

وزارة التعليم العالي و البحث العلمي

MINISTÈRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

جامعة سطيف 1

UNIVERSITÉ SETIF 1

**MÉMOIRE**

Présenté au Département d'Électrotechnique

Faculté de Technologie

Pour l'obtention du Diplôme de

**Magistère En Automatique**

Par

***AKKA Khaled***

Thème

***Développement d'un contrôleur flou optimal pour des  
systèmes non linéaires***

*Soutenu publiquement le : 28/05 /2014 devant la commission d'examen composée de:*

<i>RAHMANI Lazhar</i>	<i>Prof. à l'université Ferhat ABBAS de Sétif</i>	<i>Président</i>
<i>KHABER Farid</i>	<i>Prof. à l'université Ferhat ABBAS de Sétif</i>	<i>Rapporteur</i>
<i>LAMAMRA Athmane</i>	<i>M. C. à l'université Ferhat ABBAS de Sétif</i>	<i>Examineur</i>
<i>HEMSAS Kamel Eddine</i>	<i>M. C. à l'université Ferhat ABBAS de Sétif</i>	<i>Examineur</i>

# *REMERCIEMENTS*

Le travail présenté dans ce mémoire a été effectué au Laboratoire de la Qualité d'Energie dans les Réseaux Electriques (Q.U.E.R.E) du département d'Electrotechnique à l'université Ferhat Abbas de Sétif.

J'adresse mes sincères remerciements à Monsieur ***KHABER Farid***, Professeur à l'université Ferhat Abbas de Sétif, pour la proposition du sujet de ce mémoire et pour ses conseils et ses critiques objectifs tout au long de ce travail.

Je remercie également Monsieur ***RAHMANI Lazhar***, Professeur à l'université Ferhat Abbas de Sétif, pour avoir accepté la présidence du jury, Messieurs ***LAMAMRA Athmane*** et ***HEMSAS Kamel Eddine***, maitres de conférence à l'université Ferhat Abbas de Sétif d'avoir accepté d'examiner et d'évaluer ce travail.

Je veux également remercier tous les membres de ma famille pour leurs encouragements, leur soutien moral et la patience qu'ils m'ont manifestés durant mes études.

Enfin, je remercie tous mes amis (es).

---

*TABLE DES MATIERES*

---

<b>INTRODUCTION GENERALE</b> - - - - -	1
<b>CHAPITRE I    COMMANDE OPTIMALE LINEAIRE QUADRATIQUE                   ET OPTIMISATION PAR ESSAIM PARTICULAIRE</b>	
I. Introduction - - - - -	6
I.1 Optimisation par essaim particulaire - - - - -	6
I.1.1 Introduction - - - - -	6
I.1.2 Problème d’optimisation - - - - -	6
I.1.3 Fonction objectif - - - - -	7
I.1.4 Variables de décision - - - - -	7
I.1.6 Minimum global - - - - -	7
I.1.5 Minimum local - - - - -	7
I.1.6 Optimisation mono-objectif - - - - -	8
I.1.7 Optimisation multi-objectif - - - - -	8
I.1.8 Principe général de l’optimisation par essaim particulaire - - - - -	9
I.1.9 Formalisation de l’optimisation par essaim particulaire - - - - -	10
I.1.10 Améliorations de l’optimisation par essaim particulaire - - - - -	11
I.1.10.1 Confinement des particules - - - - -	11
I.1.10.2 Coefficient de constriction - - - - -	11
I.1.10.3 Coefficient d’inertie - - - - -	12
I.1.10.4 Topologie de voisinage - - - - -	12
I.2 Commande optimale linéaire quadratique - - - - -	13
I.2.1 Introduction - - - - -	13
I.2.2 Linéarisation - - - - -	14

## TABLE DES MATIERES

---

I.2.2.1	Linéarisation d'une fonction	14
I.2.2.2	linéarisation d'un système	15
I.2.2.3	Linéarisation du système autour d'un point de fonctionnement	16
I.2.3	Notions de contrôlabilité et d'observabilité	16
I.2.3.1	Contrôlabilité	16
I.2.3.2	Observabilité	17
I.2.4	Formulation du problème LQ et sa solution	17
I.2.4.1	Commande LQ à horizon fini	17
I.2.4.2	Commande LQ à horizon infini	19
I.2.5	Choix des matrices de pondération	22
I.3	Applications	24
I.3.1	Exemple 1	24
I.3.2	Exemple 2	27
I.4	Conclusion	30
 <b>CHAPITRE II    COMMANDE ADAPTATIVE FLOUE POUR DES                   SYSTEMES NON LINEAIRES</b>		
II.1	Introduction	31
II.2	Définitions	31
II.2.1	Commande adaptative	31
II.2.1.1	Commande adaptative indirecte	32
II.2.1.2	Commande adaptative directe	32
II.2.2	Commande floue	32
II.2.2.1	Systèmes flous	33
II.2.2.1.1	Fuzzification	33

## TABLE DES MATIERES

---

II.2.2.1.1.1 Fuzzification par singleton - - - - -	33
II.2.2.1.1.2 Fuzzification non singleton - - - - -	34
II.2.2.1.2 Base de règles - - - - -	34
II.2.2.1.3 Moteur d'inférence - - - - -	34
II.2.2.1.4 Défuzzification - - - - -	35
II.2.2.2 Structure générale et différents types de modèles flous - - - - -	35
II.2.2.2.1 Système flou de Mamdani - - - - -	35
II.2.2.2.2 Système flou de Takagi-Sugeno - - - - -	36
II.2.3 Approximation floue - - - - -	36
II.3 Commande adaptative floue - - - - -	37
II.3.1 Commande adaptative floue indirecte - - - - -	38
II.3.1.1 Formulation du problème - - - - -	38
II.3.1.1.1 Supposition 1 - - - - -	41
II.3.1.1.2 Supposition 2 - - - - -	43
II.3.1.2 Conception du contrôleur adaptatif flou - - - - -	43
II.3.1.3 Exemple de simulation - - - - -	45
II.3.1.3.1 Paramètres d'initialisation - - - - -	45
II.3.1.3.2 Mise en œuvre de l'algorithme de conception - - - - -	46
II.3.1.3.3 Résultats de simulation - - - - -	46
II.3.2 Commande adaptative floue direct - - - - -	48
II.3.2.1 Formulation du problème- - - - -	48
II.3.2.2 Exemple de simulation - - - - -	55
II.3.2.2.1 Paramètres d'initialisation - - - - -	55



## TABLE DES MATIERES

---

III.6 Conclusion - - - - -	87
<b>CONCLUSION GENERALE</b> - - - - -	88
<b>BIBLIOGRAPHIE</b> - - - - -	90

---

*INTRODUCTION GENERALE*

---

## INTRODUCTION GENERALE

Après la deuxième guerre mondiale et les progrès technologiques considérables qui ont suivi cette tragédie, l'automatique a connu un véritable essor. L'élaboration de lois de commande de systèmes a fait l'objet de nombreux travaux de recherche. Les systèmes à commander devenant de plus en plus complexes, rendant ainsi l'obtention d'un modèle du procédé pratiquement impossible, et par conséquent la mise en œuvre des lois de commande correspondantes devient une tâche très difficile.

L'un des enjeux de l'automatique est de proposer un contrôleur adapté au système à piloter garantissant la réalisation de la tâche souhaitée. De nombreuses approches, pour les systèmes continus, ont été proposées pour la synthèse de lois de commande et peuvent être classées en deux catégories dites « linéaires » où « non linéaires ».

La commande des systèmes non linéaires reposait sur la théorie de l'automatique linéaire qui consiste à linéariser la dynamique non linéaire du système autour d'un point de fonctionnement de telle sorte que les outils de la commande linéaire puissent être exploités pour la synthèse d'une loi de commande assurant les performances recherchées [1]. Grâce à sa facilité d'implémentation et de mise en œuvre, cette approche a remporté un grand succès dans les domaines industriels. La commande optimale représente la technique de contrôle classique la plus utilisée dans cette approche. Cette commande traduite par le contrôle linéaire quadratique (LQ) est étudiée et détaillée dans [2], [3], [4]. En fait, pour un système dynamique donné et dont les équations sont connues, le problème de commande optimale consiste à trouver une loi minimisant un critère de performance donné. L'obtention d'un bon compromis entre l'énergie d'entrée et l'énergie de sortie nécessite un choix convenable des matrices de pondération ( $Q$ ,  $R$ ) du régulateur LQR. Actuellement, dans la majorité des travaux, ce choix est réalisé par l'utilisation des techniques d'optimisation.

Un problème d'optimisation est défini par un ensemble de variables, une fonction "objectif" et un ensemble de contraintes. L'espace de recherche est l'ensemble des solutions possibles du problème. Résoudre un problème d'optimisation consiste à trouver la ou les meilleures solutions (en minimisant et/ou maximisant la/les fonctions objectifs du problème posé), tout en satisfaisant un ensemble de contraintes définies par l'utilisateur. Certains problèmes d'optimisation sont qualifiés de difficiles, et leur résolution, dans un temps

raisonnable, nécessite l'utilisation d'algorithmes sophistiqués, comme les méthodes approchées (Les heuristiques et les métaheuristiques). Parmi les métaheuristiques destinées à résoudre ces problèmes, plus précisément l'optimisation par essaim particulaire (OEP, ou PSO en anglais) qui est apparue en 1995 [5].

L'optimisation par essaim particulaire est une méthode d'optimisation stochastique qui est inspirée d'un comportement social des animaux évoluant en essaim. Ce comportement social est modélisé par une équation mathématique permettant de guider les « particules » durant le processus de déplacement. Le déplacement d'une particule est influencé par trois composantes : la composante d'inertie, la composante cognitive et la composante sociale. Chacune de ces composantes reflète une partie de l'équation. L'OEP présente l'avantage d'être efficace sur une grande variété de problèmes, sans pour autant que l'utilisateur ait à modifier la structure de base de l'algorithme [5], [6].

L'automatique linéaire repose sur la linéarisation du système autour d'un point de fonctionnement donné, c'est-à-dire le système n'est valable que dans une plage de fonctionnement restreinte autour de ce point. Alors, les performances du modèle linéaire se dégradent dès qu'on s'en éloigne. L'automatique non linéaire basée sur des approches non linéaires permet de garantir les performances du système dans une région plus large de l'espace d'état [7], [8]. Parmi ces approches, on distingue, les approches qui ne nécessitent pas la connaissance d'un modèle formalisé, comme la commande floue.

La logique floue a été largement utilisée pour sa capacité à résoudre les problèmes de modélisation et de commande des systèmes non linéaires. De plus, elle permet, d'une part, d'exploiter efficacement l'expertise humaine à travers les différentes informations linguistiques et, d'autre part, d'utiliser des techniques issues de la commande des systèmes linéaires comme la commande adaptative.

Concevoir un système asservi avec de bonnes performances nécessite que le correcteur soit adapté convenablement, non seulement pour un point de fonctionnement particulier appelé point nominal, mais aussi pour un ensemble de points de fonctionnement de l'espace d'état.

Avec les développements successifs de l'automatique moderne, on s'est aperçu qu'un régulateur statique ne pouvait pas toujours fournir un comportement acceptable du système en toute situation. En particulier, lorsque les processus à commander possèdent des paramètres

variant dans le temps, la conception d'un régulateur satisfaisant toujours les spécifications requises s'avère très difficile. La notion d'ajustement automatique des paramètres d'un régulateur commence à pointer vers la fin des années 40. A ce moment précis et jusqu'en 1960, le terme de « Commande Adaptative » n'a pas encore la signification que l'on connaît aujourd'hui, mais traduit plutôt la capacité d'un régulateur à ajuster ses paramètres aux comportements dynamique et statique d'un procédé. Ce n'est qu'avec l'apparition des ordinateurs puis des microprocesseurs que la commande adaptative trouvera un essor certain, concrétisé en 1981 par la sortie du premier régulateur numérique adaptatif. Même si la mise en œuvre d'un algorithme adaptatif est relativement lourde, avec les progrès réalisés en micro-électronique, permettant de faire une implantation plus rapide, plus simple, plus performante et moins chère. Les différentes méthodes de commande adaptative se différencient par la structure choisie pour réaliser la mise à jour en temps réel des paramètres du régulateur en fonction des variations du processus à commander. On distingue la commande adaptative directe dont laquelle les paramètres du régulateur sont ajustés directement en temps réel, et la commande adaptative indirecte, qui suppose une estimation du processus par une procédure d'identification [9], [10].

La synthèse de la commande par l'approche floue ne nécessite pas de modélisation préalable, mais seulement des informations émanant d'experts humains. L'utilisation des systèmes flous dans la commande a été introduite depuis les années 70 par Mamdani et son équipe [11]. La mise en œuvre de ces contrôleurs était heuristique et n'obéissait à aucune méthodologie précise. Malgré son succès, la logique floue a été mal acceptée par les automaticiens du fait qu'elle ne prend pas en compte les critères classiques de l'automatique (stabilité, robustesse). Il fallait attendre les années 80 pour avoir un type de contrôleur plus compatible avec les approches connues. En effet Takagi et Sugeno ont présenté un système flou dont la partie conséquence (conclusion) est fonctionnelle, ce qui permet d'utiliser les méthodes classiques de l'automatique [12]. Néanmoins, pour maintenir les mêmes performances en présence de grandes variations structurelles, l'utilisation d'une commande adaptative floue s'avère nécessaire. Dans ces techniques, la propriété d'approximation universelle que possèdent les systèmes flous [13], [14] est exploitée. La stabilité et la robustesse de la structure de commande proposée sont étudiées par la méthode de Lyapunov.

Dans la littérature, la commande adaptative floue a reçu beaucoup d'attention de plusieurs filières et aussi dans plusieurs applications où des performances intéressantes ont été démontrées. Les contrôleurs adaptatifs flous stables sont proposés en premier par Wang [13].

Après, plusieurs approches de la commande adaptative floue pour les systèmes non linéaires ont été développées [15], [16], [17]. Ces approches ont été appliquées seulement à une classe de systèmes non linéaires où les non linéarités inconnues paraissent sur la même équation d'état.

Plusieurs approches basées sur les systèmes flous de Takagi-Sugeno (TS) ont fournis des résultats satisfaisants dans le domaine de la commande des systèmes non linéaires [16], [17], [18]. Cette méthodologie, basée sur les systèmes flous TS, peut être appliquée d'une manière efficace pour le développement de structures de commande floues optimales. La caractéristique principale d'un modèle flou de Takagi-Sugeno est qu'il va exprimer la dynamique locale de chaque règle floue par un modèle linéaire. Cette approche a été appliquée avec beaucoup de succès pour la commande des systèmes non linéaires [18], [19].

Une approche typique de la commande des systèmes utilisant les modèles TS est la compensation parallèle distribuée PDC (Parallel Distributed Compensation) [20], [21]. La majorité des travaux qui s'inspirent de cette loi de commande ont une démarche basée sur la synthèse de Lyapunov. La formulation de celle-ci aboutit à un ensemble d'inégalités écrites, dans la mesure du possible, sous la forme d'inégalités matricielles linéaires (LMI). Les LMI, s'il s'avère qu'elles admettent une solution, peuvent être résolues à l'aide d'outils issus du domaine de l'optimisation convexe.

Pour atteindre un certain degré de performance, il peut être proposé un contrôleur optimal quadratique où le problème de la commande est réduit à la minimisation d'un critère de performance quadratique [22]. La conception des contrôleurs PDC peut être faite par deux approches, la minimisation directe du critère de performance (commande PDC sous optimale) [23], [24] et la minimisation de la borne supérieure du critère de performance (commande PDC optimale) sous les contraintes de stabilité.

Le thème principal de ce travail est le développement de structures de commande optimale floue pour prendre en charge des systèmes non linéaires mono-entrée mono-sortie (SISO) et multi-entrées multi-sorties (MIMO). La mise au point de cette stratégie de commande nécessite la combinaison de l'approche floue et de la commande optimale linéaire quadratique. La stabilité du système en boucle fermée est étudiée au sens de Lyapunov. L'efficacité de ces structures de commande est mise en évidence sur des exemples de simulation sous l'environnement Matlab.

Le mémoire présenté est composé essentiellement de trois chapitres:

Dans le premier chapitre, consacré à la commande optimale linéaire quadratique et l'optimisation par essaim particulière, nous commencerons par énoncer les fondements de l'optimisation, et nous présentons en détails l'algorithme de l'OEP. Ensuite nous donnerons des rappels sur la linéarisation, la notion de commandabilité et d'observabilité, et nous terminerons par une étude détaillée de la commande linéaire quadratique.

Dans le deuxième chapitre nous présenterons un aperçu sur l'approche floue, la commande adaptative et la commande floue. Ensuite nous développerons deux lois de commande adaptative floue, l'une indirecte pour un système non linéaire mono-variable, et l'autre directe pour un système non linéaire multi-variables.

Le troisième chapitre traite la synthèse de la commande optimale pour des modèles flous T-S. Deux approches de synthèse seront présentées afin de garantir l'optimalité et la stabilité. La première approche est élaborée dans le cas de la minimisation du critère de performance pour chaque sous-système, ce qui implique la stabilité locale. La stabilité globale sera testée, dans le cas d'instabilité nous changerons les points de fonctionnement. La deuxième approche consiste à minimiser une borne supérieure du critère de performance sous les contraintes de la stabilité du système global en boucle fermée, la synthèse de la commande exploite le formalisme des LMIs.

Le bilan des travaux réalisés et les perspectives envisagées est donné en conclusion.

---

*CHAPITRE I*

*COMMANDE OPTIMALE  
LINEAIRE QUADRATIQUE ET  
OPTIMISATION PAR ESSAIM  
PARTICULAIRE*

---

## COMMANDE OPTIMALE LINEAIRE QUADRATIQUE ET OPTIMISATION PAR ESSAIM PARTICULAIRE

### I. Introduction

La théorie de la commande optimale a été largement utilisée pour résoudre les divers problèmes de l'automatique linéaire. La mise en œuvre de la commande optimale linéaire quadratique basée sur la minimisation d'un critère de performance quadratique a reçu beaucoup d'intérêt des chercheurs. Lors de la synthèse du régulateur LQR, le choix des matrices de pondération n'obéit pas à une démarche systématique. Pour surmonter ce problème, les automaticiens ont commencé à introduire les techniques d'optimisation stochastique, comme l'optimisation par essaim particulaire, pour proposer un choix optimal de ces matrices.

#### I.1 Optimisation par essaim particulaire

##### I.1.1 Introduction

L'optimisation par essaim particulaire est apparue en 1995 [5], est une méthode d'optimisation stochastique qui est inspiré d'un comportement social des animaux évoluant en essaim. Ce comportement social est modélisé par une équation mathématique permettant de guider les « particules » durant le processus de déplacement.

Dans cette première section du chapitre I, nous allons donner les concepts de base de l'optimisation, tels que la fonction objectif, les variables de décision, le minimum local, le minimum global, l'optimisation mono-objectif et multi-objectif. Ensuite, nous allons traiter en détail l'algorithme d'optimisation par essaim particulaire. La deuxième section est consacrée à la commande optimale linéaire quadratique.

##### I.1.2 Problème d'optimisation

Un problème d'optimisation au sens général est défini par un ensemble de variables, une fonction objectif  $f$  et un ensemble de contraintes d'égalité (ou d'inégalité) que les variables doivent satisfaire. L'ensemble des solutions possibles du problème forme l'espace de recherche  $D$ , où chaque dimension correspond à une variable. L'espace de recherche  $D$  est

fini puisque le décideur précise exactement le domaine de définition de chaque variable entre autres pour des raisons de temps de calcul. Suivant le problème posé, nous cherchons à minimiser ou maximiser la fonction “objectif”  $f$ . Un problème d’optimisation peut être statique ou dynamique (i.e. la fonction objectif change avec le temps), mono-objectif ou multi-objectif (i.e. plusieurs fonctions objectifs doivent être optimisées) et avec ou sans contraintes. Ainsi, un problème d’optimisation peut être, par exemple, à la fois continu et dynamique.

### I.1.3 Fonction objectif

On l’appelle aussi fonction de coût, critère d’optimisation ou fonction fitness, c’est la fonction  $f$  que l’algorithme d’optimisation va devoir optimiser (trouver leur optimum).

### I.1.4 Variables de décision

Elles sont regroupées dans le vecteur  $\vec{x}$ . C’est en faisant varier ce vecteur que l’on recherche un optimum de la fonction  $f$ .

### I.1.6 Minimum global

On a la fonction  $f: \Omega \subseteq \mathfrak{R}^n \rightarrow \mathfrak{R}$ , tel que  $\Omega \neq \emptyset$ . Pour  $x^* \in \Omega$ , on dit que  $x^*$  est un optimum global si et seulement si:

$$\forall x \in \Omega: f(x^*) \leq f(x) \quad (\text{I.1})$$

Tel que:

- ✓  $x^*$ : l’optimum global,
- ✓  $\Omega$ : la région faisable ( $\Omega \in s$ ),
- ✓  $s$ : l’espace de recherche global.

### I.1.5 Minimum local

Un point  $x^*$  est un minimum local de la fonction  $f$  si et seulement si:

$$f(x^*) < f(x), \quad \forall x \in V(x^*) \text{ et } x^* \neq x \quad (\text{I.2})$$

D’où  $V(x^*)$  définit un voisinage de  $x^*$ .

Trois minimums locaux et un minimum global sont illustrés dans la figure I.1.

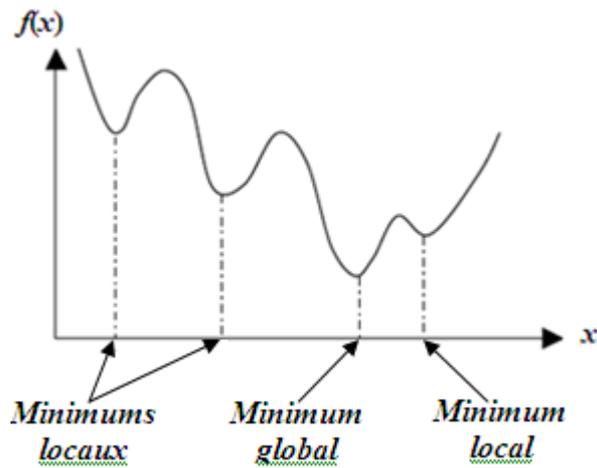


Figure I.1. Illustration des différents minima d'une fonction objective.

### I.1.6 Optimisation mono-objectif

Un problème d'optimisation mono-objectif est présenté sous la forme suivante:

$$\begin{cases} \text{Minimiser } f(\bar{x}) \\ \text{Sous les contraintes} \begin{cases} \bar{g}(\bar{x}) \leq 0 & (m \text{ contraintes d'inégalité}) \\ \bar{h}(\bar{x}) = 0 & (p \text{ contraintes d'égalité}) \end{cases} \end{cases} \quad (\text{I.3})$$

Avec  $\bar{x} \in \mathfrak{R}^n$ ,  $\bar{g}(\bar{x}) \in \mathfrak{R}^m$ ,  $\bar{h}(\bar{x}) \in \mathfrak{R}^p$ .

Les vecteurs  $\bar{g}(\bar{x})$  et  $\bar{h}(\bar{x})$  représentent respectivement  $m$  contraintes d'inégalité et  $p$  contraintes d'égalité.

### I.1.7 Optimisation multi-objectif

Un problème d'optimisation multi-objectif peut se définir comme suit:

$$\begin{cases} \text{Minimiser } y = F(x) = [f_1(x), f_2(x), \dots, f_N(x)]^T \\ \text{Sous les contraintes } g_i(x) \leq 0 \quad i = 1, 2, \dots, M \end{cases} \quad (\text{I.4})$$

Chercher le vecteur  $x^* = [x_1^*, x_2^*, \dots, x_n^*]^T$  qui satisfait les contraintes  $g_i$  tel que

$$x = [x_1, x_2, \dots, x_p]^T \in \Omega.$$

D'où

- ✓  $y$  représente le vecteur des fonctions objectives,
- ✓  $x$  représente le vecteur des variables de décision dans l'espace de recherche  $\Omega$ .

Les problèmes d'optimisations multi-objectifs n'ont pas une seule solution mais un ensemble de solutions dites solutions de compromis. Généralement, il n'existe aucune solution pour laquelle tous les objectifs sont optimisés, et souvent c'est le facteur humain qui décide l'optimalité d'une solution.

### I.1.8 Principe général de l'optimisation par essaim particulaire

L'optimisation par essaim particulaire (OEP), ou Particle Swarm Optimization (PSO) en anglais, est un algorithme évolutionnaire qui utilise une population de solutions candidates pour développer une solution optimale au problème. Cet algorithme a été proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995 [5]. Il s'inspire à l'origine du monde du vivant, plus précisément du comportement social des animaux évoluant en essaim, tels que les bancs de poissons et les vols groupés d'oiseaux. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une « intelligence » limitée, et ne dispose que d'une connaissance locale de sa situation dans l'essaim. L'information locale et la mémoire de chaque individu sont utilisées pour décider son déplacement [6].

L'essaim de particules correspond à une population d'agents simples, appelés particules. Chaque particule est considérée comme une solution du problème, où elle possède une position et une vitesse. De plus, chaque particule possède une mémoire lui permettant de se souvenir de sa meilleure performance et de la meilleure performance atteinte par les particules « voisines » (informatrices) [5], [6].

La stratégie de déplacement d'une particule est illustrée dans la figure I.2.

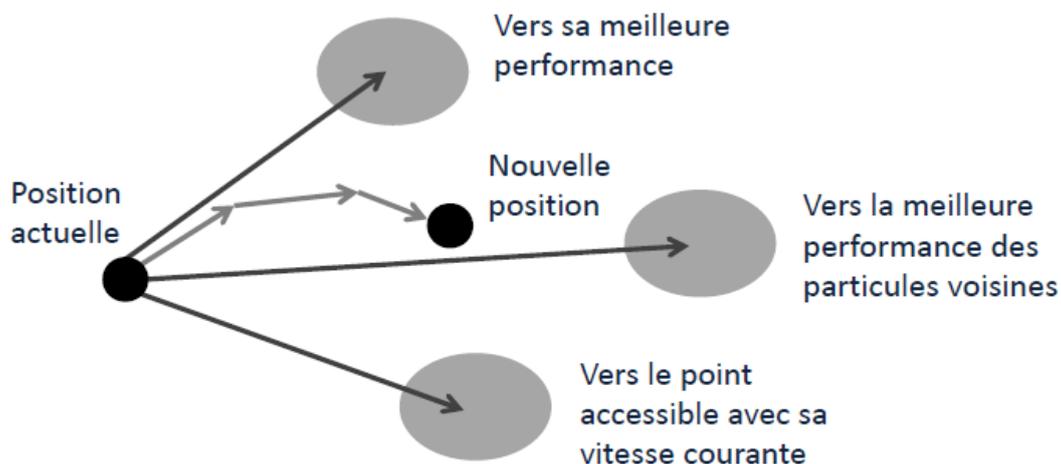


Figure I.2. Déplacement d'une particule.

### I.1.9 Formalisation de l'optimisation par essaim particulaire

Dans un espace de recherche de dimension  $D$ , la particule  $i$  de l'essaim est modélisée par son vecteur position  $\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  et par son vecteur vitesse  $\vec{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ . La qualité de sa position est déterminée par la valeur de la fonction "objectif" en ce point. Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on note  $\vec{Pbest}_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$ . La meilleure position atteinte par les particules de l'essaim est notée  $\vec{Gbest} = (gbest_1, gbest_2, \dots, gbest_D)$ . Nous nous référons à la version globale de PSO, où toutes les particules de l'essaim sont considérées comme voisines de la particule  $i$  [6].

