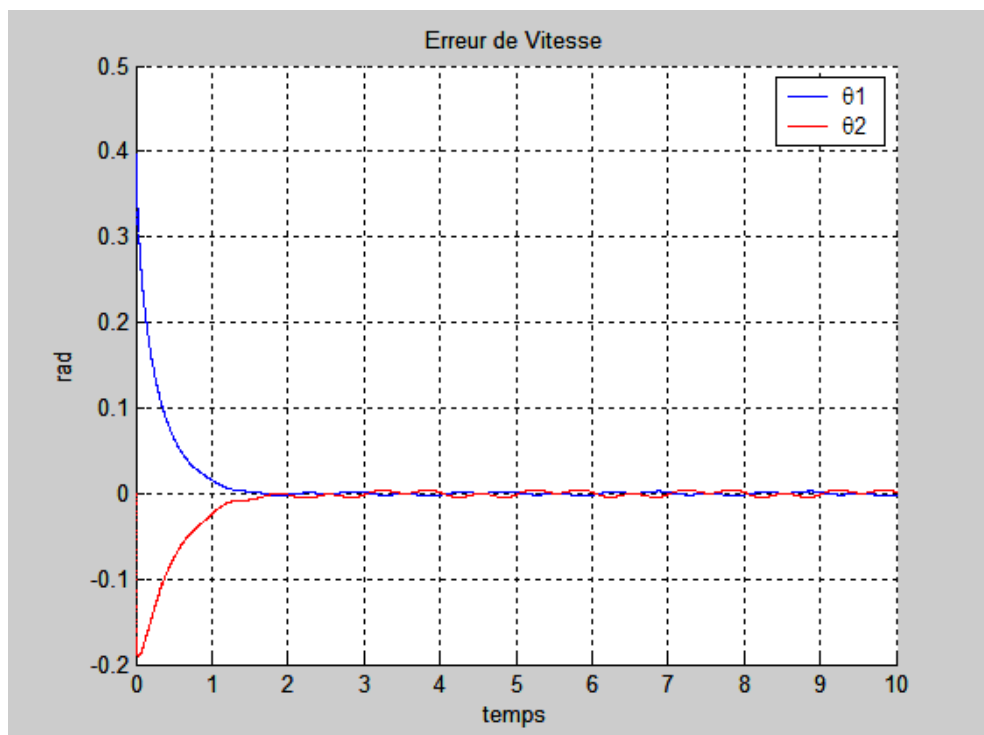


Figure III.26: Erreur de vitesse

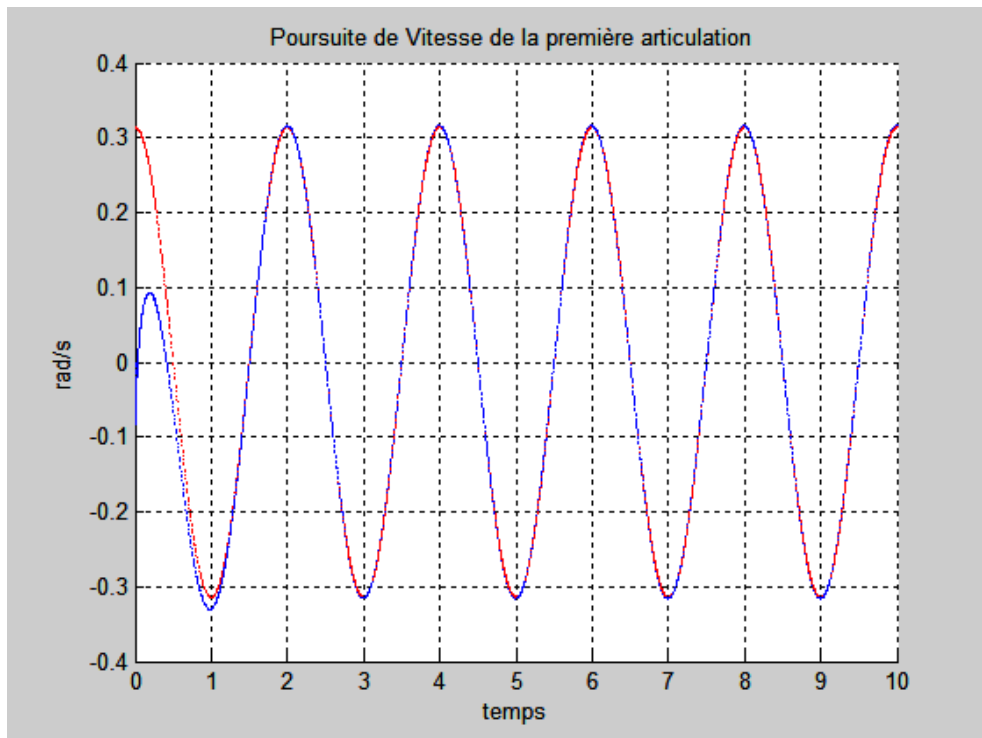


Figure III.27: Poursuite de vitesse de la première articulation

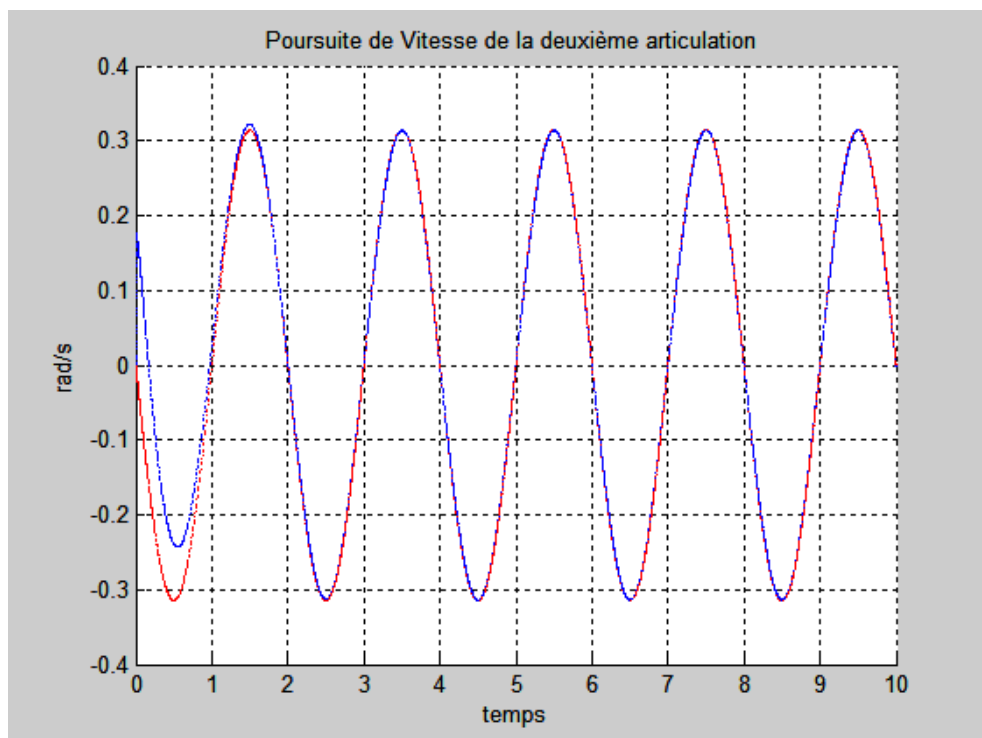


Figure III.28: Poursuite de vitesse de la deuxième articulation

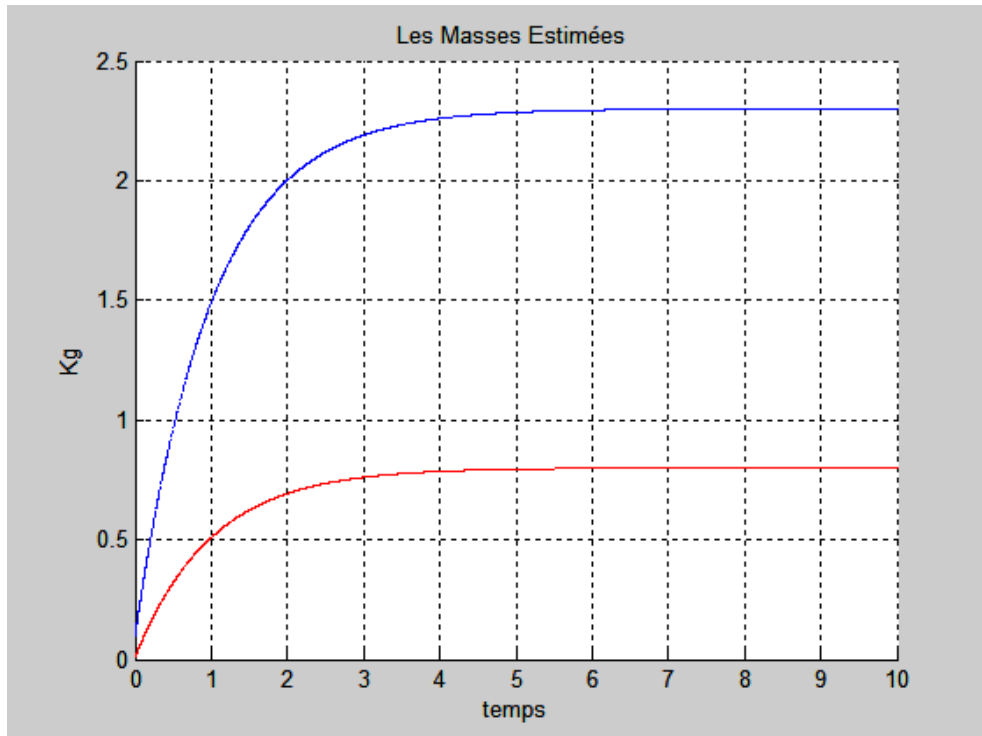


Figure III.29: Estimation des masses

Dans la commande adaptative inertielle les erreurs de poursuite (en position et vitesse) sont graduellement réduites jusqu'à 2,4(s), ceci est parce qu'il nécessite des restrictions concernant l'inversion de la matrice M des inerties. Après les 2.4s l'erreur reste à osciller dans un intervalle de 0.004 jusqu'à 0.008 pour la poursuite de position et 0.008 jusqu'à 0.02 pour la poursuite vitesse. On remarque aussi que cette technique prend plus de temps (2.4 s) pour adapter les paramatres de modèle par rapport à la technique précédente, même chose pour les erreurs de poursuite (en position et vitesse) sont un peu plus grand.

3.9 Conclusion

Bien que la commande de la plupart des robots industriels actuels soit encore conçue à partir des théories de l'automatique linéaire, des méthodes plus avancées, tenant compte du caractère non linéaire des structures articulées, doivent être envisagées pour les applications exigeant de grandes performances dynamiques (temps de cycle, précision dynamique...).

Pour cette raison, nous avons au cours de ce chapitre donné le principe de quatre méthodes répondant à ces critères : la commande par calcul de couple avec PD et avec PID, et la commande adaptative par calcul de couple et adaptative inertielle. La mise en œuvre de ces méthodes faisant appel au calcul en ligne du modèle dynamique utilise avantageusement pour le calcul dynamique inverse et aussi pour l'identification des paramètres dynamiques.

Dans toutes les lois de commande étudiées, nous avons considéré que les positions et vitesses articulaires sont disponibles. En général les robots sont équipés de capteur de haute précision pour la mesure des variables articulaires. Les génératrices tachymétriques utilisées pour la mesure des vitesses articulaires fournissent des signaux entachés d'un bruit de mesure souvent considérable. La solution la plus simple pour résoudre ce problème consiste à obtenir le signal de vitesse par une différenciation numérique du signal de position. D'autres techniques plus sophistiquées consistent à construire un observateur qui estime la vitesse en fonction des couples appliqués aux actionneurs et les positions articulaires [Canudas de Wit 92], [Nicosia 90], [Berguis 93], [Khelifi 95] [Cherki 96].

La meilleure solution est d'utiliser une commande basée sur l'intelligence artificielle, qui est l'objectif du chapitre suivant.

CHAPITRE 4

Les techniques de l'intelligence artificielle

4.1. Introduction

Le contrôle intelligent des systèmes robotiques est un problème difficile parce que la dynamique de ces systèmes est fortement non linéaire. Le contrôleur le plus simple pour les robots manipulateurs industriels est le régulateur Proportionnel Intégral et Dérivé (PID). [Frank L et al, 2004], [Wisama. K et al, 1999], [Mark W et al, 2002], [Kelly R et al, 2005]. Généralement ce type de correcteur est conçu pour des modèles de robots décrits par des équations différentielles dynamiques (indépendantes). Ce contrôle est également connu en tant que contrôle dynamique, parce qu'il est basé sur l'utilisation du modèle dynamique du robot [SPONG. G et al, 1989]. Bien que ces correcteurs soient largement utilisés dans les manipulateurs industriels (bras de robots), selon les tâches à réaliser, ils ne donnent pas toujours les meilleures performances sachant que la mise en œuvre de ces commandes exige une connaissance précise du modèle complet [EGELAND. O, 1986]. Un des plus grands problèmes dans ce domaine est qu'il y a toujours des incertitudes dans les environnements non structurés, ces incertitudes sont principalement dues à l'imprécision des capteurs et à l'imprévisibilité des caractéristiques d'environnement et de la complexité de la dynamique [Craig. J, 1989] [Lewis. F, 1993] [Castillo. O, 2000]. Ces inconvénients du contrôle ont motivé les chercheurs à développer de nouvelles stratégies de commande intelligente et adaptative pour améliorer le rendement des systèmes contrôlés.

Les techniques de l'intelligence artificielle sont considérées comme une solution très intéressante pour les systèmes non linéaire où il est difficile d'établir un modèle mathématique [Yen. J et al, 1992], [Yang. X et al, 2005], [Surmann. H et al, 1995], [Martinez. A et al, 1994].

4.2. Commande basée sur l'intelligence artificielle

La commande basée sur l'intelligence artificielle représente un domaine de recherche très active et vaste à cause des avantages qu'offrent ces outils surtout pour les systèmes non linéaires. Elle est fondée sur l'exploitation des capacités d'apprentissage, d'approximation et d'optimisation qui caractérise ces outils. Parmi Ces outils on cite les réseaux de neurones artificiels, la logique floue et les algorithmes génétiques [Li. W et al, 2001]. Il y a aussi la combinaison de plus d'une méthode pour obtenir des algorithmes hybrides qui englobent les différents avantages de ces méthodes.

Les techniques de l'intelligence artificielle basée sur la logique floue sont considérées comme une solution très intéressante pour les systèmes non linéaire où il est difficile d'établir un modèle mathématique [Martinez. A et al, 1994], [Surmann. H et al, 1995], [Zimmermann. H et al, 1999], [Yang. X et al, 2005], [Yen. J, Pfluger. N, 1992], [Fatmi. A et al, 2006], [Zavlangas. P-G et al, 2000].

Le contrôleur de logique floue peut très bien décrire le comportement d'un système désiré et traduire l'expérience humaine sous forme des variables linguistiques et des règles floues (si-alors), mais il manque de la capacité d'apprentissage et ne peuvent pas de s'ajuster sur un nouvel environnement [Zimmermann. H, 1999]. D'une part, les réseaux de neurones artificiels offrent une capacité d'apprentissage avec leur propriété d'approximation universelle, mais ils ne peuvent pas interpréter et vérifier si la solution obtenue est faisable [I.E.E.E, 1992], [Narendra. K et al, 1990]. Les deux approches sont complémentaires. L'objectif de cet chapitre est de proposer une méthode de commande d'un bras manipulateur à deux degré de liberté, utilisant un système d'inférence adaptative neuro-flou (ANFIS), qui combine les concepts de la logique floue et les réseaux de neurones pour constituer un contrôleur intelligent hybride appelé ANFIS.

Dans ce chapitre le principe général et la théorie de base de la logique floue et les réseaux de neurones artificielle sera présenté en annexe I. Ensuite, on va décrit les différentes types d'implémentation des réseaux neuro et de la logique floue.

4.3. Neuro-Flou

4.3.1 Structure de commande neuro-flou

Avec les réseaux de neurones et la logique floue il existe diverses combinaisons de ces deux méthodes ont été développées depuis 1988. Elles ont donné naissance aux systèmes neuro-flous qui sont le plus souvent orientées vers la commande de système complexe et les problèmes de classification [Benmiloud, 2006]. La figure IV.24 résume le principe du système neuro-flou qui représente l'intersection entre la logique floue et les réseaux de neurones. Cette section introduit les principes de base et les fondations de l'architecture de réseau d'ANFIS et de sa règle d'apprentissage hybride.

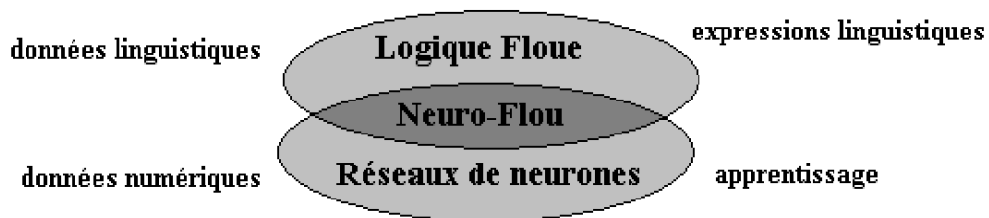


Figure IV.1 : Principe du système neuro-flou

4.3.2 Définition

Un système neuro-flou hybride est un réseau de neurone avec un signal flou, des poids flous, une fonction de transfert floue et un module d'adaptation de la structure de RNF par un apprentissage d'un ensemble de données.

4.3.3 Principe de fonctionnement

Les réseaux de neuro-flous hybride apprennent des rapports et des modèles en utilisant un algorithme d'apprentissage supervisé qui examine les données dans un ensemble de la formation qui consiste en exemples d'entrées et leurs sorties associées. Pendant la phase d'apprentissage, un réseau neuro-flou hybride modifie sa structure interne pour refléter le rapport entre les entrées et les sorties dans l'ensemble de la formation (base de connaissance). L'exactitude d'un réseau neuro-flou est vérifiée après que le cycle de d'apprentissage soit complet en utilisant un ensemble séparé d'entrées et sorties appeler l'ensemble de la validation.

4.3.4 Architectures neuro-floues

Plusieurs architectures ont été développées depuis les années quatre-vingt et sont orientées généralement vers la commande des systèmes complexes et les problèmes de classification [Varkiani. S, 1998], [Mordjaoui. M, 2008]. Diverses architectures ont été proposées pour les réseaux neuro-flous, allant d'une fusion des deux types de méthodes à une utilisation séquentielle de l'une ou de l'autre. Nous présenterons celles qui sont les plus utilisées. En conséquence, il existe trois architectures hybrides Neuro- floues:

4.3.4.1 Première architecture

La première méthode neuro-floue consiste au codage du système d'inférence floue sous la forme d'un réseau de neurones multicouches dans lequel les poids correspondent aux paramètres du système. L'architecture du réseau dépend du type de règle et des méthodes d'inférences, d'agrégation et de défuzzification choisies.

Par exemple, pour des règles de la forme " si V_1 est A_{i1} et V_2 est A_{i2} alors W est w_i ", on fait appel à un réseau de neurones qui admet pour entrées les valeurs x_1 et x_2 prises par les variables V_1 et V_2 et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenance A_{i1} pour x_1 et A_{i2} pour x_2 , et à celui de la valeur prise par la conjonction des conditions de chaque règle utilisant un opérateur de conjonction adéquat. Cette situation est schématisée sur la figure IV.25.

Les fonctions d'appartenance intervenant dans les règles sont considérées comme des paramètres ajustés par l'intermédiaire des poids entrant dans la première couche cachée. Les conclusions w_i des règles sont également des paramètres ajustables par l'intermédiaire des poids associés à la dernière couche.

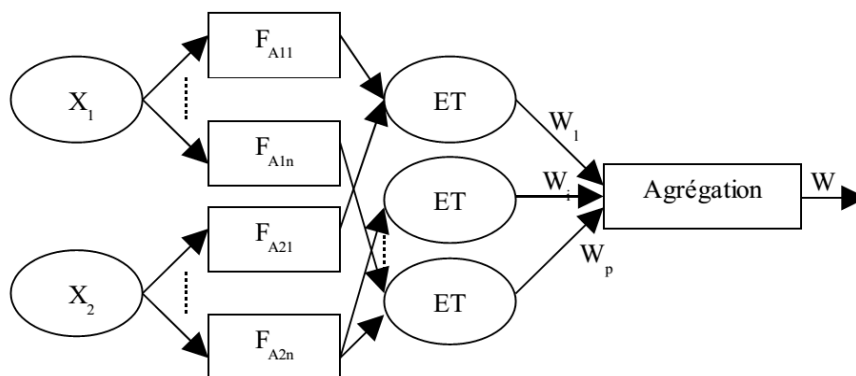


Figure IV.2 : Première architecture neuro-floue

Comme il est indiqué sur la figure des fonctions d'appartenances, les données vont subir trois étapes de calcul :

La première couche : s'occupe du calcul des degrés d'appartenance de chaque entrée.

Les paramètres de cette couche vont caractériser la fonction d'appartenance.

Le calcul des degrés d'appartenance s'effectue par des neurones spécialisés, dont la fonction d'activation est la fonction d'appartenance. La fonction la plus utilisée dans ce type d'architecture est la fonction gaussienne ; elle est donnée par :

$$y = e^{-(ax+b)^2} \quad (IV.7)$$

Donc les neurones de la première couche ont la forme suivante :

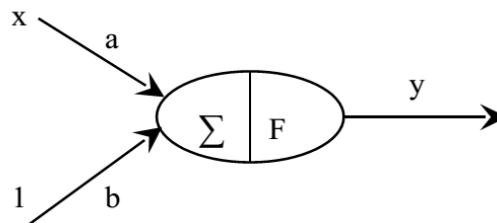


Figure IV.3 : Forme d'un neurone

La deuxième couche cachée : évalue en parallèle les prémisses des règles. Les paramètres de cette couche définissent dans ce cas l'opérateur de conjonction ET.

Plusieurs méthodes ont été proposées pour l'évaluation floue de l'opérateur ET (cette évaluation est appelée aussi calcul de la valeur de vérité). Généralement on utilise celle qui a été proposée par Lukasiewicz, cette dernière présente une certaine facilité d'implémentation neuronale moyennement satisfaisante. Elle est définie par :

$$ET [fA_1(x_1), fA_2(x_2)] = \text{Max} (0, fA_1(x_1) + fA_2(x_2) - 1) \quad (IV.8)$$

Si on pose :

$$u = fA_1(x_1) + fA_2(x_2) - 1 \quad (IV.9)$$

On aura :

$$ET = [fA_1(x_1), fA_2(x_2)] = \text{Max} (0, u) \quad (IV.10)$$

La fonction $\text{Max}(0, u)$ peut être évaluée approximativement par une fonction continue sigmoïde, elle est donnée par :

$$f = \text{Max}(0, u) = \frac{1}{1 + \exp\left(\frac{-u - 0.5}{0.227}\right)} \quad (\text{IV.11})$$

Donc notre opérateur de conjonction ET va prendre la forme suivante :

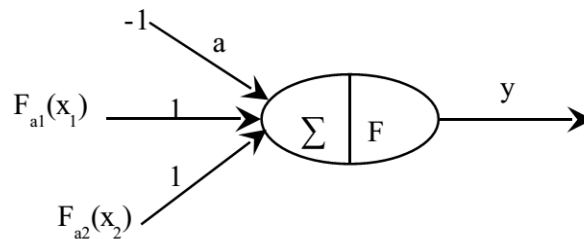


Figure IV.4 : Forme d'opérateur flou ET

La dernière couche : réalise l'opération d'inférence. Ayant calculé les valeurs de vérités produites par les opérateurs de conjonction, on doit maintenant établir une procédure permettant de déduire une sortie en fonction de ces valeurs de vérités. Cette procédure est réalisée soit en choisissant parmi ces valeurs, celle qui est la plus représentative, c'est à dire celle qui a le plus grand degré de vérité, soit en calculant le barycentre de toutes les valeurs. Une telle approche peut être effectuée par un seul neurone, où les valeurs de vérités sont pondérées par les poids synaptiques de ce neurone.

Les poids synaptiques du neurone qui calcule le barycentre (qui fait la défuzzification) sont d'une grande importance, car ce sont eux qui pondèrent les résultats des règles. Une rétro propagation serait indispensable à appliquer sur cette couche (c'est à dire sur les poids synaptiques du neurone chargé de la défuzzification).