Au départ de l'algorithme, les particules de l'essaim sont initialisées de manière aléatoire dans l'espace de recherche du problème. Ensuite, à chaque itération, chaque particule se déplace, en combinant linéairement les informations suivantes:

- ✓ La position courante
- ✓ La vitesse courante
- ✓ La distance entre la position courante et  $pbest$
- ✓ La distance entre la position courante et  $gbest$

à l'itération  $t+1$ , le déplacement d'une particule est régi par les deux équations (I.5) et (I.6) comme suit [6], [25]:

$$v_{i,j}^{t+1} = wv_{i,j}^t + c_1r_{1,i,j}^t [pbest_{i,j}^t - x_{i,j}^t] + c_2r_{2,i,j}^t [gbest_{i,j}^t - x_{i,j}^t], \quad j \in \{1, 2, \dots, D\} \quad (I.5)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}, \quad j \in \{1, 2, \dots, D\} \quad (I.6)$$

où  $v_{i,j}^t$  est la vitesse de la particule à l'itération  $t$ ,  $x_{i,j}^t$  est la position de la particule à l'itération  $t$ ,  $w$  est une constante, appelée coefficient d'inertie,  $c_1$  et  $c_2$  sont les coefficients d'accélération,  $r_1$  et  $r_2$  sont deux nombres aléatoires tirés uniformément dans  $[0, 1]$ .

Une fois le déplacement des particules effectué, les nouvelles positions sont évaluées et les deux vecteurs  $\vec{Pbest}_i$  et  $\vec{Gbest}$  sont mis à jour, à l'itération  $t+1$ , suivant les deux équations (I.7) et (I.8) (dans le cas d'une minimisation) respectivement [25]. Cette procédure est présentée dans l'Algorithme I.3, où  $N$  est le nombre de particules de l'essaim.

$$\vec{P}best_i(t+1) = \begin{cases} \vec{P}best_i(t), & \text{si } f(\vec{x}_i(t+1)) \geq \vec{P}best_i(t) \\ \vec{x}_i(t+1), & \text{sinon} \end{cases} \quad (\text{I.7})$$

$$\vec{G}best(t+1) = \underset{\vec{P}best_i}{\text{argmin}} f(\vec{P}best_i(t+1)), \quad 1 \leq i \leq N. \quad (\text{I.8})$$

Nous pouvons présenter l'OEP dans l'algorithme suivant [6], [25]:

1. **Initialiser** aléatoirement  $N$  particules: position et vitesse.
2. **Evaluer** les positions des particules
3. **Pour** chaque particule  $i$ ,  $\vec{P}best_i = \vec{x}_i$
4. **Calculer**  $\vec{G}best$  selon (I.8)
5. **tan que** le critère d'arrêt n'est pas satisfait **faire**
6.     **Déplacer** les particules selon (I.5) et (I.6)
7.     **Evaluer** les positions des particules
8.     **Mettre à jours**  $\vec{P}best_i$  et  $\vec{G}best$  selon (I.7) et (I.8)
9. **fin**

Algorithme I.3. Algorithme d'optimisation par essaim particulaire.

## I.1.10 Améliorations de l'optimisation par essaim particulaire

### I.1.10.1 Confinement des particules

Si le déplacement d'une particule soit trop rapide, il la conduise à sortir de l'espace de recherche ou à passer à côté de l'optimum, pour éviter ce problème, d'après [25] nous pouvons introduire un nouveau paramètre  $V_{\max}$ , qui permet de limiter la vitesse sur chaque dimension. C'est-à-dire nous allons limiter la distance maximale qu'une particule va parcourir au cours d'une itération. Cette méthode permet de contrôler la divergence de l'algorithme.

### I.1.10.2 Coefficient de constriction

La combinaison des paramètres  $w$ ,  $c_1$  et  $c_2$  permet de régler l'équilibre entre l'intensification et la diversification du processus de recherche. Dans [27], Clerc et Kennedy ont démontré qu'une bonne convergence peut être obtenue en rendant dépendants ces paramètres. L'utilisation d'un coefficient de constriction  $\chi$  (ou facteur de constriction) permet

de mieux contrôler la divergence de l'essaim [27]. En utilisant le coefficient de constriction, l'équation (I.5) devient:

$$v_{i,j}(t+1) = \chi(v_{i,j}(t) + \phi_1 r_1 (pbest_{i,j}(t) - x_{i,j}(t)) + \phi_2 r_2 (gbest_{i,j}(t) - x_{i,j}(t))) \quad (I.9)$$

avec:

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (I.10)$$

où  $\phi = \phi_1 + \phi_2$ ,  $\phi > 4$

Les valeurs optimales de  $\phi_1$  et  $\phi_2$  ont été déterminées dans [27], en effectuant de nombreux tests. En général, on utilise  $\phi = 4$ ,  $\phi_1 = \phi_2$ , ce qui donne un coefficient  $\chi = 0,7298844$ .

### I.1.10.3 Coefficient d'inertie

Le coefficient d'inertie  $w$ , introduit par Shi et Eberhart [25]. Le but de l'introduction de ce paramètre est de réaliser un équilibre entre la recherche locale (exploitation) et la recherche globale (exploration). L'intensité de l'exploration de l'espace de recherche dépend de la valeur du poids d'inertie, une grande valeur de  $w$  facilitant une exploration globale, alors qu'une petite valeur facilite l'exploration locale. Dans [25], les auteurs ont proposé un coefficient d'inertie dynamique qui varie au cours du temps. Il commence par une valeur proche de 0.9 et descend linéairement pour arriver à 0.4. Cette stratégie a beaucoup amélioré les performances de PSO pour plusieurs problèmes d'optimisation. Le coefficient d'inertie  $w$  varie linéairement avec le temps selon la formule suivante:

$$w = w_{min} + (w_{max} - w_{min}) \left( \frac{iter}{max_{iter}} \right) \quad (I.11)$$

où  $iter$  est l'itération courante et  $max_{iter}$  est le nombre maximal d'itération.  $w_{max}$  et  $w_{min}$  désignent respectivement les valeurs maximum et minimum du coefficient  $w$  (généralement,  $w_{max}, w_{min} \in [0,1]$ ).

### I.1.10.4 Topologie de voisinage

La communication entre les particules à l'intérieur d'un essaim est l'un des mécanismes les plus importants dans l'OEP, car elle permet l'échange de l'information entre les particules. Selon la structure du swarm, on peut distinguer plusieurs modèles de communication connus

sous le nom topologie de voisinage. Il existe de nombreuses topologies dont les suivantes sont les plus utilisées [26]: Topologie *Gbest*, Topologie en anneau, Topologie en étoile, Topologie *Von Neumann* et Topologie *Four-clusters*. Dans notre travail nous avons utilisé la topologie en étoile.

- ✓ Topologie en étoile: Cette topologie a la forme d'une roue, où la particule du centre de l'essaim, nommée focale (principale), est responsable du flux des informations. Dans cette topologie, toutes les particules sont isolées les unes des autres et elles sont liées uniquement à la particule focale. Les informations doivent être communiquées à travers cette particule, qui va ensuite utiliser ces informations pour adapter sa trajectoire. Si le résultat des ajustements montre une amélioration des performances de la particule principale, alors cette amélioration sera communiquée au reste de la population.

La figure suivante montre les différents types de la topologie de voisinage mentionnés ci-dessus.

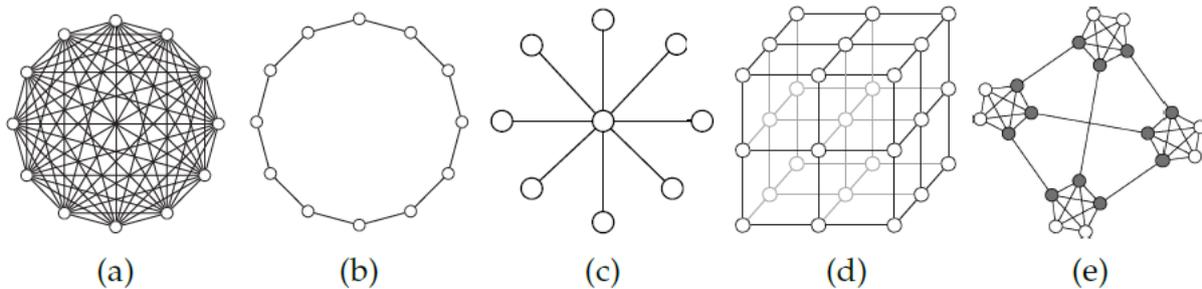


Figure I.4. Topologie statique: (a) Topologie *Gbest*, (b) Topologie en anneau, (c) Topologie en étoile, (d) Topologie *Von Neumann*, (e) Topologie *Four-clusters*.

## I.2 Commande optimale linéaire quadratique

### I.2.1 Introduction

La commande optimale représente la technique de contrôle classique largement employée par divers chercheurs. Cette commande traduite par le contrôle linéaire quadratique est étudiée et détaillée dans [1], [2] et [4]. La difficulté de conception d'un régulateur par retour d'état réside dans le choix de l'emplacement des valeurs propres en boucle fermée. La technique de placement de pôles peut prendre en charge cette difficulté, car elle fournit un contrôle explicite sur la stabilité, mais elle présente aussi un inconvénient majeur réside dans

les grandes valeurs de gains qui conduisent à des grandes amplitudes de la commande. Pour surmonter ce problème le régulateur linéaire quadratique a été introduit [3]. Notons néanmoins que les difficultés soulevées par ce genre de contrôle sont loin d'être complètement résolues, tel que le problème de choix des matrices de pondération durant la synthèse de la loi de commande.

Dans cette deuxième section du chapitre I, nous allons donner un rappel sur la linéarisation d'un système non linéaire autour d'un point donné de l'espace d'état, un aperçu sur la notion de contrôlabilité et d'observabilité. Ensuite nous allons faire une étude détaillée concernant la commande linéaire quadratique, et nous montrerons comment utiliser l'OEP pour le choix des matrices de pondération.

## I.2.2 Linéarisation

En pratique, les systèmes sont rarement linéaires. En revanche si leur vecteur d'état reste localisé dans une zone de petite taille de l'espace d'état, le système peut être assimilé à un système linéaire.

### I.2.2.1 Linéarisation d'une fonction

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$  une fonction différentiable. Au voisinage d'un point  $\bar{x} \in \mathbb{R}^n$  le développement de Taylor de  $f$  au premier ordre autour de  $\bar{x}$  nous donne

$$f(x) \approx f(\bar{x}) + \frac{df}{dx}(\bar{x})(x - \bar{x}) \quad (\text{I.12})$$

avec

$$\frac{df}{dx}(\bar{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\bar{x}) & \frac{\partial f_1}{\partial x_2}(\bar{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\bar{x}) \\ \frac{\partial f_2}{\partial x_1}(\bar{x}) & \frac{\partial f_2}{\partial x_2}(\bar{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\bar{x}) \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_k}{\partial x_1}(\bar{x}) & \frac{\partial f_k}{\partial x_2}(\bar{x}) & \cdots & \frac{\partial f_k}{\partial x_n}(\bar{x}) \end{pmatrix} \quad (\text{I.13})$$

Cette matrice est appelée matrice jacobienne. Considérons par exemple la fonction

$$f \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 x_2 \\ x_1^2 + x_1^2 \end{pmatrix}$$

La matrice jacobienne de  $f$  en  $x$  est

$$\frac{df}{dx}(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) \end{pmatrix} = \begin{pmatrix} 2x_1x_2 & x_1^2 \\ 2x_1 & 2x_2 \end{pmatrix}$$

Autour du point  $\bar{x} = (1, 2)$ , on a, d'après (I.12),

$$f \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \approx \begin{pmatrix} 2 \\ 5 \end{pmatrix} + \begin{pmatrix} 4 & 1 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x_1 - 1 \\ x_2 - 2 \end{pmatrix} = \begin{pmatrix} -4 + 4x_1 + x_2 \\ -5 + 2x_1 + 4x_2 \end{pmatrix}.$$

### I.2.2.2 linéarisation d'un système

Considérons le système décrit par ses équations d'état

$$\begin{cases} \dot{x} = f(x, u) \\ y = g(x, u) \end{cases} \quad (\text{I.14})$$

où  $x$  est de dimension  $n$ ,  $u$  est de dimension  $m$  et  $y$  est de dimension  $k$ . Posons que

$$z = \begin{pmatrix} x \\ u \end{pmatrix}, h(x, u) = \begin{pmatrix} f(x, u) \\ g(x, u) \end{pmatrix}$$

La fonction  $h(x, u)$  sera appelée la fonction d'évolution / observation. Autour du point  $\bar{z} = (\bar{x}, \bar{u})$ , nous avons d'après (I.12):

$$h(z) \approx h(\bar{z}) + \frac{dh}{dz}(\bar{z})(z - \bar{z}),$$

La matrice  $\frac{dh}{dz}(\bar{z})$  est écrite comme dans (I.13), et on peut la mettre sous la forme:

$$\frac{dh}{dz}(\bar{z}) = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

où

$A, B, C, D$  sont respectivement de dimension  $n \times n, n \times m, k \times n$ , et  $k \times m$ . Avec:

$$A = \frac{df}{dx}(\bar{x}, \bar{u}), B = \frac{df}{du}(\bar{x}, \bar{u}), C = \frac{dg}{dx}(\bar{x}, \bar{u}), D = \frac{dg}{du}(\bar{x}, \bar{u}).$$

Ainsi, autour du point  $\bar{z} = (\bar{x}, \bar{u})$ , nous avons

$$\begin{pmatrix} f(x, u) \\ g(x, u) \end{pmatrix} \approx h(\bar{x}, \bar{u}) + \frac{dh}{dz}(\bar{z}) \begin{pmatrix} x - \bar{x} \\ u - \bar{u} \end{pmatrix} = \begin{pmatrix} f(\bar{x}, \bar{u}) \\ g(\bar{x}, \bar{u}) \end{pmatrix} + \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} x - \bar{x} \\ u - \bar{u} \end{pmatrix}.$$

Autour du point  $(\bar{x}, \bar{u})$ , le comportement de (I.14) s'approxime donc par les équations d'état suivantes:

$$\begin{cases} \dot{x} = f(\bar{x}, \bar{u}) + A(x - \bar{x}) + B(u - \bar{u}) \\ y = g(\bar{x}, \bar{u}) + C(x - \bar{x}) + D(u - \bar{u}) \end{cases} \quad (\text{I.15})$$

### I.2.2.3 Linéarisation du système autour d'un point de fonctionnement

Un point  $(\bar{x}, \bar{u})$  est un point de fonctionnement (aussi appelé point de polarisation) si  $f(\bar{x}, \bar{u}) = 0$ , si  $\bar{u} = 0$ , on parle de point d'équilibre. Remarquons tout d'abord que si  $x = \bar{x}$  et si  $u = \bar{u}$ , alors  $\dot{x} = 0$ , c'est-à-dire que le système n'évolue plus si on maintient la commande  $u = \bar{u}$  et s'il est dans l'état  $\bar{x}$ . Dans ce cas, la sortie  $y$  a pour valeur  $y = \bar{y} = g(\bar{x}, \bar{u})$ . Autour du point de fonctionnement  $(\bar{x}, \bar{u})$ , d'après (I.15), le système (I.14) est approximé par:

$$\begin{cases} \dot{x} = A(x - \bar{x}) + B(u - \bar{u}) \\ y = \bar{y} + C(x - \bar{x}) + D(u - \bar{u}) \end{cases}$$

Pour des petites variations de  $\bar{u}$ ,  $\bar{x}$  et  $\bar{y}$ , on a:

$$\begin{cases} \frac{d}{dt} \bar{x} = A\bar{x} + B\bar{u} \\ \bar{y} = C\bar{x} + D\bar{u} \end{cases} \quad (\text{I.16})$$

Le système ainsi formé est appelé système linéaire autour du point de fonctionnement  $(\bar{x}, \bar{u})$

## I.2.3 Notions de contrôlabilité et d'observabilité

Les concepts de contrôlabilité et d'observabilité jouent un rôle très important dans l'étude des problèmes de contrôle et filtrage. Le concept de la contrôlabilité étudie la possibilité de régler le comportement du système considéré dans le but de forcer l'état à prendre les valeurs désirées pendant un temps fini. Par contre, l'observabilité d'un système peut-être définie comme étant la possibilité de prédire l'état du système à tout instant compris dans l'intervalle du temps de fonctionnement.

### I.2.3.1 Contrôlabilité

On étudie la contrôlabilité du système linéaire de dimension  $n$  défini dans l'espace d'état par le système d'équations de la forme suivante:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (\text{I.17})$$

où  $x \in \mathbb{R}^{n \times 1}$  est le vecteur d'état,  $A \in \mathbb{R}^{n \times n}$  est la matrice d'état,  $B \in \mathbb{R}^{n \times m}$  est la matrice de commande,  $u \in \mathbb{R}^{m \times 1}$  est le vecteur de commande,  $y \in \mathbb{R}^{p \times 1}$  est le vecteur de sortie,  $C \in \mathbb{R}^{p \times n}$  est la matrice de sortie, ou  $p$  représente le nombre de sortie et  $D \in \mathbb{R}^{p \times m}$  est la matrice de transmission directe. Les matrices  $A$  et  $B$  sont des matrices indépendants du temps. Dans ce cas, la dynamique du système est dite invariante dans le temps. L'intervalle de temps allant du temps présent ( $t = 0$ ) et tendant vers l'infini est alors l'intervalle d'intérêt sur lequel est étudiée la contrôlabilité du système.

Il ya plusieurs conditions équivalentes qui garantissent la commandabilité du système (I.17), nous avons utilisé la condition suivante:

\* Le système (I.17) est commandable si et seulement si le  $\text{rang} \left( \underbrace{B \mid AB \mid A^2B \mid \dots \mid A^{n-1}B}_{\Gamma_{com}} \right) = n$ ,

$\Gamma_{com}$  est la matrice de commandabilité ,obtenue en juxtaposant les unes à côté des autres les  $n$  matrices  $B, AB, A^2B, \dots, A^{n-1}B$ .

### I.2.3.2 Observabilité

Par dualité avec le concept de contrôlabilité, le système linéaire invariant définit dans (I.17) est observable si et seulement si la condition suivante est satisfaite:

\* Le  $\text{rang} \left( \underbrace{\begin{matrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{matrix}}_{\Gamma_{obs}} \right) = n$ ,  $\Gamma_{obs}$  est la matrice d'observabilité, obtenue en mettant les une en

dessous des autres les  $n$  matrices  $C, CA, \dots, CA^{n-1}$ .

## I.2.4 Formulation du problème LQ et sa solution

L'objectif de la commande linéaire quadratique (LQ) est de formuler le problème de commande par retour d'état en termes d'optimisation d'un critère qui traduit un compromis entre l'effort demandé à la commande et les contraintes qu'on veut imposer à l'état.

### I.2.4.1 Commande LQ à horizon fini

On considère le système linéaire (I.17), avec  $x_0$  l'état initial à l'instant  $t_0$ .

On définit la commande optimale à horizon fini  $u_{opt}(t)$  comme la commande qui minimise le critère quadratique [4]:

$$J = \frac{1}{2} \left\{ \int_{t_0}^{t_1} [x^T(t)Qx(t) + u^T(t)Ru(t)] dt + x^T(t_1)Sx(t_1) \right\} \quad (I.18)$$

avec  $Q \geq 0$  est la matrice de pondération symétrique qui donne un poids différent à chaque composante du vecteur d'état dans le critère,  $R > 0$  est la matrice de pondération symétrique de la commande qui affecte un poids différent à chaque composante de vecteur de commande,  $S \geq 0$ .

Pour résoudre le problème de minimisation de  $J$  on définit le Hamiltonien par la relation suivante:

$$H = \frac{1}{2} (x^T Q x + u^T R u) + \lambda^T (A x + B u) \quad (I.19)$$

$\lambda^T$  est choisit de sorte à avoir  $\dot{\lambda} = -\frac{\partial H}{\partial x}$

Les conditions d'optimalité se présentent alors comme suit:

$$\checkmark \text{ Equation d'état } \dot{x} = \frac{\partial H}{\partial \lambda} = A x + B u \quad (I.20)$$

$$\checkmark \text{ Equation d'état adjointe } -\dot{\lambda} = \frac{\partial H}{\partial x} = Q x + A^T \lambda \quad (I.21)$$

$$\checkmark \text{ Loi de commande } , \frac{\partial H}{\partial u} = R u + B^T \lambda = 0 \quad (I.22)$$

d'après (I.22):

$$u_{opt} = -R^{-1} B^T \lambda(t) \quad (I.23)$$

En remplaçant (I.23) dans (I.20) on obtient:

$$\dot{x} = A x - B R^{-1} B^T \lambda \quad (I.24)$$

On construit ensuite le système dit Hamiltonien homogène, on associant les équations (I.21) et (I.24):

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -B R^{-1} B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} \quad (I.25)$$

À tout instant  $t$ , on peut écrire

$$\lambda(t) = P(t)x(t) \quad (I.26)$$

où,  $P$  est une matrice semi-définie positive à calculer.

En dérivant l'équation (I.26) et utilisant l'expression de l'équation d'état adjointe (I.21), on aboutit à l'expression suivante appelée équation de Riccati:

$$-\dot{P} = A^T P + PA - PBR^{-1}B^T P + Q \quad (\text{I.27})$$

En régime permanent, l'équation de Riccati (I.27) devient:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (\text{I.28})$$

Ainsi, à partir de l'équation de Riccati (I.28) on calcule la matrice  $P$ . En utilisant cette matrice dans l'expression (I.26), on sera ensuite capable de calculer la loi de commande en remplaçant (I.26) dans (I.23):

$$u_{opt} = -R^{-1}B^T Px(t) = -Kx(t) \quad (\text{I.29})$$

$$K = R^{-1}B^T P \quad (\text{I.30})$$

où,  $K$  est le gain de la commande optimale.

#### I.2.4.2 Commande LQ à horizon infini

Le problème est de calculer le gain  $K$  qui permet de déterminer la commande par retour d'état tel que (I.31) [4], soit le minimum possible.

$$J = \int_{t_0}^{\infty} [x^T(t)Qx + u^T(t)Ru(t)] dt \quad (\text{I.31})$$

Soit  $u_{opt}$  la commande optimale, solution du problème. Toute commande  $u(t)$  peut s'écrire [4]:

$$u(t) = u_{opt}(t) + \varepsilon \tilde{u}(t), \quad \varepsilon \in \mathbb{R} \quad (\text{I.32})$$

Le système est linéaire donc il vérifie le théorème de superposition:

$$x(t) = x_{opt}(t) + \varepsilon \tilde{x}(t) \quad (\text{I.33})$$

où  $x_{opt}(t)$  est la solution optimale du système (c'est la trajectoire d'état obtenue lorsque on applique la commande optimale).

$$\dot{x}_{opt}(t) = Ax_{opt}(t) + Bu_{opt}(t) \quad (\text{I.34})$$

Ce qui nous permet d'écrire de la même manière

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B\tilde{u}(t) \quad (\text{I.35})$$

On suppose que  $x(0) \neq 0$ , (état initial indépendant de  $u(t)$ ). Donc  $x(0) = x_{opt}(0)$  et  $\tilde{x}(0) = 0$ .

Pour  $t \geq 0$ , la solution s'écrit:

$$x_{opt}(t) = e^{At} x_{opt}(0) + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau \quad (\text{I.36})$$

$$\tilde{x}(t) = \int_0^t e^{A(t-\tau)} B \tilde{u}(\tau) d\tau \quad (\text{I.37})$$

Alors :

$$J = \int_0^\infty \left[ x_{opt}(t) + \varepsilon \tilde{x}(t) \right]^T Q \left[ x_{opt}(t) + \varepsilon \tilde{x}(t) \right] dt + \int_0^\infty \left[ u_{opt}(t) + \varepsilon \tilde{u}(t) \right]^T R \left[ u_{opt}(t) + \varepsilon \tilde{u}(t) \right] dt \quad (\text{I.38})$$

$$J = \int_0^\infty \left[ x_{opt}^T(t) Q x_{opt}(t) + u_{opt}^T(t) R u_{opt}(t) \right] dt + 2\varepsilon \int_0^\infty \left[ \tilde{x}^T(t) Q x_{opt}(t) + \tilde{u}^T(t) R u_{opt}(t) \right] dt + \varepsilon^2 \int_0^\infty \left[ \tilde{x}^T(t) Q \tilde{x}(t) + \tilde{u}^T(t) R \tilde{u}(t) \right] dt \quad (\text{I.39})$$

Puisque  $u_{opt}(t)$  est la commande qui minimise  $J$ . Donc toute autre commande conduit à augmenter la valeur de  $J$ .

D'après (I.32), pour que  $u(t) = u_{opt}(t)$ , il faut que  $\varepsilon = 0$  c'est-à-dire le minimum de  $J$  se trouve en  $\varepsilon = 0$ .

Donc pour trouver le minimum de  $J$ , il faut résoudre:

$$\left. \frac{\partial J}{\partial \varepsilon} \right|_{\varepsilon=0} = 0 \quad (\text{I.40})$$

c'est à dire:

$$2 \int_0^\infty \left[ \tilde{x}^T(t) Q x_{opt}(t) + \tilde{u}^T(t) R u_{opt}(t) \right] dt = 0 \quad (\text{I.41})$$

En remplaçant  $\tilde{x}$  par sa valeur d'après (I.37) dans (I.41):

$$\int_0^\infty \left\{ \int_0^t \tilde{u}^T(\tau) B^T e^{A^T(t-\tau)} d\tau \right\} Q x_{opt}(t) + \tilde{u}^T(t) R u_{opt}(t) dt = 0 \quad (\text{I.42})$$

après calcul (permutation d'intégrales), on obtient:

$$\int_0^\infty \tilde{u}^T(t) \left\{ B^T \int_t^\infty e^{A^T(t-\tau)} Q x_{opt}(\tau) d\tau + R u_{opt}(t) \right\} dt = 0 \quad (\text{I.43})$$

on définit alors:

$$\lambda(t) = \int_t^\infty e^{A^T(t-\tau)} Q x_{opt}(\tau) d\tau \quad (\text{I.44})$$

$\lambda(t) \in \mathbb{R}^{n \times 1}$  est un vecteur qui peut être considéré comme un vecteur d'état d'un système dynamique dont l'équation d'évolution serait:

$$\begin{aligned} \dot{\lambda}(t) &= -A^T \lambda(t) - Q x_{opt}(t) \\ \lim_{t \rightarrow \infty} \lambda(t) &= 0 \end{aligned} \quad (\text{I.45})$$

L'équation (I.42) devient

$$\int_0^{\infty} \tilde{u}^T(t) \{B^T \lambda(t) + Ru_{opt}(t)\} dt = 0 \quad (I.46)$$

qui doit être vérifiée quel que soit  $\tilde{u}^T(t)$  à tout instant  $t \geq 0$ , c'est-à-dire:

$$B^T \lambda(t) + Ru_{opt}(t) = 0, \quad \forall t > 0 \quad (I.47)$$

Finalement on obtient:

$$u_{opt}(t) = -R^{-1} B^T \lambda(t) \quad (I.48)$$

En remplaçant (I.48) dans (I.34) on obtient:

$$\dot{x}_{opt}(t) = Ax_{opt}(t) - BR^{-1} B^T \lambda(t) \quad (I.49)$$

On construit ensuite le système dit Hamiltonien homogène, on associant les équations (I.45) et (I.49) :

$$\begin{bmatrix} \dot{x}_{opt} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1} B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x_{opt} \\ \lambda \end{bmatrix} \quad (I.50)$$

Il reste donc à calculer une matrice constante  $P$ , telle que:

$$\lambda(t) = Px_{opt}(t) \quad (I.51)$$

où,  $P$  est une matrice semi-définie positive.

En dérivant l'équation (I.51) et utilisant l'expression de l'équation d'état adjointe (I.45), on aboutit à l'expression suivante appelée équation de Riccati:

$$-\dot{P} = A^T P + PA - PBR^{-1} B^T P + Q \quad (I.52)$$

En régime permanent, l'équation de Riccati (I.52) devient:

$$A^T P + PA - PBR^{-1} B^T P + Q = 0 \quad (I.53)$$

Une fois on a obtenu la matrice  $P$ , on la remplace dans (I.51) pour déterminer la valeur de  $\lambda(t)$  qui nous permet de calculer la commande optimale selon (I.48):

$$u_{opt} = -R^{-1} B^T P x(t) = -Kx(t) \quad (I.54)$$

À partir de (I.54), nous calculons le gain de la commande optimale comme suit:

$$K = R^{-1} B^T P \quad (I.55)$$

Dans notre travail nous avons utilisé la commande LQ à horizon infini, dans lequel le critère de performance (I.31) a été généralisé de sorte que la commande optimale permet l'évolution du système à partir de n'importe quel état initial  $x(t_0) = x_0$  à un état de référence  $x(\infty) = x_r$ .

Le critère de performance à minimiser est donné sous la forme suivante:

$$J = \int_0^{\infty} ((x - x_r)^T Q (x - x_r) + (u - u_r)^T R (u - u_r)) dt \quad (I.56)$$

où  $u_r$  représente la commande nécessaire pour maintenir la stabilité du système dans l'état d'équilibre  $x_r$  et qui peut être calculée comme suit:

$$0 = Ax_r + Bu_r \Rightarrow u_r = -B^{-1} Ax_r \quad (I.57)$$

La loi de commande optimale dans ce cas est exprimée par l'expression (I.58):

$$u(t) - u_r = -K(x(t) - x_r)$$

$$u(t) = (K - B^{-1}A)x_r - Kx(t) \quad (I.58)$$

### I.2.5 Choix des matrices de pondération

Le problème majeur dans la synthèse de la commande LQ réside dans le choix des matrices de pondération ( $Q, R$ ) permettant d'atteindre les spécifications désirées. On trouve dans littérature deux méthodes pour réaliser ce choix.

- ✓ La première méthode est la règle de Bryson [28] qui suggère de choisir des matrices de pondérations diagonales comme suit:

$$Q = \text{diag}(q_1, q_2, \dots, q_n) \quad (I.59)$$

$$R = \text{diag}(r_1, r_2, \dots, r_m) \quad (I.60)$$

avec

$$q_i = (1 / \text{sup}(x_i))^2, \quad i = 1, 2, 3, \dots, n \quad (I.61)$$

$$r_i = (1 / \text{sup}(u_i))^2, \quad i = 1, 2, 3, \dots, m \quad (I.62)$$

où,  $\text{sup}(x_i)$  est la valeur maximale acceptable de la  $i^{\text{ème}}$  état,  $\text{sup}(u_i)$  est la valeur maximale acceptable de la  $i^{\text{ème}}$  signal de control.

Bryson indique également que cette règle ne fournit que des valeurs initiales, que l'on peut ensuite améliorer par simulations successives.

- ✓ la deuxième méthode [4], consiste à choisir des matrices de pondération symétrique et diagonales. Au départ, on choisit généralement des pondérations égales aux matrices identité que l'on peut ensuite améliorer par simulation successives.

D'après les deux méthodes précédentes le choix des matrices de pondérations sera raffiné par essais et erreurs en simulation. Alors les deux méthodes prennent du temps et peuvent conduire à de grandes valeurs de gain et une commande plus énergétique.

Dans notre travail pour surmonter ces problèmes nous avons utilisé l'OEP de sorte que le choix des matrices de pondération correspond à une commande minimale en conservant les performances. La procédure de mise en œuvre de notre méthode de sélection peut être décrite comme suit:

- ✓ sélectionner des matrices de pondération diagonales définies positives:  
 $Q = \text{diag}(q_1, q_2, \dots, q_n)$ ,  $i = 1, 2, 3, \dots, n$  et  $R = \text{diag}(r_1, r_2, \dots, r_m)$ ,  $i = 1, 2, 3, \dots, m$
- ✓ mettre les éléments de la diagonale de  $Q$  et  $R$  dans un seul vecteur  
 $V = (q_1, q_2, \dots, q_n, r_1, r_2, \dots, r_m)$ .
- ✓ spécifier la taille de la population ou de l'essaim.
- ✓ choisir le nombre maximal d'itérations.
- ✓ calculer la dimension du problème qui égale à  $(n + m)$ .
- ✓ spécifier la fonction « objectif » (fitness) qui peut être égale à la fonction coût (I.31) elle même ou égale à différentes formes de (I.31) comme dans [29].
- ✓ suivre les étapes de l'algorithme (I.3) mais avec une topologie de voisinage en étoile.

Nous pouvons représenter la structure de la commande LQ optimisée par L'OEP en schéma bloc dans la figure suivante:

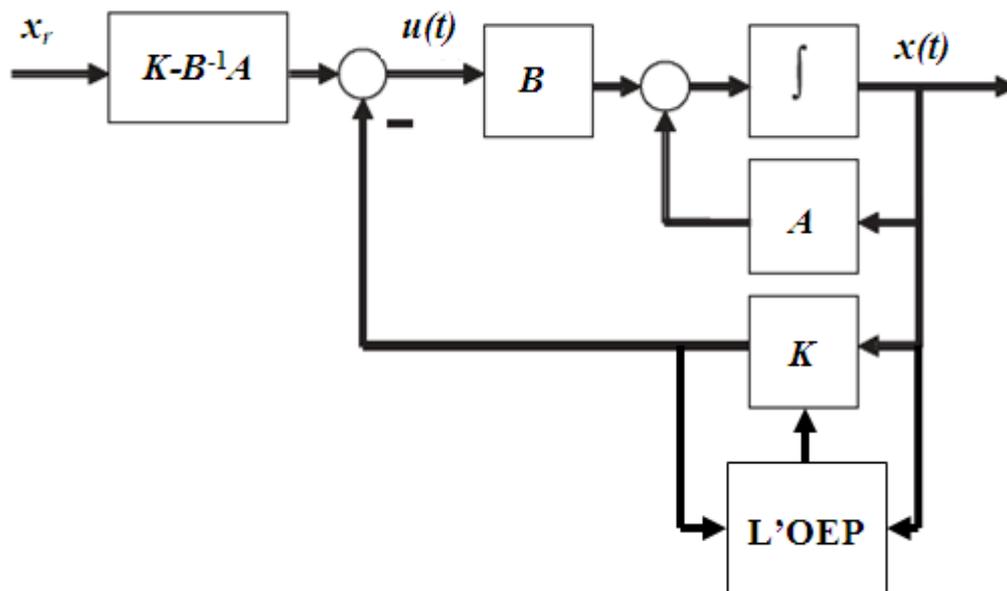


Figure I.5. Schéma bloc du régulateur LQR optimisé par PSO

### I.3 Applications

Dans cette partie, il s'agit d'appliquer notre commande à la poursuite de trajectoire de deux systèmes non linéaires. Le premier est le pendule inversé comme un système mono-entrée mono-sortie (SISO), et le second est le robot manipulateur à deux degré de liberté comme un système multi-entrées multi-sorties.

#### I.3.1 Exemple 1

Il s'agit de commander un pendule inversé, dont la dynamique est d'écrite par les équations suivantes [30]:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{g \sin(x_1) - a m l x_2^2 \sin(2x_1) / 2 - a \cos(x_1) u}{4l / 3 - a m l \cos^2(x_1)} \end{cases} \quad (\text{I.63})$$

où  $x_1$  est l'angle du pendule (en radians),  $x_2$  est la vitesse angulaire du pendule,  $g = 9.8 \text{ m/s}^2$  est la gravité universelle,  $m = 0.1 \text{ kg}$  est la masse du pendule,  $M = 1 \text{ kg}$  est la masse du chariot,  $l = 0.5 \text{ m}$  est la longueur du pendule,  $a = l / (m + M)$ ,  $u$  représente la force appliquée sur le chariot.

Nous linéarisons (I.63) autour de point de fonctionnement ( $x = 0, u = 0$ ), utilisant (I.13) nous obtenons le modèle linéaire suivant:

$$A = \begin{bmatrix} 0 & 1 \\ g / 4l / 3 - a m l & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -a / g / 4l / 3 - a m l \end{bmatrix}, C = [1 \ 0], D = 0.$$

Notre objectif est de forcer la sortie du pendule  $y = x_1(t)$  (l'angle  $\theta$ ), à suivre un signal de référence  $x_{r1}(t)$ , où  $x_r(t) = [x_{r1}(t) \ x_{r2}(t)]^T = \left[ \frac{\pi}{30} \sin(t) \ \frac{\pi}{30} \cos(t) \right]^T$ , pour les conditions initiales  $x(0) = [\pi / 15 \ 0]^T$ .

Les paramètres de simulation de l'OEP sont donnés comme suit:

- ✓ la taille de la population:  $N = 20$ .
- ✓ la dimension de l'espace de recherche:  $D = 3$ .
- ✓ le maximum d'itération:  $it\acute{e}r_{max} = 40$ .
- ✓ la fonction d'inertie  $w = 0.78$ .
- ✓ les facteurs de construction  $c_1 = 1.2$ ,  $c_2 = 1.8$ .

Après simulation nous avons trouvé les résultats suivants:

### Résultats de simulation

- ✓ les matrices de pondération

$$Q = \begin{bmatrix} 4.5 & 0 \\ 0 & 0.89 \end{bmatrix}, R = 0.1.$$

- ✓ les courbes

Les variations, en fonction du temps, de l'angle  $\theta(t) = x_1(t)$ , de la vitesse angulaire  $\dot{\theta}(t) = x_2(t)$ , de l'erreur de poursuite  $e_1(t), e_2(t)$ , et de la commande  $u(t)$  sont représentées par les figures (I.6), (I.7), (I.8), (I.9), et (I.10) respectivement.

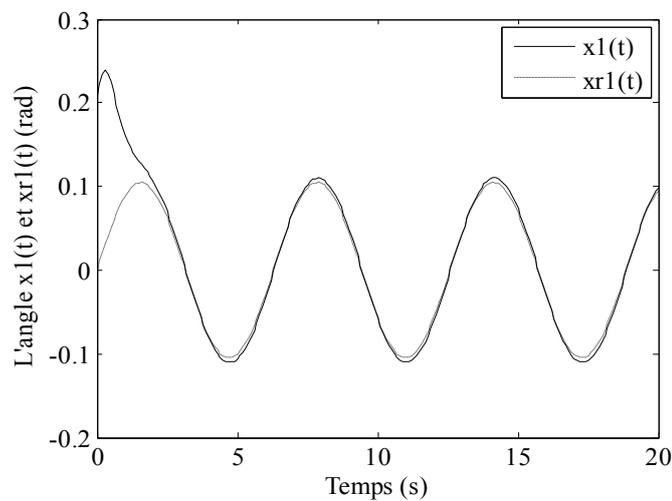


Figure I.6. Evolution de l'angle  $x_1(t)$  et le signal de référence  $x_{r1}(t)$ .

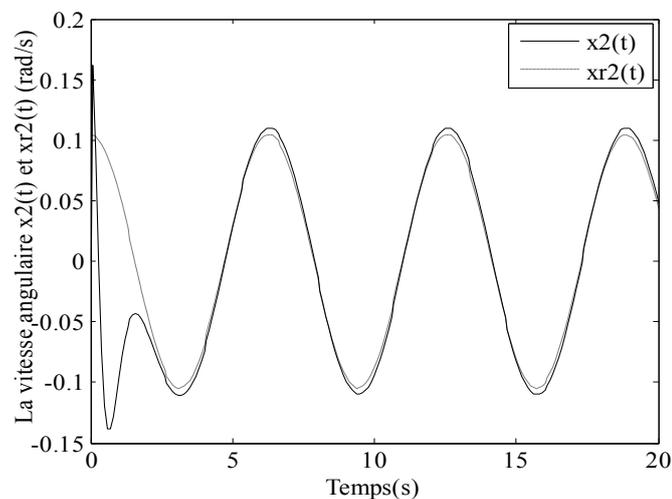


Figure I.7. Evolution de la vitesse angulaire  $x_2(t)$  et le signal de référence  $x_{r2}(t)$ .

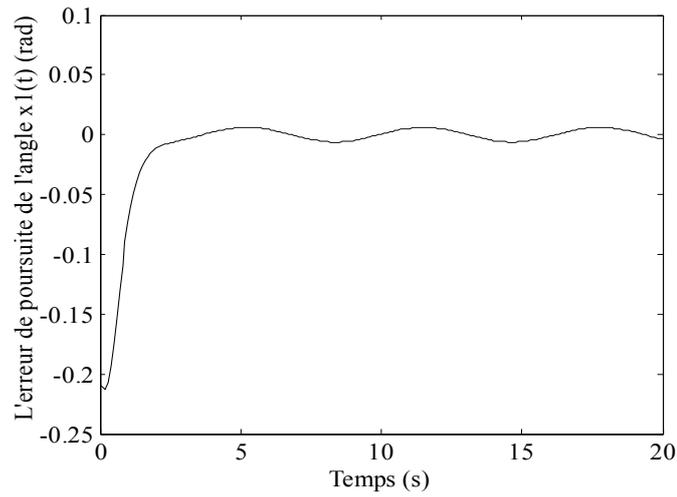


Figure I.8. L'erreur de poursuite  $e_1(t)$  de l'angle  $x_1(t)$ .

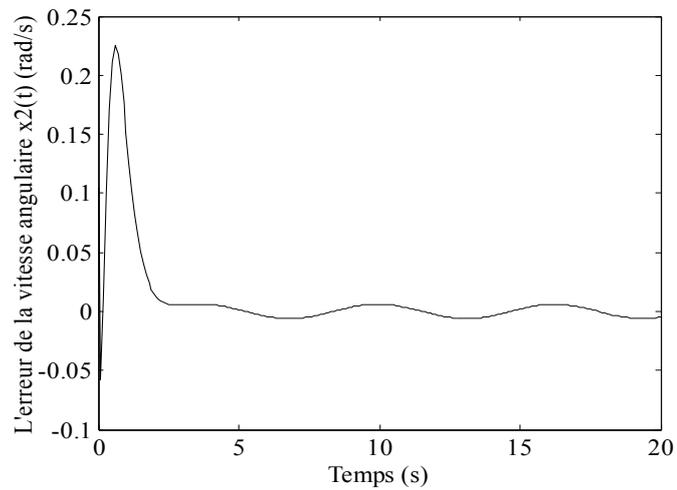


Figure I.9. L'erreur de poursuite  $e_2(t)$  de la vitesse angulaire  $x_2(t)$ .

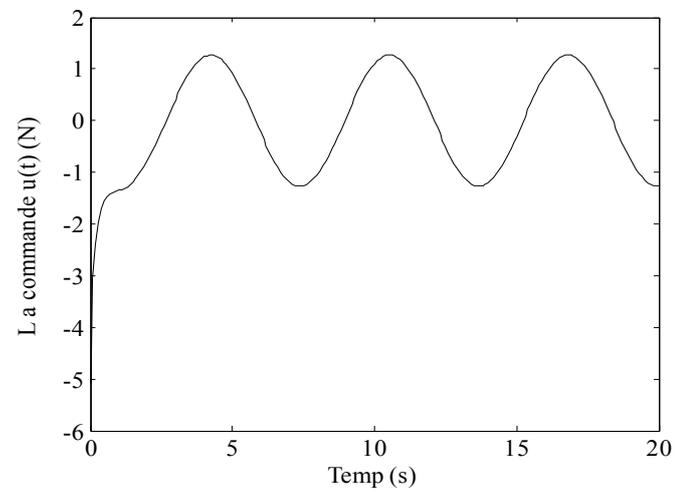


Figure I.10. La commande  $u(t)$ .

Les figures (I.6) et (I.7) mettent en évidence les capacités du contrôleur LQR développé sur un pendule inversé. En effet, ce régulateur est capable d'assurer la poursuite en un temps court. Les figures (I.8) et (I.9) montrent que l'erreur de poursuite est faible. Dans la figure (I.10) le signal de commande présente une allure lisse et continue facile à implémenter.

### I.3.2 Exemple 2

Il s'agit de commander un robot manipulateur à deux degré de liberté, dont la dynamique est régie par l'ensemble des équations différentielles suivantes [31], [32]:

$$H(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \quad (\text{I.64})$$

où:  $H(q) = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$ : matrice d'inertie symétrique définie positive,  $[q_1, q_2]^T$  est le vecteur

d'angle d'articulation,  $C(q, \dot{q})\dot{q} = \begin{bmatrix} -h\dot{q}_2 & -h\dot{q}_2 - h\dot{q}_1 \\ h\dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$ : vecteur regroupant les forces de

Coriolis et les forces centrifuges,  $G(q)$ : vecteur regroupant les forces de gravité ( $G(q) \equiv 0$  puisque nous sommes dans le plan horizontal),  $[\tau_1, \tau_2]^T$  est le vecteur de couple appliqué au niveau des articulations du manipulateur.

$$H_{11} = a_1 + 2a_3 \cos(q_2) + 2a_4 \sin(q_2), \quad H_{12} = H_{21} = a_2 + a_3 \cos(q_2) + a_4 \sin(q_2), \quad H_{22} = a_2,$$

$$h = a_3 \sin(q_2) - a_4 \cos(q_2), \quad a_1 = I_1 + m_1 l_{c1}^2 + l_e + m_e l_{ce}^2 + m_e l_1^2, \quad a_2 = I_e + m_e l_{ce}^2, \quad a_3 = m_e l_1 l_{ce} \cos(\delta_e),$$

$$, \quad a_4 = m_e l_1 l_{ce} \sin(\delta_e).$$

$$I_1 = 0.12, \quad l_1 = 1, \quad l_c = 0.5, \quad I_e = 0.25, \quad l_{ce} = 0.6, \quad l_{c1} = 0.5, \quad m_1 = 1, \quad m_e = 2, \quad \delta_e = 30^\circ$$

Puisque la matrice  $H$  est strictement définie positive, (I.64) peut être écrit comme suit:

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}^{-1} \left( \begin{bmatrix} h\dot{q}_2 & h(\dot{q}_1 + \dot{q}_2) \\ -h\dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \right) \quad (\text{I.65})$$

on prend  $x_1 = q_1, x_2 = q_2, x_3 = \dot{q}_1$ , et  $x_4 = \dot{q}_2$ .

Le modèle linéaire de (I.65) obtenu autour de point de fonctionnement 0 en utilisant (I.13) est le suivant:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -0.099(10)^{-4} & -0.048(10)^{-4} \\ 0 & 0 & 0.267(10)^{-4} & 0.099(10)^{-4} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.796 & -1.648 \\ -1.648 & 4.445 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Notre objectif est de forcer les sorties du robot  $y_1 = x_1(t)$  (l'angle d'articulation  $q_1(t)$ ),  $y_2 = x_2(t)$  (l'angle d'articulation  $q_2(t)$ ), à suivre les signaux de référence  $x_{r1}(t)$  et  $x_{r2}(t)$ .

Où  $x_r(t) = [x_{r1}(t), x_{r2}(t), \dot{x}_{r1}(t), \dot{x}_{r2}(t)]^T = \left[ \sin\left(\frac{2\pi}{25}t\right), \sin\left(\frac{2\pi}{38}t\right), \frac{2\pi}{25}\cos\left(\frac{2\pi}{25}t\right), \frac{2\pi}{38}\cos\left(\frac{2\pi}{38}t\right) \right]^T$ , et

les conditions initiales sont:  $x(0) = [-2, -2, -2, 2]$ .

Les paramètres de simulation de l'OEP sont donnés comme suit:

- ✓ la taille de la population:  $N = 60$ .
- ✓ la dimension de l'espace de recherche:  $D = 6$ .
- ✓ le maximum d'itération:  $it\acute{e}r_{max} = 100$ .
- ✓ la fonction d'inertie:  $w = 0.82$ .
- ✓ les facteurs de construction  $c_1 = 1.2$ ,  $c_2 = 1.8$ .

### Résultats de simulation

- ✓ les matrices de pondération

$$Q = \begin{bmatrix} 1.4 & 0 & 0 & 0 \\ 0 & 0.95 & 0 & 0 \\ 0 & 0 & 0.84 & 0 \\ 0 & 0 & 0 & 0.53 \end{bmatrix}, R = \begin{bmatrix} 0.93 & 0 \\ 0 & 0.18 \end{bmatrix}.$$

- ✓ les courbes

Les résultats de simulation, des sorties du système  $q_1(t) = x_1(t)$ ,  $q_2(t) = x_2(t)$  (les angles d'articulations du robot), des erreurs de poursuite  $e_1(t), e_2(t)$  et du couple  $\tau(t)$  sont représentées par les figures (I.11), (I.12), (I.13), (I.14) et (I.15) respectivement.

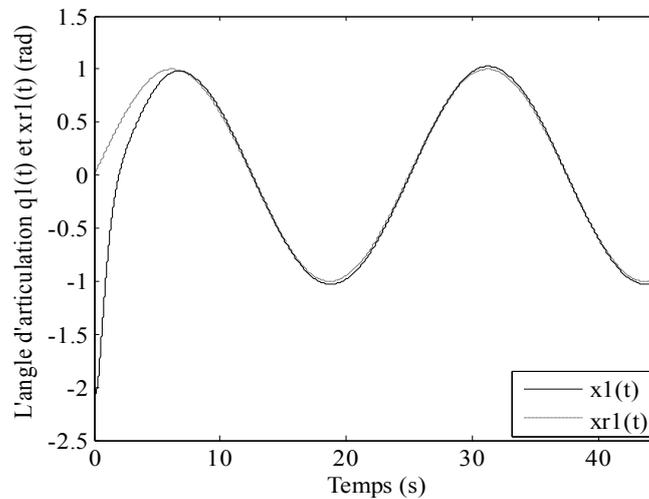


Figure I.11. Evolution de l'angle d'articulation  $x_1(t)$  et le signal de référence  $x_{r1}(t)$ .

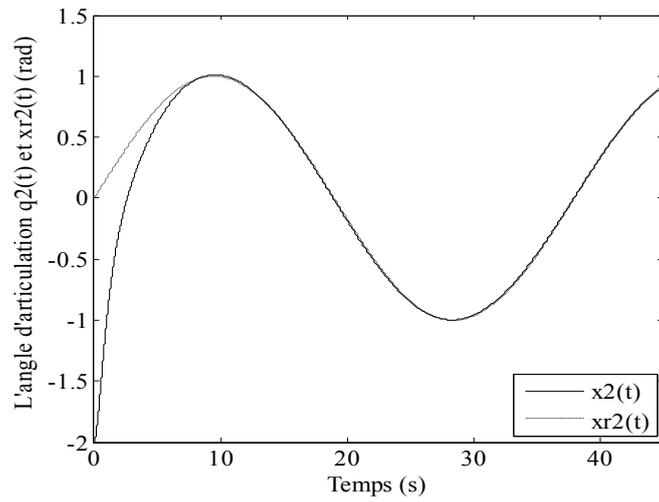


Figure I.12. Evolution de l'angle d'articulation  $x_2(t)$  et le signal de référence  $x_{r2}(t)$

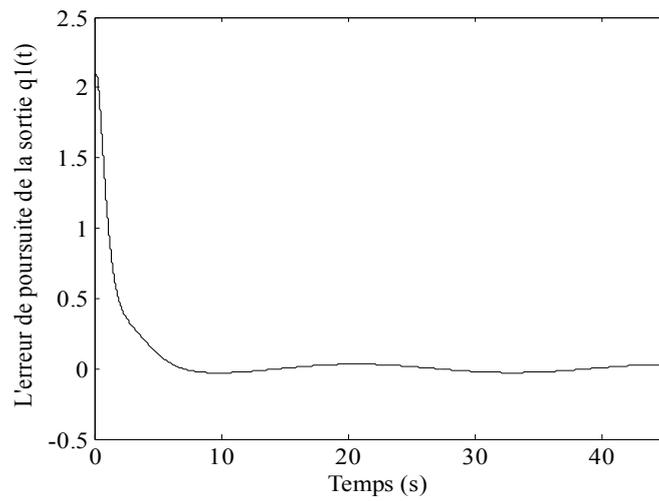


Figure I.13. L'erreur de poursuite  $e_1(t)$  de l'angle d'articulation  $x_1(t)$

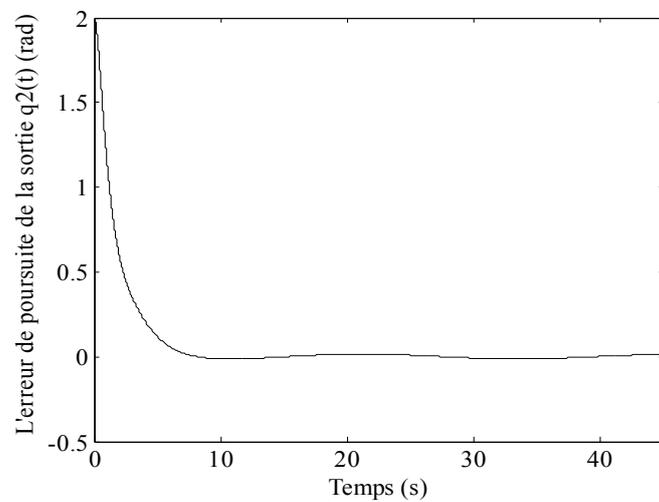


Figure I.14. L'erreur de poursuite  $e_2(t)$  de l'angle d'articulation  $x_2(t)$ .

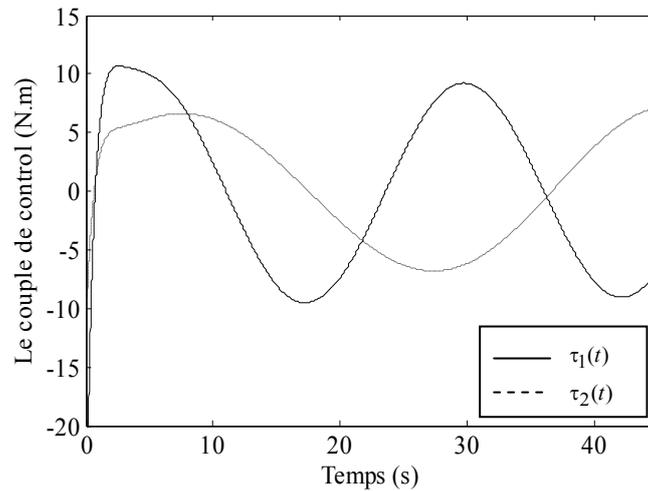


Figure I.15. Le couple de contrôle  $\tau_1(t)$  et  $\tau_2(t)$ .

Les figures (I.11), (I.12), (I.13) et (I.14) montrent de bonnes performances de poursuite et une convergence rapide des erreurs de poursuite vers zéro. D'après la figure (I.15) nous constatons que les signaux de couple de contrôle présentent des allures de formes lisses, continues et faciles à implémenter.

#### I.4 Conclusion

Dans ce chapitre, nous avons introduit l'optimisation par essaim particulaire comme une nouvelle technique pour le choix des matrices de pondération lors de la synthèse du régulateur LQR, où on a généralisé le critère de performance afin d'assurer la poursuite d'un signal de référence.

Pour mettre en évidence les capacités de notre régulateur LQR optimisé par l'OEP, deux exemples de simulation sont présentés, le premier est un système non linéaire SISO (le pendule inversé), le second est un système non linéaire MIMO (le robot manipulateur à deux degrés de liberté). Les résultats de simulation obtenus montrent de bonnes performances de poursuite et une convergence rapide de l'erreur de poursuite, avec des signaux de commande de formes lisses, continus et facile à implémenter.

L'étude faite dans ce chapitre suppose la disponibilité d'un modèle pour le système qui en général n'est valable qu'autour d'un point de fonctionnement. En l'absence d'un modèle, une nouvelle approche de commande s'avère nécessaire. La commande adaptative floue qui sera étudiée dans le chapitre II représente un choix convenable. Pour que le système soit valable dans une région plus large de l'espace d'état, nous allons proposer une approche multi-modèles à base de systèmes flous de Takagi-sugeno dans le chapitre III.