Algorithme d'apprentissage : Cet algorithme est basé sur l'algorithme de la rétro propagation du gradient descendant. Il est le même que celui qui a été utilisé pour l'apprentissage des réseaux de neurones type PMC (annexe). La seule différence qui existe, est que les entrées du neurone chargé de la défuzzification seront normalisées.

Cette normalisation est nécessaire car elle permet de manipuler des valeurs inférieures ou égales à l'unité. Une telle approche est réalisée par la division de chaque entrée de ce neurone, sur la somme de toutes ses entrées.

4.3.4.2 Deuxième architecture

Cette méthode utilise des réseaux de neurones et des systèmes flous associés en série ou en parallèle. Plusieurs variantes sont ainsi possibles :

- Le réseau de neurones fonctionne en amont du système flou (figure IV.28). Les variantes d'entrées du système flou sont déterminées à partir des sorties du réseau de neurones (dans le cas où elles ne sont pas mesurables directement) ou encore un réseau de neurones effectue une tâche de classification ou de reconnaissance de formes, suivie d'un système flou d'aide à la décision.

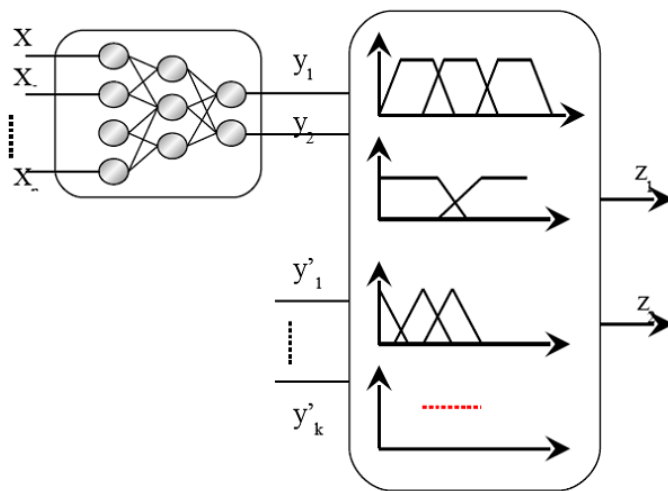


Figure IV.5 : Deuxième architecture des réseaux neuro-flou réalisation en série

Un réseau de neurones qui fonctionne en aval du système flou (figure IV.29), dans le but d'ajuster les sorties d'un système de commande floue à de nouvelles connaissances obtenues, les variables de sorties étant les erreurs sur les variables de sortie du système flou.

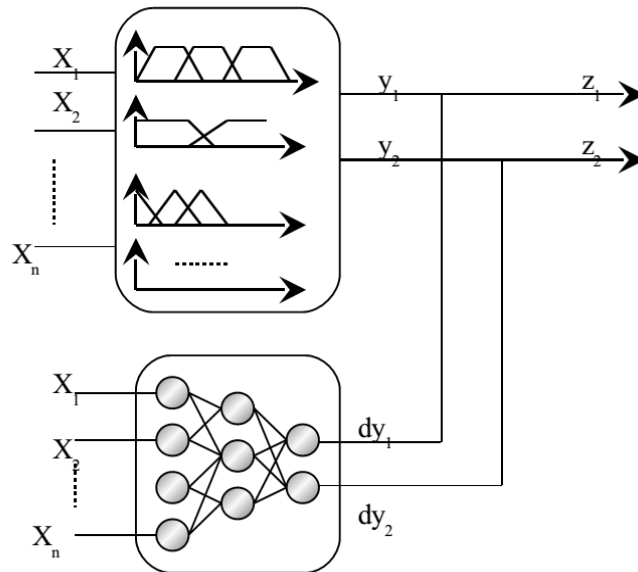


Figure IV.6 : Réseau de neurones fonctionnant en aval d'un système flou

4.3.4.3 Troisième architecture

Ce type d'association entre réseau de neurones et systèmes flous, correspond à l'utilisation des réseaux de neurones pour remplacer toutes ou quelques composantes d'un système flou. De tels réseaux servent à l'apprentissage des fonctions d'appartenance, au calcul de l'inférence, à la réalisation de la phase d'agrégation et de défuzzification. Ils peuvent réaliser l'extraction des règles floues, en analysant la corrélation qui existe entre les entrées et les sorties du réseau de neurones.

Ces approches ont une grande importance car elles sont capables de résoudre des problèmes importants :

- La détermination et l'apprentissage des fonctions d'appartenances.
- La détermination des règles.
- L'adaptation à l'environnement du système.

4.3.5 Les types d'implémentation des réseaux neuro-flous

Dans la littérature il y a une panoplie de systèmes neuro-flou comme : les systèmes neuro-flou coopératives, concurrents, les systèmes dit hybrides ou neuro-flou adaptatif, le modèle NEFCLASS, ANFIS, FALCON et GARIC ...etc [Ould. A. D, 2005], [Mordjaoui. M, 2008].

Selon l'interaction entre la logique floue et les réseaux de neurones, et compte tenu de nos objectifs, nous avons retenu: systèmes d'inférence adaptatifs neuro-flou (ANFIS).

4.3.6 Systèmes d'inférence neuro-flou adaptatifs (ANFIS)

Le système d'inférence adaptatifs neuro-flou ANFIS (Adaptative Network Fuzzy Inference System), développé dans les années 90 par Jang [Jang. J et al, 1993], combine les concepts de la logique floue et les réseaux de neurones pour former un système intelligent hybride qui augmente automatiquement la capacité d'apprentissage et d'adaptation [Lin. C. T et al, 1996]. Des systèmes hybrides ont été employés par des chercheurs pour la modélisation et la prévision dans les divers secteurs de la technologie [Castillo. O, 2000], [Jang. J et al, 1993] [Constantin V. A, 1995], [Abdulkadir Sengur, 2008]. L'idée fondamentale derrière ces techniques adaptatives est de fournir une méthode pour que le procédé de modélisation flou apprenne l'information.

4.3.6.1 Architecture d'ANFIS

La structure ANFIS est constituée de cinq couches comme le montre la figure (IV.30). Pour simplifier la compréhension, nous considérons un système à deux entrées x_1 et x_2 et une sortie y .

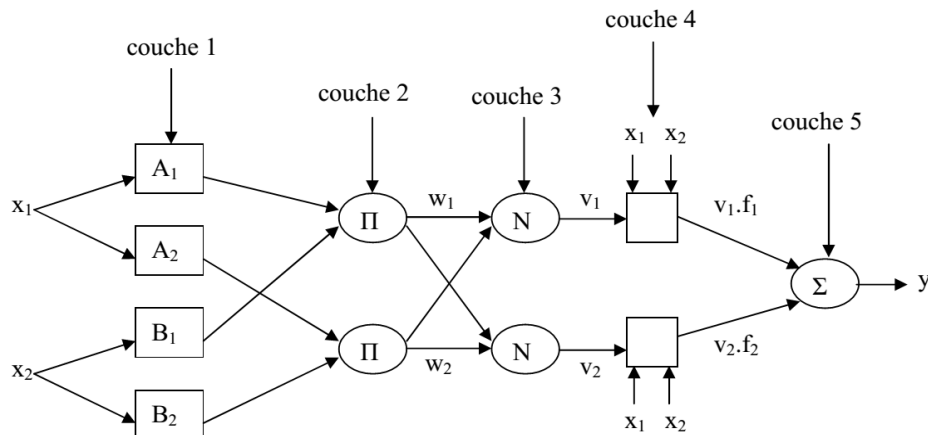


Figure IV.7 : Les différentes couches d'ANFIS

Dans laquelle un cercle indique un nœud fixe, et un carré indique un nœud adaptatif. Les x_1 et x_2 sont les entrées, et y sont les ensembles flous, la sortie dans la région floue spécifique par les règles floues. Parmi différents types de modèle flou, le modèle flou de Sugeno est le plus largement appliqué pour son haute interprétabilité et efficacité de calcul. Pour un modèle flou de Sugeno du premier ordre, un ensemble typique de règles avec deux règles flou (si-alors) peut être exprimé comme :

Rule 1: if (x is A_1) and (y is B_1) then $f_1 = (p_1x + q_1y + r_1)$

Rule 2: if (x is A_2) and (y is B_2) then $f_2 = (p_2x + q_2y + r_2)$

Ou : A_i et B_i sont les positionnements flou dans l'antécédent (Prémises), et p_i, q_i, r_i sont les paramètres de conception qui sont déterminés pendant le procédé d'apprentissage.

L'architecture d'ANFIS se compose de cinq couches, figure (IV.31). la première et la quatrième couche se compose des nœuds adaptatifs tandis que la seconde, la troisième et la cinquième couche se composent de nœuds fixes. Les nœuds adaptatifs sont associés à leurs paramètres respectifs, qui sont mis à jour avec chaque itération, tandis que les nœuds fixes sont exempts de tous paramètres [Constantin. V. A, 1995], [Kim. J. N, 1999], [Lin. C. T, 1996].

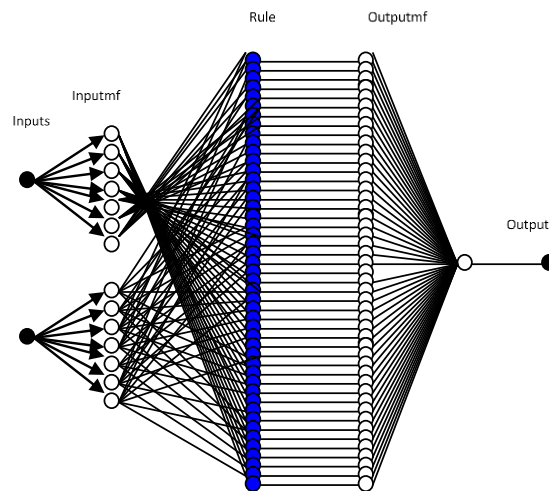


Figure IV.8 : Architecture ANFIS

Couche 1 : Génération du degré d'appartenance :

Chaque nœud de cette couche possède des paramètres ajustables. La fonction du nœud est identique à la fonction d'appartenance à un sous-ensemble flou de l'univers de discours des entrées. Dans le modèle de Jang [JAN.95], les fonctions d'appartenance sont des gaussiennes.

Chaque nœud i dans la première couche utilise une fonction de nœud donnée par :

$$\begin{cases} O_i^1 = \mu_{A_i}(x) & i = 1, 2 \\ O_i^1 = \mu_{B_i}(x) & i = 1, 2 \end{cases}$$

Ou : μ_A et μ_B peuvent être n'importe quelle fonction d'appartenance flou (MF).

Couche 2 : Génération du poids de la règle i :

Chaque nœud circulaire appelé Π dans cette couche correspond à une règle floue Sugeno. Il reçoit les sorties des nœuds de fuzzification et calcule son activation. La conjonction des antécédents est réalisée avec l'opérateur produit.

Chaque nœud dans cette couche calcule la force d'activation d'une règle par multiplication :

$$O_i^2 = w_i = \mu_{A_i}(x) * \mu_{B_i}(y) \quad i = 1, 2$$

Couche 3 : Normalisation des poids de la règle i :

Chaque nœud de cette couche est un nœud circulaire appelé N . Il calcule le degré d'activation normalisé d'une règle floue donnée. La valeur obtenue représente la contribution de la règle floue au résultat final.

Le $i^{\text{ème}}$ nœud dans cette couche calcule le rapport de la force d'activation de la $i^{\text{ème}}$ règle par rapport à la somme des forces d'activation de toutes les règles :

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1, 2$$

Ou : \bar{w}_i représente la forces d'activation normalisée.

Couche 4 : Calcul de la sortie des règles :

Chaque nœud de cette couche est un nœud carré. Dans cette couche, chaque nœud i a la fonction suivante :

$$O_i^4 = \bar{w}_i z_i = \bar{w}_i (p_i x + q_i y + r_i) \quad i = 1, 2$$

Ou : \bar{w}_i est la sortie de la couche 3, et (p_i, q_i, r_i) est l'ensemble des paramètres ajustables de sortie de la règle i .

Couche 5 : Faire la somme de toutes les entrées à partir de la couche 4:

Comprend un seul nœud qui fournit la sortie de l'ANFIS en calculant la somme des sorties de tous les nœuds de défuzzification, c'est-à-dire :

le seul nœud dans cette couche calcule la sortie globale comme somme de tous les signaux entrants, ce qui est exprimée par :

$$O_i^5 = \sum_i \bar{w} z_i = \frac{\sum_{i=1}^2 w_i z_i}{\sum_{i=1}^2 w_i} \quad i = 1, 2$$

La sortie f sur le schéma 2 peut être réécrite comme : [Constantin. V. A, 1995], [Kim. J. N, 1999].

$$z = (\bar{w}_1 x)p_1 + (\bar{w}_1 y)q_1 + (\bar{w}_1)r_1 + (\bar{w}_2 x)p_2 + (\bar{w}_2 y)q_2 + (\bar{w}_2)r_2$$

L'organigramme de la marche à suivre d'ANFIS est affiché sur la figure IV.32.

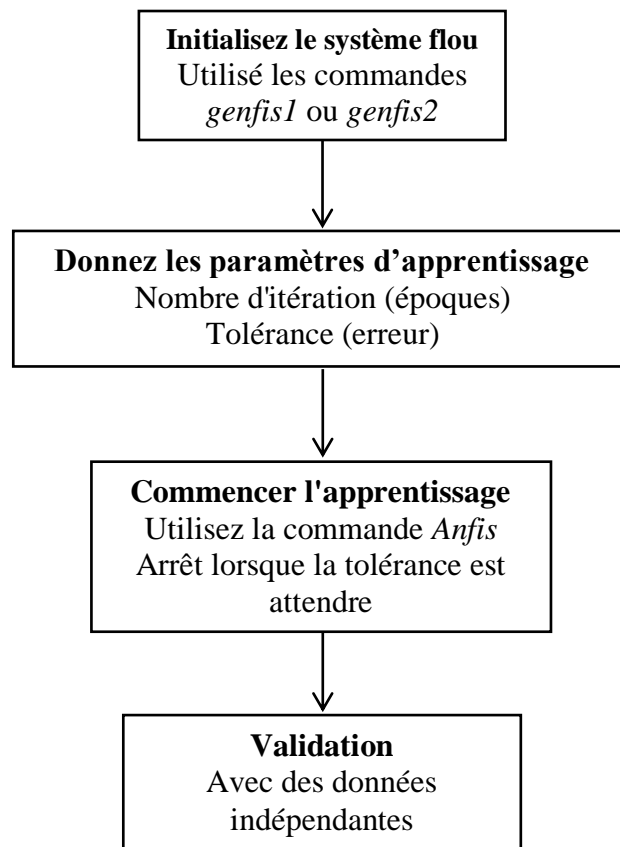


Figure IV.9 Marche à suivre d'ANFIS

4.3.6.2 Algorithme d'apprentissage

L'approche proposée emploie un algorithme hybride c.-à-d. la combinaison de la méthode de descente de gradient de rétro-propagation et de la méthode des moindres carrés. Le procédé d'adaptation des paramètres de l'ANFIS est divisé en deux étapes. Pour la première étape de l'entraînement des paramètres conséquents, la méthode des moindres carrés (LS) est utilisée,

vu que la sortie de l'ANFIS est une combinaison linéaire des paramètres conséquents. Les paramètres de prémisse sont fixés à cette étape. Après ajustement des paramètres conséquents, comme deuxième étape l'erreur d'approximation est rétro propagée (BP) au travers de chaque couche pour mettre à jour les paramètres de prémisse. Cette partie de la procédure d'adaptation est basée sur le principe de la descente du gradient [Zhao. G, 2008], [Anupam. D et al, 2009].

On a utilisé la structure d'ANFIS avec la première modèle de commande de Sugeno. Des fonctions d'appartenance triangulaires sont utilisées et au niveau de la fuzzification une règle d'inférence du type produit. Les paramètres associés aux fonctions d'appartenance vont changer durant le processus d'apprentissage, comme pour un réseau de neurone. L'ajustement est facilité par un vecteur de gradient, qui fournit une mesure de la qualité avec laquelle le FIS modélise les données d'entrée-sortie pour un ensemble donné de paramètres. Une fois que le vecteur de gradient est obtenu, n'importe laquelle des différentes routines d'optimisation pourrait être appliquée pour l'ajustement des paramètres de façon à réduire l'erreur entre les sorties réelles et désirées. Cela permet au système flou d'apprendre à partir des données qu'il modélise. L'avantage de cette approche par rapport au paradigme du pur flou, est que le besoin de l'opérateur humain pour accorder le système en ajustant les limites des fonctions d'appartenance n'est plus nécessaire. Cet article propose une application de l'ANFIS. Les meilleures gammes et formes de fonctions d'appartenance obtenues avec des résultats de simulation de l'ANFIS sont données pour montrer les avantages de la méthode proposée.

4.3.7 Conception de contrôleur basée sur ANFIS

Le modèle utilisé pour les simulations est prise à partir de [Andrew. C, 2000], [Fanping. B, Bin. Y, 2000], avec des modifications mineures pour répondre aux spécificités de notre robot expérimental (figure IV.33). Certains paramètres sont obtenus directement à partir du notre bras manipulateur expérimental (gains de régulation, mesures physiques de longueurs de liaison, etc.), et d'autres à partir de spécifications techniques, et les estimations des paramètres. [Chun-Ta. C, 2012]. Un travail préliminaire a été fait pour l'étalonnage des positions communes, la détermination de la position initiale et les limites de chaque position.



Figure IV.10 : Robot manipulateur hydraulique FEEDBACK HRA934

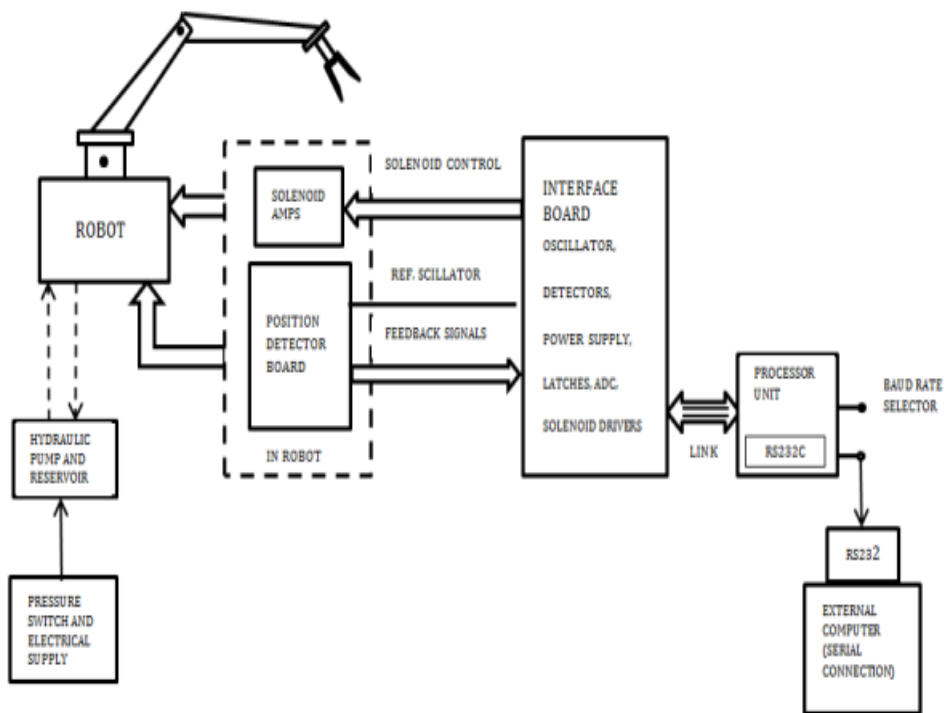


Figure IV.11 : Schéma fonctionnel du bras expérimental

Dans cette étude, il a été décidé de limiter les degrés de liberté de manipulateur utilisé dans les simulations à deux degrés de liberté, à savoir l'épaule et le coude. Les systèmes de contrôle et les conclusions développées ici pour le manipulateur restreint peuvent être étendus à six degrés de liberté. Ce modèle englobe la dynamique de corps rigide des links du robot, la dynamique des actionneurs hydrauliques et de la dynamique de contact [Guang LI, WU Min, 2005]. Le modèle dynamique du robot complet est présenté dans une forme fermée avec le

modèle de l'actionneur hydraulique étant dérivé de telle sorte qu'il est applicable à un manipulateur avec un nombre arbitraire de joints.

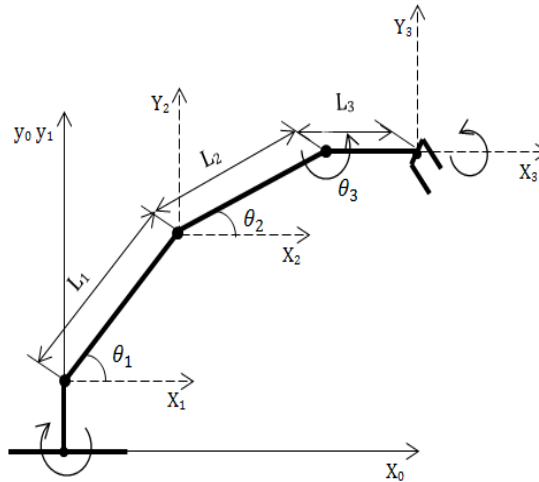


Figure IV.12 : Vue de plan de la configuration HRA934 restreinte

Ce manipulateur est composé de deux liaisons rigides de masses m_1 et m_2 , et des longueurs l_1 et l_2 et les angles des articulations sont respectivement θ_1 et θ_2 .

L'équation (IV.12) décrit le modèle dynamique du manipulateur.

$$\tau = M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) + B(\dot{q}) + {}^c J^T(q) {}^c F \quad (IV.12)$$

$\tau(t)$: est le vecteur $n \times 1$ des couples appliqués par le bras.

$M(q)$: est la matrice d'inertie $n \times n$ bien déterminée positive et symétrique.

$V(\theta, \dot{\theta})\dot{\theta}$: est le vecteur des forces centrifuges et de Coriolis.

$G(q)$: est le vecteur $n \times 1$ des couples de la gravité.

$B(\dot{q})$: est le vecteur $n \times 1$ de frottement agissant à chaque joint.

${}^c J^T(q)$: est le jacobienne du manipulateur.

${}^c F$: est le vecteur $n \times 1$ des forces et couples de l'organe terminale.

q, \dot{q}, \ddot{q} : sont respectivement les vecteurs $n \times 1$ de la position, vitesse et accélération de la communes.

n : correspond au nombre de degrés de liberté du robot.

Le problème de contrôle peut être donné comme : pour une trajectoire désirée liée donnée $q_d, \dot{q}_d, \ddot{q}_d$ concevoir les couples d'entrée de commande tels que les états du robot suivent

leurs références avec tous les signaux impliqués dans la boucle fermée restent bornées : $\lim_{t \rightarrow \infty} q(t) = q_d(t)$.

4.3.7.1 ANFIS basée sur le calcul de couple (type PD)

Le plus commun et l'une des techniques de contrôle non linéaires les plus anciennes pour le contrôle de robots manipulateurs est le contrôle de couple de calcul (CTC) [Weiwei S, Shuang C, 2009], [Zuoshi S et al, 2005], [Tien D et al, 2013], récemment, a là été renouvelé l'intérêt pour ce contrôleur étant utilisé avec les contrôleurs flous. La stratégie du contrôle d'ANFIS consiste à d'ajuster, en permanence, les paramètres des gains de correcteurs. Le schéma de commande du système est représenté sur la figure IV 36.

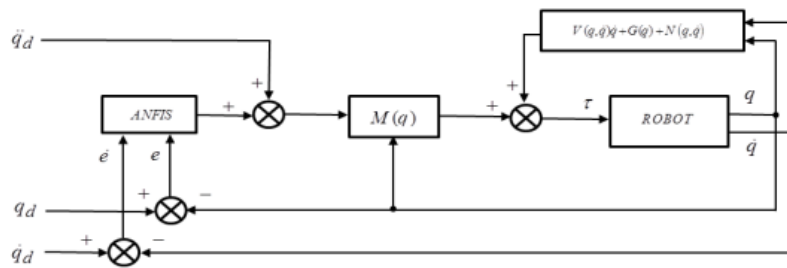


Figure IV.13 : Le schéma fonctionnel global du système de commande

Le contrôle se fait dans l'espace des joints. Chaque joint a son propre contrôleur PD.