---

*CHAPITRE II*

*COMMANDE ADAPTATIVE  
FLOUE POUR DES SYSTEMES  
NON LINEAIRES*

---

## COMMANDE ADAPTATIVE FLOUE POUR DES SYSTEMES NON LINEAIRES

### II.1 Introduction

La logique floue est une description mathématique d'un processus basée sur la théorie des ensembles flous. Cette théorie introduite en 1975 par le professeur Lotfi Zadeh [33] permet de traiter des propositions ou des états par plusieurs niveaux de vérité.

La commande adaptative est une technique qui consiste à identifier un processus et son environnement à partir de quelques mesures entrées-sorties, et d'utiliser ensuite le modèle obtenu pour concevoir une action de commande. Nous trouvons dans la littérature deux approches de commande adaptative: commande adaptative directe (DAC), et commande adaptative indirecte (IAC) [34], [35]. Depuis le régulateur auto ajustable de Procky et Mamdani [36], divers systèmes adaptatifs flous ont été élaborés et plusieurs résultats ont été publiés [37], [38]. Les systèmes adaptatifs flous possèdent l'avantage d'être capable d'incorporer les informations numériques et les informations linguistiques en même temps. Il a été démontré que les systèmes flous peuvent approximer n'importe quelle fonction continue avec une précision arbitraire donnée. Récemment, plusieurs travaux de la commande adaptative floue ont été proposés [39], [40], et leur stabilité a été analysée par la méthode de Lyapunov.

Dans ce chapitre nous présenterons quelques notions de base concernant la commande adaptative et la commande floue. Ainsi nous élaborerons une loi de commande adaptative floue indirecte pour un système non linéaire monovarié. Ensuite nous proposerons une loi de commande adaptative floue directe pour un système non linéaire multivarié.

### II.2 Définitions

#### II.2.1 Commande adaptative

Adapter veut dire changer un comportement en conformité avec de nouvelles circonstances. La commande adaptative est définie comme une commande permettant au régulateur de s'adapter de lui-même aux changements du processus. En conséquence,

l'objectif de la commande adaptative est l'ajustement automatique en ligne des régulateurs des boucles de commande, afin de réaliser ou maintenir de façon identique un certain niveau de performances, quand les paramètres du procédé à commander varient dans le temps. Un système adaptatif peut également s'envisager comme une structure à deux boucles, une boucle principale classique qui prend en compte les variations des signaux d'entrée et de sortie, et une boucle secondaire qui réagit aux variations des paramètres du processus: c'est elle qui rend le système adaptatif. Les différentes méthodes de commande adaptative se différencient par la structure choisie pour réaliser la mise à jour en temps réel des paramètres du régulateur en fonction des variations du processus à commander.

### **II.2.1.1 Commande adaptative indirecte**

Le principe de la commande adaptative indirecte repose sur l'estimation des paramètres du processus par une procédure d'identification. Cette commande connue aussi sous le nom de commande auto-ajustable.

### **II.2.1.2 Commande adaptative directe**

L'idée de cette commande consiste à recalculer les paramètres du régulateur, mais sans identifier explicitement les paramètres du système, c'est-à-dire les paramètres du régulateur sont ajustés directement et en temps réel.

## **II.2.2 Commande floue**

La commande floue fait de plus en plus l'objet de réalisations effectives, en particulier dans le domaine industriel. Elle permet de traiter les problèmes de commande de processus à partir des connaissances des experts ou opérateurs qualifiés.

Lorsque l'on ne dispose pas d'un modèle défini au sens classique du terme, la commande floue peut être considérée comme une démarche adaptée. Dans ce cas, on élabore la loi de commande en codant les connaissances des experts, opérateurs, pilotes, etc., qui savent contrôler le processus.

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

### II.2.2.1 Systèmes flous

Les systèmes flous permettent d'exploiter et de manipuler efficacement les informations linguistiques émanant de l'expert humain grâce à un fondement théorique important [41], [42]. En plus, le système mis en œuvre peut être intégré facilement dans une boucle de commande ou d'identification. La structure de base d'un système flou se divise en trois parties principales comme le montre la figure (II.1).

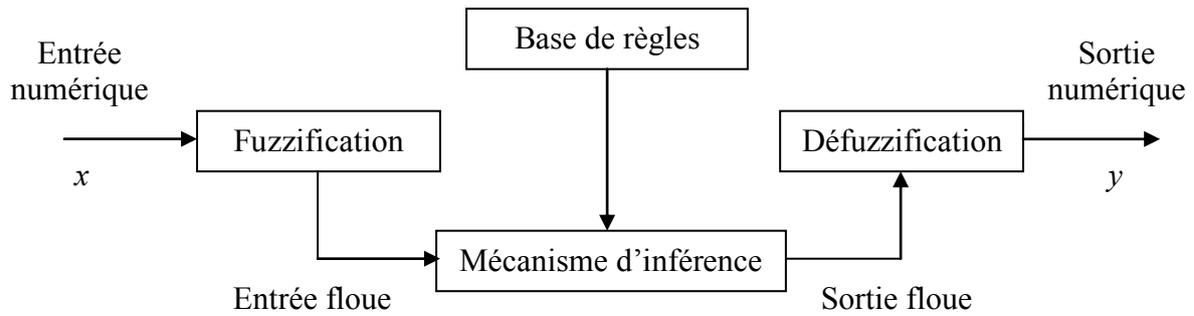


Figure II.1. Structure d'un système flou

#### II.2.2.1.1 Fuzzification

La fuzzification de la valeur précise d'une variable consiste à caractériser le degré avec lequel cette mesure appartient à un ensemble flou donné, c'est à dire le passage d'une grandeur précise à une variable floue ou variable linguistique. On distingue deux types de fuzzification

##### II.2.2.1.1.1 Fuzzification par singleton

Cette démarche est utilisée lorsque la valeur fournie par le capteur est supposée précise. La fuzzification par singleton d'une valeur  $u_0 \in U$  est alors la partie floue numérique  $F_x$  dans  $U$  dont la fonction d'appartenance est définie par:

$$\begin{cases} u_{F_x}(u) = 1 & \text{si } u = u_0 \\ u_{F_x}(u) = 0 & \text{si } u \neq u_0 \end{cases} \quad (\text{II.1})$$

où,  $U$  est l'univers de discours,  $F_x$  est l'ensemble flou et  $u_{F_x}(u)$  représente la fonction d'appartenance.

Cette stratégie est largement utilisée dans les applications de commande floue, car elle est facile à implémenter.

### II.2.2.1.1.2 Fuzzification non singleton

Dans cette démarche, on considère que le capteur peut engendrer une erreur de mesure ou être exposé à un bruit. La fonction d'appartenance de  $F_x$  dépend dans ce cas de l'erreur ou du bruit, par exemple, elle peut être définie par une fonction gaussienne:

$$u_{F_x}(u) = \exp\left(\frac{-(u-u_0)^T(u-u_0)}{\delta^2}\right) \quad (\text{II.2})$$

où  $\delta$  est un paramètre caractérisant la forme de  $u_{F_x}$ .

### II.2.2.1.2 Base de règles

Ce bloc contient les règles de commande dont le nombre et l'expression dépendent, notamment, de la partition des univers de discours des variables d'entrée et de sortie. La partition de l'espace des entrées (variables des prémisses des règles) et de l'espace des sorties (variables de conclusion des règles), consiste à définir le nombre d'ensembles flous décrivant chaque variable, ainsi que leur répartition sur leurs univers de discours respectifs.

En général, les contrôleurs flous utilisent des règles de la forme:

Si  $x_1$  est  $A_1$  et si  $x_2$  est  $A_2$  et .....et  $x_n$  est  $A_n$  alors  $u_1$  est  $B_1$  et  $u_2$  est  $B_2$  et.....et  $u_m$  est  $B_m$

où  $x_1, x_2, \dots, x_n$ , sont les variables physiques caractéristiques du processus,  $u_1, u_2, \dots, u_m$

sont les variables de commande du système et  $A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m$ , sont des valeurs linguistiques.

### II.2.2.1.3 Moteur d'inférence

L'inférence floue transforme un degré d'appartenance en un autre degré d'appartenance. L'implication et l'inférence floue sont associées pour définir l'ensemble flou caractérisant la commande. Il existe plusieurs possibilités pour réaliser les opérateurs qui combinent les valeurs d'entrée et les valeurs de sortie, c'est ce qu'on appelle la méthode d'inférence. Les méthodes les plus utilisées sont:

- ✓ Méthode d'inférence MAX-MIN.
- ✓ Méthode d'inférence MAX-PROD.
- ✓ Méthode d'inférence SOMME-PROD.

#### II.2.2.1.4 Défuzzification

Comme nous avons vu dans la section précédente, les méthodes d'inférence fournissent un résultat qui est une fonction d'appartenance. Mais, la sortie du contrôleur est en général une grandeur continue, prenant sa valeur dans un intervalle. La défuzzification est le traitement qui permet de définir une correspondance entre le résultat de l'inférence et la grandeur continue fournie en sortie. On distingue trois méthodes:

- ✓ Défuzzification par le centre de gravité.
- ✓ Défuzzification le centre des aires.
- ✓ Défuzzification par le centre des maximums.

#### II.2.2.2 Structure générale et différents types de modèles flous

En général les systèmes flous s'appuient sur une représentation de la connaissance sous forme de règles « Si...Alors » qui permettent de représenter les relations entre les variables d'entrée et de sortie dont l'expression générique est de la forme:

**Si** antécédent **Alors** conséquent

Dans un premier temps et afin de faciliter l'interprétation, on peut considérer l'antécédent (prémisse) comme une description linguistique qui indique les conditions de validité du phénomène représenté. Pour sa part, le conséquent (conclusion) représente le comportement associé aux conditions de validité décrites par l'antécédent.

Selon la structure particulière de la proposition conséquente, on peut distinguer deux types de modèles flous basés sur des règles [43]:

##### II.2.2.2.1 Système flou de Mamdani

Dans ce type de modèle flou l'antécédent et le conséquent sont tout les deux des propositions floues qui utilisent des variables linguistiques [44], [45].

L'avantage majeur de ces modèles est que les prémisses et les conclusions des règles sont de nature entièrement linguistique. Ceci les rend facilement compréhensibles, et donne une interprétation sémantique claire au modèle. En revanche, un grand nombre de règles est souvent nécessaire pour approcher le comportement global d'un système. Par conséquent, le nombre de paramètres à identifier est élevé [46]. Ce nombre exige une base d'informations importante et complète pour déterminer tous les paramètres. Ceci veut dire, un nombre de données d'identification important, mais aussi des données réparties sur la totalité des espaces d'entrée.

### II.2.2.2 Système flou de Takagi-Sugeno

Un autre type de modèle flou, approprié pour l'approximation d'une classe générale de systèmes non linéaires est celui proposé par Takagi et Sugeno [12]. Ce type de modèle est, comme celui de Mamdani, construit à partir d'une base de règles "Si...Alors...", dans laquelle si la prémisse est toujours exprimée linguistiquement, le conséquent utilise des variables numériques plutôt que des variables linguistiques. Le conséquent peut s'exprimer par exemple, sous la forme d'une constante, d'un polynôme ou de manière plus générale d'une fonction ou d'une équation différentielle dépendant des variables associées à l'antécédent. Ce type de modèle flou sera étudié en détail dans le chapitre III.

### II.2.3 Approximation floue

Le choix de la méthode de fuzzification, la stratégie d'inférence et de la méthode de défuzzification permet d'établir différentes classes de systèmes flous. Lors de la conception des contrôleurs flous, qui sont à base de systèmes flous, ces classes seront prises en considération. Parmi ces classes, celle utilisant une fuzzification par singleton, une défuzzification par le centre de gravité, et le produit d'inférence, représentent toutes les fonctions:  $f : U \subset R^n \longrightarrow R$  de la forme suivante [14]:

$$y(x) = \frac{\sum_{l=1}^k \bar{y}^l \left( \prod_{i=1}^n \mu_{F_i^l}(x_i) \right)}{\sum_{i=1}^k \left( \prod_{i=1}^n \mu_{F_i^l}(x_i) \right)} \quad (\text{II.3})$$

Où  $x = (x_1, \dots, x_n)^T \in U$ ,  $\bar{y}^l$  est un point dans  $R$  où  $\mu_{G^l}(y)$  atteint sa valeur maximale (en général, on prendra  $\mu_{G^l}(\bar{y}^l) = 1$ ),  $F_i^l$  et  $G^l$  sont les ensembles flous. Si les fonctions d'appartenance  $\mu_{F_i^l}(x_i)$  sont fixées, et que les paramètres  $\bar{y}^l$  sont considérés comme des paramètres ajustables, alors (II.3) peut être réécrite sous la forme compacte suivante [14]:

$$y(x) = \theta^T \xi(x) \quad (\text{II.4})$$

Où  $\theta = (\bar{y}^1, \dots, \bar{y}^M)^T$  est un vecteur de paramètres et  $\xi(x) = (\xi^1(x), \dots, \xi^M(x))^T$  est un vecteur (des fonctions floues du base) régressif avec le régresseur  $\xi^l(x)$  défini comme :

$$\xi^l(x) = \frac{\prod_{i=1}^n \mu_{F_i^l}(x_i)}{\sum_{l=1}^M \left( \prod_{i=1}^n \mu_{F_i^l}(x_i) \right)} \quad (\text{II.5})$$

Les contrôleurs adaptatifs flous basés sur la relation précédente seront relativement faciles à concevoir et à analyser, mais l'inconvénient d'utiliser cette forme réside dans le fait de ne pas pouvoir ajuster les fonctions d'appartenance  $\mu_{F_i^l}(x_i)$  durant la procédure d'adaptation. Pour cela on introduit une deuxième classe de systèmes flous définis par une fuzzification par singleton, défuzzification par centre de gravité et en utilisant le produit d'inférence et des fonctions d'appartenance gaussiennes.

Ce nouveau système est représenté par l'ensemble de toutes les fonctions  $f : U \subset R^n \longrightarrow R$  de la forme:

$$y(x) = \frac{\sum_{l=1}^M \bar{y}^l \left[ \prod_{i=1}^n \exp \left( -\frac{1}{2} \left( \frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]}{\sum_{l=1}^M \left[ \prod_{i=1}^n \exp \left( -\frac{1}{2} \left( \frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]} \quad (\text{II.6})$$

où les paramètres  $\bar{y}^l, \bar{x}_i^l$  et  $\sigma_i^l > 0$  sont des paramètres réglable.

Ces systèmes flous sont utilisés comme étant des blocs de base pour la construction de contrôleurs adaptatifs flous car il à été prouvé que ces systèmes sont des approximateurs universels [47], c'est à dire qu'ils sont capables d'approximer toute fonction réelle continue  $g$  sur un ensemble compact  $U$  avec une précision arbitraire donnée.

### II.3 Commande adaptative floue

Les contrôleurs flous ont la propriété d'approximer n'importe quelle fonction non linéaire mais l'obtention de ces contrôleurs n'était pas toujours évidente. C'est pourquoi dans les années 90, l'adaptatif flou a fait son apparition [13] et [47].

Le plus souvent, les régulateurs flous sont utilisés dans des systèmes qui possèdent des variations inconnues intrinsèques. L'objectif est donc de conserver de bonnes performances du système complet en adaptant le régulateur en fonction des variations du système.

Les contrôleurs adaptatifs flous offrent l'avantage d'être capable d'incorporer les informations numériques et les informations linguistiques en même temps pendant les étapes de constrictioin et d'apprentissage.

### II.3.1 Commande adaptative floue indirecte

Dans cette méthode on approxime d'abord le modèle du processus par des systèmes adaptatifs flous puis on synthétise la loi de commande à partir du modèle approximé.

#### II.3.1.1 Formulation du problème

Soit un système dynamique non linéaire de la forme:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_n = f(x_1, x_2, \dots, x_n) + g(x_1, x_2, \dots, x_n)u \\ y = x_1 \end{cases} \quad (\text{II.7})$$

qui peut être aussi écrit sous la forme:

$$\begin{cases} \dot{x}^{(n)} = f(x, \dot{x}, \dots, x^{(n-1)}) + g(x, \dot{x}, \dots, x^{(n-1)})u \\ y = x \end{cases} \quad (\text{II.8})$$

où  $f$  et  $g$  sont des fonctions continues inconnues,  $u \in \mathfrak{R}$  et  $y \in \mathfrak{R}$  sont l'entrée et la sortie du système, respectivement,  $x = (x_1, x_2, \dots, x_n)^T = (x, \dot{x}, \dots, x^{(n-1)})^T \in \mathfrak{R}^n$  est le vecteur d'état du système qui est disponible pour la mesure. Pour que (II.7) soit contrôlable, il faut que  $g(x) \neq 0$  pour  $x$  appartenant à une certaine région de contrôlabilité  $U_c \in \mathfrak{R}^n$ . Puisque  $g(x)$  est continue, nous allons supposer que  $0 < g(x) < \infty$  pour tous  $x \in U_c$ .

L'objectif de la commande est de forcer la sortie  $y$  à suivre un signal de référence borné  $x_r(t) = y_m(t)$ , sous la contrainte que tous les signaux impliqués doivent être bornés. Plus spécifiquement il s'agit de déterminer la commande par retour d'état  $u = u(x/\theta)$  et une loi d'adaptation pour ajuster les paramètres du vecteur  $\theta$  tel que :

- ✓ Le système en boucle fermée soit globalement stable et robuste dans le sens que tous les signaux  $x(t)$ ,  $\theta(t)$  et  $u(x/\theta)$ , soient uniformément bornés, c'est-à-dire que

$|x(t)| \leq M_x < \infty$ ,  $|\theta(t)| \leq M_\theta < \infty$  et  $|u(x/\theta)| \leq M_u < \infty$  pour  $t \geq 0$ , où  $M_x$ ,  $M_\theta$  et  $M_u$  sont des paramètres spécifiés par le concepteur.

- ✓ L'erreur de poursuite  $e = y_m - y$  devrait être la plus petite possible sous la contrainte précédente.

Pour commencer, on considère le signal d'erreur  $e = (e, \dot{e}, \dots, e^{(n-1)})$  et le vecteur de gain  $k = (k_n, \dots, k_1)^T \in \mathfrak{R}^n$  telles que toutes les racines du polynôme caractéristique  $h(s) = s^n + k_1 s^{n-1} + \dots + k_n$  soient dans le demi-plan gauche (système stable).

Si les fonctions  $f$  et  $g$  sont connues, alors la loi de commande a la forme suivante [48]:

$$u = \frac{1}{g(x)} \left[ -f(x) + y_m^{(n)} + k^T e \right] \quad (\text{II.9})$$

Ce qui donne:

$$e^n + k_1 e^{(n-1)} + \dots + k_n e = 0 \quad (\text{II.10})$$

Qui implique que  $\lim_{t \rightarrow \infty} e(t) = 0$ , qui est l'un des objectifs de la commande.

Si  $f$  et  $g$  sont inconnues, on les approxime par les systèmes flous, de la forme (II.3),  $\hat{f}(x/\theta_f)$  et  $\hat{g}(x/\theta_g)$  respectivement. La commande dans ce cas sera:

$$u_c = \frac{1}{\hat{g}(x/\theta_g)} \left[ -\hat{f}(x/\theta_f) + y_m^{(n)} + k^T e \right] \quad (\text{II.11})$$

En appliquant (II.11) à (II.8) on trouve:

$$e^{(n)} = -k^T e + \left[ \hat{f}(x/\theta_f) - f(x) \right] + \left[ \hat{g}(x/\theta_g) - g(x) \right] u_c \quad (\text{II.12})$$

qui est équivalent à:

$$\dot{e} = A_c e + b_c \left[ (\hat{f}(x/\theta_f) - f(x)) + (\hat{g}(x/\theta_g) - g(x)) u_c \right] \quad (\text{II.13})$$

où

$$A_c = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -k_n & -k_{n-1} & -k_{n-2} & \dots & -k_2 & -k_1 \end{bmatrix} \text{ et } b_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{II.14})$$

Tant que  $|sI - A_c| = s^{(n)} + k_1 s^{(n-1)} + \dots + k_n$  est stable ( $A_c$  est stable), on déduit qu'il existe une matrice unique définie positive  $P \in \mathfrak{R}^{n \times n}$  qui vérifie l'équation de Lyapunov:

$$A_c^T P + P A_c = -Q \quad (\text{II.15})$$

où  $Q$  est une matrice symétrique arbitraire définie positive de dimensions  $n \times n$ .

Soit la fonction quadratique de lyapunov candidate  $V_e = \frac{1}{2} e^T P e$ , on détermine la dérivée de  $V_e$

comme suit:  $\dot{V}_e = \frac{1}{2} \dot{e}^T P e + \frac{1}{2} e^T P \dot{e}$ , en utilisant (II.13) et (II.15) on aura:

$$\dot{V}_e = -\frac{1}{2} e^T Q e + e^T P b_c \left[ (\hat{f}(x/\theta_f) - f(x)) + (\hat{g}(x/\theta_g) - g(x)) u_c \right] \quad (\text{II.16})$$

Pour que  $x_i = y_m^{(i-1)} - e^{(i-1)}$  soit borné, on exige que  $V_e$  soit bornée, qui signifie que  $\dot{V}_e \leq 0$  lorsque  $V_e$  est supérieur à une importante constante  $\tilde{V}$ . Cependant, de (II.16) on voit qu'il est très difficile de concevoir  $u_c$  tel que le dernier terme de (II.16) soit négatif. Pour résoudre ce problème, on ajoute un autre terme de commande [14], [48],  $u_s$  à  $u_c$ .

La nouvelle commande résultante est alors:

$$u = u_c + u_s \quad (\text{II.17})$$

Ce nouveau terme ( $u_s$ ) est un contrôle de surveillance appelé superviseur.

En substituant (II.17) dans (II.8), on obtient la nouvelle équation de l'erreur:

$$\dot{e} = A_c e + b_c \left[ (\hat{f}(x/\theta_f) - f(x)) + (\hat{g}(x/\theta_g) - g(x)) u_c - g(x) u_s \right] \quad (\text{II.18})$$

En utilisant (II.18) et (II.15) on obtient:

$$\begin{aligned}\dot{V}_e &= -\frac{1}{2}e^T Qe + e^T P b_c \left[ (\hat{f}(x/\theta_f) - f(x)) + (\hat{g}(x/\theta_g) - g(x))u_c - g(x)u_s \right] \\ &\leq -\frac{1}{2}e^T Qe + |e^T P b| \left[ |\hat{f}(x/\theta_f)| + |f(x)| + |\hat{g}(x/\theta_g)u_c| + |g(x)u_c| \right] - e^T P b_c g(x)u_s\end{aligned}\quad (\text{II.19})$$

Pour concevoir la commande  $u_s$  tel que le dernier terme de (II.19) soit négatif, on a besoin de connaître les limites des fonctions  $f$  et  $g$ . Pour ce faire on considère la supposition suivante:

### II.3.1.1.1 Supposition 1

On détermine  $f^U(x)$ ,  $g^U(x)$  et  $g_L(x)$  tels que  $|f(x)| \leq f^U(x)$  et  $g_L(x) \leq g(x) \leq g^U(x)$  pour  $x \in U_c$ , où  $f^U(x) < \infty$ ,  $g^U(x) < \infty$  et  $g_L(x) > 0$ .

Sachant ces limites, le superviseur  $u_s$  est choisi comme suit [14], [48]:

$$u_s = I_1^* \operatorname{sgn}(e^T P b_c) \frac{1}{g_L(x)} \cdot \left[ |\hat{f}(x/\theta_f)| + f^U(x) + |\hat{g}(x/\theta_g)u_c| + |g^U(x)u_c| \right] \quad (\text{II.20})$$

$$\text{où } \begin{cases} I_1^* = 1 \text{ si } V_e > \tilde{V} \\ I_1^* = 0 \text{ si } V_e \leq \tilde{V} \end{cases} \quad \text{et} \quad \begin{cases} \operatorname{sgn}(e^T P b_c) = 1 \text{ si } (e^T P b_c) \geq 0 \\ \operatorname{sgn}(e^T P b_c) = -1 \text{ si } (e^T P b_c) < 0 \end{cases}$$

En substituant (II.20) dans (II.19) et en considérant le cas  $V_e > \tilde{V}$ , on a:

$$\begin{aligned}\dot{V}_e &= -\frac{1}{2}e^T Qe + |e^T P b_c| \times \left[ |\hat{f}| + |f| + |\hat{g}u_c| + |g u_c| - \frac{g}{g_L} \times (|\hat{f}| + f^U + |\hat{g}u_c| + |g^U u_c|) \right] \\ &\leq -\frac{1}{2}e^T Qe \leq 0\end{aligned}\quad (\text{II.21})$$

La prochaine étape consiste à remplacer  $\hat{f}$  et  $\hat{g}$  par leurs expressions floues données par (II.3).

Soient les paramètres optimaux suivants [14], [48]:

$$\begin{cases} \theta_f^* = \arg \min_{\theta_f \in \Omega_f} \left[ \sup_{x \in U_c} |\hat{f}(x/\theta_f) - f(x)| \right] \\ \theta_g^* = \arg \min_{\theta_g \in \Omega_g} \left[ \sup_{x \in U_c} |\hat{g}(x/\theta_g) - g(x)| \right] \end{cases} \quad (\text{II.22})$$

où  $\Omega_f$  et  $\Omega_g$  sont les ensembles des contraintes pour  $\theta_f$  et  $\theta_g$  respectivement, spécifiés par le concepteur.  $\Omega_f$  et  $\Omega_g$  sont définis par (II.23) et (II.24) respectivement.

$$\Omega_f = \{\theta : |\theta_f| \leq M_f\} \quad (\text{II.23})$$

$$\Omega_g = \{\theta_g : |\theta_g| \leq M_g, \bar{y}^l \geq \varepsilon\} \quad (\text{II.24})$$

où  $M_f$ ,  $M_g$  et  $\varepsilon$  sont des constantes positives spécifiées par le concepteur.

L'erreur d'approximation minimale définit comme suit:

$$w = (\hat{f}(x/\theta_f^*) - f(x)) + (\hat{g}(x/\theta_g^*) - g(x))u_c \quad (\text{II.25})$$

Alors, L'équation dynamique de l'erreur (II.18) peut être réécrite comme suit:

$$\dot{e} = A_c e - b_c g(x)u_s + b_c \left[ (\hat{f}(x/\theta_f) - \hat{f}(x/\theta_f^*)) + (\hat{g}(x/\theta_g) - \hat{g}(x/\theta_g^*))u_c + w \right] \quad (\text{II.26})$$

Si  $\hat{f}$  et  $\hat{g}$  sont du type (II.3), alors (II.26) peut être réécrite de la façon suivante:

$$\dot{e} = A_c e - b_c g(x)u_s + b_c w + b_c \left[ \phi_f^T \xi(x) + \phi_g^T \xi(x)u_c \right] \quad (\text{II.27})$$

où  $\phi_f = \theta_f - \theta_f^*$ ,  $\phi_g = \theta_g - \theta_g^*$  et  $\xi(x)$  représente les fonctions floues de base donnée par (II.5).

Considérons la fonction de Lyapunov :

$$V = \frac{1}{2} e^T P e + \frac{1}{2\gamma_1} \phi_f^T \phi_f + \frac{1}{2\gamma_2} \phi_g^T \phi_g \quad (\text{II.28})$$

où  $\gamma_1$  et  $\gamma_2$  sont des constantes positives.

La dérivée de  $V$  par rapport au temps est:

$$\begin{aligned} \dot{V} = & -\frac{1}{2} e^T Q e - g(x)e^T P b_c u_s + e^T P b_c w + \frac{1}{2\gamma_1} \phi_f^T (\dot{\theta}_f + \gamma_1 e^T P b_c \xi(x)) \\ & + \frac{1}{2\gamma_2} \phi_g^T (\dot{\theta}_g + \gamma_2 e^T P b_c \xi(x)u_c) \end{aligned} \quad (\text{II.29})$$

où on utilise (II.15) et  $\dot{\phi}_f = \dot{\theta}_f$ ,  $\dot{\phi}_g = \dot{\theta}_g$ , à partir de (II.20) et on sait que  $g(x) > 0$ , alors on a:

$$g(x)e^T P b_c u_s \geq 0.$$

Si on choisit les lois d'adaptation [14], [48]:

$$\dot{\theta}_f = -\gamma_1 e^T P b_c \xi(x) \quad (\text{II.30})$$

$$\dot{\theta}_g = -\gamma_2 e^T P b_c \xi(x) u_c \quad (\text{II.31})$$

Donc à partir de (II.29) on aura:

$$\dot{V} \leq -\frac{1}{2} e^T Q e + e^T P b_c w \quad (\text{II.32})$$

Le terme  $e^T P b_c w$  est de l'ordre de  $w$ , qui est négligeable devant l'erreur  $e$ . Ce qui permet de conclure que la loi de commande synthétisée garantit la stabilité du système au sens de Lyapunov ( $\dot{V} < 0$ ).

### II.3.1.1.2 Supposition 2

On dispose de descriptions linguistiques pour les fonctions inconnues  $f$  et  $g$  sous la forme:

$$R_f^{(r)} : \text{If } x_1 \text{ is } A_1^r \text{ and } \dots \text{ and } x_n \text{ is } A_n^r \text{ Then } f(x) \text{ is } C^r \quad (\text{II.33})$$

$$R_g^{(s)} : \text{If } x_1 \text{ is } B_1^s \text{ and } \dots \text{ and } x_n \text{ is } B_n^s \text{ Then } g(x) \text{ is } D^s \quad (\text{II.34})$$

avec  $A_i^r$ ,  $B_i^s$ ,  $C^r$  et  $D^s$  sont des ensembles flous dans  $\mathfrak{R}$ ,  $r=1,2,\dots,L_f$  et  $s=1,2,\dots,L_g$ . Si  $L_f = L_g = 0$ , alors il n'y a pas des descriptions linguistiques concernant  $f$  et  $g$ . Donc cette supposition n'est pas nécessaire.

### II.3.1.2 Conception du contrôleur adaptatif flou

Les fonctions  $\hat{f}(x/\theta_f)$  et  $\hat{g}(x/\theta_g)$  sont sélectionnées selon la forme (II.3). Nous allons présenter les étapes détaillées durant la conception du contrôleur adaptatif flou.

#### ✓ Etape 1: processus autonome

1. spécifier les  $k_1, \dots, k_n$  telles que toutes les racines de  $s^n + k_1 s^{n-1} + \dots + k_n$  sont dans le demi-plan gauche, et spécifier une matrice  $Q_{(n \times n)}$  définie positive.
2. résoudre l'équation de Lyapunov, pour obtenir la matrice symétrique  $P > 0$ .

3. spécifier les paramètres  $M_f, M_g, \varepsilon$  et  $\tilde{V}$ , basés sur des contraintes pratiques.

✓ **Etape 2: conception du contrôleur initial**

1. Définir les  $m_i$  ensembles flous  $F_i^{l_i}$  dont les fonctions d'appartenances sont

$$\mu_{F_i^{l_i}}, \text{ où } l_i = 1, 2, \dots, m_i \text{ et } i = 1, 2, \dots, n.$$

2. Construire les bases de règles floues pour les systèmes  $\hat{f}(x/\theta_f)$  et  $\hat{g}(x/\theta_g)$ , chacune consiste en  $m_1 \times m_2 \times \dots \times m_n$  règle dont les parties IF comprennent

toutes les combinaisons possibles des  $F_i^{l_i}$ . Spécifiquement les règles floues de base de  $\hat{f}$  et  $\hat{g}$  sont constituées de règles de la forme:

$$\begin{aligned} R_f^{(l_1, l_2, \dots, l_n)} : & \text{ If } x_1 \text{ is } F_1^{l_1} \text{ and } \dots \text{ and } x_n \text{ is } F_n^{l_n} \\ & \text{ Then } \hat{f}(x/\theta_f) \text{ is } G^{(l_1, l_2, \dots, l_n)} \end{aligned} \quad (\text{II.35})$$

$$\begin{aligned} R_g^{(l_1, l_2, \dots, l_n)} : & \text{ If } x_1 \text{ is } F_1^{l_1} \text{ and } \dots \text{ and } x_n \text{ is } F_n^{l_n} \\ & \text{ Then } \hat{g}(x/\theta_g) \text{ is } H^{(l_1, l_2, \dots, l_n)} \end{aligned} \quad (\text{II.36})$$

où  $l_i = 1, 2, \dots, m_i$ ,  $i = 1, 2, \dots, n$ ,  $G^{(l_1, l_2, \dots, l_n)}$  et  $H^{(l_1, l_2, \dots, l_n)}$  sont des ensembles flous dans  $\mathfrak{R}$ .

3. construction de fonctions floues de base

$$\xi^{(l_1, l_2, \dots, l_n)}(x) = \frac{\prod_{i=1}^n \mu_{F_i^{l_i}}(x_i)}{\sum_{l_1=1}^{m_1} \dots \sum_{l_n=1}^{m_n} \left( \prod_{i=1}^n \mu_{F_i^{l_i}}(x_i) \right)} \quad (\text{II.37})$$

Les fonctions  $\hat{f}(x/\theta_f)$  et  $\hat{g}(x/\theta_g)$  sont construites comme suit:

$$\hat{f}(x/\theta_f) = \theta_f^T \xi(x) \quad (\text{II.38})$$

$$\hat{g}(x/\theta_g) = \theta_g^T \xi(x) \quad (\text{II.39})$$

✓ **Etape 3 : Adaptation on line**

1. Appliquer la commande par retour d'état (II.17) au système (II.7), où  $u_c$  est données par (II.11),  $u_s$  est donnée par (II.20),  $\hat{f}$  et  $\hat{g}$  sont données par (II.38) et (II.39) respectivement.

2. Utiliser les lois d'adaptation suivantes pour ajuster le vecteur de paramètres  $\theta_f$ .

$$\dot{\theta}_f = \begin{cases} -\gamma_1 e^T P b_c \xi(x) & \text{si } (|\theta_f| < M_f) \text{ ou} \\ & (|\theta_f| = M_f \text{ et } e^T P b_c \theta_f^T \xi(x) \geq 0) \\ -\gamma_1 e^T P b_c \xi(x) + \gamma_1 e^T P b_c \frac{\theta_f \theta_f^T \xi(x)}{|\theta_f|^2} & \text{si} \\ & (|\theta_f| = M_f \text{ et } e^T P b_c \theta_f^T \xi(x) < 0) \end{cases} \quad (\text{II.40})$$

3. Utiliser les lois d'adaptation suivantes pour ajuster le vecteur de paramètres  $\theta_g$ , chaque fois qu'un élément  $\theta_{g_i}$  de  $\theta_g$  égal à  $\varepsilon$  on utilise:

$$\dot{\theta}_{g_i} = \begin{cases} -\gamma_2 e^T P b_c \xi_i(x) u_c & \text{si } e^T P b_c \xi_i(x) u_c < 0 \\ 0 & \text{si } e^T P b_c \xi_i(x) u_c \geq 0 \end{cases} \quad (\text{II.41})$$

où  $\xi_i(x)$  est le  $i^{\text{ème}}$  composant de  $\xi(x)$ .

autrement, on utilise:

$$\dot{\theta}_g = \begin{cases} -\gamma_2 e^T P b_c \xi(x) u_c & \text{si } (|\theta_g| < M_g) \text{ ou} \\ & (|\theta_g| = M_g \text{ et } e^T P b_c \theta_g^T \xi(x) u_c \geq 0) \\ -\gamma_2 e^T P b_c \xi(x) u_c + \gamma_2 e^T P b_c \frac{\theta_g \theta_g^T \xi(x)}{|\theta_g|^2} u_c & \text{si} \\ & (|\theta_g| = M_g \text{ et } e^T P b_c \theta_g^T \xi(x) u_c < 0) \end{cases} \quad (\text{II.42})$$

### II.3.1.3 Exemple de simulation

Il s'agit de développer un contrôleur adaptatif flou pour commander un pendule inversé dont l'équation dynamique et les paramètres de simulation du système sont déjà présentées dans le chapitre I (§ I.3.1).

#### II.3.1.3.1 Paramètres d'initialisation

-Le signal de référence  $x_{r1}(t) = y_m(t) = \frac{\pi}{30} \sin(t)$ .

-Les limites des fonctions  $f$  et  $g$  sont :

$$|f(x_1, x_2)| = \left| \frac{g \sin(x_1) - \frac{m l x_2^2 \cos x_1 \sin x_1}{m_c + m}}{l \left( \frac{4}{3} - \frac{m \cos^2(x_1)}{m_c + m} \right)} \right| \leq \frac{9.8 + \frac{0.025}{1.1} x_2^2}{\frac{2}{3} - \frac{0.05}{1.1}} = 15.78 + 0.036 x_2^2 = f^U(x_1, x_2) \quad (\text{II.43})$$

$$|g(x_1, x_2)| = \left| \frac{\cos(x_1) \frac{1}{m_c + m}}{l \left( \frac{4}{3} - \frac{m \cos^2(x_1)}{m_c + m} \right)} \right| \leq \frac{1}{1.1 \left( \frac{2}{3} - \frac{0.05}{1.1} \right)} = 1.46 = g^U(x_1, x_2) \quad (\text{II.44})$$

On exige que les états et la commande soient bornés comme suit:

$$|x_1| \leq \pi/6, \quad |x_2| \leq \pi/6 \quad \text{et} \quad |u| \leq 180.$$

$$|g(x_1, x_2)| \geq \frac{\cos(\pi/6)}{1.1 \left( \frac{2}{3} + \frac{0.05}{1.1} \cos^2(\pi/6) \right)} \leq \frac{1}{1.1 \left( \frac{2}{3} - \frac{0.05}{1.1} \right)} = 1.12 = g_L(x_1, x_2) \quad (\text{II.45})$$

-Les valeurs constantes  $M_f = 16$ ,  $M_g = 1.6$ ,  $\varepsilon = 0.7$ ,  $\gamma_1 = 50$ ,  $\gamma_2 = 1$  et  $\tilde{V} = 0.267$ .

### II.3.1.3.2 Mise en œuvre de l'algorithme de conception

On considère le vecteur de gain  $k = [2 \ 1]$  assurant la stabilité du système. La résolution de

$$\text{l'équation de Lyapunov donne (pour } Q = \text{diag}(10,10)) P = \begin{bmatrix} 15 & 5 \\ 5 & 5 \end{bmatrix}.$$

Les fonctions d'appartenance sont choisies de forme gaussiennes ( $m_1=m_2=5$ ,  $i=1,2$  et  $j=1,\dots,5$ ).

$$\mu_{F_i^j}(x_i) = \exp \left[ - \left( \frac{x_i + \pi/6 - (j-1)\pi/12}{\pi/24} \right)^2 \right].$$

### II.3.1.3.3 Résultats de simulation

Les figures (II.2) et (II.3) montrent les allures de l'angle  $x_1(t)$  et sa référence  $x_{r1}(t)$ , la vitesse angulaire  $x_2(t)$  et sa référence  $x_{r2}(t)$  respectivement. Les figures (II.4) et (II.5) illustrent les variations en fonction du temps de l'erreur de poursuite et la commande respectivement.

D'après les résultats de simulations obtenus on peut constater que notre contrôleur est capable de réaliser une bonne poursuite avec une bonne vitesse d'adaptation et une petite erreur de poursuite. L'allure de la commande montre qu'elle est continue et facile à mise en œuvre.

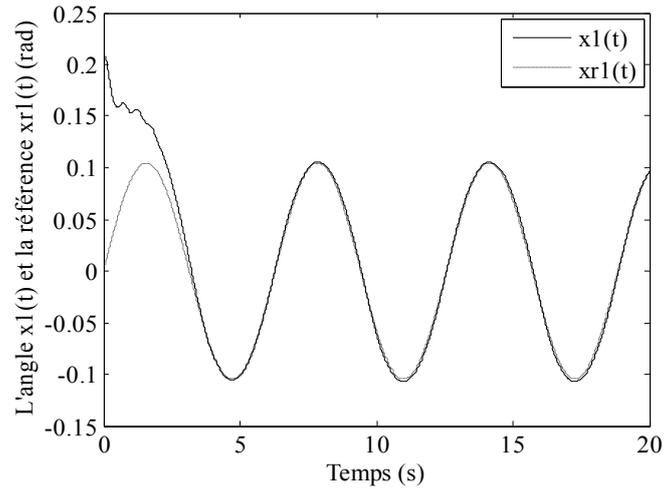


Figure II.2. Evolution de l'angle  $x_1(t)$  et le signal de référence  $x_{r1}(t)$

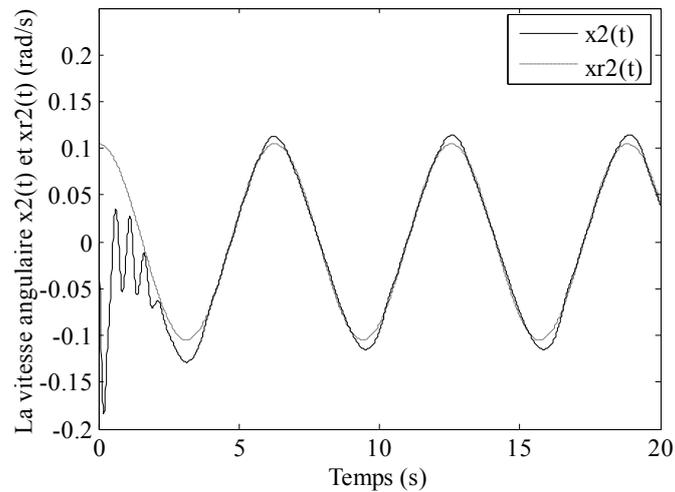


Figure II.3. Evolution de la vitesse angulaire  $x_2(t)$  et le signal de référence  $x_{r2}(t)$

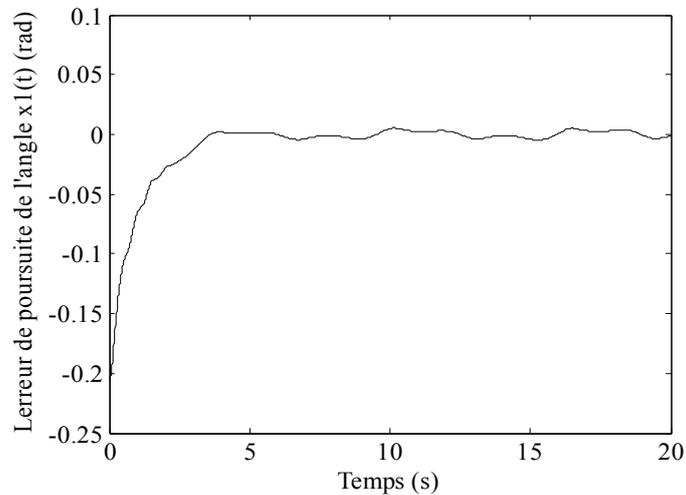
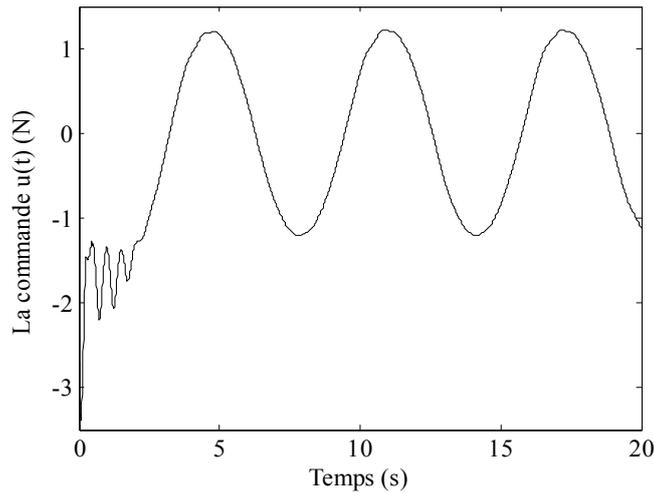


Figure II.4. L'erreur de poursuite  $e_1(t)$  de l'angle  $x_1(t)$ .

Figure I.15. La commande  $u(t)$ .

### II.3.2 Commande adaptative floue direct

La loi de commande est directement approximée par un système adaptatif flou, c'est-à-dire les systèmes flous sont utilisés comme des régulateurs.

#### II.3.2.1 Formulation du problème

Nous considérons une classe de systèmes non linéaires multi-entrées multi-sorties (MIMO) modélisés par [49]:

$$\begin{cases} \dot{x}_1^{(r_1)} = f_1(x) + \sum_{j=1}^p g_{1j}(x)u_j \\ \vdots \\ \dot{x}_p^{(r_p)} = f_p(x) + \sum_{j=1}^p g_{pj}(x)u_j \end{cases} \quad (\text{II.46})$$

où  $x = [x_1, \dot{x}_1, \dots, x_1^{(r_1-1)}, \dots, x_p, \dot{x}_p, \dots, x_p^{(r_p-1)}]^T \in \mathfrak{R}^n$  avec  $n = \sum_{i=1}^p r_i$  est le vecteur d'état global qui

est supposé disponible pour la mesure,  $u = [u_1, u_2, \dots, u_p]^T \in \mathfrak{R}^p$  est le vecteur de commande,

$y = [y_1, y_2, \dots, y_p]^T \in \mathfrak{R}^p$  est le vecteur de sortie, et  $f_i(x), g_{ij}(x)$  avec  $i, j = 1, \dots, p$  sont des fonctions non linéaires lisses inconnus.

Si on écrit:

$$y^{(r)} = \left[ y_1^{(r_1)}, \dots, y_p^{(r_p)} \right]^T, \quad f(x) = \left[ f_1(x), \dots, f_n(x) \right]^T \quad \text{et} \quad G(x) = \begin{bmatrix} g_{11}(x) & \dots & g_{1p}(x) \\ \vdots & \ddots & \vdots \\ g_{p1}(x) & \dots & g_{pp}(x) \end{bmatrix}.$$

Alors, le système dynamique (II.46) peut s'écrire sous la forme compacte suivante:

$$y^{(r)} = f(x) + G(x)u \quad (\text{II.47})$$

L'objectif du contrôle est de concevoir une loi de commande adaptative  $u_i(t)$  pour le système (II.46) de telle sorte que la sortie  $y_i(t)$  suit une trajectoire de référence  $y_{mi}(t) = x_{ri}(t)$  sous la contrainte que tous les signaux impliqués doivent être bornés.

Durant cette étude on a besoin aux hypothèses suivantes:

Hypothèse 1 [49]: La matrice  $G(x)$  est symétrique définie positive et bornée comme suit:

$$0 < \underline{g}I_p \leq G(x) \leq \bar{g}I_p, \quad \text{où } I_p \text{ est la matrice identité, } \underline{g} \text{ et } \bar{g} \text{ sont des constantes positives.}$$

Hypothèse 2 [49]: Le signal de référence  $y_{mi}(t)$  est un signal borné, connu et dérivable jusqu'à l'ordre  $r_i$ .

Notons que l'hypothèse 1 est une condition suffisante qui assure que la matrice  $G(x)$  est toujours singulière, par conséquent, le système (II.46) peut être linéarisé par un retour d'état.

On définit les erreurs de poursuite par:

$$\begin{cases} e_1(t) = y_{m1}(t) - y_1(t) \\ \vdots \\ e_p(t) = y_{mp}(t) - y_p(t) \end{cases} \quad (\text{II.48})$$

ainsi que les erreurs de poursuite filtrées:

$$\begin{cases} s_1(t) = \left( \frac{d}{dt} + \lambda_1 \right)^{r_1-1} e_1(t), \quad \lambda_1 > 0 \\ \vdots \\ s_p(t) = \left( \frac{d}{dt} + \lambda_p \right)^{r_p-1} e_p(t), \quad \lambda_p > 0 \end{cases} \quad (\text{II.49})$$

D'après (II.49),  $s_i(t) = 0$  représente une équation différentielle linéaire dont la solution implique que l'erreur de poursuite  $e_i(t)$  et ses dérivés jusqu'à l'ordre  $r_i - 1$  convergent vers zéro [34]. Donc, l'objectif de contrôle devient la conception d'un dispositif de commande pour

maintenir l'erreur de poursuite filtrée  $s_i(t)$  à zéro. Les bornes sur  $s_i(t)$  peut être directement traduit en des bornes sur  $e_i(t)$ , selon [34], si  $|s_i(t)| \leq \Phi_i$  où  $\Phi_i$  est une constante positive alors,  $|e_i^{(j)}(t)| \leq 2^j \lambda_i^{j-\eta+1} \Phi_i$ ,  $j = 0, \dots, r_i - 1$  et  $i = 1, \dots, p$ . Ces limites peuvent être réduites en augmentant les paramètres  $\lambda_i$ .

Les dérivées par rapport au temps des erreurs filtrées (II.49) peuvent être écrites comme suit:

$$\begin{cases} \dot{s}_1 = v_1 - f_1(x) - \sum_{j=1}^p g_{1j}(x)u_j \\ \vdots \\ \dot{s}_p = v_p - f_p(x) - \sum_{j=1}^p g_{pj}(x)u_j \end{cases} \quad (\text{II.50})$$

où  $v_1, \dots, v_p$ , sont donnés par:

$$\begin{cases} v_1 = y_{m1}^{(r_1)} + \beta_{1,r_1-1} e_1^{(r_1-1)} + \dots + \beta_{1,1} \dot{e}_1 \\ \vdots \\ v_p = y_{mp}^{(r_p)} + \beta_{p,r_p-1} e_p^{(r_p-1)} + \dots + \beta_{p,1} \dot{e}_p \end{cases} \quad (\text{II.51})$$

avec:  $\beta_{i,j} = C_{r_i-1}^{j-1} \lambda_i^{r_i-j}$ ,  $j = 0, \dots, r_i - 1$  et  $i = 1, \dots, p$ .

Notons que:  $s = [s_1, \dots, s_p]^T$  et  $v = [v_1, \dots, v_p]^T$ . Alors l'équation (II.50) peut être écrite sous la forme suivante:

$$\dot{s} = v - f(x) - G(x)u \quad (\text{II.52})$$

Si les fonctions non linéaires  $f(x)$  et  $G(x)$  sont connues, pour atteindre les objectifs de contrôle, on peut utiliser la loi commande non linéaire idéale suivante:

$$u^* = G^{-1}(x)(-f(x) + v + Ks + K_0 \tanh(s / \varepsilon_0)) \quad (\text{II.53})$$

où,  $K = \text{diag}(K_1, \dots, K_p)$ ,  $K_0 = \text{diag}(K_{01}, \dots, K_{0p})$ , avec  $K_i > 0$  et  $K_{0i} > 0$ ,  $i = 1, \dots, p$ ,  $\varepsilon_0$  est une petite constante positive, et  $\tanh(\cdot)$  est la fonction tangente hyperbolique définit pour le vecteur  $s = [s_1, \dots, s_p]^T$  par  $\tanh(s / \varepsilon_0) = [\tanh(s_1 / \varepsilon_0), \dots, \tanh(s_p / \varepsilon_0)]^T$ .

En effet, lorsque nous choisissons l'entrée de commande  $u = u^*$  l'équation (II.52) devient:

$$\dot{s} = -Ks - K_0 \tanh(s / \varepsilon_0) \quad (\text{II.54})$$

ou, de façon équivalente

$$\dot{s}_i = -K_i s_i - K_{0i} \tanh(s_i / \varepsilon_0) \quad (\text{II.55})$$

À partir de laquelle nous pouvons conclure que  $s_i(t) \rightarrow 0$  si  $t \rightarrow \infty$  et, par conséquent,  $e_i(t)$  et ses dérivés jusqu'à l'ordre  $r_i - 1$  convergent vers zéros.

Il est clair que si  $f(x)$  et  $G(x)$  sont inconnues, la loi de commande non linéaire proposée (II.53) n'est pas réalisable. Dans ce cas, afin de surmonter cette difficulté de conception nous proposons d'utiliser des systèmes flous de la forme (II.3) pour approximer tout la loi de commande idéale (II.53). L'erreur entre le contrôleur flou et le contrôleur idéal sera utilisée pour mettre à jour les paramètres libres du contrôleur flou.

Pour concevoir la loi de commande, nous représentons chaque composante du vecteur de commande idéale  $u^* = [u_1^*, \dots, u_p^*]$  par un système flou de la forme (II.4) comme suit:

$$u_i^*(x) = \xi_i^T(x) \theta_i^* + w_i(x) \quad (\text{II.56})$$

où  $w_i(x)$  est l'erreur d'approximation,  $\theta_i^*$  est le vecteur idéal des paramètres inconnus et  $\xi_i^T(x)$  est un vecteur de fonction floue de base donné par l'équation (II.5).

Notons que,

$$w(x) = [w_1(x), \dots, w_p(x)]^T, \theta^* = [\theta_1^{*T}, \dots, \theta_p^{*T}]^T \text{ et } \xi(x) = \text{diag}[\xi_1(x), \dots, \xi_p(x)]$$

Par conséquent, on peut écrire (II.53) comme suit:

$$u^*(x) = \xi^T(x) \theta^* + w(x) \quad (\text{II.57})$$

Puisque le vecteur idéal de paramètre  $\theta^*$  est inconnu, nous allons utiliser son estimation  $\theta$  pour former la commande adaptative:

$$u(x) = \xi^T(x) \theta \quad (\text{II.58})$$

L'étape suivante consiste à concevoir une loi d'adaptation pour ajuster le vecteur de paramètres libres  $\theta$  de telle sorte que la loi de commande  $u(x)$  présente une bonne approximation de la loi de commande idéale  $u^*(x)$ .

On définit l'erreur de la commande par:

$$e_u = u^* - u \quad (\text{II.59})$$

L'erreur  $e_u$  représente l'écart entre la fonction inconnue  $u^*(x)$  et son approximation floue en ligne (II.58), tandis que l'erreur d'approximation floue  $w(x)$  représente le minimum possible de l'écart entre  $u^*(x)$  et  $u(x)$ , c'est-à-dire  $w(x)$  représente le minimum possible de  $e_u$ .

Utilisant (II.57) et (II.58), (II.59) devient:

$$e_u = u^* - \xi^T(x)\theta = \xi^T(x)\tilde{\theta} + w(x) \quad (\text{II.60})$$

où  $\tilde{\theta} = \theta^* - \theta$  est le vecteur d'erreur d'estimation des paramètres.

Par l'addition et la soustraction de  $G(x)u^*$  au côté de droite de (II.52), on obtient l'équation d'erreur suivante:

$$\dot{s} = v - f(x) - G(x)u + G(x)u^* - G(x)u^* \quad (\text{II.61})$$

avec (II.53) et (II.60), (II.61) devient:

$$\dot{s} = -Ks - K_0 \tanh(s / \varepsilon_0) + G(x)e_u \quad (\text{II.62})$$

Maintenant, nous considérons une fonction de coût quadratique, qui mesure l'écart entre le contrôleur idéal et le contrôleur flou, définie comme suit:

$$J(\theta) = \frac{1}{2} e_u^T G(x) e_u = \frac{1}{2} (u^* - \xi^T(x)\theta)^T G(x) (u^* - \xi^T(x)\theta) \quad (\text{II.63})$$

Nous utilisons la méthode de descente de gradient comme dans [34] pour minimiser la fonction de coût (II.63) par rapport au vecteur de paramètres ajustable  $\theta$ . On aura la trajectoire minimale du vecteur  $\theta(t)$  générée par l'équation différentielle suivante:

$$\dot{\theta} = -\eta \nabla_{\theta} J(\theta) \quad (\text{II.64})$$

où  $\eta$  est un paramètre constant positif.