Nous choisissons une commande qui stabilise le system, de sorte que l'erreur $e(t)$ tende vers zéro. Nous supposons que les articulations du robot sont reliées ensemble avec des articulations rotoïdes.

Le contrôleur neuro-flou développé possède deux entrées, l'erreur e et la variation de l'erreur \dot{e} , définie comme :

$$\begin{cases} e(t) = q_d(t) - q(t) \\ \dot{e}(t) = \dot{q}_d(t) - \dot{q}(t) \end{cases}$$

Où : $q_d(t)$ et $\dot{q}_d(t)$ Sont respectivement les vecteurs de position et de vitesse désirée.

Pour assurer la linéarisation et le découplage du système non linéaire décrit par l'équation (IV.12) dans la boucle fermée, nous introduisons une commande linéarisant (calcul de couple)

basé sur la connaissance du modèle du robot. Le choix d'un couple appliqué au robot est donnée par:

$$\tau = M(q)\tau_0 + V(q, \dot{q})\dot{q} + G(q) + B(\dot{q}) + {}^c J^T(q) * {}^c F \quad (IV.13)$$

Où : τ_0 est une entrée auxiliaire du contrôleur choisi, avec substitution de (IV.13) (IV.12), on obtient:

$$\ddot{q} = \tau_0 \quad (IV.14)$$

Une commande proportionnelle dérivée (PD) est un choix typique pour τ_0 , donnée par l'équation :

$$\tau_0 = \ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) \quad (IV.15)$$

Par le remplacement de (IV.15) (IV.14), nous obtenons l'équation d'erreur suivant:

$$\ddot{e} + K_d \dot{e} + K_p e = 0$$

K_p et K_v sont des matrices diagonales définies positives, de sorte que le système à boucle fermée linéaire devient découplée.

Cette équation a une solution pour une erreur $e(t)$ qui tend exponentiellement vers zéro. Le système en boucle fermée avec ce contrôleur, où le modèle du robot est connu avec précision, est asymptotiquement stable. Dans le cas d'une connaissance imprécise des paramètres du robot et / ou la présence de certaines dynamiques non modélisées, la commande par calcul de couple montre ses limites. La solution proposé est d'utilisé un système d'inférence adaptative neuro-flou (ANFIS) dont le rôle est d'ajuster, en permanence, les paramètres K_p et K_v pour compenser les parties négligées du modèle dynamique. Ainsi :

$$\tau_0 = \ddot{q}_d + \tau_{ANFIS} \quad (IV.16)$$

$$\tau_{ANFIS} = K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q)$$

Voir application de cette technique dans la section 5.

4.3.7.2 ANFIS directe type PD plus une correction intégrale

Dans cette structure, le système adaptatif neuro- flou n'est appliqué qu'au signal proportionnel et dérivé du contrôleur linéaire PID.

Pour la conception d'ANFIS, une base de données a été créé à partir de diverses trajectoires possibles du notre bras hydraulique simulé (même chose pour les bras hydraulique

expérimentale). Un ensemble de 70% des données ont été utilisées pour l'apprentissage, et le reste utilisé pour le teste. La valeur des gains sont utilisés pour l'entrée et la sortie de l'ANFIS. Ces gains sont choisis par essais erreurs afin d'obtenir une bonne réponse.

$$\tau = \Delta e + \tau_{ANFIS} \quad (IV.17)$$

Δe : C'est l'intégral de l'erreur (équivalent de K_i).

Le contrôle se fait dans l'espace des joints. Chaque joint a son propre contrôleur PID. Le contrôleur ANFIS est pour ajuster les paramètres K_p et K_v , plus une action intégrale conventionnelle (I) pour la précision du suivi de trajectoires. La principale caractéristique de ces contrôleurs est que dans ces systèmes de commande la sortie finale appliquée à la fin est la somme de sortie individuelle de deux contrôleurs. Le schéma fonctionnel global du système de commande est affiché sur la figure (IV.37).

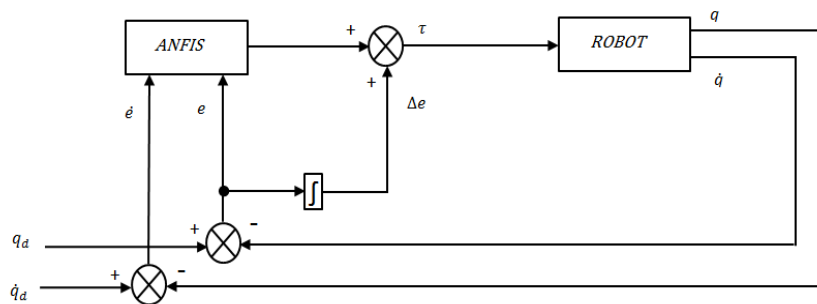


Figure IV.14 : Le schéma fonctionnel global du système de commande

Le signal intégré utilise la méthode linéaire classique. Le but simple du signal intégré est d'éliminer l'erreur d'état stationnaire. La réponse transitoire est affectée principalement par le signal proportionnel et le signal dérivé. Pour l'amélioration de la réponse transitoire, les gains variables sont appliqués sur les parties proportionnelles et dérivées à l'aide de système flou à deux entrées.

Voir application de cette technique dans la section 5.

5. Simulation et validation expérimentale

5.1 Simulation

Nous présenterons dans cette section les résultats de simulation de notre système en utilisant Matlab 7.8 et l'outil de simulation (Neuro-Fuzzy logic toolbox and Robotics Toolbox 9.8 for MATLAB). Dans cette simulation nous utiliserons une trajectoire de référence donnée par :

$$\begin{cases} \theta_1 = \sin(t) \\ \theta_2 = \cos(t) \end{cases}$$

Dans cette section, nous présentons le résultat de comparaison (en simulation) entre les deux techniques basés sur ANFIS (Commande Adaptative Neuro flou basée sur le calcul de couple (type PD), et Commande Adaptative Neuro flou directe type PD plus une correction intégrale). La technique la mieux adaptée à la simulation sera appliqué à notre robot expérimental.

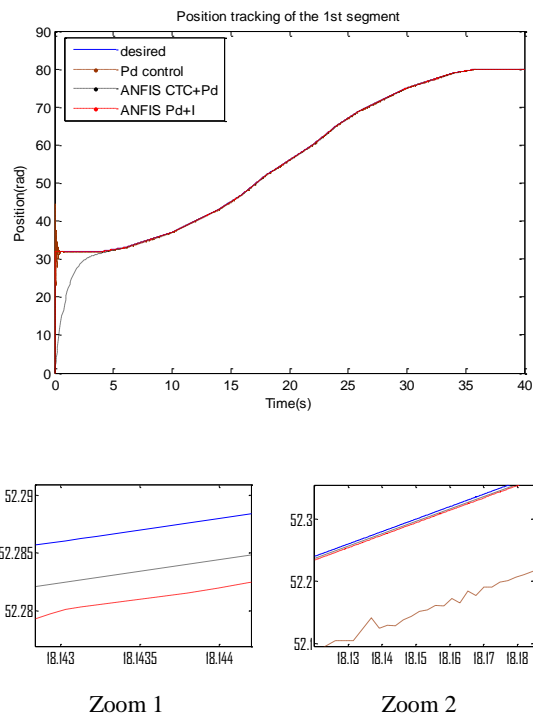


Figure V.1 : Poursuite de position de la première articulation

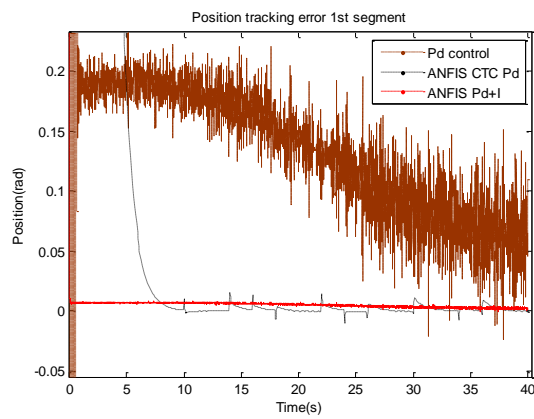
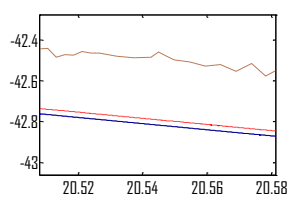
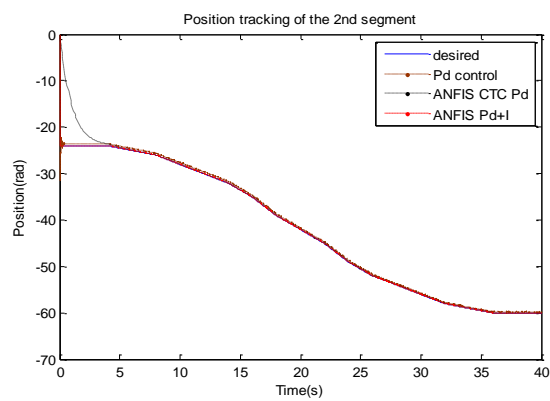
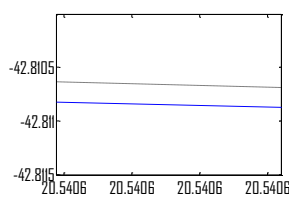


Figure V.2 : Erreur de position de la première articulation



Zoom 1



Zoom 2

Figure V.3 : Poursuite de position de la deuxième articulation

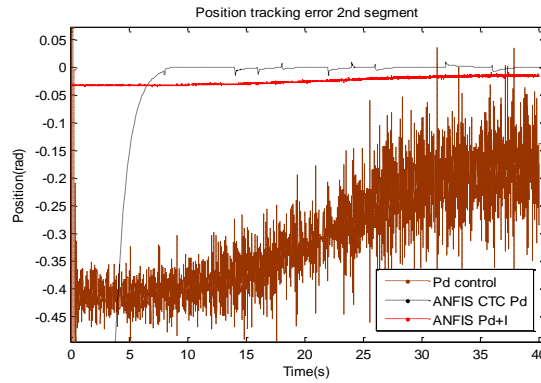


Figure V.4 : Erreur de position de la deuxième articulation

La figure V.1 présente la trajectoire désirée pour le link1 et les résultats des trois techniques de contrôle pour la poursuite en position.

Figure V.2 présente l'erreur de poursuite en position de la première articulation.

Pour contrôleur PD: Pendant près de 1s l'erreur de poursuite est oscillante et grande, à 2s son amplitude est de 0,2. Radian, et commence à diminuer tandis que l'oscillation persiste.

Pour ANFIS CTC + PD: au démarrage l'erreur de poursuite est très grande (environ 30 radian). Ensuite elle commence à diminuer pour suivre la trajectoire de référence après 7s.

Pour ANFIS PD + I: depuis le début l'erreur de poursuite converge asymptotiquement vers 0. Les figures V.3 et V.4 montrent le même comportement.

Comme mentionné ci-dessus, la loi de commande par calcul de couple a la capacité de rendre le système asymptotiquement stable, si la dynamique du robot est connue avec précision (qui n'est pas le cas de notre robot), dans les figures (V.1, V.2, V.3 et V.4) nous pouvons voir qu'au début il y a une grande différence entre la trajectoire de référence et les trajectoires obtenues par le contrôleur PD classique et le contrôleur adaptative neuro flou basée sur le calcul de couple (type PD), et ce n'est pas le cas pour le contrôleur adaptative neuro flou directe type PD plus une correction intégrale, où depuis le début l'erreur de poursuite est proche de zéro.

Ces simulations montrent l'efficacité du contrôle adaptatif neuro flou directe type PD plus une correction intégrale, dans le cas de modèle dynamique incertain en comparaison avec les techniques de commande basées sur le calcul de couple.

5.2 Expérimentale

On prend en considération les résultats obtenus par simulation, nous mettrons en œuvre le contrôleur adaptative neuro flou directe type PD plus une correction intégrale au robot expérimental et on voit si elle donne toujours de bons résultats.

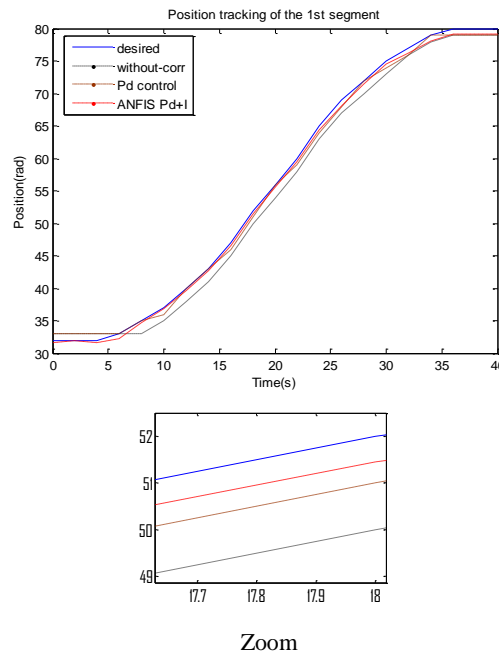


Figure V.5 : Poursuite de position de la première articulation

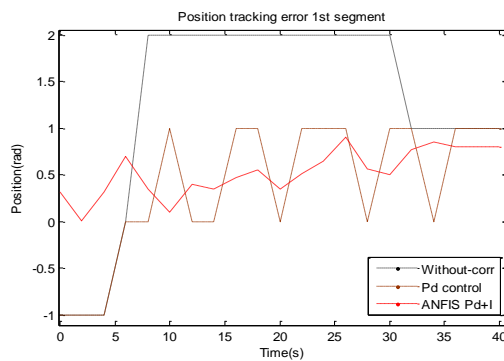


Figure V.6 : Erreur de position de la première articulation

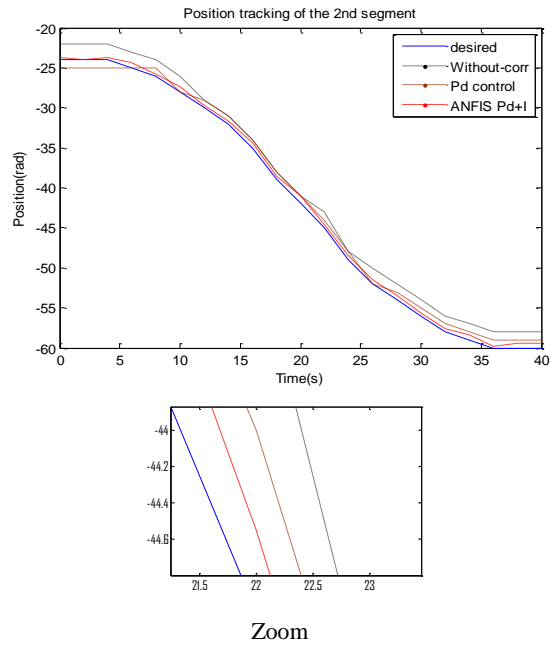


Figure V.7 : Poursuite de position de la deuxième articulation

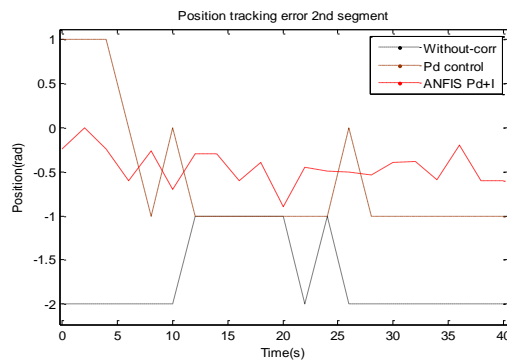


Figure V.8 : Erreur de position de la deuxième articulation

Sur la figure V.5 nous comparons les trois méthodes de poursuite en position, et on cherche la méthode la plus efficace et la plus précise.

Sur la figure V.6 nous remarquons que l'erreur de poursuite en position du système non corrigée se comporte comme un signal triangulaire écrêté et varie par étapes entre -1 et 2 radian, de même pour le régulateur PD, l'erreur de poursuite de position varie entre 0 et 1,3 radian, cela est dû au vannes de régulation avec zone morte et les gains de flux non linéaires. Par contre, l'erreur de poursuite de l'ANFIS directe type PD plus une correction intégrale est faible et varie sans à-coups ou doucement (smoothly) et de façon plus souple sans écrêtage.

Les figures V.7 et V.8 montrent le même comportement.

Compte tenu de l'état de délabrement du robot utilisé et malgré la présence d'actionneurs hydrauliques régulés par des valves de contrôle directionnelles proportionnelles avec zone morte et les flux à gain non linéaire, la technique donne de bons résultats.

Conclusion générale et Perspectives

Conclusion générale et Perspectives

Le contrôle intelligent des systèmes robotiques est un problème difficile parce que la dynamique de ces systèmes est fortement non linéaire et complexes. Il est difficile de les modéliser ou de les commander par des méthodes de l'automatique linéaire, ainsi que la synthèse des différentes commandes des robots manipulateurs par les méthodes conventionnelles est basée principalement sur le modèle dynamique (précis). La disponibilité d'un modèle dynamique précis n'est pas toujours évidente, vu les bruits externes agissant sur le robot, et les incertitudes paramétriques décrivant les propriétés dynamiques. L'apport principal de ce travail réside dans le développement de méthodologies de commande à base des techniques de l'intelligence artificielle, plus précisément l'approches neuro-floues. Cette approche combine les concepts de la logique floue et les réseaux de neurones artificiels pour constituer un contrôleur intelligent hybride appelé ANFIS (Adaptive Neuro Fuzzy Inference System). Ce dernier est utilisé pour la commande d'un bras manipulateur hydraulique à 6 ddl. Nous proposons deux techniques de contrôle basé sur ANFIS : Commande adaptative neuro flou basée sur le calcul de couple (type PD), et commande adaptative neuro flou directe type PD plus une correction intégrale.

Il nous a fallu d'abord définir un modèle mathématique de notre bras manipulateur. Cette modélisation est implémentée dans le chapitre 2 où il y a une présentation des différents modèles utilisés pour décrire le mouvement des articulations d'un bras manipulateur, couvrant ainsi les aspects des modèles géométrique, cinématique et dynamique. Puis l'élaboration du modèle dynamique du robot considéré par l'association des modèles dynamiques mécanique et hydraulique de ce robot. Par la suite, le chapitre 3 présente un état de l'art sur la commande des robots, avec la description détaillée des différentes stratégies de commandes classiques, et l'application en simulation des commandes les plus connues telles que la commande par calcul de couple avec correction PD et avec PID, la commande adaptative par calcul de couple, et la commande adaptative inertielle. Le chapitre 4, présente les outils de l'intelligence artificielle (réseaux de neurones, logique floue et neuro-flou), qui nous permettront de faire une étude comparative (en simulation) de différentes techniques de commande à base de réseaux de neurones, et d'ANFIS, Afin de déterminer la commande qui convient le mieux à la commande du robot expérimentale FEEDBACK HRA934. Parmi les deux méthodes basées sur l'ANFIS la commande adaptative neuro flou directe type PD plus une correction intégrale est implémentée sur le robot expérimental à l'aide du logiciel Matlab/Simulink.

Les travaux présentés dans cette thèse ouvrent un certain nombre de perspectives. A court terme, nous pensons appliquer l'approche d'analyse de la commande utilisée dans cette thèse à un robot plus sophistiqué, et utiliser d'autre combinaison des outils de l'intelligence artificielle.

Annexe

A.1. LOGIQUE FLOU

A.1.1 Introduction

La majorité des systèmes industriels sont complexes et difficiles à contrôler. Ces problèmes sont dus au non linéarité et à la variation des paramètres de ces systèmes, ainsi que la plupart des systèmes non linéaires sont modélisables sous des hypothèses parfois assez restrictives qui peuvent rendre difficiles la mise en œuvre des schémas de commande résultants. Il est donc nécessaire de prendre en compte toutes les informations imprécises et incertaines relatives au système. Afin d'apporter une solution à ces problèmes, une nouvelle stratégie de commande basée sur la "Théorie de la logique floue" ou plus généralement le traitement des incertitudes, a été élaborée [Zadeh. L. A, 65] [HENAO. H et al, 96] [RODRIGUES. M et al, 2001] [A.H. H.AMIN et al, 2002]. L'origine du développement de cette théorie a été le contrôle des processus basé sur l'expertise de l'opérateur.

La théorie des sous-ensembles flous développée par Zadeh en 1965 [Zadeh. L. A, 65], a permis de traiter les imprécisions et les incertitudes. De nombreuses applications sont alors développées dans divers domaines, là où aucun modèle déterministe n'existe ou est difficile à obtenir. Lotfi A. Zadeh en 1965 prétend que la notion d'ensemble flou fournit un point de départ convenable pour la construction d'une structure en parallèle aux ensembles ordinaires, mais plus générale avec un plus grand éventail d'applicabilité surtout dans le domaine de la classification des patrons et de l'analyse de l'information. Une telle structure permet le traitement de problèmes où la source d'imprécision est à l'origine de critères mal définis et non de variables aléatoires- Le terme "imprécision" est utilisée dans le sens d'ambiguïté plutôt qu'un manque d'information sur la valeur d'un paramètre. La théorie des ensembles flous offre donc une structure mathématique dans laquelle des concepts vagues peuvent être précisément et rigoureusement étudiés. Elle peut également être considérée comme un langage de modélisation convenable à des situations caractérisées de relations, critères ou phénomènes flous. Il n'existe pas de définition unique à la notion du flou; elle peut signifier différentes choses selon le contexte d'application et la façon dont elle est mesurée. Cette section a servi pour monter les différentes bases de la logique floue ainsi que le principe de fonctionnement d'un contrôleur à base de cette logique.

A.1.2 La logique floue

La logique floue, depuis les travaux de Lofti Zadeh [Zadeh 1973], a connu un réel succès non seulement dans la modélisation mais aussi dans la commande de systèmes complexes non linéaires [Takagi 1985]. Des applications utilisant les systèmes flous ont été développées dans des domaines variés : traitement d'images [Asgharzadeh 1998] où la logique floue est utilisée pour améliorer les caractéristiques d'images numériques couleur (luminosité, teinte, brillance), robotique mobile [Pagès 1998] où la logique floue intervient au niveau de la commande de robots par simulation. Dans [Titli 1994], les principes de la logique floue sont rappelés et les nombreux domaines d'application sont détaillés (commande, mesure...). Les modèles flous ont la propriété d'approximer n'importe quelle fonction non linéaire [Wang 1994] et l'autre avantage est qu'il est possible de "se passer" d'un modèle explicite du procédé. Cette approche est basée sur deux concepts : celui de la décomposition d'un univers de discours d'une ou de plusieurs variables mesurées sous forme de symboles linguistiques : «petit», «moyen», «grand»... et des règles provenant de l'expertise de l'opérateur humain, qui expriment, là aussi, sous forme d'un langage, comment doivent évoluer les commandes du système en fonction de la ou des variables observées :

«Si l'erreur est positivement grande et la variation de l'erreur est positivement grande Alors la variation de la commande est très négative».

En outre, les systèmes flous ont été classés en trois groupes : les systèmes flous linguistiques ou systèmes de Mamdani, les systèmes flous relationnels et les systèmes à conséquence fonctionnelle ou encore connus sous le nom de systèmes flous de type Takagi-Sugeno-Kang (TSK). Les contrôleurs flous de type TSK sont quasiment omniprésents dans la littérature relative à la commande floue mais nous retrouvons aussi les systèmes flous de Mamdani avec l'introduction des régulateurs flous à structure PID. Dans [Procyk 1979], Procyk et Mamdani introduisent le SOC : Self-Organized Controller, capable de contrôler une large variété de systèmes non linéaires. Le SOC, avec des règles linguistiques, observe l'environnement tout en fournissant les lois de commande appropriées ; il utilise les résultats de ces commandes afin de les améliorer par la suite. De nombreux travaux portent sur ces deux types de contrôleurs mais citons les ouvrages de référence : [Driankov 1993], [Palm 1997]. Cependant, l'obtention de ces contrôleurs n'est pas forcément triviale : le choix judicieux des conclusions des contrôleurs flous de type TSK n'est pas toujours évident. De plus, des problèmes liés à la stabilité sont vite apparus et des méthodes qualitatives et géométriques ont été développées. Le lecteur pourra se référer à [Kandel 1999] qui analyse en détail les différentes approches de

la stabilité des systèmes flous et à [Sugeno 1999] qui traite la stabilité des systèmes flous avec des conclusions de type singleton dans le cas continu et le cas discret.