D'après (II.63), le gradient de  $J(\theta)$  par rapport au vecteur  $\theta$  est donné par:

$$\frac{\partial J(\theta)}{\partial \theta} = -\xi(x)G(x)e_u \quad (\text{II.65})$$

Par conséquent, (II.64) devient:

$$\dot{\theta} = \eta \xi(x)G(x)e_u \quad (\text{II.66})$$

Nous rappelons ici que le contrôleur idéal  $u^*$  est inconnu, donc le signal d'erreur  $e_u$  défini dans (II.59) n'est pas disponible. L'équation (II.62) sera utilisée pour surmonter cette difficulté de conception. En effet, à partir de (II.62), nous voyons que même si le vecteur d'erreur  $e_u$  n'est pas disponible, le vecteur  $G(x)e_u$  est disponible, et il est donné par:

$$G(x)e_u = \dot{s} + Ks + K_0 \tanh(s / \varepsilon_0) \quad (\text{II.67})$$

à partir de (II.67), on peut écrire (II.66) comme suit:

$$\dot{\theta} = \eta \xi(x) \{ \dot{s} + Ks + K_0 \tanh(s / \varepsilon_0) \} \quad (\text{II.68})$$

[50] a montré qu'une loi d'adaptation de la forme (II.68) ne garantit pas la bornitude de  $\tilde{\theta}$  en présence des erreurs d'approximation, qui sont inévitables dans tels systèmes adaptatifs. Alors, pour améliorer la robustesse de la loi d'adaptation (II.68) en présence des erreurs d'approximation, nous la modifier en introduisant le terme  $\sigma$ -modification comme suit [50]:

$$\dot{\theta} = \eta \xi(x) \{ \dot{s} + Ks + K_0 \tanh(s / \varepsilon_0) \} - \eta \sigma \theta \quad (\text{II.69})$$

où  $\sigma$  est une petite constante positive.

Le théorème suivant résume les résultats de stabilité du système de contrôle adaptatif direct proposé.

**Théorème 1** [49]: Considérons le système (II.46) avec la loi de commande (II.53). Supposons que les hypothèses 1, 2 sont vérifiées, l'erreur d'approximation  $w(x)$  dans (II.60) est bornée comme suit:  $\|w(x)\| \leq \bar{w}$  où  $\bar{w}$  est une constante positive, et les paramètres libres du vecteur  $\theta$

sont ajustés selon (II.69). Donc tous les signaux en boucle fermée sont uniformément bornés, et les erreurs de poursuite sont très proches au voisinage de l'origine.

### Démonstration

Utilisant la fonction de Lyapunov candidate suivante:

$$V = \frac{1}{2} s^T s + \frac{1}{2\eta} \tilde{\theta}^T \tilde{\theta} \quad (\text{II.70})$$

Nous calculons la dérivé de (II.70) par rapport au temps, et en utilisant (II.62) et (II.69), on obtient:

$$\dot{V} = -s^T K s - s^T K_0 \tanh(s / \varepsilon_0) + s^T G(x) e_u - \tilde{\theta}^T (\xi(x) G(x) e_u - \sigma \theta) \quad (\text{II.71})$$

avec (II.60), (II.71) devient:

$$\dot{V} = -s^T K s - s^T K_0 \tanh(s / \varepsilon_0) + s^T G(x) e_u - e_u^T G(x) e_u + w^T(x) G(x) e_u + \sigma \tilde{\theta}^T \theta \quad (\text{II.72})$$

En utilisant les inégalités suivantes:

$$\begin{cases} \sigma \tilde{\theta}^T \theta \leq -\frac{\sigma}{2} \|\tilde{\theta}\|^2 + \frac{\sigma}{2} \|\theta^*\|^2 \\ w^T(x) G(x) e_u \leq \frac{1}{4} e_u^T G(x) e_u + w^T(x) G(x) w(x) \\ s^T G(x) e_u \leq \frac{1}{2} e_u^T G(x) e_u + \frac{1}{2} s^T G(x) s \end{cases} \quad (\text{II.73})$$

on aura

$$\begin{aligned} \dot{V} \leq & -s^T \left( K - \frac{1}{2} G(x) \right) s - s^T K_0 \tanh(s / \varepsilon_0) - \frac{1}{4} e_u^T G(x) e_u - \frac{\sigma}{2} \|\tilde{\theta}\|^2 \\ & + w^T(x) G(x) w(x) + \frac{\sigma}{2} \|\theta^*\|^2 \end{aligned} \quad (\text{II.74})$$

Puisque  $w(x)$  et  $G(x)$  sont supposés bornées dans cette étude et  $\theta^*$  est un vecteur constant, nous pouvons définir une borne constante positive  $\psi$  comme suit:

$$\psi = \sup(w^T(x) G(x) w(x)) + \frac{\sigma}{2} \|\theta^*\|^2 \quad (\text{II.75})$$

Alors, (II.74) peut être simplifiée à:

$$\dot{V} \leq -s^T \left( K - \frac{1}{2} \bar{g} I_p \right) s - \frac{\sigma}{2} \|\tilde{\theta}\|^2 - s^T K_0 \tanh(s / \varepsilon_0) - \frac{1}{4} e_u^T G(x) e_u + \psi \quad (\text{II.76})$$

Nous supposons ici que chaque paramètre de conception  $K_i$  est choisi de tel sorte que  $K_i \geq \bar{g} / 2$ , et on prend  $\kappa = \min_{1 \leq i \leq p} (K_i - \bar{g} / 2)$ . Par conséquent (II.76) devient bornée par:

$$\dot{V} \leq -\alpha V - s^T K_0 \tanh(s / \varepsilon_0) - \frac{1}{4} e_u^T G(x) e_u + \psi \leq -\alpha V + \psi \quad (\text{II.77})$$

où  $\alpha = \min(\kappa, \sigma\eta)$ .

à partir de (II.77), on peut établir que la fonction candidate de Lyapunov satisfait la condition suivante:

$$0 \leq V(t) \leq \left( V(0) - \frac{\Psi}{\alpha} \right) e^{-\alpha t} + \frac{\Psi}{\alpha} \quad (\text{II.78})$$

Cette dernière condition implique que  $s(t)$  et  $\tilde{\theta}(t)$  sont uniformément bornées,  $\|s\| \leq \sqrt{2\Psi/\alpha}$ , on a donc les erreurs de poursuite bornées comme suit:  $|e_i^{(j)}(t)| \leq 2^j \lambda_i^{j-\eta+1} \sqrt{2\Psi/\alpha}$ ,  $j = 0, \dots, r_i - 1$  et  $i = 1, \dots, p$  [34].

### II.3.2.2 Exemple de simulation

Dans cette section, nous allons appliquer la commande adaptative floue directe proposée sur un système non linéaire MIMO (robot manipulateur à deux degré de liberté), dont la dynamique du système et ses paramètres sont déjà mentionnées dans le chapitre I (§ I.3.2).

#### II.3.2.2.1 Paramètres d'initialisation

-L'objectif du contrôle est de forcer les sorties du système  $q_1(t)$  et  $q_2(t)$  pour suivre les

trajectoires désirés:  $x_{r1}(t) = y_{m1}(t) = \sin\left(\frac{2\pi}{25}t\right)$ ,  $x_{r2}(t) = y_{m2}(t) = \sin\left(\frac{2\pi}{38}t\right)$ .

-Les paramètres constantes:  $\lambda_1 = 1$ ,  $\lambda_2 = 1$ ,  $\varepsilon_0 = 0.01$ ,  $\sigma = 0.001$ ,  $\eta = 5$ .

### II.3.2.2.2 Mise en œuvre de l'algorithme de conception

-On considère les matrices de gains:  $K = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ,  $K_0 = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$ .

-Les fonctions d'appartenance sont choisies de forme gaussienne comme suit:

$$\mu_{F_j^1}(x_j) = \exp\left[-\frac{1}{2}\left(\frac{x_j + 1.25}{0.6}\right)^2\right], \quad \mu_{F_j^2}(x_j) = \exp\left[-\frac{1}{2}\left(\frac{x_j}{0.6}\right)^2\right],$$

$$\mu_{F_j^3}(x_j) = \exp\left[-\frac{1}{2}\left(\frac{x_j - 1.25}{0.6}\right)^2\right], \quad j = 1, \dots, 4.$$

### II.3.2.2.3 Résultats de simulation

Les résultats présentés sur les figures (II.16) et (II.17) illustrent la poursuite au niveau du deux articulations  $q_1$  et  $q_2$  respectivement. Les figures (II.18) et (II.19) montrent les variations des erreurs de poursuite. La figure (II.20) présente le couple de contrôle.

Les résultats de simulation montrent que le système rejoint facilement la trajectoire de référence dans un temps court, ainsi que les erreurs de poursuite sont très faibles et les signaux de control sont faciles à implémenter.

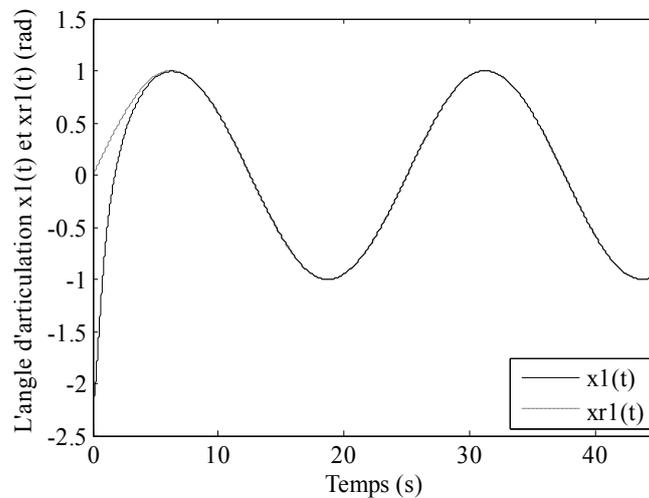


Figure III.16. Evolution de l'angle d'articulation  $x_1(t)$  et le signal de référence  $x_{r1}(t)$ .

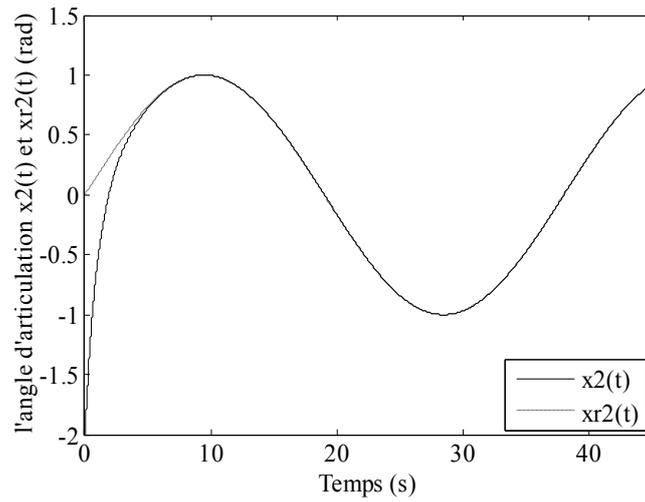


Figure III.17. Evolution de l'angle d'articulation  $x_2(t)$  et le signal de référence  $x_{r2}(t)$ .

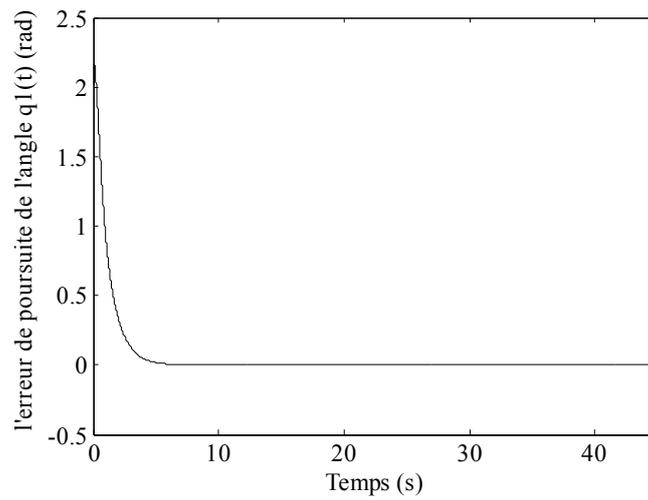


Figure III.18. L'erreur de poursuite  $e_1(t)$  de l'angle d'articulation  $x_1(t)$

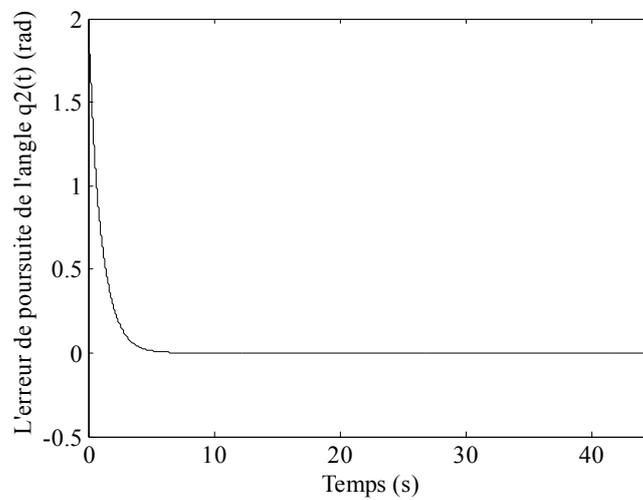


Figure III.19. L'erreur de poursuite  $e_2(t)$  de l'angle d'articulation  $x_2(t)$ .

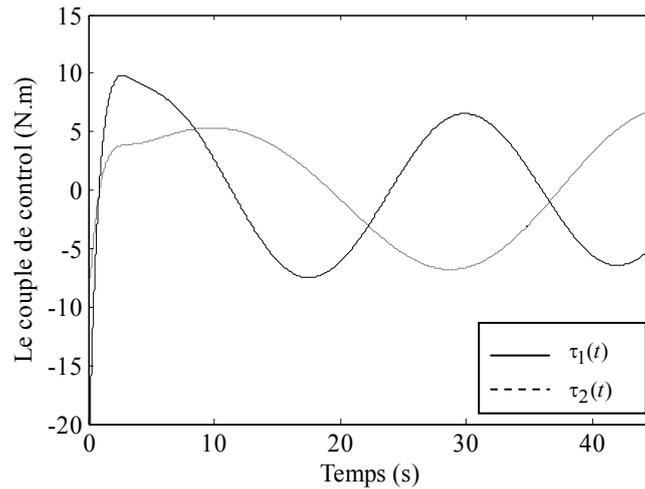


Figure III.20. Le couple de contrôle  $\tau_1(t)$  et  $\tau_2(t)$ .

## II.4 Conclusion

Dans ce chapitre, nous avons d'abord donné les notions de base de la théorie de la commande adaptative floue. Puis nous avons développé deux contrôleurs adaptatifs flous, l'un indirect et l'autre direct, capables d'incorporer directement les informations linguistique décrivant le fonctionnement du système, et garantir leur stabilité globale en boucle fermée dans le sens que tous les signaux impliqués soient uniformément bornés.

Pour valider nos approches et mettre en évidence les capacités de nos contrôleurs adaptatifs flous développés, deux exemples de simulation sont présentés, le premier est un pendule inversé commandé par un contrôleur adaptatif flou indirect, le second est un robot manipulateur à deux degrés de liberté piloté par un contrôleur adaptatif flou direct. Les résultats de simulation obtenus montrent de bonnes performances de poursuite avec une convergence rapide des erreurs de poursuite.

Dans le prochain chapitre, nous allons utiliser l'approche PDC pour combiner la commande floue avec la commande optimale linéaire quadratique en raison de concevoir des contrôleurs optimaux flous pour des systèmes non linéaires monovariables et multivariables.

---

*CHAPITRE III*

*COMMANDE OPTIMALE DES  
SYSTEMES FLOUS DE TAKAGI-  
SUGENO*

---

## COMMANDE OPTIMALE DES SYSTEMES FLOUS DE TAKAGI-SUGENO

### III.1 Introduction

Une manière élégante de représenter un système non linéaire repose sur une représentation multi-modèles. Celle-ci consiste en une collection de systèmes linéaires autour de différents points de fonctionnement interconnectés par des fonctions de pondération non linéaires normalisées appelées fonctions d'activation. Plusieurs catégories de multi-modèles existent dans la littérature, la plus utilisée est l'approche de modélisation de type Takagi-Sugeno (T- S). Une approche typique de la commande des systèmes utilisant ce type de modèle est la commande par retour d'état appelée commande PDC [19], [21], [23], [30]. La majorité des travaux qui s'inspirent de cette loi de commande ont une démarche basée sur la deuxième méthode de Lyapunov. Récemment, quelques chercheurs [24] ont proposé des approches basées sur la théorie de la commande LQ et les inégalités matricielles (LMIs) pour concevoir des contrôleurs PDC sous- optimaux et optimaux quadratiques pour les systèmes flous de T-S.

Dans ce chapitre nous allons commencer par un bref aperçu sur les modèles flous de type Takagi-Sugeno et leur stabilité basée sur une loi de commande de type PDC. Ensuite nous allons élaborer deux lois de commande PDC, la première est sous-optimale, la deuxième est optimale. A la fin du chapitre une étude comparative des résultats trouvés dans les trois chapitres sera présentée.

### III.2 Présentation des modèles flous de type Takagi-Sugeno (T-S)

Le modèle flou de Takagi-Sugeno (TS) d'un système dynamique est décrit par un ensemble de règles floues de type « Si –Alors » [12], [51]. Sa caractéristique principale est qu'il représente localement les relations entrées sorties d'un système en exprimant chaque conclusion par un modèle local invariant dans le temps (représentation d'état linéaire).

La  $i^{\text{ème}}$  règle du modèle flou s'écrit:

$$\text{Si } x_1(t) \text{ est } M_{i1} \text{ et } \dots \text{ et } x_n(t) \text{ est } M_{in} \text{ alors } \dot{x} = A_i x + B_i u \quad (\text{III.1})$$

où  $x \in \mathfrak{R}^n$  représente le vecteur d'état.  $M_{ij}$ ,  $j = 1, \dots, n$  est le  $j^{\text{ème}}$  ensemble flou de la  $i^{\text{ème}}$

règle,  $A_i \in \mathfrak{R}^{n \times n}$ ,  $B_i \in \mathfrak{R}^{n \times m}$  sont la matrice d'état et la matrice d'entrée du système respectivement,  $u \in \mathfrak{R}^m$  est le vecteur des commandes.

A chaque règle est attribué un poids noté  $w_i(x)$ . Ce poids dépend du degré d'appartenance des variables de prémisses  $x_j(t)$  aux sous-ensembles flous  $\mu_{ij}(x_j(t))$ .

$$w_i(x) = \prod_{j=1}^n \mu_{ij}(x_j) \quad (\text{III.2})$$

où  $\mu_{ij}(x_j)$  est la fonction d'appartenance de la  $j^{\text{ème}}$  ensemble flou dans la  $i^{\text{ème}}$  règle. On a alors les propriétés suivantes:

$$\begin{cases} \sum_{i=1}^r w_i(x(t)) > 0 \\ w_i(x(t)) \geq 0, \forall t \end{cases} \quad (\text{III.3})$$

$r$  est le nombre des règles floues.

On définit le coefficient  $\alpha_i$  par:

$$\alpha_i = \frac{w_i}{\sum_{i=1}^r w_i} \quad (\text{III.4})$$

Le coefficient  $\alpha_i(x(t))$  représente la fonction d'activation de la  $i^{\text{ème}}$  règle du modèle flou.

Pour  $i=1, \dots, r$ , ces fonctions vérifient la propriété d'une somme convexe, c'est-à-dire:

$$\begin{cases} \sum_{i=1}^r \alpha_i(x(t)) = 1 \\ \alpha_i(x(t)) \geq 0 \end{cases} \quad (\text{III.5})$$

Une fuzzification par singleton, une défuzzification par le centre de gravité, et le produit d'inférence, nous permettent d'écrire le modèle flou global comme suit:

$$\dot{x} = \frac{\sum_{i=1}^r w_i(x)(A_i x + B_i u)}{\sum_{i=1}^r w_i(x)} \quad (\text{III.6})$$

nous substituons (III.4) dans (III.6) le modèle flou global sera écrit sous la forme suivante:

$$\dot{x} = \sum_{i=1}^r \alpha_i(x)(A_i x + B_i u) \quad i=1, \dots, r \quad (\text{III.7})$$

### III.2 Stabilité des modèles T-S

On s'intéresse à la stabilité et à la stabilisation quadratique des modèles flous T-S, en utilisant la fonction de Lyapunov suivante:

$$V(x(t)) = x^T(t)Px(t), \quad P > 0 \quad (\text{III.8})$$

Le théorème suivant basé sur la seconde méthode de Lyapunov donne les conditions suffisantes permettant de garantir la stabilité de modèles flous continus.

**Théorème 1** [31] :

Le modèle flou (III.7) est asymptotiquement stable s'il existe une matrice commune  $P = P^T > 0$  telle que:

$$A_i^T P + P A_i < 0, \quad i = 1, \dots, r \quad (\text{III.9})$$

#### III.2.1 Stabilité des modèles T-S par retour d'état

Afin d'assurer la stabilité d'un modèle flou T-S en boucle fermée, on réalise la synthèse d'une loi de commande adéquate. Plusieurs lois de commande floues ont été proposées dans la littérature. Les plus répandues se basent sur des lois de commande de type compensation parallèle distribuée.

#### III.2.2 Compensation parallèle distribuée

La commande de type compensation parallèle distribuée, consiste à calculer une loi de commande linéaire par retour d'état, pour chaque sous-modèle du modèle flou. La détermination d'une loi de commande revient à déterminer pour chaque modèle local des gains matriciels, en utilisant par exemple une synthèse quadratique ou des LMI (*Linear Matrix Inequalities*) [30].

La commande de type PDC a pour but d'intégrer dans une seule loi de commande globale les lois de commande individuelles issues de l'approche multi-modèles. La partie antécédent des règles reste la même que pour le modèle flou T-S tandis que la partie conséquent est remplacée par une loi de commande par retour d'état. Dans ce contexte, chaque sous modèle du modèle flou de type T-S est stabilisé localement par une loi linéaire. La loi de commande globale est une fusion floue des lois de commande linéaires. Pour l'application de la commande de type PDC il est nécessaire que tous les sous-modèles linéaires soient stabilisables. La figure (III.1) illustre le concept de ce type de commande.

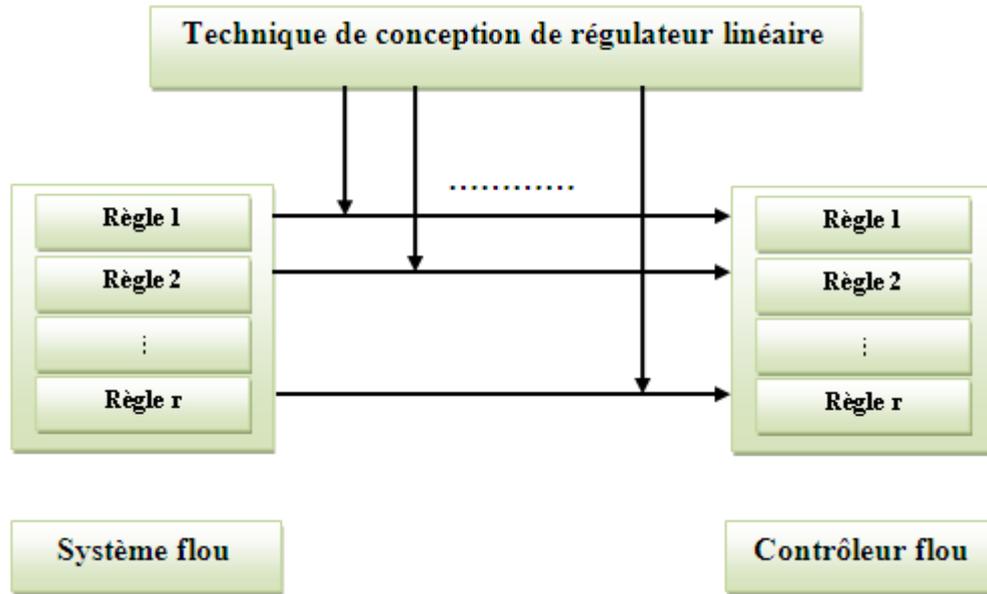


Figure III.1. Représentation du concept de compensation parallèle distribuée (PDC).

### III.2.3 Stabilisation des modèles T-S par une loi de commande de type PDC

Le régulateur flou PDC partage les mêmes ensembles flous que le modèle, donc, il garde les mêmes parties prémisses ainsi que les mêmes fonctions d'appartenance, la réalisation du régulateur se fait de la façon suivante:

Si  $x_1(t)$  est  $M_{i1}$  et .....et  $x_n(t)$  est  $M_{in}$  alors  $u = -K_i x$  (III.10)

Le contrôleur flou global est inféré comme suit:

$$u = -\sum_{i=1}^r \alpha_i(x) K_i x \quad (III.11)$$

Pour obtenir l'expression du système en boucle fermée, il suffit de substituer (III.11) dans (III.7). On a donc:

$$\dot{x} = \sum_{i=1}^r \alpha_i(x) (A_i x - B_i \sum_{j=1}^r \alpha_j(x) K_j x)$$

$$\dot{x} = \sum_{i=1}^r \sum_{j=1}^r \alpha_i(x) \alpha_j(x) (A_i - B_i K_j) x$$

(III.12)

En appliquant le théorème 1 à (III.12), il est possible de tirer des conditions de stabilité pour le système en boucle fermée. Le théorème suivant provient directement du théorème 1.

**Théorème 2** [31]: Le système flou en boucle fermée (III.12) est globalement, asymptotiquement stable s'il existe une matrice commune  $P = P^T > 0$  qui satisfait les inégalités de Lyapunov suivante:

$$\begin{cases} (A_i - B_i K_i)^T P + P(A_i - B_i K_i) < 0 & i = 1, \dots, r \\ G_{ij}^T P + P G_{ij} < 0 & i < j \leq r \end{cases} \quad (\text{III.13})$$

avec

$$G_{ij} = A_i - B_i K_j + A_j - B_j K_i \quad (\text{III.14})$$

L'obtention du régulateur flou PDC consiste donc à déterminer les matrices de gains de retour d'état  $K_i$  ( $i = 1, \dots, r$ ), satisfaisant les conditions du théorème 2.

Une façon de déterminer la matrice  $P$  et les gains de commande simultanément est l'utilisation des outils issus de l'optimisation convexe, et plus particulièrement des LMI. Certains outils LMI sont utilisables à l'aide du logiciel MATLAB [52].

En pré-multipliant et post- multipliant les deux côtés de les inégalités (III.13) par  $P^{-1}$  et on pose le changement de variable:

$$\begin{cases} Y = P^{-1} \\ X_i = K_i Y \end{cases} \quad (\text{III.15})$$

on obtient LMIs suivantes:

$$\begin{cases} Y A_i^T + A_i Y - B_i X_i - X_i^T B_i^T < 0 & i = 1, \dots, r \\ Y(A_i + A_j)^T + (A_i + A_j)Y - (B_i X_j + B_j X_i) - (B_i X_j + B_j X_i)^T < 0 & i < j \leq r \end{cases} \quad (\text{III.16})$$

La stabilité du système (III.12) est garantie si LMIs ci-dessus ont une solution commune définie positif.