A.1.3 Vocabulaire de la logique floue

La logique floue est une branche des mathématiques, et à ce titre toute une série de notions fondamentales sont développées. Ces notions permettent de justifier et de démontrer certains principes de base. Dans ce qui suit, on ne retiendra que les éléments indispensables à la compréhension du principe du réglage par la logique floue.

A.1.3.1 Ensemble Flou

Dans un ensemble de référence E, depuis [Zadeh 65] et [Kaufmann 73], un sous-ensemble flou de ce référentiel E est caractérisé par une fonction d'appartenance μ de E dans l'intervalle des nombres réels [0, 1] (degré d'appartenance qui est l'extension de la fonction caractéristique d'un sous-ensemble classique). En fait un sous-ensemble flou (nous dirons plus brièvement un ensemble flou) est formellement défini par l'application μ , mais pour se ramener au langage des mathématiques classiques, nous parlerons d'un ensemble flou A, et noterons μ_A sa fonction d'appartenance. La notion de sous-ensembles flous a pour but de permettre des gradations dans l'appartenance d'un élément à une classe, c'est-à-dire d'autoriser un élément à appartenir plus ou moins à une classe prédéfinie.

Soient U: L'univers du discours.

A: un sous-ensemble de U.

- Théorie classique des ensembles:

Si μ_A est la fonction d'appartenance de l'ensemble A

$$\forall x \in U \quad \mu_A(x) = 0 \quad \text{si } x \notin A$$
$$\mu_A(x) = 1 \quad \text{si } x \in A$$

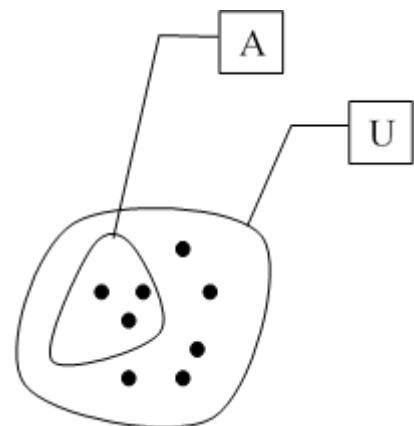
- Concept d'ensemble flou:

Si μ_A est la fonction d'appartenance de l'ensemble flou

$$\forall x \in U \quad \mu_A(x) \in [0; 1]$$

Si $\mu_A(x) = 0,30$

x appartient à l'ensemble flou A avec un degré d'appartenance de 30%



Donc :

- Degré d'appartenance = valeur de vérité.
- Un ensemble flou est totalement déterminé par sa fonction d'appartenance.

A.1.3.2 Représentation des sous-ensembles flous

Soit une variable x (la taille) et un univers de référence ou de discours U (les individus).

Un sous-ensemble flou A est défini par une fonction d'appartenance $\mu_A(x)$ qui décrit le degré avec lequel l'élément x appartient à A .

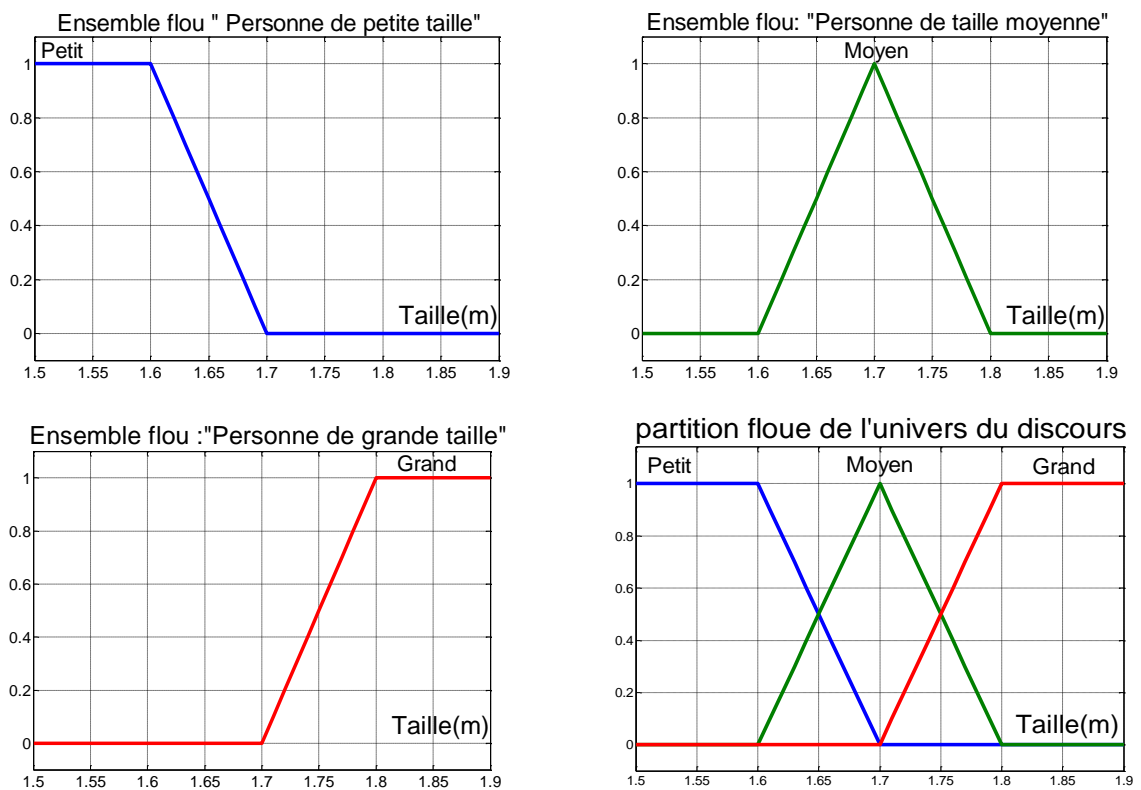


Figure A.1 : Exemples d'ensembles flous

On parle de variable linguistique taille et de valeurs linguistiques petit, moyen et grand.

A.1.3.3 Variable et valeurs linguistique

Contrairement aux variables binaires qui sont définies par les deux états « vrai » ou « faux », les variables floues présentent toute une gradation entre la valeur « vrai » et la valeur « faux ». La description d'une certaine situation imprécise ou incertaine peut contenir des expressions floues comme par exemple : très grand, grand, moyen, petit. Ces expressions forment les

valeurs d'une variable \tilde{v} , appelée "linguistique", soumise à des fonctions appelées fonctions d'appartenance, donc une variable linguistique sert à modéliser les connaissances imprécises ou vagues sur une variable dont la valeur précise peut être inconnue.

Exemple :

Logique floue est basée sur des variables floues dites variables linguistiques à valeurs linguistiques dans l'univers du discours U.

Chaque valeur linguistique constitue alors un ensemble flou de l'univers du discours.

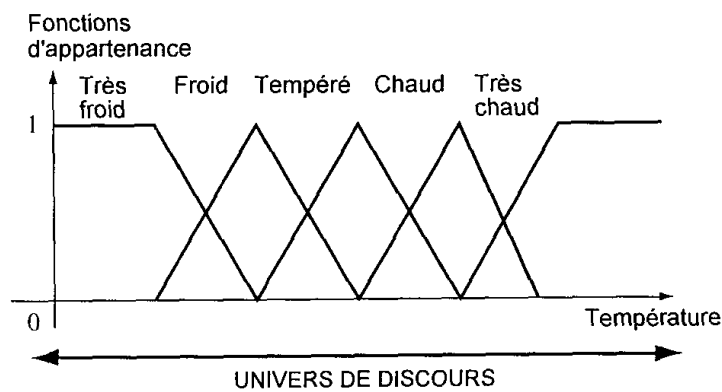


Figure A.2 : Exemple de variables floues

Univers du discours : Gamme de température de 0°C à 200°C.

Variable linguistique : La température.

Valeurs linguistiques : « Très froid » « Froid » « Tempéré » « Chaud » « Très Chaud »

A.1.3.4 Fonctions d'appartenance

Un ensemble flou est caractérisé par une fonction d'appartenance $\mu_E(\tilde{v})$ prenant des valeurs dans l'intervalle $[0,1]$ qui détermine le degré d'appartenance, aussi appelé possibilité ou coefficient d'appartenance. Soit la variable \tilde{v} varie dans un univers de discours, ce dernier est partagé en sous-ensembles flous de façon que dans chaque zone il y ait une situation dominante. Ces zones sont décrites par des fonctions convexes, généralement sous forme triangulaire ou trapézoïdale, elles admettent comme argument la position de la variable \tilde{v} dans l'univers de discours, et comme sortie le degré d'appartenance de \tilde{v} à la situation décrite par la fonction notée :

$\mu_E(\tilde{v})$: Degré d'appartenance de \tilde{v} au sous ensemble E.

Le choix de la répartition des fonctions, leurs chevauchements ainsi que leurs formes doit être judicieux comme indiqué dans [Büh, 94]. La figure IV.1 donne quelques fonctions d'appartenance.

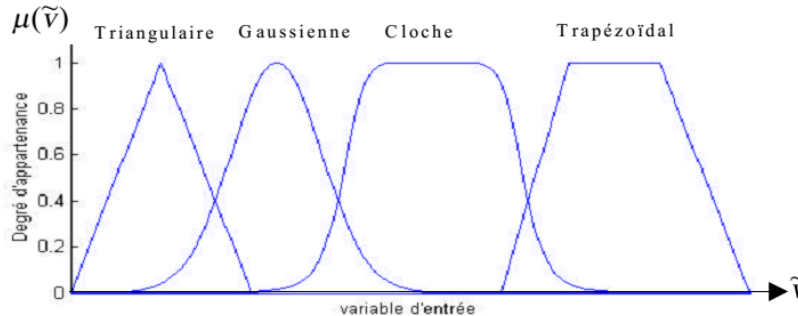


Figure A.3 : Exemple de fonctions d'appartenance

A.1.3.5 Les opérateurs flous

La description d'une situation où il y a plus qu'une variable qui intervient, nécessite l'utilisation des opérateurs logique tels que : "ET", "OU", et "NON" qui dans la théorie de la logique floue peuvent être interprétés par les opérations respectivement : "Minimum", "Maximum" et "complément à un".

La logique booléenne standard est un cas particulier de la logique floue, tous les résultats obtenus en logique classique doivent être retrouvés par la logique floue.

A est l'ensemble flou des personnes petites.

B est l'ensemble flou des personnes moyennes.

Comme pour la théorie classique des ensembles, on définit la réunion, l'intersection, le complément....d'ensembles flous.

La réunion : L'ensemble des personnes petites OU moyennes est un ensemble flou de fonction d'appartenance :

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad \forall x \in U$$

L'intersection : L'ensemble des personnes petites ET moyennes est un ensemble flou de fonction d'appartenance :

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad \forall x \in U$$

Le complément : L'ensemble des personnes NON petites est un ensemble flou de fonction d'appartenance :

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \forall x \in U$$

A.1.3.6 Propriétés des ensembles flous

Comme dans le cas des ensembles «classiques», les ensembles flous possèdent certaines propriétés.

Commutativité: $A \cup B = B \cup A$, $A \cap B = B \cap A$

Associativité : $A \cup (B \cap C) = (A \cup B) \cap C$, $A \cap (B \cup C) = (A \cap B) \cup C$

Distributivité : $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Idempotence : $A \cup A = A$, $A \cap A = A$

Identité : $A \cup \varnothing = A$, $A \cup 1_U = 1_U$, $A \cap \varnothing = \varnothing$, $A \cap 1_U = A$

Deux exceptions notables :

Les deux propriétés suivantes ne sont pas « classiques » :

- L'intersection d'un ensemble flou et de son complément n'est pas vide

Loi de contradiction : $A \cap \bar{A} \neq \varnothing$

- L'union d'un ensemble flou et de son complément ne donne pas l'univers du discours

Loi du "excluded middle": $A \cup \bar{A} \neq U$

A.1.4 La commande floue

Lorsque le nouveau concept de la logique floue a été proposé par Zadeh, ses premières applications étaient dans le domaine du contrôle des systèmes fait par des experts humains, mais il déborde maintenant ce cadre par ses multiples applications, partout où une modélisation mathématique est difficile.

Le contrôle flou utilise la logique floue comme une démarche qui peut couvrir la stratégie du contrôle linguistique. Il est intégré dans la partie qui gère les données de commandes et de contrôle de la partie opérative du système, appelée contrôleur flou (figure A.4).

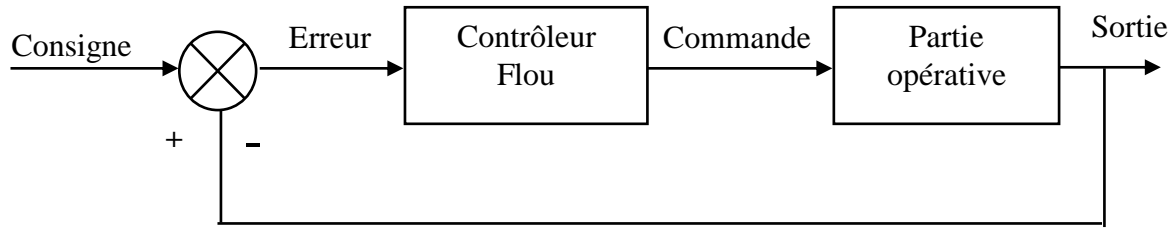


Figure A.4: Système à commande floue

Les contrôleurs flous sont des systèmes experts basés sur des règles Si-Alors et pour lesquelles les prémisses et les conclusions sont exprimées à l'aide de termes linguistiques. La proximité de ces règles au langage naturel dote ces systèmes flous d'une grande lisibilité et facilite ainsi l'introduction de toute connaissance à priori dans la base de règles.

Par opposition aux contrôleurs classiques, les contrôleurs flous ne nécessitent pas de modèles analytiques de la tâche à résoudre. Leur mise au point repose généralement sur l'expression, dans la base de règles linguistiques, de la logique suivie par les experts du domaine pour prendre des décisions.

Pourquoi un contrôle flou ?

- Un contrôleur standard (PID ou autres) demande toujours un modèle le plus précis possible (équations différentielles).
- Un contrôleur flou ne demande pas de modèle du système à régler. Les algorithmes de réglage se basent sur des règles linguistiques de la forme Si ... Alors ...

En fait, ces règles peuvent être exprimées en utilisant le langage de tous les jours et la connaissance intuitive d'un opérateur humain. Ce qui conduit à deux avantages :

- Le contrôle flou reste CLAIR pour tous les opérateurs et utilisateurs de la machine.
- Les opérateurs peuvent aisément interpréter les effets ou conséquences de chaque règle.

Donc : pour les systèmes mal connus ou complexes (non-linéaires), un contrôleur flou s'avère très intéressant.

A.1.5 Caractéristiques de la commande floue

Son but est de traiter les problèmes de commande de processus avec une approche différente de l'automatique classique. Le plus souvent elle se sert de la connaissance des experts du processus. La spécificité de la commande floue réside dans les points suivants :

- ✚ La connaissance mathématique du fonctionnement du processus n'est pas nécessaire. C'est le savoir-faire de l'opérateur qui est pris en compte.
- ✚ Des variables subjectives sont utilisables. Il est possible de modéliser les sens humains.

L'intérêt de la commande floue apparaît clairement dans le cas de système mal connu ou difficile à décrire. De même manière lorsque les variables sont décrites de façon imprécise ou en langage naturel. Intérêt de la commande floue :

- ✚ La commande est simple à réaliser, facilement adaptable aux conditions de fonctionnement. La plupart du temps un petit nombre de règle suffit à décrire le système.
- ✚ La coordination de plusieurs objectifs est possible (système multi variable).
- ✚ Cette commande est reconnue comme robuste.

A.1.6 Structure de base d'un contrôleur flou

Contrairement aux techniques de réglage classique, le réglage par la logique floue n'utilise pas des formules ou des relations mathématiques bien déterminées ou précises. Mais, il manipule des inférences avec plusieurs règles floues à base des opérateurs flous ET, OU, ALORS,...etc, appliquées à des variables linguistiques.

On peut distinguer trois parties principales constituant la structure d'un régulateur flou (voir figure A.5) [Dell'Aquila. A et al, 2003], [Mamdani. E, 75] et [Yan. J, 94]:

- Une interface de fuzzification.
- Un mécanisme d'inférence.
- Et une interface de Defuzzification.

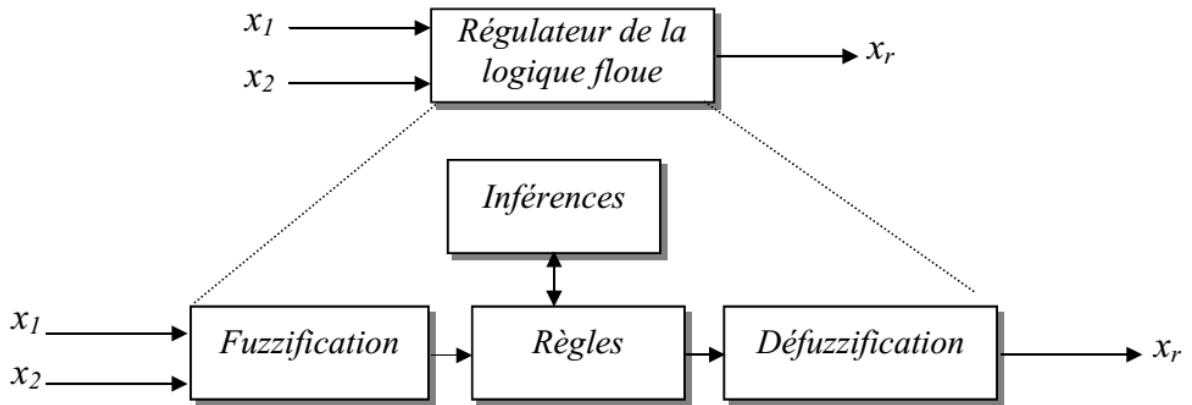


Figure A.5 : Structure interne d'un régulateur de la logique floue

La figure (A.5) représente, à titre d'illustration la structure d'un régulateur flou à deux entrées et une sortie : ou x_1 et x_2 représentent les variables d'entrée, et x_r celle de sortie ou la commande.

A.1.7 Principe

La résolution d'un problème par la logique floue comprend trois étapes :

- 1) la quantification 'floue' des entrées du système, procéder à la partition en sous-ensembles flous des différents univers de discours que le système impose, et transformer les variables réelles en variables floues (fuzzification).
- 2) l'établissement des règles liant les sorties aux entrées à base des règles floues afin de déterminer la base de règle qui va caractériser le fonctionnement désiré du système, donc utiliser les variables floues dans un mécanisme d'inférence qui crée et détermine les variables floues de sortie, en utilisant des opérations sur les fonctions d'appartenance.
- 3) la combinaison de ces règles pour la génération des sorties floues qui consiste à extraire une valeur réelle de sortie de la fonction d'appartenance du sous-ensemble flou de sortie établi par le mécanisme d'inférence (défuzzification)

La conception d'un régulateur flou doit donc passer par plusieurs étapes illustrées sur la figure de la page suivante :

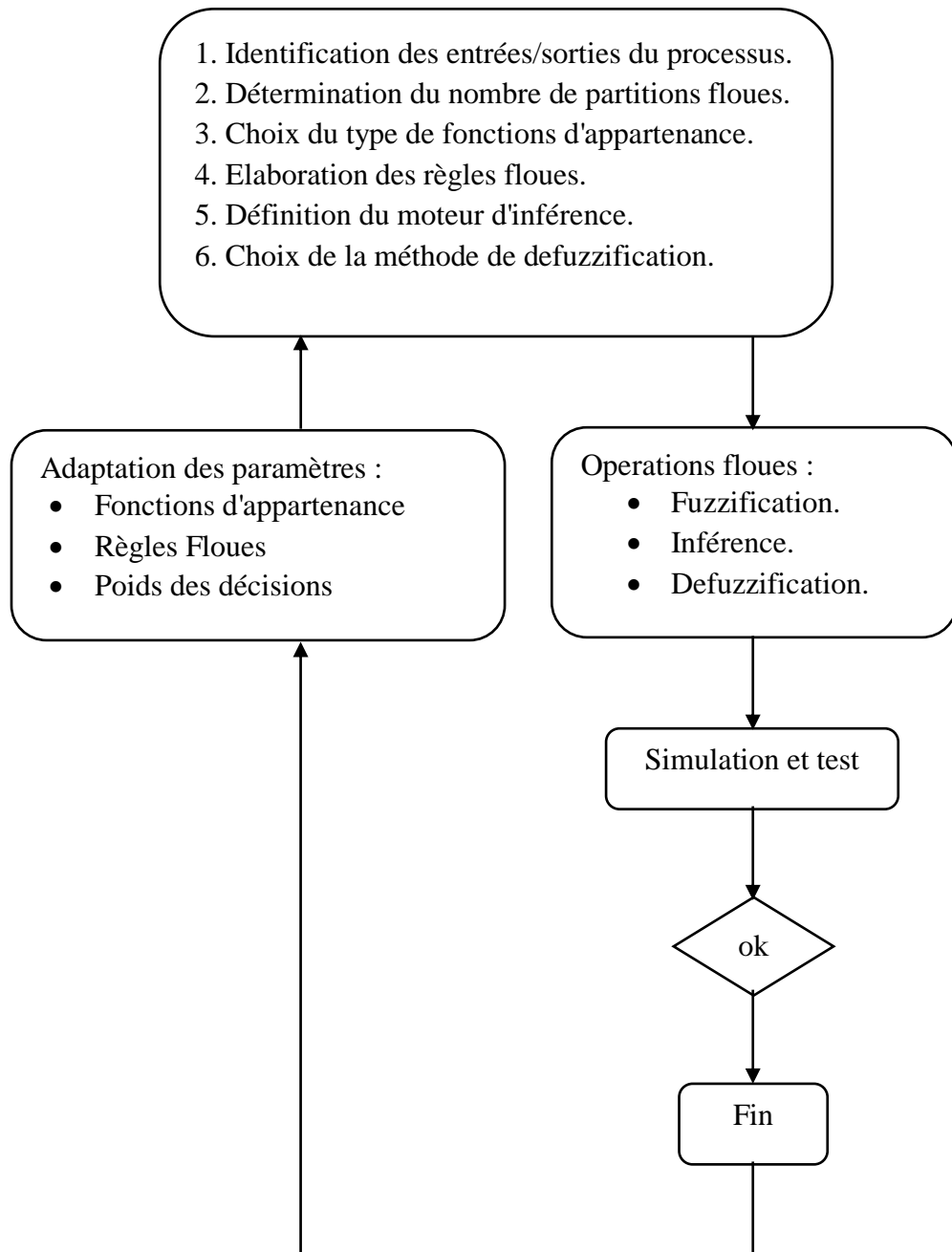


Figure A.6 : Conception d'un régulateur flou

A.1.7.1 Fuzzification

C'est une opération qui consiste à transformer les données numériques d'un phénomène à des valeurs linguistiques sur un domaine normalisé qui facilite le calcul. Donc la fuzzification proprement dite consiste à définir des fonctions d'appartenances pour les différentes variables, et réalisant le passage d'une grandeur physique à une grandeur linguistique [Ying. H, 2000].

A partir de ces domaines numériques appelés univers de discours et pour chaque grandeur d'entrée ou de sortie, on peut calculer les degrés d'appartenance aux sous-ensembles flous de la variable linguistique correspondant [Dell'Aquila. A et al, 2003].

- **Comment fuzzifier**

Pour fuzzifier, il faut donner:

- L'univers du discours.
i.e.: Plage de variations possibles de l'entrée considérée.
- Une partition en classe floue de cet univers.
- Les fonctions d'appartenances de chacune de ces classes.

Exemple :

La fuzzification assure la transformation des grandeurs physiques (entrées) en valeurs linguistiques qui peuvent être traitées, ensuite, par le mécanisme d'inférence et ce grâce aux fonctions d'appartenances des valeurs linguistiques concernées par la valeur physique. Dans la figure A.7 un exemple de fuzzification :

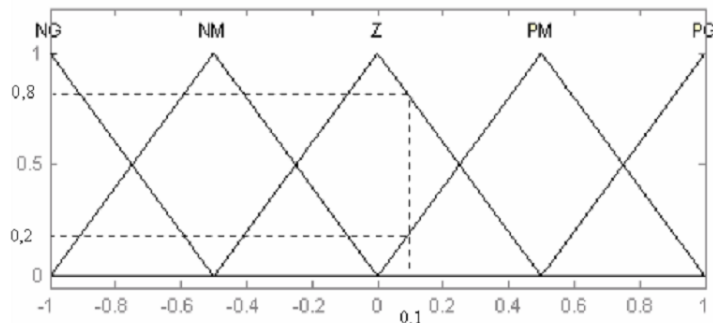


Figure A.7: Exemple de fuzzification

Là où : négative grande (NG), négative moyenne (NM), zéro (Z), positive moyenne (PM), positive grande (PG).

Par exemple si l'erreur physique en entrée est égale à 0.1 Après fuzzification l'erreur est égale à Z avec un degré 0.8 et est égale aussi à PM avec un degré 0.2.

A.1.7.2 Mécanisme d'inférence floue

Cette étape consiste à relier les variables physiques d'entrée du régulateur (grandeurs mesurées ou estimées), qui sont transformées en variables linguistiques pendant l'étape de

fuzzification ; à la variable de sortie du contrôleur sous sa forme linguistique, par des règles mentales traduisant une action ou une décision linguistique sur la commande à la sortie du régulateur, face à toute situation se présentant à l'entrée de ce régulateur [Dell'Aquila. A et al, 2003] [Schupbach. R. M, 2003]. Ces inférences sont basées sur plusieurs règles établies par l'expertise et le savoir-faire humain concernant le système à régler. Elles sont structurées sous forme compacte dans une matrice multidimensionnelle dite matrice d'inférence. On exprime les inférences généralement par une description linguistique et symbolique à base de règles pré définies dans la matrice d'inférence.

Base de connaissances :

La base de connaissances comprend les connaissances de l'expert humain pour le contrôle du système ainsi que le domaine de variation des variables d'E/S. Elle est donc constituée de :

- **Base de données :**

On regroupe dans ce bloc, l'ensemble des définitions utilisées dans les règles du contrôleur flou (univers du discours, partition flou, choix des opérateurs,...)

- **La base des règles floues**

La base de l'expert est généralement exprimée par des règles de la forme ‘‘ SI – ALORS ‘’. La base de des règles est donc une collection de règles floues :

Règle : Si x_1 est (Prémisses) et ... et x_n est (Prémisses) alors y (Conclusion).

Exemple : Soit deux entrées x_1 et x_2 et une sortie x_R , toutes deux définies par les 5 sous-ensembles de l'exemple précédent.

Description de la base de règles :

Si (x_1 NG ET x_2 EZ), Alors x_R PG ou
Si (x_1 NG ET x_2 PM), Alors x_R PM ou
Si (x_1 NM ET x_2 EZ), Alors x_R PM ou
Si (x_1 NM ET x_2 PM), Alors x_R EZ ou
Si (x_1 NM ET x_2 PG), Alors x_R NM ou

...

Si (x_1 PG ET x_2 EZ), Alors x_R NG.

Chaque règle est composée d'une condition précédée du symbole 'SI' appelée prémisse, et d'une conclusion (action, décision, opération ou commande) précédée du symbole 'ALORS'.

Représentations sous forme de tableau ou matrice :

Tableau A.1 : Tableau ou matrice des règles

X_R		X_1				
		NG	NM	EZ	PM	PG
X_2	NG		PG	PM		
	NM			PM	EZ	NM
	EZ	PG	PM	EZ	NM	NG
	PM	PM	EZ	NM		
	PG		NM	NG		

On n'est pas obligé de compléter toute la table. Les règles sont élaborées par un expert et sa connaissance du problème ...

Composition de règles :

Sachant que pour un état donné des entrées, plusieurs règles peuvent être validées pour fournir des consignes aux sorties, il faut disposer d'une méthode de composition pour obtenir le résultat final. Cette tâche appelée agrégation fait correspondre à chaque règle un degré qui dépend de sa prémisse. La méthode d'inférence Max – Min utilise l'agrégation comme une première étape pour calculer les ensembles flous de la sortie. Il existe plusieurs méthodes d'inférence floue qui dépendent des implications de la forme des fonctions.

L'ensemble des règles se présente sous la forme d'une énumération du type :

Si condition 1 et/ou condition 2 (et/ou...) **alors** action sur les sorties.

Si condition 3 et/ou condition 4 (et/ou...) **alors** action sur les sorties.

Si condition 5 et/ou condition 6 (et/ou...) **alors** action sur les sorties.

La combinaison de ces différentes règles se fait à l'aide de l'opérateur ou. La justification du choix de l'opérateur se fonde sur la pratique du langage courant : en effet, une telle énumération est comprise dans le sens :

Si... alors...

Ou

Si... alors...

Ou

...

Bien que l'opérateur **ou** ne soit pas explicitement mentionné.

✚ Activation des règles :

Une règle est activée dès qu'elle a une prémisse ayant une valeur de vérité non nulle. Plusieurs règles peuvent être activées simultanément et préconiser des actions avec différents degrés de validités; ces actions peuvent être contradictoires. Il convient d'agréger les règles pour fournir une appartenance de la variable floue de sortie à une classe floue consolidée.

✚ Moteur d'Inférence :

En utilisant la base de connaissance et la logique floue, le moteur d'inférence appelée aussi Unité de raisonnement, qui représente le cerveau du contrôleur flou, détermine les commandes floues à appliquer au système.

La fonction principale d'un moteur d'inférence est de déterminer la valeur globale de la variable de sortie de contrôle calculée à partir de la contribution de chaque règle. Le moteur d'inférence est constitué de deux fonctions principales qui sont L'inférence floue et l'agrégateur de règles [Foudil. A, 2003].

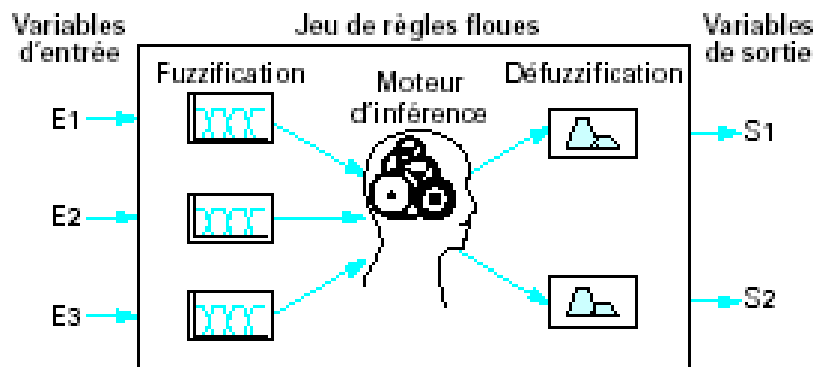


Figure A.8: Moteur d'inférence ou l'unité de raisonnement

A partir des entrées floues et les différentes règles activées cette unité génère une conclusion.

Le traitement numérique des règles d'inférence qui permet d'obtenir la sortie linguistique ou floue du régulateur se fait par différentes méthodes, on cite principalement [Zhongfu. Z et al, 2004]:

- la méthode d'inférence : max-min.
- la méthode d'inférence : max-prod.
- et la méthode d'inférence : somme-prod.

Chacune de ces trois méthodes utilise un traitement numérique propre des opérateurs de la logique floue [Schupbach. R. M, 2003]:

- ✚ Pour la méthode d'inférence Max-Min : l'opérateur ET est réalisé par la formation du minimum, l'opérateur OU est réalisé par la formation du maximum, et ALORS, (l'implication) est réalisée par la formation du minimum.

Au niveau de la condition : $ET \rightarrow \text{Min}$ $OU \rightarrow \text{Max}$

Au niveau de la conclusion : $ou \rightarrow \text{Max}$ $Alors \rightarrow \text{Min}$

- ✚ Pour la méthode d'inférence Max-Produit : l'opérateur ET est réalisé par la formation du produit, l'opérateur OU est réalisé par la formation du maximum, et ALORS (l'implication) est réalisée par la formation du produit.

Au niveau de la condition : $ET \rightarrow \text{Min}$ $OU \rightarrow \text{Max}$

Au niveau de la conclusion : $ou \rightarrow \text{Max}$ $Alors \rightarrow \text{Prod}$

- ✚ Pour la méthode d'inférence Somme-Produit : on réalise au niveau de la condition, l'opérateur OU par la formation de la somme (valeur moyenne), et l'opérateur ET par la formation du produit. Pour la conclusion, l'opérateur ALORS est réalisé par un produit.

Au niveau de la condition : $ET \rightarrow \text{Prod}$ $OU \rightarrow \text{Somme}$

Au niveau de la conclusion : $ou \rightarrow \text{Somme}$ $Alors \rightarrow \text{Prod}$

Dans le cas de la méthode somme-produit, l'action des différentes règles sont liées entre elles par l'opérateur OU qui est réalisé par la formation de la moyenne arithmétique (somme moyenne).

Généralement il existe deux méthodes principales d'implication floue:

➤ Méthode de Mamdani: $\mu'_{conclusion}(y) = \underset{y}{MIN} (\mu_{prémisse}(x_0), \mu_{conclusion}(y))$

Chaque règle floue i est résolue seule par : $\mu_{S_i} = \min(\mu_A(e), \mu_B(\Delta e), \mu_C(z))$

L'ensemble flou solution est décrit par : $\mu_S(c) = \max(\mu_{S_1}(c), \mu_{S_2}(c), \dots, \mu_{S_n}(c))$

A, B et C représentent ici les valeurs linguistiques comme NG, NM, Z...

➤ Méthode de Larsen : $\mu'_{conclusion}(y) = \mu_{prémisse}(x_0) \times \mu_{conclusion}(y)$

Chaque règle floue i est résolue seule par : $\mu_{S_i} = \min(\mu_A(e), \mu_B(\Delta e), \mu_C(z))$

L'ensemble flou solution est décrit par : $\mu_S(c) = \max(\mu_{S_1}(c), \mu_{S_2}(c), \dots, \mu_{S_n}(c))$

A, B et C représentent ici les valeurs linguistiques comme NG, NM, Z...

A.1.7.3 Interface de défuzzification

Les méthodes d'inférence fournissent une fonction d'appartenance résultante pour la variable de sortie. Il s'agit donc d'une information floue qu'il faut transformer en grandeur physique. La commande nécessitant un signal précis, il faudra donc transformer la fonction d'appartenance résultante obtenue à la sortie du moteur d'inférence en une valeur précise. Cette opération est appelée défuzzification [Nabila .B, 2005]. Parmi les méthodes utilisées dans la littérature [Buh, 94], [Pas, 98], [Yin, 00], on peut citer :

- Méthode du maximum.
- Méthode de la moyenne des maxima.
- Méthode du centroïde.
- Méthode de la somme pondérée.

.Méthode du maximum : La sortie correspond à l'abscisse du maximum de la fonction d'appartenance résultante. Cette méthode simple et rapide mais elle introduit des ambiguïtés et une discontinuité de la sortie.

Méthode de la moyenne des maxima :

Dans le cas où plusieurs sous-ensembles auraient la même hauteur maximale, on réalise leur moyenne. Une des ambiguïtés de la méthode du maximum est donc supprimée.

Méthode du centroïde :

La sortie correspond à l'abscisse du centre de gravité de la surface de la fonction d'appartenance résultante. Il existe deux méthodes :

- On prend l'union des sous-ensembles flous de sortie et on en tire le centroïde global (calculs très lourds).
- On prend chaque sous-ensemble séparément et on calcul son centroïde, puis on réalise la moyenne de tous les centroïdes.

Dans cette on n'a plus de discontinuités et d'ambiguïtés, mais cette méthode est plus complexe et demande des calculs plus importants.

Méthode de la somme pondérée :

Compromis entre les deux méthodes précédentes. On calcule individuellement les sorties relatives à chaque règle selon le principe de la moyenne des maxima, puis on réalise leur moyenne pondérée.

- ❖ Dans la plus part des cas les deux méthodes principales de défuzzification :

Méthode du centre de gravité (COG) :

En commande floue, la défuzzification COG est presque toujours utilisée. Elle prend en compte l'influence de l'ensemble des valeurs proposées par la solution floue.

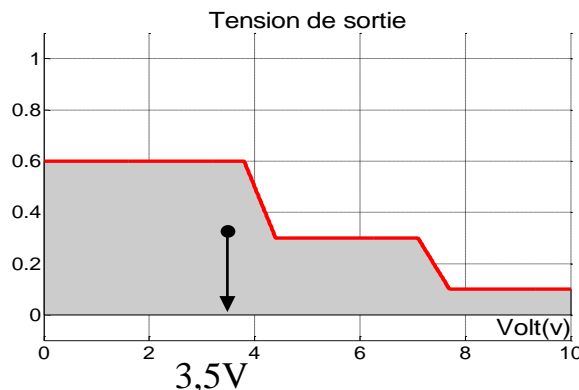


Figure A.9: La défuzzification par la méthode du centre de gravité (COG)

C'est l'abscisse du centre de gravité de la surface sous le courbe résultat.

$$\text{sortie} = \frac{\int_U y \cdot \mu(y) \cdot dy}{\int_U \mu(y) \cdot dy}$$

U = Univers du discours

= Toutes les valeurs de sorties considérées

Méthode moyenne des maximums (MM) :

La défuzzification MM est plutôt utilisée lorsqu'il s'agit de discriminer une valeur de sortie (Ex: reconnaissance de formes).

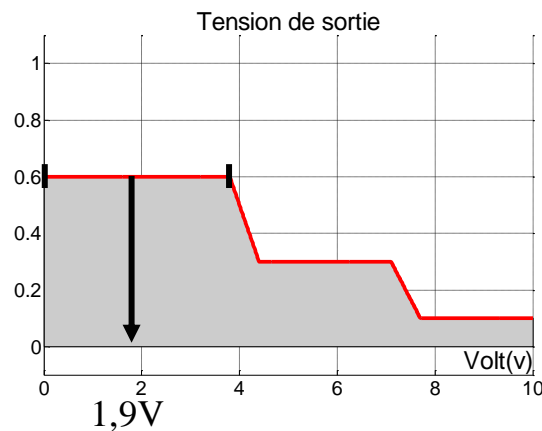


Figure A.10: La défuzzification par la méthode de la moyenne des maximums (MM)

C'est la moyenne des valeurs de sorties les plus vraisemblables

$$\text{sortie} = \frac{\int_S y \cdot dy}{\int_S dy}$$

$$\text{où } S = \left\{ y_0 \in U / \mu(y_0) = \underset{y \in U}{\text{SUP}} (\mu(y)) \right\}$$

A.1.8 Régulateur flou de Takagi et Sugeno

La plupart des implications floues utilisent des variables floues avec fonctions d'appartenance monomodales. L'utilisation de l'agrégation par inférence des règles ne permet pas d'avoir de bon résultat dans la commande de système multi variable. Il faut alors augmenter considérablement le nombre de variables floues et donc le nombre de règles.

La méthode de Mamdani, datant de 1975, est beaucoup plus répandue dans la théorie du contrôle de procédés. La méthode de Sugeno (1985) en diffère au niveau de la définition de la variable de sortie et, par conséquent, des méthodes de défuzzification. Rien ne change au niveau de la fuzzification des variables d'entrée. La variable de sortie prend, soit une valeur constante (singleton) indépendant des valeurs des entrées, soit une combinaison linéaire de celles-ci. Ce singleton sera, lors de l'étape de défuzzification, pondéré par les degrés d'appartenance des variables d'entrées. L'implication floue présentée par Takagi et Sugeno est basée sur la partition floue de l'ensemble des entrées. Dans chaque sous-ensemble flou une équation linéaire lie les entrées et les sorties. La sortie non floue de ce raisonnement flou est donnée par l'agrégation des valeurs floues résultantes des règles utilisées par les entrées.

La règle générale d'une règle de type Sugeno est, pour un système à 2 entrées e_1 et e_2

$$\text{Si } e_1 \text{ est } A \text{ ET } e_2 \text{ est } B \text{ ALOR } S \text{ sortie} = p * e_1 + q * e_2 + r$$

Avec : A et B qui sont des fonctions d'appartenance, respectivement, de e_1 et e_2 , et p, q, r des constantes choisies par l'utilisateur pour définir la combinaison linéaire des entrées.

La méthode de Mamdani :

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

La méthode de Sugeno :

- Elle fonctionne bien avec les techniques linéaires (par exemple, régulateur PID).
- Elle fonctionne bien avec l'optimisation et les techniques adaptatives.
- Elle garantit la continuité de la surface de la sortie.
- C'est bien assorti aux analyses mathématiques.

A.1.9 Conclusion

Les outils fournis par la logique floue permettent une modélisation des phénomènes pouvant en un certain sens s'approcher du raisonnement humain. Le fait de transcender le (tout ou rien) des ordinateurs introduit une souplesse faisant la puissance des outils flous dans de nombreux domaines. La logique floue va prendre de plus en plus de place dans le domaine du

réglage, cette affirmation est motivée par le fait que ce principe ne nécessite pas de modèle du système à régler et qu'il utilise les connaissances de l'opérateur (expert).

Ce chapitre a servi pour monter les différentes bases de la logique floue ainsi que le principe de fonctionnement d'un contrôleur à base de cette logique. Parmi les avantages et inconvénients de ce contrôleur on peut citer :

Avantage :

- Pas besoin de modèle.
- Implémentation de connaissances linguistiques.
- Maîtrise de systèmes à comportement complexe (non linéarité, ...).
- Solutions logicielles (μ p ou PC) et matérielles disponibles.

Inconvénients :

- Manque de directives précises pour la conception.
- Approche artisanale et non systématique.
- Impossibilité de démonstration de la stabilité du système.
- Possibilité d'apparition de règles d'inférence contradictoires.

Le contrôleur de logique floue peut très bien décrire le comportement d'un système désiré et traduire l'expérience humaine sous forme des variables linguistiques et des règles floues (si-alors), la collaboration d'un expert est très précieuse lors de la conception d'un contrôleur flou, mais il manque de la capacité d'apprentissage et ne peuvent pas de s'ajuster sur un nouvel environnement [Zimmermann. H, 1999]. Par conséquent, les réseaux de neurones artificiels offrent une bonne capacité d'apprentissage avec leur propriété d'approximation universelle. Dans la prochaine section on va voir l'un des outils de l'intelligence artificielle (les réseaux de neurones artificiels) avec les différents types d'apprentissage qui offrent cette technique.

A.2 RESEAUX DE NEURONES

A.2.1 Introduction

Les réseaux de neurones artificiels (RNA), apparus dans la littérature dès 1943 [MCC, 43], ils sont toujours en cours de développement. Ces réseaux de neurones artificiels sont issus de la combinaison entre la neurobiologie comme idée de base, la physique comme champs d'application et les mathématiques avec l'informatique en tant que moyens de réalisation. Ils mènent à élaborer une technique de traitement de données qui fera bientôt partie de la boîte à outils de tout ingénieur préoccupé de tirer le maximum d'informations pertinentes des données qu'il possède: faire des prévisions, élaborer des modèles, reconnaître des formes ou des signaux, etc...

La recherche sur les réseaux de neurones a connu un développement important ces dernières années. Cet axe de recherche a tiré l'attention de beaucoup de chercheurs de différentes disciplines. L'objectif principal de la recherche sur réseaux de neurones était d'accroître nos connaissances sur le mécanisme cérébral via l'élaboration de systèmes artificiels capables de reproduire des calculs complexes, similaires à ceux qu'effectue le cerveau humain, donc l'étude approfondie des concepts de base de ces réseaux est toujours nécessaire pour qu'on puisse aboutir à des solutions pour des applications différentes. En effet, ces travaux de recherche ont montré que les réseaux de neurones sont des approximateurs universels, tant du côté architecture où plusieurs modèles sont proposés, que du côté algorithmes d'apprentissages utilisés pour entraîner ces réseaux, ce qui permet de modéliser n'importe quel système non linéaire, et ils sont principalement, dotés de deux propriétés importantes : l'apprentissage et la généralisation. Ces travaux de recherche ont donné lieu à des applications très intéressantes des réseaux de neurones dans plusieurs domaines, et en particulier le domaine de la commande des systèmes non linéaires.

Les réseaux de neurones ont eu beaucoup d'applications dans la commande (neurocontrôle), et la littérature dans le domaine est devenue tout à fait volumineuse, les lecteurs intéressés peuvent se référer à des travaux d'aperçu général comme ceux de [Hunt, 91] et [Miller, 90]. [Andersen, 85], [Miller, 90], [Suykens, 96].

Le but de cette section est de présenter certains des paradigmes les plus largement utilisés de commande neuronale et de discuter comment ils effectuent leur opération de commande.

D'abord, nous faisons une présentation générale des réseaux de neurones, nous donnons un aperçu sur les éléments de base qui entrent dans leur constitution, à savoir le modèle du

neurone biologique et celui du neurone formel, ainsi que la définition d'un réseau de neurone et ses propriétés. Les structures de connexions entre les neurones, et les différents types d'apprentissage sont présentées.

A.2.2 Du neurone biologique au modèle mathématique

A.2.2.1 Le neurone biologique

L'unité de base de l'organisation du système nerveux est la cellule nerveuse ou le neurone. Cette cellule, bien qu'on en ait observé de nombreuses variantes, présente toujours la même fonctionnalité de transfert d'un signal électrique. Sa structure est d'ailleurs parfaitement adaptée à cette tâche ; on peut, grossièrement, la diviser en trois parties : les dendrites, le corps cellulaire (qui porte le noyau) et l'axone (figure A.11).

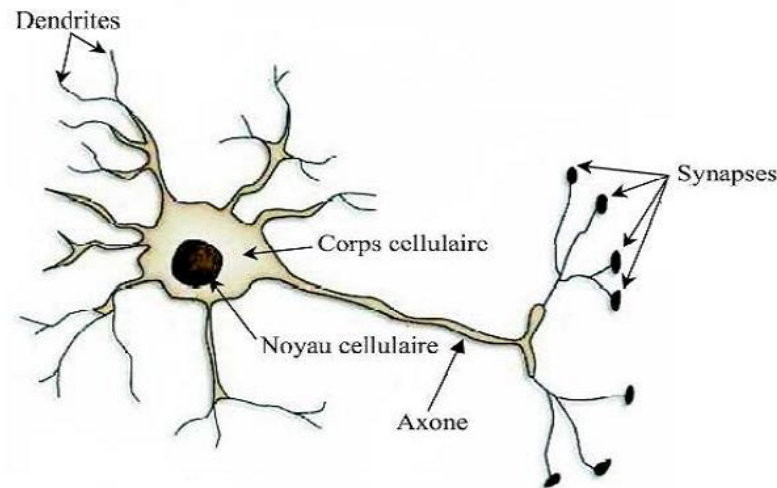


Figure A. 11 : Représentation schématique d'un neurone biologique

Les dendrites sont considérées comme les portes d'entrées des neurones. Elles sont les réceptrices principales du neurone permettant de capter les signaux lui parvenant. Ensuite, un traitement de toutes ces informations a lieu au niveau du corps cellulaire. Les principales lignes de transmission du système nerveux sont les axones. Ils servent de moyen de transport pour les signaux émis par le neurone, vers d'autres neurones. L'influx nerveux, qui a longtemps été considéré comme un signal continu, est en réalité un train de différences de potentiels d'environ 110 mV. Il est collecté sur d'autres cellules par l'intermédiaire des synapses et conduit par les dendrites jusqu'à la cellule elle-même. L'information va transiter de neurone en neurone le long de l'axone jusqu'aux muscles via un potentiel d'action (un signal électrique). La synapse désigne les zones de contact entre neurones, zones spécialisées

dans la transmission de l'information. Elles peuvent être inhibitrice ou excitatrice de signal pondérant l'information arrivant au corps cellulaire par dendrites.

A.2.2.2 Le neurone artificiel (ou neurone formel)

Par analogie avec la biologie, un neurone artificiel Figure A.12. Le neurone artificiel (ou cellule) est un processeur élémentaire. Les modèles mathématiques des neurones artificiels imitent le fonctionnement des neurones biologiques à divers niveaux de détail. Ici, un modèle simple sera considéré, qui est fondamentalement une fonction statique avec plusieurs entrées (représentant les dendrites) et une sortie (l'axone). Chaque entrée est associée à un facteur de poids (solidité du lien synaptique). La somme pondérée des entrées est appliquée à une fonction non linéaire, qui s'appelle la fonction d'activation. La valeur de cette fonction est la sortie du neurone.