Dans notre travail nous considérons que la stabilité seulement n'est pas suffisante, et que le contrôleur doit assurer certain degré de performance. Pour ceci nous allons utiliser le régulateur LQR pour déterminer les gains de commande des sous-systèmes du système flou (III.12), c'est-à-dire chaque sous modèle du modèle flou T-S est localement stable par une loi de commande optimale LQ. Cette technique connue sous les noms: « commande LQ floue parallèle distribuée » [24], « commande linéaire quadratique floue (FLQR) » [53], et « commande PDC sous optimale » [54].

### III.3 Commande PDC sous optimale

La commande PDC sous optimale consiste à calculer une loi de commande par retour d'état pour chaque modèle local où la détermination des gains de commande sera accomplie en minimisant le critère de performance quadratique (I.56). La loi de commande globale est une fusion floue des lois de commande linéaires optimales. On dit commande sous optimale car la loi de commande par retour d'état n'est optimale que pour chacun des sous-modèles linéaires. Dans la synthèse de ce type de commande le calcul des matrices de gains et la matrice  $P$  n'est pas simultanément, puisque nous commençons par le calcul des matrices des gains (stabilité locale est garantie), ensuite nous testons la stabilité globale en utilisant le théorème 2, si le système (III.12) n'est pas globalement asymptotiquement stable, d'après les auteurs dans [53] et [54] il faut changer un peu dans les valeurs des matrices de gains ou dans les points de fonctionnement pour atteindre la stabilité globale. Les variations dans les valeurs de gains dépendent des matrices de pondération et suivant le chapitre I l'OEP nous permet de réaliser un bon choix des matrices de pondération, donc nous avons une grande possibilité que les équations du théorème 2 soient également vérifiées.

En utilisant la loi de commande (I.58) pour chaque sous-système linéaire, on aura le contrôleur global écrit sous la forme suivante:

$$u = \sum_{i=1}^r \alpha_i(x) \left[ (K_i - B_i^{-1} A_i) x_r - K_i x(t) \right] \quad (\text{III.17})$$

En substituant (III.17) dans (III.7) on aura l'équation du système en boucle fermée comme suit:

$$\begin{aligned} \dot{x} &= \sum_{i=1}^r \alpha_i(x) (A_i x + B_i \sum_{j=1}^r \alpha_j(x) [(K_j - B_j^{-1} A_j) x_r - K_j x(t)]) \\ \dot{x} &= \sum_{i=1}^r \sum_{j=1}^r \alpha_i(x) \alpha_j(x) (A_i - B_i K_j) (x - x_r) \end{aligned} \quad (\text{III.18})$$

où  $x_r$  est la référence.

#### III.3.1 Applications

Dans cette section, il s'agit d'appliquer notre commande PDC sous optimale à la poursuite de trajectoire de deux systèmes non linéaires. On va utiliser les systèmes non

linéaires du chapitre I, avec les mêmes conditions initiales, et les mêmes signaux de références.

### III.3.1.1 Exemple 1: pendule inversé

Pour faciliter la mise en œuvre, nous utilisons le modèle linearisé sur deux points de fonctionnement selon deux règles floues de Takagi-Sugeno. Notons si  $x = \pm\pi/2$ , le système est non commandable, pour remédier à ce problème on va faire la linéarisation autour  $88^\circ$  au lieu de  $90^\circ$ .

Les deux règles floues sont:

Règle 1: Si  $x_1$  est proche de 0 alors  $\dot{x} = A_1x + B_1u$

Règle 2: Si  $x_1$  est proche de  $x = \pm\frac{\pi}{2}$  alors  $\dot{x} = A_2x + B_2$

avec:

$$A_1 = \begin{bmatrix} 0 & 1 \\ g/4l/3 - aml & 0 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ -a/g/4l/3 - aml \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 1 \\ 2g/\pi(4l/3 - aml\beta^2) & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ -a\beta/g/4l/3 - aml\beta^2 \end{bmatrix}, \beta = \cos(88^\circ).$$

où proche de zéro et proche de  $\pm\pi/2$  sont les ensembles flous d'entrée définis par les fonctions d'appartenance  $\mu_1(x_1) = 1 - \frac{2}{\pi}|x_1|$  et  $\mu_2(x_1) = \frac{2}{\pi}|x_1|$  respectivement, représentées dans la figure (III.2).

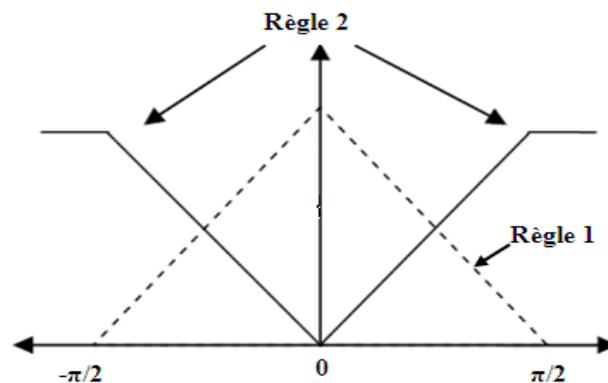


Figure III.2. Fonctions d'appartenance  $\mu_i(x_1)$

Les règles du contrôleur sont définies par:

Règle 1: Si  $x_1$  est proche de 0 alors  $u(t) = (K_1 - B_1^{-1}A_1)x_r - K_1x(t)$

Règle 2: Si  $x_1$  est proche de  $x = \pm \frac{\pi}{2}$  alors  $u(t) = (K_2 - B_2^{-1}A_2)x_r - K_2x(t)$

La loi de commande globale est donnée par l'équation suivante:

$$u(t) = \left[ \mu_1(x_1)(K_1 - B_1^{-1}A_1) + \mu_2(x_1)(K_2 - B_2^{-1}A_2) \right] x_r - \left[ \mu_1(x_1)K_1 + \mu_2(x_1)K_2 \right] x(t) \quad (\text{III.19})$$

Avec les mêmes paramètres de simulation de l'OEP mentionnés dans le chapitre I (§ I.3.1), nous choisissons les matrices de pondération ( $Q_1, R_1$ ) et ( $Q_2, R_2$ ) pour les deux sous-systèmes.

### Résultats de simulation

- ✓ les matrices de pondération

$$Q_1 = \begin{bmatrix} 4.5 & 0 \\ 0 & 0.89 \end{bmatrix}, R_1 = 0.1, Q_2 = \begin{bmatrix} 0.91 & 0 \\ 0 & 0.54 \end{bmatrix}, R_2 = 0.82.$$

- ✓ les gains de commande

$$K_1 = [-21.7667 \quad -5.5351], K_2 = [-57.8061 \quad -8.9414].$$

- ✓ la matrice  $P$  symétrique définie positif:  $P = \begin{bmatrix} 2.2921 & 0.1387 \\ 0.1387 & 0.0714 \end{bmatrix}$ ,

- ✓ les courbes

Les résultats de simulation de l'angle  $\theta(t) = x_1(t)$ , de la vitesse angulaire  $\dot{\theta}(t) = x_2(t)$ , de les erreurs de poursuite  $e_1(t), e_2(t)$ , et de la commande  $u(t)$  sont représentées par les figures (III.3), (III.4), (III.5), (III.6) et (III.7) respectivement.

D'après les figures (III.3) et (III.4) nous pouvons constater que notre contrôleur garanti de bonnes performances de poursuite.

Les figures (III.5) et (III.6) montrent une convergence rapide des erreurs de poursuite, et on peut remarquer que ces derniers sont très faibles.

La figure (III.7) montre que le signal de commande est continu et facile à implémenter.

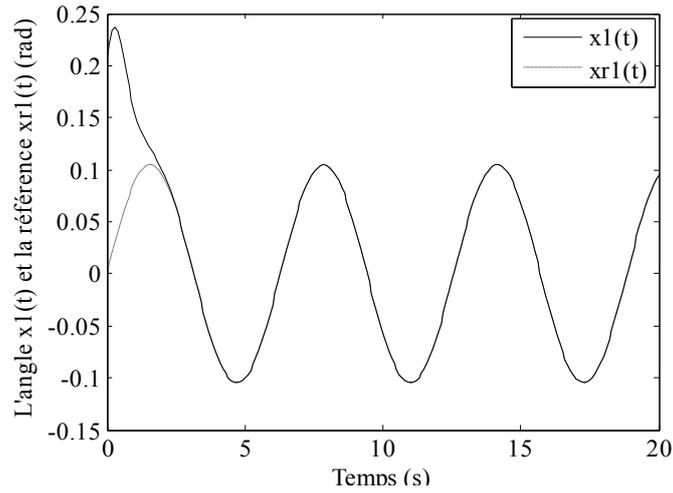


Figure III.3. Evolution de l'angle  $x_1(t)$  et le signal de référence  $x_{r1}(t)$

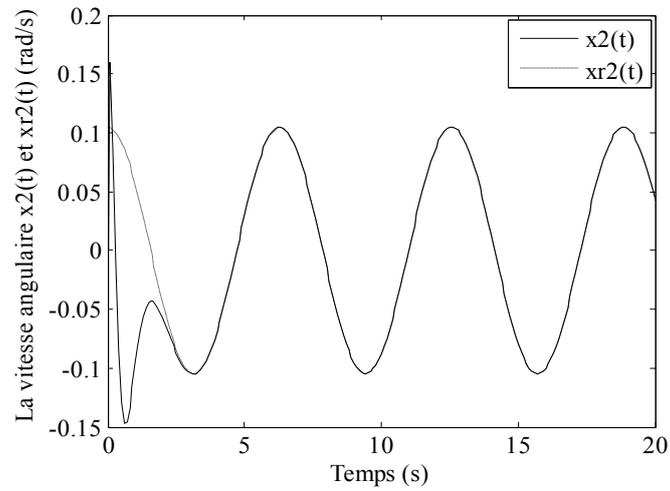


Figure III.4. Evolution de la vitesse angulaire  $x_2(t)$  et le signal de référence  $x_{r2}(t)$

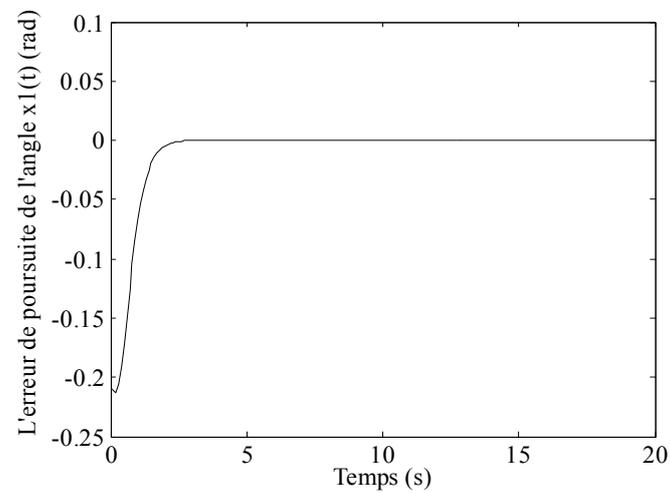


Figure III.5. L'erreur de poursuite  $e_1(t)$  de l'angle  $x_1(t)$ .

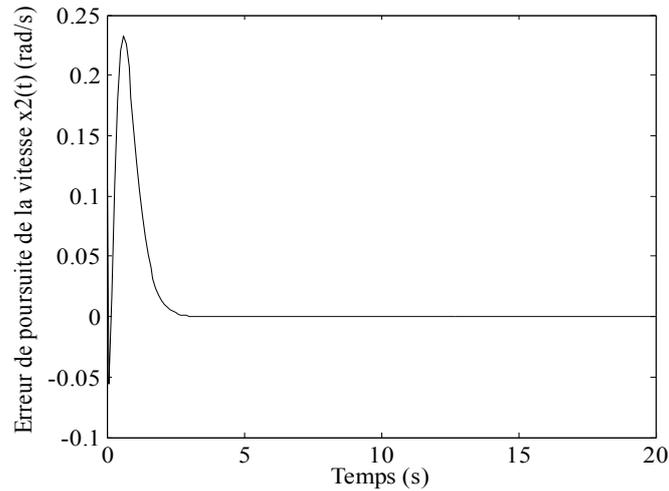


Figure III.6. L'erreur de poursuite  $e_2(t)$  de la vitesse angulaire  $x_2(t)$ .

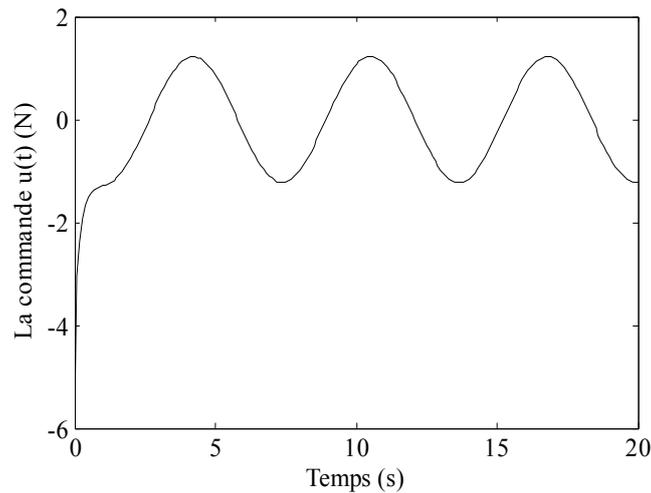


Figure III.7. La commande  $u(t)$ .

### III.3.1.2 Exemple 2 (robot manipulateur)

Nous allons utiliser le modèle linearisé sur deux points de fonctionnement selon quatre règles floues de Takagi-Sugeno comme suit:

Règle 1: Si  $x_1$  est proche de 0 et si  $x_2$  est proche de 0 alors  $\dot{x} = A_1x + B_1u$

Règle 2: Si  $x_1$  est proche de  $\pm \frac{\pi}{2}$  et si  $x_2$  est proche de 0 alors  $\dot{x} = A_2x + B_2u$

Règle 3: Si  $x_1$  est proche de 0 et si  $x_2$  est proche de  $\pm \frac{\pi}{2}$  alors  $\dot{x} = A_3x + B_3u$

Règle 4: Si  $x_1$  est proche de  $\pm \frac{\pi}{2}$  et si  $x_2$  est proche de  $\pm \frac{\pi}{2}$  alors  $\dot{x} = A_4x + B_4u$

avec :

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -0.099(10)^{-4} & -0.048(10)^{-4} \\ 0 & 0 & 0.267(10)^{-4} & 0.099(10)^{-4} \end{bmatrix}, B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.796 & -1.648 \\ -1.648 & 4.445 \end{bmatrix}, A_2 = A_1 \text{ et } B_2 = B_1.$$

$$A_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0.084(10)^{-4} & -0.052(10)^{-4} \\ 0 & 0 & -0.243(10)^{-4} & -0.084(10)^{-4} \end{bmatrix}, B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.503 & -0.801 \\ -0.801 & 2.243 \end{bmatrix}, A_4 = A_3 \text{ et } B_4 = B_3.$$

Nous utilisons les mêmes ensembles flous d'entrées comme dans l'exemple 1 précédent.

Les règles floues de notre contrôleur T-S sont:

Règle 1: Si  $x_1$  est proche de 0 et si  $x_2$  est proche de 0 alors  $u(t) = (K_1 - B_1^{-1}A_1)x_r - K_1x(t)$

Règle 2: Si  $x_1$  est proche de  $\pm \frac{\pi}{2}$  et si  $x_2$  est proche de 0 alors  $u(t) = (K_2 - B_2^{-1}A_2)x_r - K_2x(t)$

Règle 3: Si  $x_1$  est proche de 0 et si  $x_2$  est proche de  $\pm \frac{\pi}{2}$  alors  $u(t) = (K_3 - B_3^{-1}A_3)x_r - K_3x(t)$

Règle 4: Si  $x_1$  est proche de  $\pm \frac{\pi}{2}$  et si  $x_2$  est proche de  $\pm \frac{\pi}{2}$  alors  $u(t) = (K_4 - B_4^{-1}A_4)x_r - K_4x(t)$

La loi de commande globale est donnée par l'équation suivante:

$$u = [\mu_1(x_1)\mu_1(x_2) + \mu_2(x_1)\mu_1(x_2)] [(K_1 - B_1^{-1}A_1)x_r - K_1x(t)] + [\mu_1(x_1)\mu_2(x_2) + \mu_2(x_1)\mu_2(x_2)] [(K_3 - B_3^{-1}A_3)x_r - K_3x(t)] \quad (\text{III.20})$$

Nous utilisons les mêmes paramètres de simulation de l'OEP mentionnés dans le chapitre I (§ I.3.2), après simulation nous avons obtenu les résultats suivantes:

### Résultats de simulation

✓ les matrices de pondération

$$Q_1 = \begin{bmatrix} 1.4 & 0 & 0 & 0 \\ 0 & 0.95 & 0 & 0 \\ 0 & 0 & 0.84 & 0 \\ 0 & 0 & 0 & 0.53 \end{bmatrix}, R_1 = \begin{bmatrix} 0.93 & 0 \\ 0 & 0.18 \end{bmatrix}, Q_3 = \begin{bmatrix} 0.86 & 0 & 0 & 0 \\ 0 & 0.74 & 0 & 0 \\ 0 & 0 & 0.48 & 0 \\ 0 & 0 & 0 & 0.42 \end{bmatrix},$$

$$R_3 = \begin{bmatrix} 1.04 & 0 \\ 0 & 0.21 \end{bmatrix}.$$

- ✓ les gains de commande

$$K_1 = K_2 = \begin{bmatrix} 8.6575 & 0.1703 & 13.8440 & 1.4173 \\ -0.2199 & 6.7060 & 1.1315 & 10.5681 \end{bmatrix},$$

$$K_3 = K_4 = \begin{bmatrix} 8.7584 & 0.1406 & 13.3128 & 1.1329 \\ -0.1815 & 6.7067 & 0.8969 & 10.5909 \end{bmatrix}.$$

- ✓ la matrice  $P$  symétrique définie positif

$$P = \begin{bmatrix} 1.1960 & 0.0995 & 0.4647 & 0.1718 \\ 0.0995 & 0.7111 & 0.1440 & 0.0685 \\ 0.4647 & 0.1440 & 0.7729 & 0.2891 \\ 0.1718 & 0.0685 & 0.2891 & 0.1310 \end{bmatrix}.$$

- ✓ les courbes

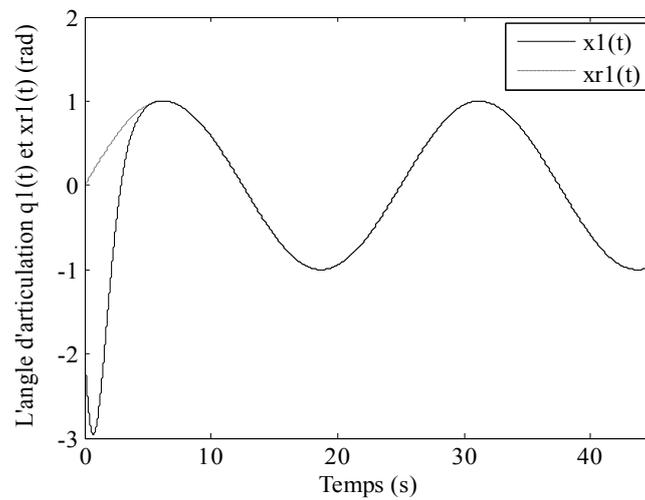


Figure III.8. Evolution de l'angle d'articulation  $x_1(t)$  et le signal de référence  $x_{r1}(t)$ .

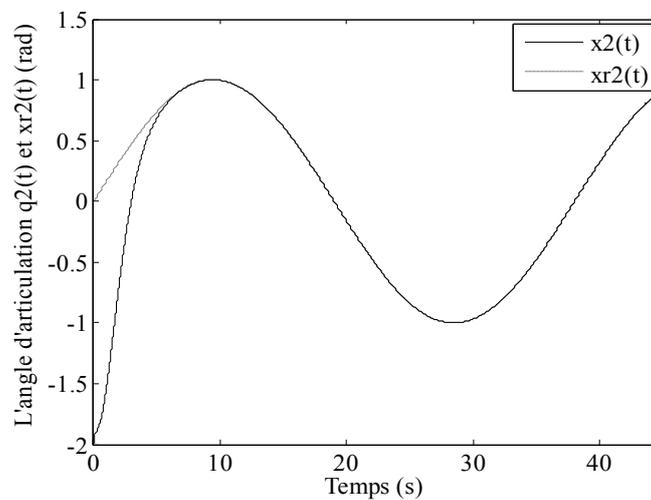


Figure III.9. Evolution de l'angle d'articulation  $x_2(t)$  et le signal de référence  $x_{r2}(t)$ .

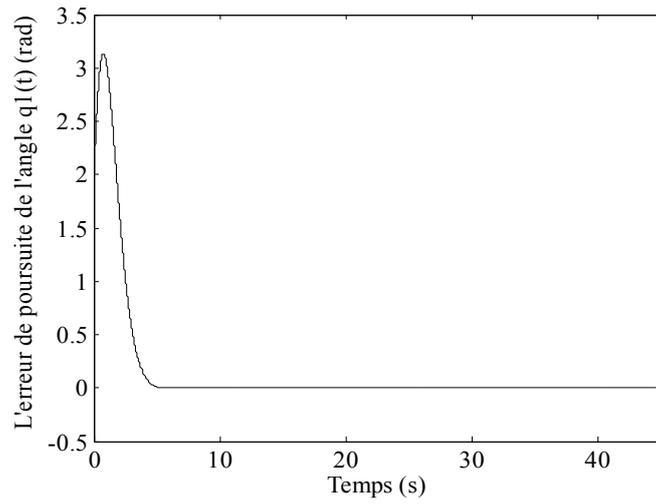


Figure I.10. L'erreur de poursuite  $e_1(t)$  de l'angle d'articulation  $x_1(t)$

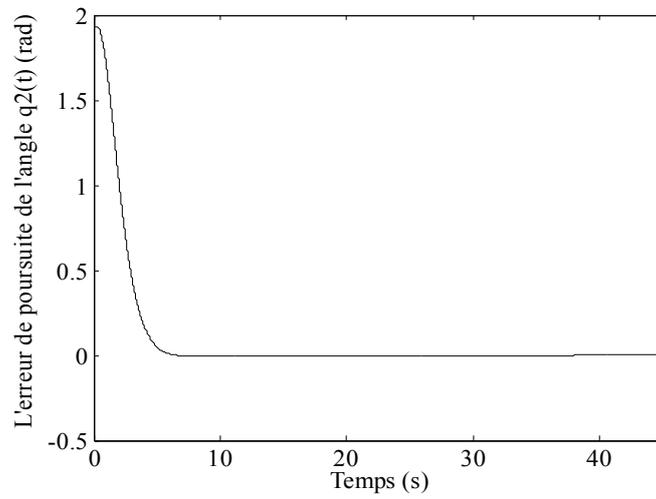


Figure III.11. L'erreur de poursuite  $e_2(t)$  de l'angle d'articulation  $x_2(t)$ .

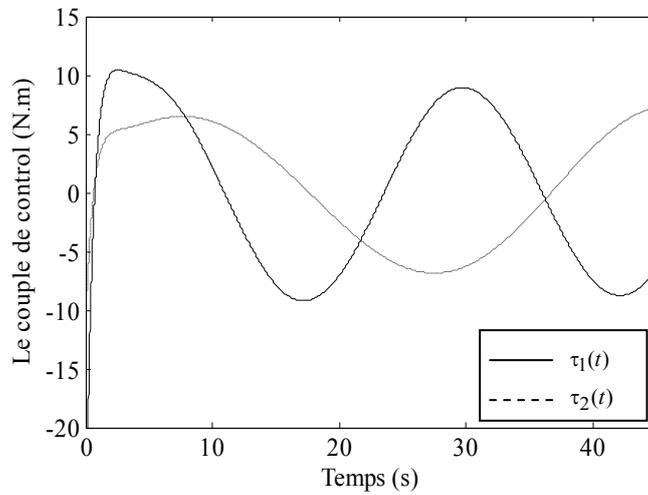


Figure III.12. Le couple de contrôle  $\tau_1(t)$  et  $\tau_2(t)$ .

D'après les résultats des figures (III.8) et (III.9), on remarque une très bonne convergence des états du robot vers leurs références.

Les figures (III.10) et (III.11) montrent que les erreurs de poursuite sont très faibles et convergent rapidement vers zéro.

La figure (III.12) illustre que les deux signaux du couple de contrôle ont une forme lisse, continue et facile à implémenter.

### III.4 Commande PDC optimale

Dans la technique de commande précédente nous avons mentionné que la loi de commande n'est optimale que pour les systèmes linéaires locaux, et que nous avons besoin de faire un test sur la stabilité du système en boucle fermée (stabilité globale). Pour rendre la loi de commande globalement optimale, il faut que la commande LQ soit applicable à tous les sous-systèmes linéaires simultanément, c'est pour ça qu'on va reformuler le problème LQR afin de trouver une fonction de Lyapunov commune pour l'ensemble des systèmes locaux [55]. Ceci est accompli par la représentation du problème sous une forme permet l'utilisation des outils de l'optimisation convexe (les LMIs).

Dans la synthèse du régulateur LQR, le problème de minimisation du critère de performance (I.56) se réduit à trouver une solution  $P$  positive de l'équation de Lyapunov (I.53). Cette fois on va résoudre l'équation de Lyapunov (III.21) dans le cas du système en boucle fermée (III.18).

L'équation de Lyapunov en boucle fermée est donnée comme suit:

$$(A - BK)^T P + P(A - BK) + Q + K^T R K = 0 \quad (III.21)$$

#### Démonstration

D'après (I.53) et (I.55) on a:  $A^T P + PA - PBR^{-1}B^T P + Q = 0$  et  $K = R^{-1}B^T P$  respectivement. En rajoutant et retranchant  $PBR^{-1}B^T P$  dans (I.53) on aura:

$$(A^T P - PBR^{-1}B^T P) + (PA - PBR^{-1}B^T P) + Q + PBR^{-1}B^T P = 0,$$

$$(A^T - \underbrace{PBR^{-1}B^T}_{K^T})P + P(A - \underbrace{BR^{-1}B^T}_{K})P + Q + \overbrace{PBR^{-1}R}^I R^{-1}B^T P = 0,$$

$$(A^T - K^T B^T)P + P(A - BK) + Q + \underbrace{PBR^{-1}}_{K^T} R \underbrace{R^{-1}B^T}_{K} P = 0,$$

$$(A - BK)^T P + P(A - BK) + Q + K^T R K = 0.$$

### III.4.1 Représentation du problème LQR via LMI

On peut écrire le minimum du critère de performance (I.31) comme suit [56]:

$$\min \{J\} = x(0)^T P x(0) \quad (\text{III.22})$$

Si nous écrivons l'équation de Lyapunov (III.21) sous forme d'inégalité matricielle au lieu d'une égalité, la solution de l'inégalité sera une borne supérieure sur le critère de performance  $J$ , et par la minimisation de cette borne supérieure on peut atteindre le  $\min\{J\}$ . Ce que nous avons dit précédemment est concernant un seul système linéaire invariant, mais on peut l'étendre au cas du système décrit par l'équation (III.18). Notons que le critère de performance  $J$  est dépend des conditions initiales, alors pour éviter ça l'auteur dans [55] a donné la proposition suivante:

$$\begin{cases} E\{0\} = 0 \\ E\{x(0)x(0)^T\} = I \end{cases} \quad (\text{III.23})$$

où  $E$  est l'opérateur de l'espérance mathématique.

L'objectif est de minimiser la valeur prévue de l'indice de performance  $J$  par rapport à toutes les conditions initiales possibles de moyenne nulle et de covariance égale à l'identité.