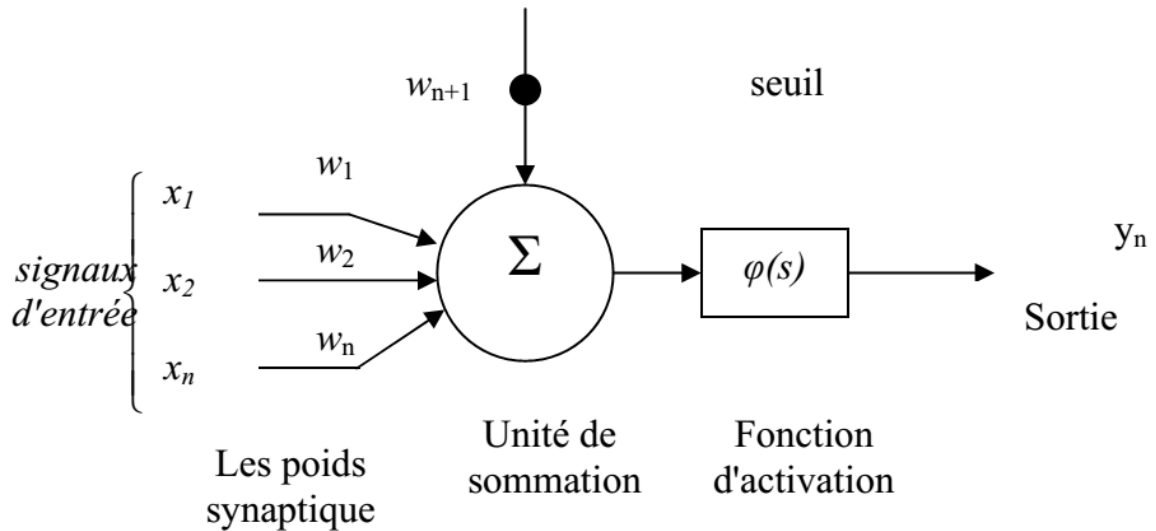


Figure A. 12 : Représentation schématique d'un neurone artificiel (formel)

Ce modèle est décrit mathématiquement par les équations suivantes:

$$z = \sum_{i=1}^p w_i x_i = W^T X. \quad (\text{A.1})$$

Où $w_i x_i \varphi$ et y_n sont respectivement, les entrées, les poids synaptiques, la fonction d'activation et la sortie du neurone.

Parfois, un seuil est employé quand on calcule l'activation du neurone :

$$z = \sum_{i=1}^p w_i x_i + d = \begin{bmatrix} W^T & d \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix} \quad (\text{A.2})$$

On peut voir que le seuil peut être considéré comme un poids supplémentaire provenant d'une entrée (unité) constante.

A.2.2.3 Fonctionnement d'un RNA

Le neurone calcule la somme de ses entrées puis cette valeur passe à travers la fonction d'activation pour produire sa sortie.

En règle générale, le calcul de la valeur de cette fonction peut se décomposer en deux étapes :

- Une combinaison linéaire des entrées :

$$v = w_0 + \sum_{i=1}^n w_i x_i \quad (\text{A.3})$$

Les w_i sont appelés poids synaptiques, w_0 est appelé biais. Le biais peut être considéré comme la pondération de l'entrée 0 fixée à 1. v est appelé potentiel du neurone. f : est la fonction d'activation du neurone.

- La sortie du neurone est :

$$y = f(v) = f\left(\sum_{i=0}^n w_i x_i\right) \quad (\text{A.4})$$

A.2.2.4 Fonction d'activation

La fonction d'activation (ou fonction de seuillage, ou encore fonction de transfert) sert à introduire une non linéarité dans le fonctionnement du neurone. Les fonctions de seuillage présentent généralement trois intervalles :

- en dessous du seuil, le neurone est non actif (souvent dans ce cas, sa sortie vaut 0 ou -1).
- aux alentours du seuil, une phase de transition.
- au-dessus du seuil, le neurone est actif (souvent dans ce cas, sa sortie vaut 1).

La fonction d'activation produit l'activation du neurone z dans un certain intervalle $[0, 1]$ ou $[-1, 1]$. Les fonctions le plus souvent utilisées sont, les fonctions à seuils, sigmoïdales, ou tangente hyperboliques (Figure A.13).

- Fonction de seuil (type dure)

$$\sigma(z) = \begin{cases} 0 & \text{pour } z < 0 \\ 1 & \text{pour } z \geq 0 \end{cases}$$

- Fonction (saturée) linéaire par morceaux

$$\sigma(z) = \begin{cases} 0 & \text{pour } z < -\alpha \\ \frac{1}{2\alpha}(z + \alpha) & \text{pour } -\alpha \leq z \leq \alpha \\ 1 & \text{pour } z > \alpha \end{cases}$$

- Fonction sigmoïde

$$\sigma(z) = \frac{1}{1 + \exp(-sz)}$$

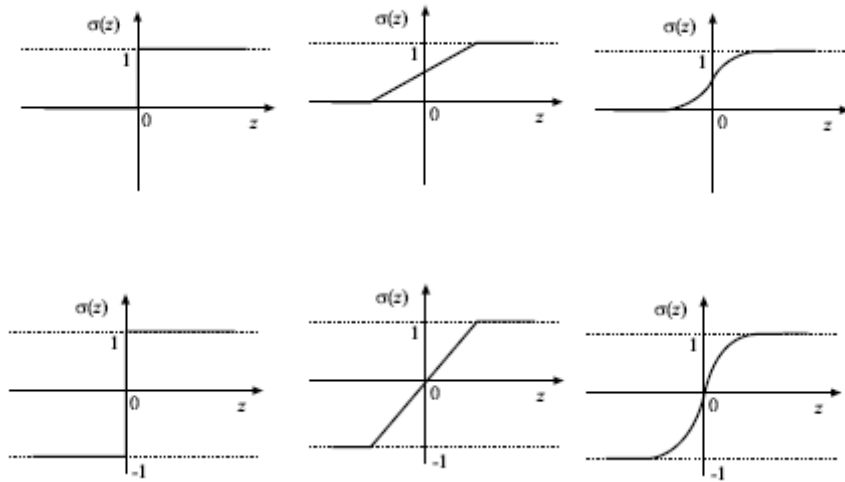


Figure A.13 : Différent types de fonctions d'activations

Ici, s est une constante déterminant la raideur de la courbe sigmoïdale. Pour $s \rightarrow 0$ la sigmoïde est très plate et pour $s \rightarrow \infty$ elle approche une fonction de seuil. La Figure A.14 montre trois courbes pour différentes valeurs de s . En général on prend $s = 1$ (La courbe en trait foncé sur la figure A.14).

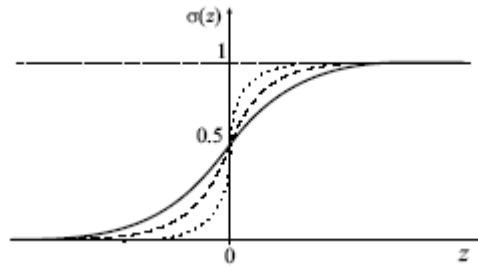


Figure A.14 : Fonction d'activation sigmoïdale

- Tangent hyperbolique

$$\sigma(z) = \frac{1 - \exp(-2z)}{1 + \exp(-2z)}$$

- Fonctions stochastiques :

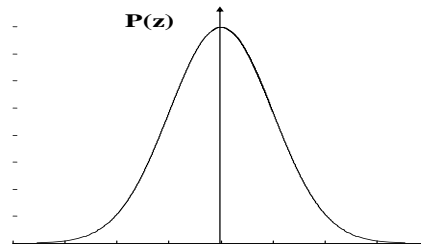


Figure A.15 : Fonction Gaussienne

La sortie v d'un neurone probabiliste peut prendre deux états, $\{0, 1\}$ ou $\{-1, 1\}$. La probabilité d'être dans un état donné dépend d'une fonction de probabilité qui est paramétrée par l'entrée total z du neurone.

$$P_r(v=1) = \frac{1}{1 + \exp\left(\frac{k \cdot z}{T}\right)}$$

Où k est une constante choisie en fonction du problème et où T est souvent appelé température par analogie avec les formules de physique statistique.

A.2.3 Architecture des réseaux de neurones

Un réseau de neurone est un ensemble d'éléments de traitement de l'information, avec une topologie spécifique d'interconnexions (architecteur du réseau) entre ces éléments et une loi

d'apprentissage pour adapter les poids de connexions (poids synaptiques), il est caractérisé par un parallélisme à gain très fin et à forte connectivité.

Nous entendons par là que dans un réseau de neurones donné, l'information est traitée par grand nombre de processeurs élémentaires très simples, chacun étant relié à d'autres processus. Ce processus très simple est un neurone formel désigné ainsi son fonctionnement s'inspire d'une modélisation des cellules biologiques. Ils sont dotés de deux propriétés importantes qui sont à l'origine de leur intérêt pratique des domaines très divers:

- Capacité d'adaptation ou d'apprentissage qui permet au réseau de tenir compte des nouvelles contraintes ou de nouvelles données du monde extérieur.
- Capacité de généralisation qui est son aptitude de donner une réponse satisfaisante à une entrée qui ne fait pas partie des exemples à partir desquels il apprend.

De fait, les règles d'organisation topologique des réseaux de neurones varient énormément d'un modèle connexionniste à l'autre. Nous allons essayer ici de décrire quelques-unes des grandes règles topologiques utilisées [Guersi, 2004].

A.2.3.1 Cas général

Bien que les règles d'organisation puissent être très variables, la description de la topologie est toujours donnée très simplement par l'ensemble des fonctions d'entrée des neurones composant le réseau (nous nous limitons ici au cas des fonctions d'entrée affines). En effet, on considère qu'une connexion entre deux neurones i, j (de i vers j) est inexistante lorsque la pondération correspondante est nulle ($W_{ji} = 0$). La topologie d'un réseau de n neurones (réseau de dimension n) peut alors être décrite sous la forme d'une matrice carrée : la matrice des poids W .

Cette écriture permet de décrire très simplement les différentes architectures employées. Elle permet, en particulier, de définir la notion de chemin d'activation dans un réseau de neurones : *un chemin d'activation est une suite de neurones reliés par des connexions non nulles*. Un chemin d'activation peut être bouclé (c'est-à-dire qu'un même neurone peut apparaître plusieurs fois dans la suite). En conséquence, il peut y avoir, dans un même réseau, une infinité de chemins d'activation.

Décrire en détail tous les modèles connexionnistes serait impossible étant donné les développements pléthoriques que connaît aujourd'hui cette question. Pour s'en convaincre, il

suffit de constater le nombre impressionnant d'ouvrages introductifs et de revues nouvelles consacrées à ce sujet [Davallo et Naim, 89]. Toutefois nous allons revenir sur quelques modèles classiques qui sont à la base du développement des méthodes utilisées aujourd'hui. Nous détaillons surtout des algorithmes qui sont à l'origine de l'algorithme de rétro propagation que nous avons utilisé dans ce travail. Nous allons maintenant présenter, à partir de ces notions, quelques définitions sur les organisations topologiques caractéristiques.

A.2.3.2 Les réseaux en couches

Dans les réseaux en couche, seules les cellules appartenant à la couche d'entrée ou à la couche de sortie sont associées à une information symbolique. Elles réalisent l'interface entre l'information distribuée sur les couches cachées (plus exactement sur les connexions cachées) et l'utilisateur du réseau. Dans ce cas particulier, chaque couche de connexions peut être assimilée à une fonction mathématique linéaire appliquée aux entrées, celle-ci étant répartie sur l'ensemble des connexions (chaque critère d'entrée est associé, par une pondération, aux critères de sortie). On montre cependant que toute fonction continue bornée peut être identifiée par un réseau à trois couches [FUN, 89] [HOR, 89]. La répartition de l'information entre les connexions d'une même couche (et entre les différentes couches) est étroitement liée à l'algorithme utilisé : l'algorithme le plus couramment utilisé, la rétro propagation du gradient, provoque une " dilution " générale de l'information mais ce n'est pas le cas, par exemple, pour les modèles " RBF-Nets" [CHEN, 91] qui sont basés sur la décomposition de la fonction mathématique globale en fonctions locales, propres à chaque cellule cachée, se référant à un " noyau " (" Kernel ") identifié dans l'espace d'entrée.

La distribution de l'information dans les réseaux utilisant la rétro propagation du gradient est à l'origine de la plupart des contraintes pesant sur ce type de réseaux. Ainsi l'apprentissage doit être organisé de façon à fournir au réseau une information cohérente dans le temps. Il n'est pas possible, avec de tels réseaux, d'apprendre successivement différentes informations. L'apprentissage de nouvelles informations dans un réseau déjà entraîné risque en effet de provoquer un phénomène " d'oubli catastrophique " : les pondérations préalablement fixées sont modifiées et le réseau " désapprend ". Dans ce type de réseaux, toutes les informations apprises doivent être fournies au cours d'une même séance d'apprentissage.

A.2.3.3 Les réseaux monocouches à transmission directe dit "feedforward"

Il s'agit de réseaux dont les neurones forment une couche unique. Dans sa forme la plus simple (voir Figure A.16), ce type de réseau possède une seule couche de nœuds d'entrée qui se projettent sur les neurones de la couche de sortie (la couche de calcul). L'information est donc transmise directement de l'entrée vers la sortie. Ce réseau est monocouche dans la mesure où l'on ne compte pas les nœuds sources car ils ne sont le siège d'aucune opération mathématique.

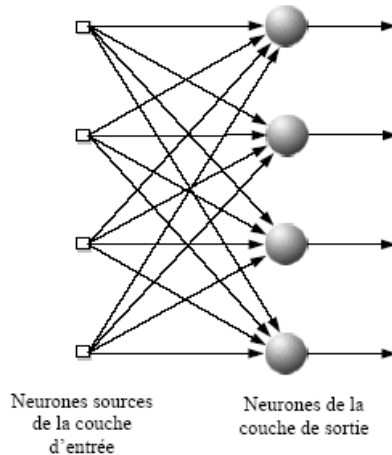


Figure A.16 : Réseau monocouche à transmission directe (feedforward)

A.2.3.4 Les réseaux multicouches à transmission directe dit "feedforward"

Cette seconde classe de réseaux de neurones se distingue de la précédente par la présence d'une ou plusieurs couches cachées, dont les nœuds de calcul sont appelés neurones cachés, ou unités cachées, on les appelle aussi "Perceptron Multi-Couche", (PMC) ou "Multi-Layer Perceptron", (MLP). La fonction des neurones cachés est d'intervenir entre l'entrée externe et la sortie du réseau. Les neurones de ces couches intermédiaires n'ont aucun contact avec l'extérieur, d'où leur nom de neurones cachés. Les nœuds sources de la couche d'entrée du réseau fournissent le vecteur d'entrée. Ce signal est appliqué aux neurones de calcul de la seconde couche (qui est aussi la première couche cachée). Les signaux de sortie de la seconde couche deviennent les entrées de la troisième et ainsi de suite. Typiquement, les neurones de chaque couche ont leurs entrées dans la couche précédente. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc de définir les concepts de neurones d'entrée et de sortie. Les signaux de sortie fournis par le jeu de neurones de la couche finale constituent la réponse du réseau aux stimuli issus des nœuds sources de la première couche.

Le réseau sera dit totalement connecter, ou à connexion complète dans le sens où chaque nœud de chaque couche est relié à tous ceux des couches suivantes (Figure A.18). Si certaines connexions manquent, on parle alors de réseaux partiellement connectés (Figure A.17). Une forme particulièrement intéressante de réseaux partiellement connectés est le réseau localement connecté. La figure A.17 illustre un exemple d'un tel réseau à une couche cachée. Chaque neurone de la couche cachée est connecté localement (partiellement) à un jeu de nœuds source, situés dans son voisinage proche. Il entretient donc des relations avec un nombre réduit et localisé de neurones de la couche avale. Ce type de réseau présente une spécialisation spatiale par rapport au vecteur signal d'entrée. Ainsi, chaque neurone de la couche répondra essentiellement à des variations locales du signal source.

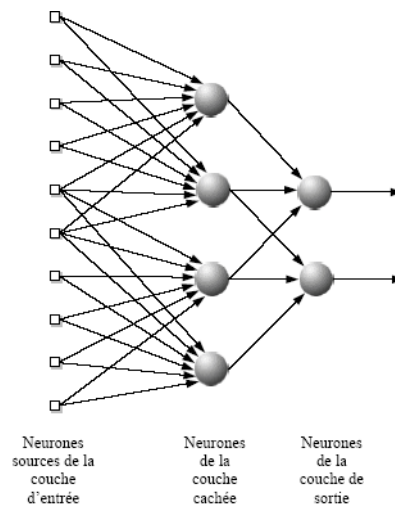


Figure A.17 : Réseau à transmission directe partiellement interconnecté

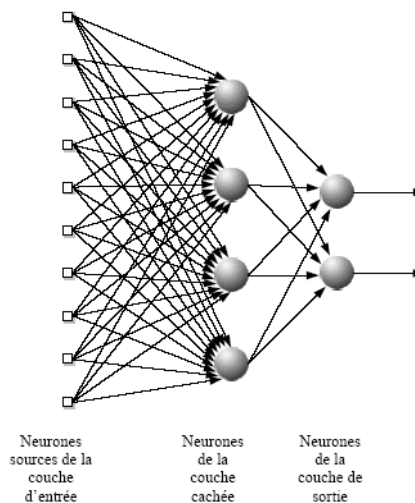


Figure A.18 : Réseau à transmission directe totalement interconnecté

A.2.3.5 Les réseaux récurrents

Un réseau de type récurrent se distingue d'un réseau à transmission directe de type feedforward en ce qu'il intègre au moins une boucle de retour. La Figure A.19 suivante illustre une telle structure. Les connexions retour partent aussi bien des neurones de sortie que des neurones cachés. La présence des boucles de retour induit l'utilisation de branches particulières contenant des éléments retard (notés Z^{-1}). Ceux-ci entraînent un comportement dynamique non linéaire du réseau, en raison de la nature non linéaire de ces neurones.

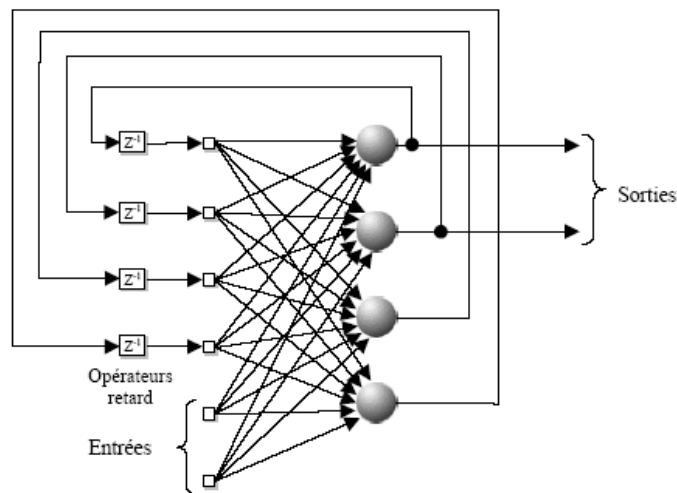


Figure A.19 : Réseau récurrent avec des neurones cachés

A.2.3.6 Les réseaux à "fonction radiale de base " ou "Radial Basis Functions "

Les " Réseaux de neurone R B F " (ou "RBF-Nets") sont, comme les réseaux que nous venons de présenter, des réseaux " feed-forward " composés de trois couches de neurones. Cependant, ils reposent sur un principe d'organisation de l'information totalement différent. C'est pourquoi nous avons choisi de les présenter séparément. A l'inverse de la plupart des réseaux de neurones présentés dans ce chapitre, les "Radial Basis Functions" ne sont pas issus de l'analogie connexionniste mais de la transposition vers les réseaux de neurones de modèles mathématiques pour l'approximation de fonctions [POG, 90].

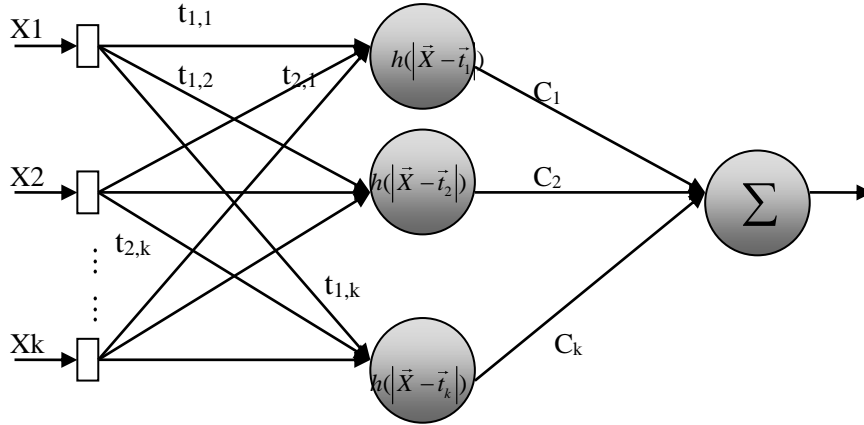


Figure A.20 : Réseau R.B.F

Soit N paires $(\vec{x}_i, |y_i)$ ($\vec{x} \in \mathfrak{R}^n, y \in \mathfrak{R}$) on cherche une fonction $f(\bullet)$ de la forme :

$$f(\vec{x}) = \sum_{i=1}^k c_i h(\|\vec{x} - \vec{t}_i\|)$$

où $h(\bullet)$ est une "fonction radiale de base" et où les \vec{t}_i sont les K centres (ou Noyaux, Kernels) devant être choisis dans \mathfrak{R}^n . Les noyaux, ainsi que les coefficients c_i , sont à déterminer à partir de l'ensemble des couples $(\vec{x}_i, |y_i)$ de sorte que $f(\bullet)$ minimise une fonction de coût composée de l'erreur quadratique et d'un terme stabilisateur destiné à lisser la fonction :

$$C[f] = \sum_{i=1}^n (y_i - f(\vec{x}_i))^2 + \lambda \|Pf\|^2$$

La fonction $h(\bullet)$ est appliquée à la distance euclidienne entre chaque centre \vec{t}_i et chaque élément \vec{x} . On choisit généralement une fonction ayant son maximum en zéro (distance nulle), la fonction la plus couramment utilisée est une gaussienne

$$h(\|\vec{x} - \vec{t}_i\|) = e^{-\theta \|\vec{x} - \vec{t}_i\|^2} \quad (A.5)$$

Mais d'autres fonctions sont utilisables et plusieurs études montrent que la fonction choisie n'est pas déterminante pour les performances de l'algorithme [CHEN, 91].

L'algorithme des "RBF" consiste alors à choisir les K centres dans l'espace de départ (généralement les centres sont choisis dans l'ensemble des éléments \vec{x}) puis à déterminer analytiquement le vecteur $\vec{c} = (c_1, \dots, c_k)^T$ minimisant la fonction de coût (équation A.3).

Si G est une matrice (n, K) telle que :

$$G = \begin{bmatrix} h(\vec{x}_1 - \vec{t}_1) \dots h(\vec{x}_1 - \vec{t}_k) \\ \vdots \vdots \\ h(\vec{x}_n - \vec{t}_1) \dots h(\vec{x}_n - \vec{t}_k) \end{bmatrix}$$

et si H est une matrice (K, K) telle que :

$$H = \begin{bmatrix} h(\vec{t}_1 - \vec{t}_1) \dots h(\vec{t}_1 - \vec{t}_k) \\ \vdots \vdots \\ h(\vec{t}_k - \vec{t}_1) \dots h(\vec{t}_k - \vec{t}_k) \end{bmatrix}$$

Alors, si $\vec{y} = (y_1, \dots, y_n)^T$ on a :

$$\vec{c} = (G^T . G + \lambda H)^{-1} . G^T . \vec{y} \quad (\text{A.6})$$

Hormis le terme stabilisateur λH l'équation (A.3) est identique à l'inversion matricielle de Moore-Penrose utilisée en interpolation linéaire.

La fonction $f(\bullet)$ peut être directement représentée sous la forme d'un réseau de neurones à trois couches comptant n neurones sur la couche d'entrée, K neurones sur la couche cachée et un seul neurone de sortie. Les connexions de la couche cachée sont pondérées par les différentes composantes des vecteurs noyaux et les connexions de la couche de sortie reçoivent une pondération égale à la valeur des coefficients c_i (figure A.20). La fonction de transfert des neurones cachés est la fonction "RBF" appliquée à la distance euclidienne entre le vecteur d'entrée et le vecteur de poids de l'unité (c'est-à-dire, pour le neurone i entre \vec{x} et \vec{t}_i). Dans le cas d'une fonction gaussienne, le facteur θ (équation A.5) correspond alors au seuil des cellules cachées.

L'analogie avec les réseaux de neurones a conduit à l'extension des "Radial Basis Functions". La fonction de transfert de la cellule de sortie, en particulier, peut-être plus complexe qu'une simple sommation des influx entrants : l'utilisation d'un seuil et d'une fonction non-linéaire (par exemple une fonction sigmoïde) permet d'améliorer les capacités d'approximation. En outre, il est toujours possible d'augmenter le nombre de neurones sur la couche de sortie et d'approximer plusieurs fonctions à partir d'un même ensemble de noyaux.

Les "réseaux RBF" présentent de nombreux intérêts dont, en premier lieu, la possibilité de calculer analytiquement les poids à partir de la base d'apprentissage ce qui permet de s'affranchir de la phase d'entraînement du réseau. Cela suppose cependant que l'on ait préalablement placé les noyaux de façon optimale ce qui n'est pas toujours possible. Des

algorithmes d'apprentissage (basés par exemple sur les moindres carrés [CHEN, 91]) permettent alors d'adapter la position des noyaux dans l'espace d'entrée.

Pour notre propos les "réseaux RBF" présentent un intérêt d'un autre type. En effet, dans ce type de réseau, l'information n'est pas totalement distribuée sur l'ensemble des connexions : la distribution de l'information est directement dépendante du nombre de noyaux et de leur position, chaque connexion supportant un élément d'information parfaitement identifiable. On peut donc considérer que les "réseaux RBF" constituent une forme particulière de réseau à information semi distribuée, voire de réseau modulaire.

A.2.4 L'apprentissage des réseaux de neurones

Comme le cerveau humain, les réseaux de neurones artificiels (RNA) peuvent apprendre par expérience. On appelle « phase d'apprentissage » des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseau, durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. Ainsi, suite à l'application séquentielle de plusieurs entrées à apprendre, les algorithmes d'apprentissage modifient la valeur des poids entre les neurones ainsi que la valeur des biais de façon à améliorer la performance du RNA et afin que celui-ci remplisse au mieux la tâche qui lui est affectée. La plupart des réseaux de neurones apprennent par l'exemple [Claude Touzet, 92]. En fait, l'ajustement des poids est plus ou moins efficace tout dépendamment de la connaissance que nous avons du système à modéliser ou du problème à résoudre.

A.2.5 Types d'apprentissage

Selon le critère d'apprentissage les réseaux de neurones se subdivisent en trois grandes classes d'apprentissage existent :

A.2.5.1 L'apprentissage supervisé

Un ingrédient essentiel de l'apprentissage supervisé (voir schéma Figure A.21) est la présence d'un " professeur " ou " expert ". D'un point de vue conceptuel, cet expert possède la connaissance de l'environnement, connaissance qui se représente par des couples d'exemples d'entrées-sorties. L'environnement est supposé inconnu du réseau. Supposons maintenant qu'à la fois l'expert et le réseau soient soumis à un stimulus de l'environnement (sous la forme d'un vecteur d'entraînement). L'expert emmagasinant la connaissance de l'environnement est apte à fournir au réseau une réponse cible (où réponse désirée). Cette

réponse désirée représente l'action optimale que doit effectuer le réseau. Les paramètres du réseau évoluent alors pour s'ajuster en fonction des influences combinées du vecteur d'apprentissage et du signal d'erreur. Ce dernier signal d'erreur étant défini comme la différence entre la réponse courante du réseau et la réponse désirée fournie par l'expert.

Le réseau doit à terme, après un certain nombre de pas d'apprentissage, émuler la connaissance de l'expert de façon optimale au sens statistique. Lorsque ce transfert de connaissance de l'expert est achevé, on se passe alors de lui, et c'est le réseau qui travaille directement avec son environnement.

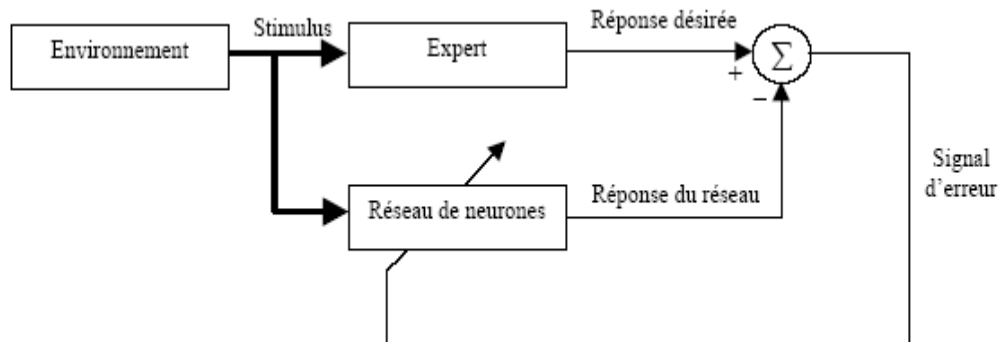


Figure A.21 : Schéma bloc-diagramme de l'apprentissage supervisé

L'apprentissage supervisé que nous venons de décrire est fondé sur un processus de correction de l'erreur d'apprentissage qui sera décrit plus loin. Il réalise un système bouclé dont l'environnement est toutefois exclu. La mesure de la performance du système est évaluée d'après l'erreur définie au sens des moindres carrés (la moyenne de la somme des erreurs au carré). Cette fonction d'erreur se visualise sous la forme d'une surface d'erreur, dont les coordonnées sont les paramètres libres. [Haykin, 91] a montré que la surface d'erreur est moyennée sur l'ensemble des couples d'entrée-sortie désirés.

Toute opération du système, sous la supervision de l'expert peut être représentée comme un point de la surface d'erreur. Pour que le réseau améliore ses performances au fil du temps en apprenant la connaissance de l'expert, ce point de la surface d'erreur doit descendre vers un point minimum de la surface d'erreur. Ce point minimum peut être un extremum local, ou global. Pour ce faire, un système à apprentissage supervisé utilise une information utile qui est le gradient de la surface d'erreur au point de fonctionnement souhaité. Ce vecteur gradient de la surface d'erreur est un vecteur qui pointe dans la direction de plus grande pente. En fait, dans le cas d'un apprentissage supervisé à base d'exemples, le système utilise une estimation instantanée du vecteur gradient. A partir de cela, un algorithme de réseau de neurones prévu pour minimiser une fonction de coût, épaulé par un jeu d'exemples d'entrées-sorties, et

possédant assez de temps d'apprentissage, est capable de réaliser de manière très satisfaisante la tâche d'intérêt pour laquelle il a été conçu (classification, approximation de fonctions...).

A.2.5.2 L'apprentissage par renforcement

L'apprentissage supervisé nécessite la définition d'une base d'exemples d'apprentissage représentative. Chaque exemple est un couple (entrée, sortie désirée) et le principe de l'algorithme est de réduire l'erreur quadratique réalisée par le réseau. Cette erreur est la différence entre la valeur de sortie désirée et la valeur de sortie obtenue. La rétropropagation du gradient opérant sur un réseau multicouche a contribué de façon importante à la popularité et à la diffusion des applications connexionnistes. Cependant, la difficulté inhérente à la construction d'une base d'apprentissage "représentative" limite son utilisation. Des recherches ont donc été conduites avec l'objectif de construire de manière automatique cette base. L'apprentissage par renforcement est une alternative possible [Touzet, 99]. La différence essentielle avec l'apprentissage supervisé réside dans la forme des exemples d'apprentissage. Du fait du caractère automatique de la construction de la base d'apprentissage, l'utilité d'une paire (entrée, sortie) doit être décrite par une équation, une règle de calcul ou une procédure. C'est à ce niveau que se situe l'intervention principale de l'opérateur. Le succès de l'application dépendra de la qualité de la fonction spécifiant l'utilité d'une paire. Cette fonction d'utilité, habituellement appelée fonction de renforcement, mesure la performance du réseau. La performance est définie comme l'utilité des sorties proposées relativement à la tâche à accomplir. L'utilité est ici un concept qualitatif, généralement codé de façon binaire : +1 = bon, -1 = mauvais et 0 lorsque l'on est incapable de se prononcer. Habituellement, la génération de la base d'apprentissage se fait en parallèle de la phase d'exploitation, l'apprentissage est incrémental. C'est pourquoi, lorsqu'une base d'apprentissage représentative a finalement été construite, l'apprentissage est terminé.

A.2.5.3 L'apprentissage non supervisé

L'apprentissage non supervisé, ou apprentissage auto organisé ne requiert pas la présence d'un expert. En d'autres termes, on n'a pas besoin d'exemples spécifiques de la fonction à apprendre (voir schéma Figure A.22). A la place, on utilise une mesure indépendante de la qualité de représentation du réseau, et ses paramètres libres sont optimisés selon ce critère de mesure. Cette forme d'apprentissage n'a donc pas besoin de couples (entrée-sortie désirée), mais simplement d'entrées [Haykin, 1994].

Pour réaliser un apprentissage non supervisé, on a recours à une règle d'apprentissage compétitive. Dans sa forme la plus simple, cette règle est de type "winner take all", c'est-à-dire que la stratégie consiste à élire le neurone qui correspond le plus au vecteur d'entrée et à modifier ses paramètres pour l'ajuster encore mieux. Les autres neurones du réseau étant inhibés

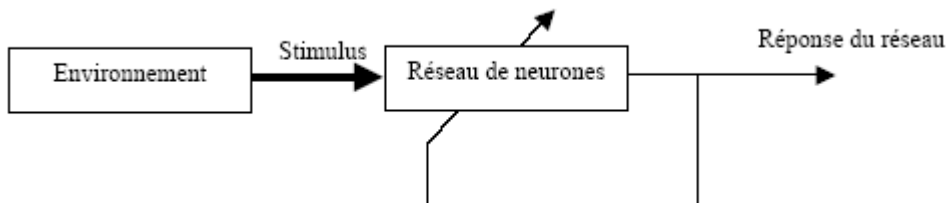


Figure A.22 : Schéma bloc-diagramme de l'apprentissage non supervisé

A.2.6 La commande neuronale

Le domaine de la commande neuronale a eu une grande poussée avec l'arrivée de l'algorithme de rétropropagation dans les années 80 et la réalisation d'un réseau simple avec une seule couche caché peut (en théorie) approcher des fonctions continues avec des niveaux arbitraires de précision [Funahashi, 89]. Beaucoup de procédures de commande neuronale ont émergé, basé, sur ces deux résultats. Indépendamment de l'algorithme de rétropropagation, il y a d'autres structures de réseau récurrents (c.-à-d. de rétroaction) et d'algorithmes qui ont été aussi bien employés (voir [Narendra, 90], [Miller, 90]).

Un système de commande neuronale est un système dans lequel un réseau de neurones artificiel est employé dans le processus de détermination du signal de commande. Le contrôleur lui-même peut être un contrôleur conventionnel utilisant des paramètres produits par un émulateur neuronal du système, ou un réseau de neurones entraîné pour copier le comportement d'un contrôleur analytique existant. Cela peut même être un mélange des deux. Une partie très cruciale de n'importe quel genre de conception de commande neuronale est la technique par laquelle des exemples d'apprentissage sont collectés et fournis au réseau pour que les poids puissent être convenablement adaptés. Puisque le mécanisme par lequel un système de commande neuronale adapte ses poids est semblable à celui des contrôleurs adaptatifs conventionnels, la plupart des contrôleurs neuronaux sont de nature réellement adaptative.

Dans beaucoup de structures de commande neuronale, le but des NN est de faire l'approximation de fonction. Si les réseaux sont entraînés par des techniques de descente du gradient, l'approximation est faite en associant les signaux d'erreur de sortie à des signaux

d'entrée particuliers. L'objet de l'apprentissage est d'essayer de réduire au minimum les erreurs de sortie, donc les signaux d'entrée sont choisis en conséquence.

A.2.7 Contrôleurs neuronaux : Classification

La classification des schémas neuronaux de commande est faite en se basant sur la méthode par laquelle le signal d'erreur pour la commande est déterminé. La figure A.23 montre la hiérarchie de différents neurocontrôleurs et les classes auxquelles ils appartiennent.

Une brève description de chaque classe est donnée ci-dessous:

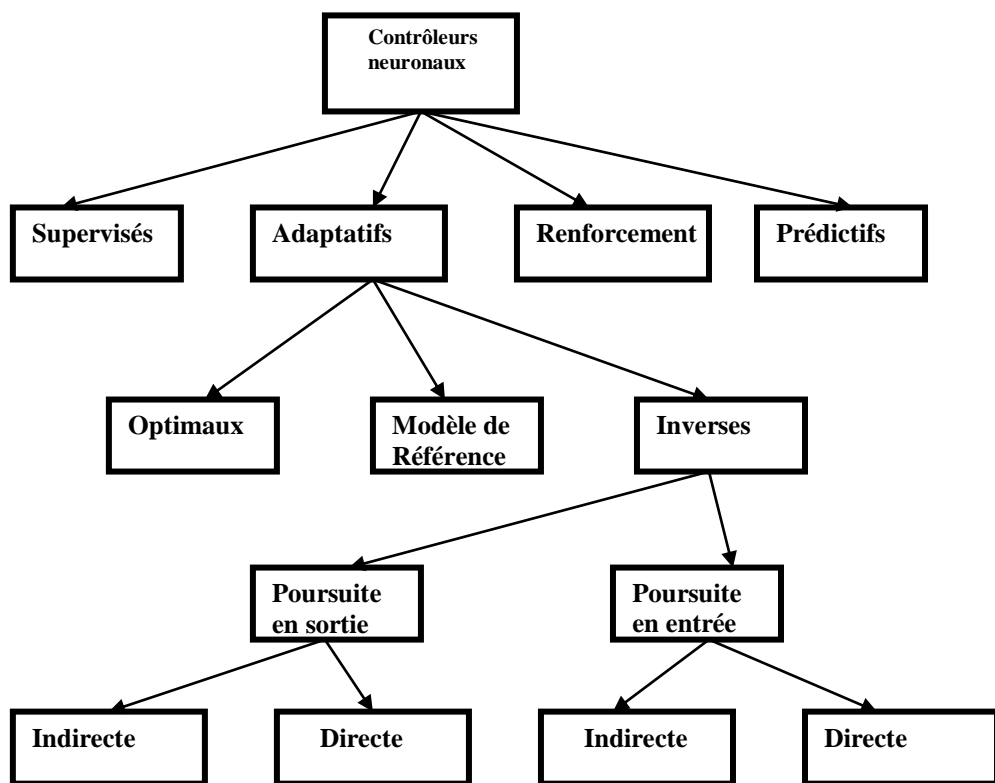


Figure A.23 : Classifications des contrôleurs

Supervisés : Les NN sont entraînés pour imiter un autre contrôleur.

Renforcement: L'apprentissage des contrôleurs est effectué par renforcement [Hertz, 91].

Prédicatif: Le contrôleur est entraîné à poursuivre la sortie d'une routine d'optimisation opérant sur un modèle émulé du processus à commander.

Adaptatif : Le contrôleur neuronal est spécifié par minimisation d'une fonction de coût.

Optimale: La fonction de coût non entièrement basée sur la sortie actuelle du processus et l'erreur sur le signal de commande sont réduits au minimum.

Modèle de référence: Le contrôleur neuronal est entraîné de telle façon que le processus global se comporte comme un modèle de référence.

Inverse: Le modèle inverse du processus est utilisé comme contrôleur.

Poursuite en sortie : "*Output matching*": Elle vise à réduire au minimum l'erreur entre la sortie du processus et le signal de référence.

Poursuite indirecte en sortie : "*Indirect output matching*": L'erreur du contrôleur est déterminée par rétropropagation de l'erreur de sortie du processus au travers du modèle neuronal de ce dernier.

Poursuite directe en sortie : "*Direct output matching*": L'erreur du contrôleur est directement déterminée à partir de l'erreur de sortie réelle du processus.

Poursuite en entrée : "*Input matching*": vise à réduire au minimum l'erreur entre un signal de commande connu et la sortie du contrôleur neuronal.

Poursuite indirecte en entrée : "*Indirect input matching*": modèles inverses du processus utilisés pour calculer l'erreur du contrôleur.

Poursuite directe en entrée : "*Direct input matching*": les erreurs du contrôleur sont calculé en utilisant le contrôleur comme modèle inverse.

A.2.8 Conclusion

Dans cette section, nous faisons une présentation générale des réseaux de neurones à partir du comportement du cerveau humain et d'un modèle neuronal biologique simple, d'abord, nous donnons un aperçu sur les éléments de base qui entrent dans leur constitution, à savoir le modèle du neurone biologique et celui du neurone formel, ainsi que la définition d'un réseau de neurone et ses propriétés, les structures de connexions entre les neurones, et les différents types d'apprentissage ainsi ses différents domaines d'utilisation.

Les réseaux de neurones formels, tels que nous les avons définis, possèdent une propriété remarquable qui est à l'origine de leur intérêt pratique dans des domaines très divers. Elle donne une bonne qualité d'apprentissage qui est vraisemblablement la propriété la plus

intéressante des réseaux neuronaux, mais ils ne peuvent pas interpréter et vérifier si la solution obtenue est faisable [Narendra, 90], [I.E.E.E, 92].

Dans la partie suivante on va utiliser un système d'inférence adaptative neuro-flou (ANFIS), qui combine les concepts de la logique floue et les réseaux de neurones pour constituer un contrôleur intelligent hybride appelé ANFIS.

Références

Bibliographiques

RÉFÉRENCES

- [**Aldon 82**] Aldon M.J., "Elaboration automatique de modèles dynamiques de robots en vue de leur conception et de leur commande". Thèse d'Etat, USTL, Montpellier, oct. 1982.
- [**Amirabdollahian et al., 2002**] F. Amirabdollahian, R.Loureiro, et W. Harwin (2002). "Minimum jerk trajectory control for rehabilitation and haptic applications". Dans IEEE International Conference on Robotics and Automation, pages 3380–3385, May.
- [**Andrew. C, 2000**] Andrew C. Clegg. "Self-tuning Position and Force Control of a Hydraulic Manipulator". PhD thesis, Heriot-Watt University Department of Computing and Electrical Engineering, Royaume-Uni, 2000.
- [**Arato. S, 1995**] Arato. S. Deo. "Inverse Kinematics and Dynamic Control Methods for Robotic Systems". These de PhD Houston .Texas.1995.
- [**Armstrong, 1979**] Armstrong W.W., "Recursive solution to the equation of motion of an N-links manipulator". Proc. 5th World Congress on Theory of Machines and Mechanisms, Montréal, 1979, p. 1343-1346.
- [**Asada et Slotine, 1986**] Asada, H. et Slotine, J. "Robot analysis and control. Wiley, New York". 1986.
- [**Bailleul et al., 1984**] Bailleul, J., Hollerbach, J., et Brockwett, R. "Programming and control of kinematically redundant manipulators". Dans IEEE Conf. On Decision and Control (CDC), 1984. Pages 768–774.
- [**Bayard, 1988**] Bayard D., Wen J.T., "New class of control laws for robotic manipulator. Part 2: Adaptive case", Int. J. Control, Vol. 47(5), 1988, p. 1387-1406.
- [**Benalia. A, 2003**] A. Benalia. "Génération de trajectoires des systèmes non linéaires en présence de contraintes". Conférence GDA 2003, Valenciennes, France.
- [**Benalia. A, 2004**] A. Benalia. "Contribution à la modélisation et la commande robuste du confort au sein d'un habitacle automobile". Thèse de Doctorat, université d'Orsay 2004.
- [**Berghuis, 93**] Berghuis H. "Model-based robot control: from theory to practice". Ph. D. Thesis, Université de Twente, Enschede, Pays Bas, 1993.
- [**Bidaud et al, 2014**] Ph. Bidaud - Philippe. Bidaud@upmc. "Cours de Modélisation des Systèmes Robotiques". Université Pierre et Marie Curie., Master Sciences de l'Ingénieur Spécialités "Systèmes Avancés et Robotiques", "Systèmes Intelligents et Robotiques".
- [**Bishop, 95**] Bishop C. M. "Neural networks for pattern recognition". Oxford University Press, Oxford, G.B, 1995.
- [**Bobrow et al., 1985**] J. E. Bobrow, S. Dubowsky, et J. S. Gibson. "Time optimal control of robotic manipulators along specified paths". The Int. J. of Robotics Research, 4:3–17. 1985.

[**Boissonnat et al., 1988**] Boissonnat, J.-D., Faverjon, B., et Merlet. "Techniques de la robotique. Perception et planification". J.-P. Hermès. 1988.

[**Borrel 79**] Borrel P., "Modèle de comportement de manipulateurs ; application à l'analyse de leurs performances et à leur commande automatique". Thèse de Troisième Cycle, USTL, Montpellier, déc. 1979.

[**Borrel, 86**] Borrel P., "Contribution à la modélisation géométrique des robots manipulateurs ; application à la conception assistée par ordinateur". Thèse d'Etat USTL Montpellier, juillet 1986.

[**Brandl, 86**] Brandl H., Johanni R., Otter M., "A very efficient algorithm for the simulation of robots and multibody systems without inversion of the mass matrix". Proc. IFAC Symp. On Theory of Robots, Vienne. Autriche, déc. 1986. p. 365-370.

[**Brooks, 1983**] Brooks, R.. "Planning collision-free motions for pick-and-place operations". International Journal of Robotics Research (IJRR), 2(4). 1983.

[**Burdick 88**] Burdick J.W., "Kinematic analysis and design of redundant manipulators". PhD Thesis, Stanford University, 1988.

[**Canudas de Wit, 92**] Canudas de Wit C., Fixot N., Astrôm K.J., "Trajectory tracking in robot manipulators via nonlinear estimated feedback". IEEE Trans, on Robotics and Automation, Vol. RA-8(1), 1992. p. 138-144.

[**Canudas de Wit et al., 1996**] Canudas de Wit, C., Siciliano, B., et Bastin, G. "Theory of robot control". Éditeurs (1996). Springer Verlag.

[**Canudas et al., 1997**] C. Canudas, B. Siciliano, et G. Bastin (1997). "Theory of Robot Control". Springer.

[**Cesareo 84**] Cesareo G., Nicolo F., Nicosia S., "DYMIR: a code for generating dynamic model of robots". Proc. IEEE Int. Conf. on Robotics, Atlanta, mars 1984, p. I 15-120.

[**Chen et al., 1989**] Y. Chen et A. Desrochers (1989). "Minimum-time control laws for robotic manipulators". Dans Proc. of the 28th Conference on Decision and Control, pages 2494–2499, Tampa, Fl.

[**Cherki, 96**] Cherki B. "Commande des robots manipulateurs par retour d'état estimé". Thèse de Doctorat. Université de Nantes et Ecole Centrale de Nantes. 1996.

[**Chevallereau, 87**] Chevallereau C., Khali 1 W., "Efficient method for the calculation of the pseudo inverse kinematic problem". Proc. IEEE Int. Conf. on Robotics and Automation, Raleigh, mars-avril 1987, p. 1842-1848.

[**Chun-Ta. C, 2012**] Chun-Ta Chen. "Hybrid approach for dynamic model identification of an electro-hydraulic parallel platform". Springer Nonlinear Dynamics, Vol. 67, No. 1, 2012, pp. 695-711.

[**Clegg, 94**] A.C. Clegg. "The Mechanics and Modelling of Hydraulically Actuated Manipulators". Research Memo RM/94/8, Dept. of Computing & Electrical Eng., Heriot-Watt University, November 1994.

[**Craig, 1985**] Craig, J., "Adaptive Control of Mechanical Manipulators". Reading, MA: Addison Wesley, 1985.

[**Craig, 1986**] Craig, J. (1986). "Introduction to robotics: mechanisms and control". Addison Wesley publishing company, Reading.

[**Craig 86a**] Craig J.J., "Introduction to robotics: mechanics and control". Addison Wesley Publishing Company, Reading, 1986.

[**Craig, 86b**] Craig J.J., Hsu P., Sastry S., "Adaptive control of mechanical manipulators". Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, avril 1986, p. 190-195.

[**Craig, 89**] J.J. Craig. "Introduction to Robotics: Mechanics and Control". Addison-Wesley, 2nd ed., 1989.

[**Cyril, 2009**] CYRIL QUENNOUELLE. "Modélisation Géométrico-Statique Des Mécanismes Parallèles Complains". Université Laval Québec : thèse de doctorat, 2009.

[**Denavit 55**] Denavit J., Hartenberg R.S., "A kinematic notation for lower pair mechanism based on matrices". Trans, of ASM E, J. of Applied Mechanics, Vol. 22. juin 1955, p. 215-221.

[**Dillon, 1973**] Dillon S.R., "Computer assisted equation generation in linkage dynamics". PhD. Thesis, Ohio State University, Aout 1973.

[**Dombre et Khalil, 1988**] Dombre, E. et Khalil, W. (1988). "Modélisation et commande des robots". Hermès.

[**Dubowsky, 1979**] Dubowsky S., Des Forges D.T., "The application of model-referenced adaptive control to robotic manipulators". Trans, of ASME, J. of Dynamic Systems, Measurement, and Control, Vol. 101. 1979, p. 193-200.

[**Egeland et al., 1988**] O. Egeland et E. Lunde (1988). "Trajectory generation for manipulators using linear quadratic optimal tracking". Dans IEEE, pages 376–381.

[**Elhami et Brookfield, 97**] M.R. Elhami and D.J. Brookfield. "Sequential Identification of Coulomb and Viscous Friction in Robot Drives". Automatica, Vol. 33, No. 3, pp. 393-401, 1997.

[**El Serafi, 91a**] El Serafi K., Khalil W., "Energy based indirect adaptive control of robots", Proc. IEEE Int. Conf. on Robotics and Automation, Sacramento, avril 1991. p. 2142-2147.

[**Er, 1993**] M. J. Er. "Recent developments and futuristic trends in robot manipulator control". Dans Asia-Pacific Workshop on Advances in Motion Control Proceedings, 1993. Pages 106–111.

[**Espiau et al., 1992**] Espiau, B., Chaumette, F., et Rives, P. (1992). "A new approach to visual servoing in robotics". IEEE Trans. on Automatic Control (ITAC), 8(3):313–326.

[**Fanping. B, Bin. Y, 2000**] Fanping Bu and Bin Yao. "Nonlinear Adaptive Robust Control of Hydraulic Actuators Regulated by Proportional Directional Control Valves with Dead band and Nonlinear Flow Gains". Proceedings of the American Control Conference Chicago, Illinois, Vol. 6, 2000, pp. 4129-4133.

[**Faverjon, 1984**] Faverjon, B. (1984). "Obstacle avoidance using an octree in the configuration space of a manipulator". Dans IEEE Conf. on Robotics and Automation (ICRA), Atlanta, USA.

[**Featherstone, 83a**] Featherstone R. "Position and velocity transformations between robot end-effector coordinates and joint angles". The Int. J. of Robotics Research. Vol. 2(2), 1983, p. 35-45.

[**Featherstone, 83b**] Featherstone R. "The calculation of robot dynamics using articulated-body inertias". The Int. J. of Robotics Research. Vol. 2(3). 1983. p. 87-101.

[**Flash et al., 1985**] T. Flash et N. Hogan (1985). "The coordination of arm movements: An experimentally confirmed mathematical model". The Journal of Neuroscience, 5 :1688–1703.

[**Fossard. A. J et al, 1993**] A. J. Fossard, D. Normand-Cyrot. "Systèmes non linéaires", Tome, Commande.

[**Fournier, 80**] Fournier A., "Génération de mouvements en robotique : application des inverses généralisées et des pseudo-inverses". Thèse d'Etat, USTL. Montpellier, avril 1980.

[**Frank L, 2004**] Frank L.Lewis, Darren M.Dawson, Chaouki T.Abdallah. "Robot Manipulator Control Theory and Practice". Second Edition, Revised and Expanded Copyright © 2004 by Marcel Dekker, Inc. chapter 4 p 188.

[**Gilbert and Ha 1984**] Gilbert, E.G., and I.J.Ha. "An approach to nonlinear feedback control with applications to robotics" IEEE Trans. Syst. Man Cybern., vol. SMC-14, no. 6, pp. s 879–884, Nov./Dec. 1984.

[**Goldenberg, 85**] Goldenberg A.A., Benhabib B., Fenton R.G. "A complete generalized solution to inverse kinematics of robots". IEEE J. of Robotics and Automation, Vol. RA-1 (1), 1985, p. 14-20.

[**Gorla et Renaud, 1984**] Gorla, B. et Renaud. M. "Modèle des robots manipulateurs. Application à leur commande". Cepadues Toulouse, 1984.

[**Guang LI, WU Min, 2005**] LI Guang and WU Min. "Modeling and controlling of a flexible hydraulic manipulator". Springer Journal of Central South University of Technology, Vol. 12, No. 5, 2005, pp. 578-583.

[**Hassan. K et Khalil, 2002**] Hassan K. Khalil. "Nonlinear Systems". 3^{ème} édition. 2002.

[**Hedjar et al., 2002**] R. Hedjar, R. Toumi, P. Boucher, et D. Dumur. "Feedback nonlinear predictive control of rigid link robot manipulators". Dans Proceedings of the American Control Conference, pages 3594–3599, Anchorage, 2002.

[**Horowitz, 80**] Horowitz R., Tomizuka M., "An adaptive control scheme for mechanical manipulators; compensation of non-linearity and decoupling control". Presentation at the Winter Meeting of ASME, Dynamic Systems and Control Division. Chicago. 1980.

[**Hsia, 86**] Hsia T.C. "Adaptive control of robot manipulators; a review". Proc. IEEE Int. Conf. on Robotics and Automation. San Francisco, avril 1986. p. 183-189.

[**Hunt et al. 1983**] Hunt, L.R., R.Su, and G.Meyer. "Global transformations of nonlinear systems". IEEE Trans. Autom. Control, vol. AC-28, no. 1, pp. 24–31, Jan. 1983.

[**Ignacio H. A, 2007**] Ignacio Herrera Aguilar. "Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service". Ecole doctorale systèmes. tel-00199179, version 1 - 18 Dec 2007.

[**Izaguirre 86**] Izaguirre A., Paul R.C.P. "Automatic generation of the dynamic equations of the robot manipulators using a LISP program". Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, avril 1986, p. 220-226.

[**J. J. Craig, 1989**] J. J. Craig, "Introduction to Robotics: Mechanics and Control". 2^{ème} édition, 1989.

[**Jean P. C, 1996**] Jean Pierre Carriou. "Commande des procédés". Génie des procédés de l'Ecole de Nancy, Paris 1996.

[**Kahn et al., 1971**] M. E. Kahn et B. E. Roth. "The near-minimum-time control of open-loop articulated kinematic chains". Trans. ASME J. Dynamic, Systems, Meas., Cont., 93:164–172. (1971).

[**Kelly, 88**] Kelly R., Ortega R. "Adaptive control of robot manipulators: an input- output approach". Proc. IEEE Int. Conf. on Robotics and Automation, Philadelphia, avril 1988. p. 699-703.

[**Khalil 76**] Khalil W. "Modélisation et commande par ordinateur du manipulateur MA- 23 ; extension à la conception par ordinateur des manipulateurs". Thèse de Docteur- Ingénieur, USTL, Montpellier, sept. 1976.

[**Khalil, 1978**] Khalil, W. "Contribution à la commande automatique des manipulateurs avec l'aide d'un modèle mathématique des mécanismes". Thèse de Doctorat, université de Montpellier (1978).

[**Khalil et al., 1979**] Khalil, W., Liegois, A., et Fournier, A. "Commande dynamique des robots". RAIRO automatique / Systems analysis and control, 13(2):189–201 (1979).

[**Khalil 86a**] Khalil W., Kleinfinger J.-F. "A new geometric notation for open and closed-loop robots". Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, avril 1986, p. 1174-1 180.

[**Khalil 89a**] Khalil W., Bennis F., Chevallereau C., Kleinfinger J.-F. "SYMORO: a software package for the symbolic modelling of robots". Proc. 20th Int. Symp, on Industrial Robots. Tokyo, oct. 1989, p. 1023-1030.

[**Khalil 97a**] Khalil W., Creusol D. "SYMORO+: a system for the symbolic modelling of robots". Robotica, Vol. 15, 1997, p. 153-161.

[**Khalil et Dombre, 1999**] Khalil, W. et Dombre, E. "Modélisation, identification et commande des robots". Hermes Science Publications, Paris, 1999.

[**Khelfi, 95**] Khelfi M.-F. "Observateurs non linéaires : application à la commande des robots manipulateurs". Thèse de Doctorat, Université Poincaré Nancy I, 1995.

[**Kim et al., 1985**] B. K. Kim et K. G. Shin. "Minimum-time path planning for robot arms and their dynamics". IEEE Trans. on Systems, Man and Cybernetics, SMC-15:213–223 (1985).

[**Kircânski 85**] Kircânski M., Vukobratovic M. "Computer-aided generation of manipulator kinematic models in symbolic form". Proc. 15th Int. Symp, on Industrial Robots, Tokyo, sept. 1985, p. 1043-1049.

[**Kreuzer 79**] Kreuzer E.J. "Dynamical analysis of mechanisms using symbolical equation manipulation". Proc. 51th World Congress on Theory of Machines and Mechanisms, Montréal, 1979, p. 599-602.

[**Kyriakopoulos et al., 1988**] K. J. Kyriakopoulos et G. N. Saridis. "Minimum jerk path generation". Dans Proc. IEEE International Conference on Robotics and Automation, 1988 pages 364–369.

[**LAN, 1979**] LANDAU Y.D. "Adaptive Control. The Model Reference Approach Dekker". Inc. 79.

[**Landau, 88**] Landau I. D., Horowitz R. "Synthesis of adaptive controllers for robot manipulators using a passive feedback system approach". Proc. IEEE Int. Conf. on Robotics and Automation, Philadelphia, avril 1988, p. 1028-1033.

[**Lewis, 93**] Lewis F.L., Abdallah C.T., Dawson D.M. "Control of robot manipulators, Macmillan". New York. 1993.

[**Li, 89**] Li W., Slotine J.-J.E. "An indirect adaptive robot controller". Systems & Control Letters, Vol. 12. 1989. p. 259-266.

[**Liégeois, 1977**] Liégeois, A. "Automatic supervisory control of the configuration and behavior of multibody mechanisms". IEEE Transactions Systems Man Cybernetics, 7:868–871 (1977).

[**Lin et al., 1983**] C. S. Lin, P. R. Chang, et J. Y. S. Luh. "Formulation and optimization of cubic polynomial joint trajectories for industrial robots". IEEE Transactions on Automatic Control, AC- 28:1066–1074 (1983).

[**Lozano et al., 2001**] R. Lozano et D. Taoutaou. "Commande adaptative et applications". Hermes Science Publications, Paris, 2001.

[**Luh et al., 1980**] Luh, J., Walker, M., et Paul, R. "Resolved acceleration control of mechanical manipulators". IEEE Trans. on Automatic Control (ITAC), 25(3):468–474. (1980).

[**Mark W et al, 2004**] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. "Robot Dynamics and Control". 2^{ème} édition, 2004.

[**Martin et al., 1999**] B. J. Martin et J. E. Bobrow. "Minimum-effort motions for open-chain manipulators with task-dependant end-effector constraints". The International Journal of Robotics Research, 18:213–224. (1999).

[**Megahed, 82**] Megahed S., Renaud M. "Minimization of the computation time necessary for the dynamic control". Proc. 12th Int. Symp, on Industrial Robots, Paris, juin 1982, p. 469-478.

[**Megahed 84**] Megahed S. "Contribution à la modélisation géométrique et dynamique des robots manipulateurs ayant une structure de chaîne cinématique simple ou complexe ; application à leur commande". Thèse d'Etat. UPS. Toulouse, juillet 1984.

[**Middleton, 88**] Middleton R.H., Goodwin G.C. "Adaptive computed torque control for rigid link manipulators". Systems & Control Letters, Vol. 10, 1988. p. 9-16.

[**Murray 84**] Murray J.J., Newman C.P. "ARM: an algebraic robot dynamic modeling program". Proc. IEEE Int. Conf, on Robotics and Automation, Atlanta, mars 1984, p. 103-104.

[**Nicosia, 84**] Nicosia S., Tomei P. "Model reference adaptive control algorithms for industrial robots". Automatica, Vol. 20(5), 1984, p. 635-644.

[**Nicosia, 90**] Nicosia S, Tomei P. "Robot control by using only joint position measurements". IEEE Trans, on Automatic Control, Vol. AC-35(5), 1990, p. 1058-1061.

[**Ortega, 1989**] Ortega R., Spong M.W. "Adaptive motion control of rigid robots: a tutorial". Automatica, Vol. 25(6), 1989, p. 877-888.

[**Patricia, 1987**] Patricia CONTI. " Contribution à la commande dynamique adaptative des robots manipulateurs". Thèse doctorat, université paul sabatier de TOULOUSE (Sciences) 1987. Rapport LA AS n° 87.350 7, avenue du Colonel-Roche 31077 TOULOUSE CEDEX p. 34.

[**Paul, 1979**] R. P. Paul. "Manipulator cartesian path control". IEEE Trans. Syst., Man Cybern. 9:702–711 (1979).

[**Paul 81**] Paul R.C.P., "Robot manipulators: mathematics, programming and control". MIT Press. Cambridge, 1981.

[**Pieper, 68**] Pieper D.L. "The kinematics of manipulators under computer control". Ph.D. Thesis, Stanford University, 1968.

[**Raghavan, 90**] Raghavan M., Roth B. "Inverse kinematics of the general 6R manipulator and related linkages". Trans, of the ASME, J. of Mechanical Design, Vol. 115, 1990, p. 502-508.

[**Renaud 75**] Renaud M. "Contribution à l'étude de la modélisation et de la commande des systèmes mécaniques articulés". Thèse de Docteur-Ingénieur, UPS, Toulouse, déc. 1975.

[**Renaud, 85**] Renaud M. "A near minimum iterative analytical procedure for obtaining a robot-manipulator dynamic model". LUTAM/IFTtoMM Symp, on Dynamics of Multi body Systems, Udine, 1985.

[**Sadegh, 87**] Sadegh N., Horowitz R. "Stability analysis of an adaptive controller loi robotic manipulators". Proc. IEEE Int. Conf. on Robotics and Automation. Raleigh, mars-avril 1987, p. 1223-1229.

[**Samson et al., 1991**] Samson, C. et Ait-Abderrahim, K. "Feedback control of a non holonomic wheeled cart in cartesian space". Dans IEEE Conf. on Robotics and Automation (ICRA), pages 1136–1141 (1991).

[**Samson et al., 1991**] Samson, C., Leborgne, M., et Espiau, B. "Robot Control: The Task-function Approach. Oxford Engineering". Oxford University Press (1991).

[**Sayers, 96**] C.P. Sayers, D.R. Yoerger, R.P. Paul and J.S. Lisiewicz. "A Manipulator Work Package for Teleoperation from Unmanned Untethered Vehicles - Current Feasibility and Future Applications". 6th IARP Workshop on Underwater Robotics, Toulon, France, 27-29 March 1996.

[**Sciavicco, 86**] Sciavicco L., Siciliano B. "Coordinate transformation; a solution algorithm for one class of robots". IEEE Trcins, on Systems, Man, and Cybernetics. Vol. SMC-16(4), 1986, p. 550-559.

[**Sciavicco et Siciliano, 1988**] Sciavicco, L. et Siciliano, B. "A solution algorithm to the inverse kinematic problem for redundant manipulators". IEEE Journal Robotics and Automation, 4:403–410 (1988).

[**Sciavicco et al, 2000**] L. Sciavicco et B. Siciliano. "Modelling and Control of Robot Manipulators". Springer, 2000.

[**Seki et al., 2004**] K. Seki et S. Tadakuma. "Minimum jerk control of power assiting robot based on human arm behavior characteristic". Dans International Conference on Systems, Man and Cybernetics, pages 722–721 (2004).

[**Sheth 71**], Sheth P.N., Uicker J.J. "A generalized symbolic notation for mechanism". Trans, of ASME, J. of Engineering for Industry, Vol. 93, 1971, p. 102-112.

[**Shin et al., 1985**] K. G. Shin et N. D. McKay. "Minimum-time control of robotic manipulators with geometric path constraints". IEEE Transactions on Automatic Control, AC-30:531–541 (1985).

[**Siciliano, 1990**] Siciliano, B. "Kinematic control of redundant robot manipulators: a tutorial". Journal of Intelligent and Robotic Systems, 3(3): 201–212 (1990).

[**Slotine, 1985**] J. J. E. Slotine. "Robust control of robot manipulators". The International Journal of Robotics Research, 4:49–64 (1985).

[**Slotine, 87**] Slotine J.-J.E., Li W. "Adaptive manipulator control: a case study". Proc. IEEE Int. Conf. on Robotics and Automation, Raleigh, mars-avril 1987, p. 1312-1400.

[**Slotine et al., 1987**] J. J. E. Slotine et W. Li. "Adaptive strategies in constrained manipulation". Dans Proc. IEEE Int. Conf. on Robotics and Automation, volume Raleigh, N.C., pages 595–601. (1987).

[**Stringer, 82**] J. D. Stringer. "Hydraulic Systems Analysis", MacMillan Press, 1982.

[**Spong, 89**] Spong M.W., Vidyasagar M. "Robot dynamics and control". John Wiley & Sons, New York. 1989.

[**Takegaki, 81a**] Takegaki M., Arimoto S. "An adaptive trajectory control of manipulators". Int. J. Control, Vol. 34(2), 1981, p. 219-230.

[**Thayer, 65**] W.J. Thayer. "Transfer Functions for MOOG Servovalves". MOOG Technical Bulletin 103, MOOG Controls, New York, 1965.

[**Tien D et al, 2013**] Tien Dung Le, Hee-Jun Kang, Young-Soo Suh, Young-Shick Ro. "An online self-gain tuning method using neural networks for nonlinear PD computed torque controller of a 2-dof parallel manipulator". Elsevier Neurocomputing, Vol. 116, 2013, pp. 53-61.

[**TOS, 1987**] TOSUNOQLU S. "A Survey of Adaptive Control Technology in Robotics". Communication Personnelle- TESÀR D.

[**Vukobratovic et Kircanski, 1984**] Vukobratovic, M. et Kircanski, M. "A dynamic approach to nominal trajectory synthesis for redundant manipulators". IEEE Trans. Systems Man Cyber. 14:580–586 (1984).

[**Walker, 82**] Walker M.W., Orin D.E. "Efficient dynamic computer simulation of robotics mechanism". Trans, of ASME, J. of Dynamic Systems. Measurement, and Control, Vol. 104. 1982, p. 205-211.

[**Weiwei S, Shuang C, 2009**] Weiwei Shang, Shuang Cong. "Nonlinear computed torque control for a high-speed planar parallel manipulator". Elsevier Mechatronics, Vol. 19, No. 6, 2009, pp. 987-992.

[**Wenger, 89**] Wenger P. "Aptitude d'un robot manipulateur à parcourir son espace de travail en présence d'obstacles". Thèse de Doctorat, ENSM. Nantes, sept. 1989.

[**Whitney, 69**] Whitney D.E. "Resolved motion rate control of manipulators and human prostheses". IEEE Trans, on Man Machine Systems, Vol. MMS-10(2). juin 1969. p. 47-53.

[**Wisama et Dombre, 1999**] Wisama Khalil, Etienne Dombre. "Modélisation, identification et commande des robots". 2^{ème} Edition revue et augmentée. Paris Hermès Science publications, 1999. ISBN 2-7462-0003-1.

[**Withney, 1977**] Withney, D. "Force feedback control of manipulator fine motions". Journal of Dynamic Systems, Measures and Control, pages 91–97 (1977).

[**Withney, 1985**] Withney, D. (1985). "Historical perspective and state of the art in robot force control". Dans IEEE Conf. On Robotics and Automation (ICRA), pages 262–268.

[**Wolovich, 84**] Wolovich W.A., Elliott H. "A computational technique for inverse kinematics". Proc. 23rd IEEE Conf. on Decision and Control, Las Vegas, dec. 1984, p. 1359-1363.

[**Wu et al., 1982**] C. H. Wu et R. P. Paul. "Resolved motion force control of robot manipulator". IEEE Trans. Syst., Man Cybern., pages 266–275 (1982).

[**Y. BAAZI, 2000**] Y. BAAZI. "Etude d'un Manipulateur à 3 Degrés de Liberté Application à la Fonction de Préhenseur". Thèse de magister, Institut d'électronique, Université de Batna, 2000.

[**Yoshikawa, 1985**] Yoshikawa, T. (1985). "Manipulability of robotic mechanisms". International Journal of Robotics Research (IJRR), 4(2):3–9.

[**Yoshikawa, 1990**] Yoshikawa, T. "Foundations of robotics: Analysis and control". MIT Press (1990).

[**Zabala 78**] Zabala Iturralde J. "Commandes des robots manipulateurs à partir de la modélisation de leurs dynamique". Thèse de Troisième Cycle, UPS, Toulouse, juillet 1978.

[**Zodiac, 96**] The Zodiac, C. Canudas de Wit. B. Siciliano, G. Bastin Eds. "Theory of robot control". (Ouvrage collectif). Springer-Verlag, Berlin, 1996.

[**Zuoshi S et al, 2005**] Zuoshi Song, Jianqiang Yi, Dongbin Zhao, Xinchun Li. "A computed torque controller for uncertain robotic manipulator systems: Fuzzy approach". Elsevier Fuzzy Sets and Systems, Vol. 154, No. 2, 2005, pp. 208-226.

ADAPTIVE TRACKING CONTROL OF HYDRAULIC ROBOT MANIPULATOR USING HYBRID INTELLIGENT SYSTEM (ANFIS)

Makhloufi Fateh ^{a,*}, Guersi Noureddine ^b, Frihi Mohamed ^c

^a Department of Mechanical Engineering. University Badji Mokhtar (UBMA), BP12-23000, Annaba, Algeria.

^{b,c} Department of electronic. University Badji Mokhtar (UBMA), BP12-23000, Annaba, Algeria.

ABSTRACT

The motion control of an experimental hydraulically actuated robot with structured uncertainty (parameter uncertainty, unknown loads, inaccuracies in the torque constants of the actuators, and others) and unstructured uncertainty (high-frequency modes, neglected time-delays, unknown friction forces, stick-slip oscillations, and unknown oil viscosity, etc...) is considered. As a solution we propose two control techniques based on ANFIS: Adaptive Neuro Fuzzy Inference System based computed torque controller (type PD), and Adaptive Neuro Fuzzy Inference System based PD plus I controller. Comparative evaluations with respect to conventional PD controller are presented to validate the controllers design. The simulated and experimental results presented emphasize that a satisfactory tracking precision could be achieved using ANFIS controllers.

Keywords

ANFIS based computed torque controller (type PD), ANFIS based computed torque controller (type PD) plus I, ANFIS based PD plus I controller, hydraulically actuated robot arm.

**International Conference on
Electromechanical Engineering (ICEE'2012)**
Skikda, Algeria, 20-22 November 2012

Adaptive Neuro-Fuzzy Inference System for Control of a Manipulator Arm with uncertain Dynamic Model

Makhloufi Fateh

Mechanical Engineering Department
Laboratory of Industrial Mechanics (LMI)
Faculty of Science of Engineer University
Badji Mokhtar Annaba
E-mail: fatehlmd@yahoo.fr

Guersi Noureddine

Department of electronics
Laboratory of Industrial automatic (LASA)
Faculty of Science of Engineer University
Badji Mokhtar Annaba
E-mail: guersi54@yahoo.fr

Abstract— In case of an imprecise knowledge of robot parameters and / or the presence of some non-modeled dynamics, control by torque calculation shows its limits. The objective of this paper is to propose a control method for manipulators, using a system of adaptive neuro-fuzzy inference (ANFIS) whose role is to adjust, continuously, the parameters K_p and K_v to compensate for the neglected parts of the dynamic model, and a conventional integral action (I) for the accuracy of trajectory tracking. To demonstrate the efficiency and accuracy of this method, a comparative evaluation with a neuronal controller (MLP) with an adaptive learning rate [14], and a conventional computed torque control is performed. The results are presented and discussed.

Keywords— Fuzzy Logic, Neural networks, ANFIS, PD control.