**Lemme 1** [56]: Pour des conditions initiales de moyenne nulle et de covariance égale à l'identité, nous avons:

$$E_{x(0)} \{x(0)^T P x(0)\} = \text{tr}(P) \quad (\text{III.24})$$

#### **Démonstration**

En utilisant la propriété:  $\text{tr}(Hba^T) = \text{tr}(a^T Hb)$  sur le côté gauche de l'équation (III.24), nous aurons:

$$E_{x(0)} \{x(0)^T P x(0)\} = E_{x(0)} \{\text{tr}(P x(0)x(0)^T)\} = \text{tr}\{P \underbrace{E_{x(0)}(x(0)x(0)^T)}_I\} = \text{tr}(P)$$

Utilisant lemme 1 on peut énoncer le théorème suivant:

**Théorème 3** [56]: Considérons le système flou en boucle fermée (III.18), nous pouvons écrire la borne supérieure sur le critère de performance comme suit:

$$J = E_{x(0)} \int_0^{\infty} (x^T Q x + u^T R u) dt < tr(P) \quad (III.25)$$

où  $P$  est la solution de l'inégalités suivantes:

$$\begin{cases} (A_i - B_i K_i)^T P + P(A_i - B_i K_i) + Q + \sum_{i=1}^r K_i^T R K_i < 0 \\ G_{ij}^T P + P G_{ij} + Q + \sum_{i=1}^r K_i^T R K_i < 0 & i = 1, \dots, r \\ & i < j \leq r \end{cases} \quad (III.26)$$

La commande  $u$  est définie comme dans l'équation (III.17).

### Démonstration

Nous savons que  $J < tr(\hat{P})$  où  $\hat{P}$  satisfait les inégalités suivantes:

$$\begin{cases} (A_i - B_i K_i)^T \hat{P} + \hat{P}(A_i - B_i K_i) + Q + (\sum_{i=1}^r \alpha_i K_i)^T R (\sum_{i=1}^r \alpha_i K_i) < 0 \\ G_{ij}^T \hat{P} + \hat{P} G_{ij} + Q + (\sum_{i=1}^r \alpha_i K_i)^T R (\sum_{i=1}^r \alpha_i K_i) < 0 & i = 1, \dots, r \\ & i < j \leq r \end{cases} \quad (III.27)$$

On a juste besoin de montrer que:

$$(\sum_{i=1}^r \alpha_i K_i)^T R (\sum_{i=1}^r \alpha_i K_i) < \sum_{i=1}^r K_i^T R K_i \quad (III.28)$$

Pour la simplicité, nous allons montrer que l'inégalité ci-dessus est valable dans le cas où on a juste deux règles pour le contrôleur, l'extension à plus de deux règles peut être faite par induction. Alors nous avons besoin de montrer que:

$$(\alpha_1 K_1 + \alpha_2 K_2)^T R (\alpha_1 K_1 + \alpha_2 K_2) < K_1^T R K_1 + K_2^T R K_2 \quad (III.29)$$

Pour illustrer ça, nous réécrivons le côté gauche de l'équation ci-dessus sous la forme quadratique suivante:

$$\begin{bmatrix} K_1^T R^{1/2} K_2^T R^{1/2} \\ \alpha_1 \alpha_2 \end{bmatrix} \begin{bmatrix} \alpha_1^2 & \alpha_1 \alpha_2 \\ \alpha_1 \alpha_2 & \alpha_2^2 \end{bmatrix} \begin{bmatrix} R^{1/2} K_1 R^{1/2} K_2 \end{bmatrix} \quad (\text{III.30})$$

Le côté droit de l'équation peut être écrit comme suit:

$$\begin{bmatrix} K_1^T R^{1/2} K_2^T R^{1/2} \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R^{1/2} K_1 R^{1/2} K_2 \end{bmatrix} \quad (\text{III.31})$$

Pour prouver le théorème, il faut montrer que:

$$\begin{bmatrix} \alpha_1^2 & \alpha_1 \alpha_2 \\ \alpha_1 \alpha_2 & \alpha_2^2 \end{bmatrix} < \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{III.32})$$

Cela est déjà satisfaite car nous avons  $0 < \alpha_i < 1$ , donc la différence entre les deux matrices est définie positif comme suit:

$$\begin{bmatrix} 1 - \alpha_1^2 & \alpha_1 \alpha_2 \\ \alpha_1 \alpha_2 & 1 - \alpha_2^2 \end{bmatrix} > 0 \quad (\text{III.33})$$

Maintenant, en utilisant le même changement de variables comme dans (III.15), et en multipliant les deux côtés de l'équation (III.26) par  $P^{-1}$  et selon le théorème 2, nous pouvons écrire (III.26) comme suit:

$$\begin{cases} N_i + YQY + \sum_{i=1}^r X_i^T R X_i < 0, & i = 1, \dots, r \\ T_{ij} + YQY + \sum_{i=1}^r X_i^T R X_i < 0, & i < j \leq r \end{cases} \quad (\text{III.34})$$

où

$$N_i = Y A_i^T + A_i Y - B_i X_i - X_i^T B_i^T \quad (\text{III.35})$$

$$T_{ij} = Y(A_i - A_j)^T + (A_i - A_j)Y - S_{ij} - S_{ij}^T \quad (\text{III.36})$$

$$S_{ij} = B_i X_j + B_j X_i \quad (\text{III.37})$$

Utilisant lemme d'LMI [57], nous pouvons écrire l'inégalité ci-dessus sous la forme suivante:

$$\left\{ \begin{array}{l} \left[ \begin{array}{ccccc} N_i & YQ^{1/2} & X_1^T R^{1/2} & \dots & X_r^T R^{1/2} \\ Q^{1/2} Y & -I_n & 0 & \dots & 0 \\ R^{1/2} X_1 & 0 & -I_m & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R^{1/2} X_r & 0 & 0 & \dots & -I_m \end{array} \right] < 0, i = 1, \dots, r \\ \\ \left[ \begin{array}{ccccc} T_{ij} & YQ^{1/2} & X_1^T R^{1/2} & \dots & X_r^T R^{1/2} \\ Q^{1/2} Y & -I_n & 0 & \dots & 0 \\ R^{1/2} X_1 & 0 & -I_m & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R^{1/2} X_r & 0 & 0 & \dots & -I_m \end{array} \right] < 0, i < j \leq r \end{array} \right. \quad (III.38)$$

Pour obtenir le minimum possible de la borne supérieure, on a le problème d'optimisation suivant:

$$\left\{ \begin{array}{l} \text{Min } tr(Y^{-1}) \\ \text{Sous les contraintes (III.38)} \end{array} \right. \quad (III.39)$$

Il s'agit d'un problème d'optimisation convexe qui peut être résolu en utilisant Matlab LMI Toolbox. Pour rendre possible l'utilisation de Matlab LMI Toolbox, on introduit une variable artificielle  $Z$  comme une borne supérieure sur  $Y^{-1}$ , et on minimise la  $tr(Z)$  au lieu de la  $tr(Y^{-1})$  [56], donc le problème sera écrit sous la forme suivante:

$$\left\{ \begin{array}{l} \text{Min } tr(Z) \\ \text{Sous les contraintes } \left\{ \begin{array}{l} (III.38) \\ \left[ \begin{array}{cc} Z & I_n \\ I_n & Y \end{array} \right] > 0 \end{array} \right. \end{array} \right. \quad (III.40)$$

Si les inégalités ci-dessus sont faisables, on peut calculer les gains de la commande par la relation suivante:

$$K_i = X_i Y^{-1} \quad (III.41)$$

### III.4.2 Applications

Pour illustrer l'approche traitée dans cette section, nous allons l'appliquer sur les deux

systemes non linéaires mentionnés précédemment, avec les mêmes conditions initiales, et les mêmes signaux de références.

### III.4.2.1 Exemple 1: pendule inversé

Nous avons suivi les mêmes étapes mentionnés dans (§ III.3.1.1), sauf pour les matrices de pondération qui sont choisies cette fois pour le système global.

#### Résultats de simulation

- ✓ Les matrices de pondération

$$Q = \begin{bmatrix} 4.06 & 0 \\ 0 & 0.75 \end{bmatrix}, R = 0.18.$$

- ✓ les gains de commande:

$$K_1 = [-18.9373 \quad -4.2633], K_2 = [-40.9196 \quad -5.3347].$$

- ✓ la matrice  $P$  symétrique définie positif:

$$P = \begin{bmatrix} 7.7880 & 2.4117 \\ 2.4117 & 1.3150 \end{bmatrix}.$$

- ✓ les courbes:

Après la simulation nous avons trouvé les résultats de la poursuite, des erreurs de poursuite et de la commande, illustrées dans les figures suivantes:

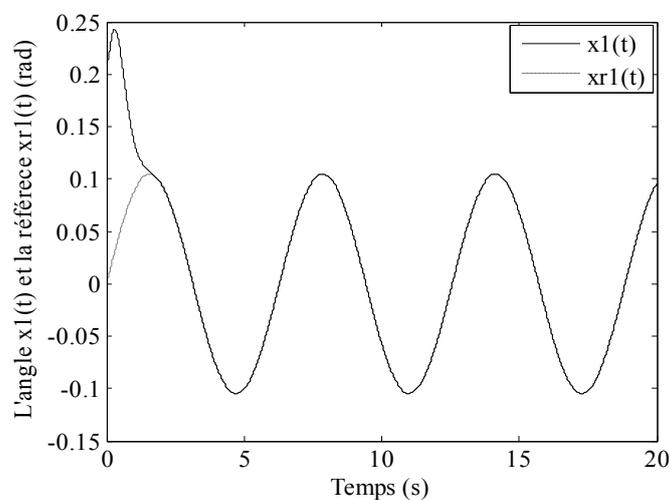


Figure III.13. Evolution de l'angle  $x_1(t)$  et le signal de référence  $x_{r1}(t)$

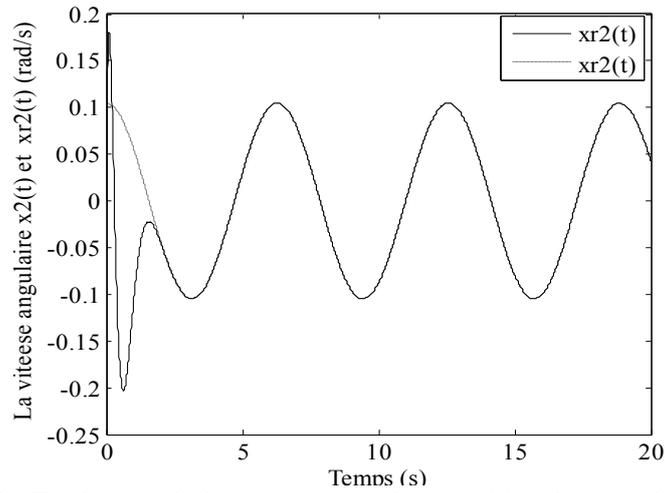


Figure III.14. Evolution de la vitesse angulaire  $x_2(t)$  et le signal de référence  $x_{r2}(t)$

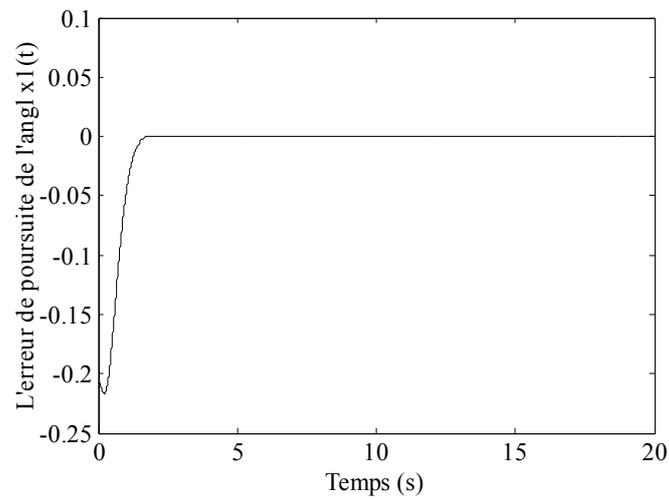


Figure III.15. L'erreur de poursuite  $e_1(t)$  de l'angle  $x_1(t)$ .

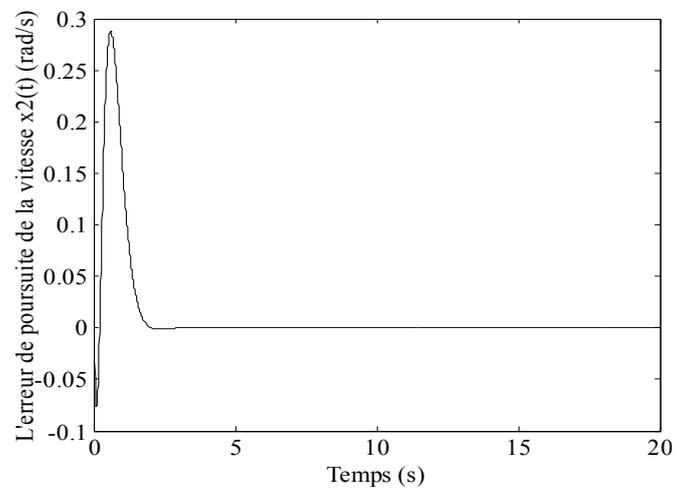


Figure III.16. L'erreur de poursuite  $e_2(t)$  de la vitesse angulaire  $x_2(t)$ .

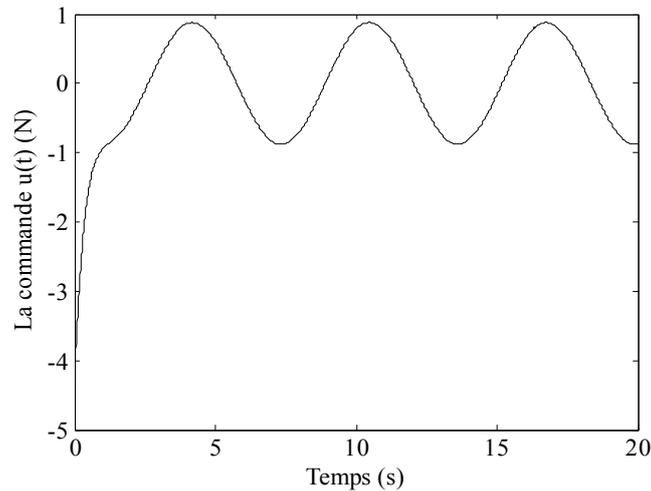


Figure III.17. La commande  $u(t)$ .

On constate à partir de les figures (III.13) et (III.14) que notre contrôleur assure de très bonnes performances de poursuite.

Les figures (III.15) et (III.16) montrent que les erreurs de poursuite sont très faible et convergent très rapide vers zéro.

On remarque dans la figure (III.17) que la commande est minimale, lisse et facile à mettre en œuvre.

### III.4.2.2 Exemple 2: robot manipulateur

Nous considérons la même synthèse du régulateur comme dans (§ III.3.1.2), avec des matrices de pondération du système global.

#### Résultats de simulation

- ✓ les matrices de pondération

$$Q = \begin{bmatrix} 1.12 & 0 & 0 & 0 \\ 0 & 0.86 & 0 & 0 \\ 0 & 0 & 0.72 & 0 \\ 0 & 0 & 0 & 0.49 \end{bmatrix}, R = \begin{bmatrix} 0.97 & 0 \\ 0 & 0.18 \end{bmatrix}.$$

- ✓ les gains de commande:

$$K_1 = K_2 = \begin{bmatrix} 5.3456 & 2.3702 & 11.5578 & 4.4624 \\ 0.1137 & 4.0833 & 2.3152 & 3.6090 \end{bmatrix},$$

$$K_3 = K_4 = \begin{bmatrix} 7.6071 & 2.1382 & 14.8190 & 5.0485 \\ 0.1958 & 6.6101 & 4.6924 & 4.9617 \end{bmatrix}.$$

✓ la matrice  $P$  symétrique définie positif:

$$P = \begin{bmatrix} 6.3714 & 1.4001 & 8.0859 & 2.7401 \\ 1.4001 & 2.2274 & 2.6443 & 1.3109 \\ 8.0859 & 2.6443 & 16.4962 & 5.8526 \\ 2.7401 & 1.3109 & 5.8526 & 2.3505 \end{bmatrix}.$$

✓ les courbes:

L'évolutions, en fonction du temps, des angles d'articulations  $q_1(t)$  et  $q_2(t)$  sont représentées sur les figures (III.18) et (III.19) respectivement, où l'on constate la convergence rapide du système vers la position de référence.

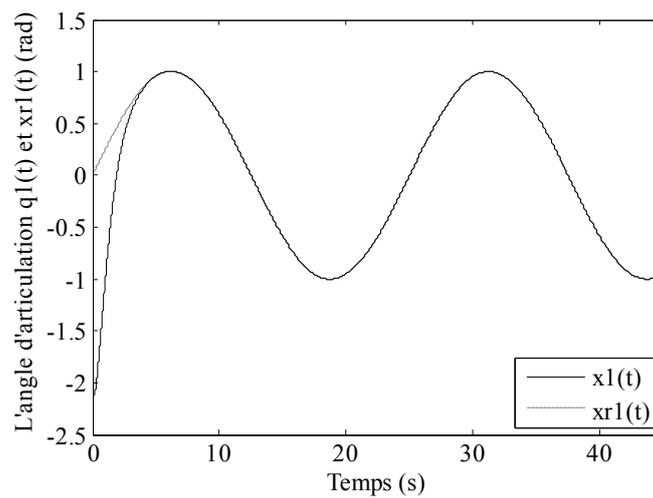


Figure III.18. Evolution de l'angle d'articulation  $x_1(t)$  et le signal de référence  $x_{r1}(t)$ .

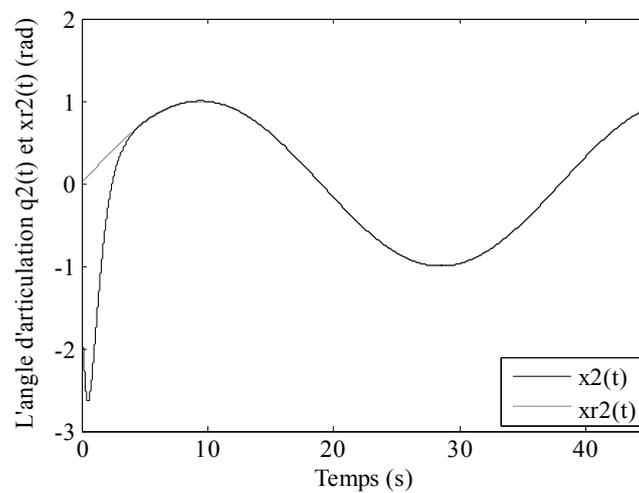


Figure III.19. Evolution de l'angle d'articulation  $x_2(t)$  et le signal de référence  $x_{r2}(t)$ .

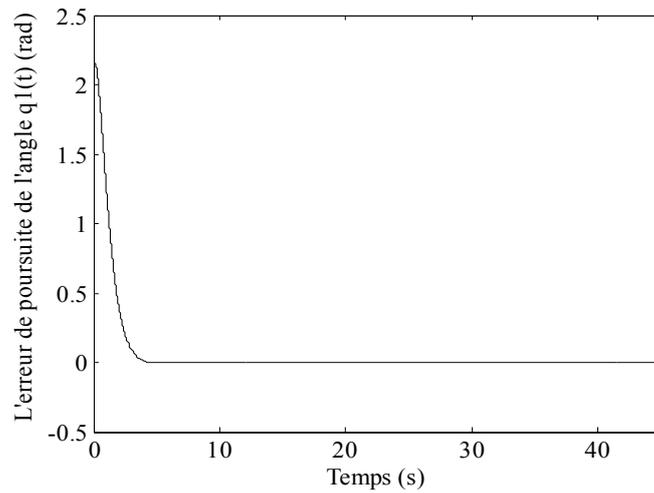


Figure III.20. L'erreur de poursuite  $e_1(t)$  de l'angle d'articulation  $x_1(t)$

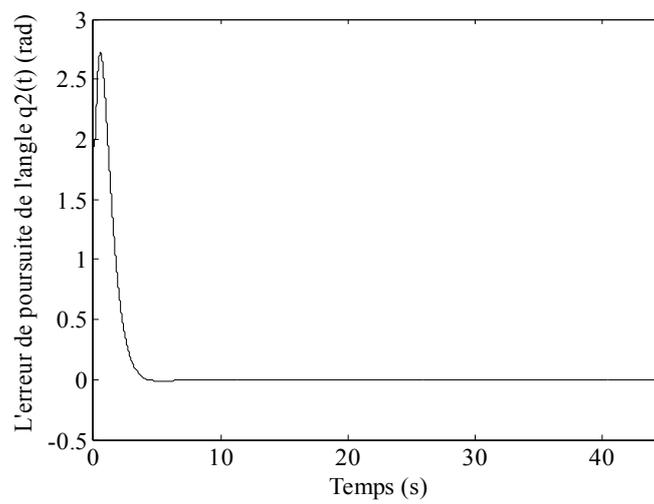


Figure III.21. L'erreur de poursuite  $e_2(t)$  de l'angle d'articulation  $x_2(t)$ .

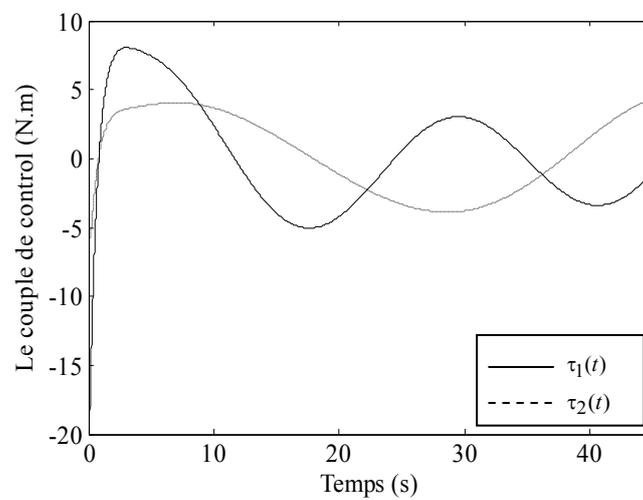


Figure III.22. Le couple de contrôle  $\tau_1(t)$  et  $\tau_2(t)$ .

Les figures (III.20) et (III.21) montrent que notre régulateur est capable d'assurer les poursuites en un temps très court avec des erreurs de poursuite très faibles.

La représentation des variations des signaux de contrôle sur la figure (III.22) montre qu'ils sont continus et faciles à implémenter.

### III.5 Etude comparative

Il s'agit de faire une étude comparative entre les résultats de simulation obtenus pour les deux applications dans les trois chapitres, c'est à dire dans le cas de la commande LQ, de la commande adaptative floue et de la commande PDC optimale. Par la suite nous allons comparer les meilleurs résultats obtenus avec ceux d'autres références.

#### III.5.1 Exemple 1: pendule inversé

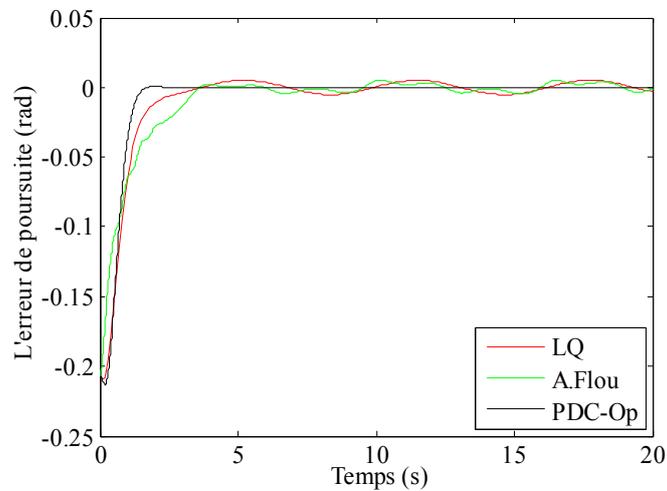


Figure III.23. L'erreur de poursuite  $e_1(t)$  de l'angle  $x_1(t)$

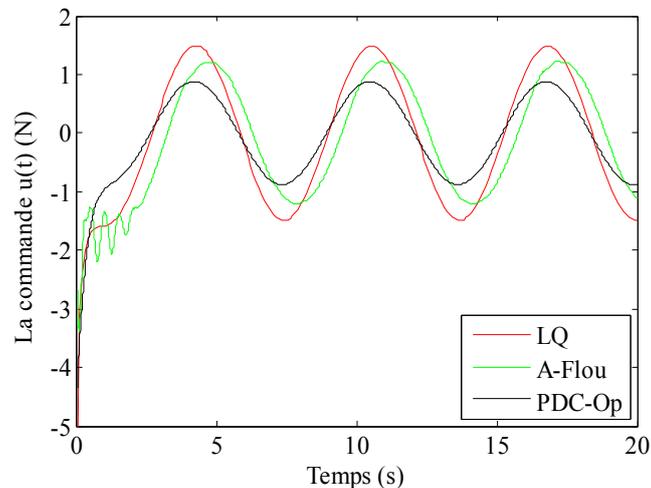


Figure III.24. La commande  $u(t)$

Tableau III.1. Critère IAE et commande maximale pour les trois types de commande

	Type de commande		
	Linéaire quadratique (LQ)	Adaptative Floue	PDC Optimale
IAE (rad)	0.2496	0.2097	0.1534
$ u_{\max} $ (N)	1.4859	1.2268	0.8770

**III.5.2 Exemple 2: robot manipulateur**

*La sortie  $q_1(t)$*

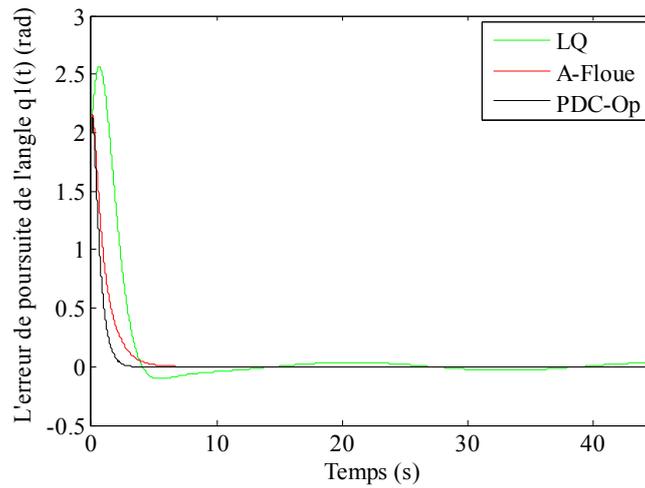


Figure III.25. L'erreur de poursuite  $e_1(t)$  de l'angle d'articulation  $x_1(t)$

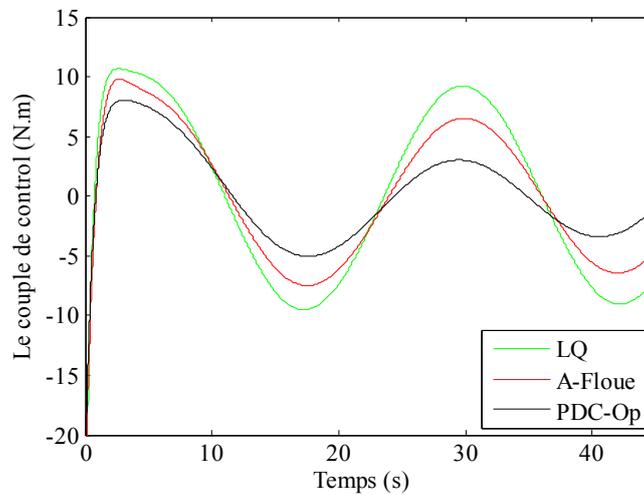


Figure III.26. La commande  $\tau_1(t)$

Tableau III.2. Critère IAE et commande maximale pour le trois types de commande

	Type de commande		
	Linéaire quadratique (LQ)	Adaptative Floue	PDC Optimale
IAE (rad)	2.3700	2.0224	1.6656
$\tau_1(t)$ (N.m)	10.6700	9.8122	7.8342

*La sortie  $q_2(t)$*

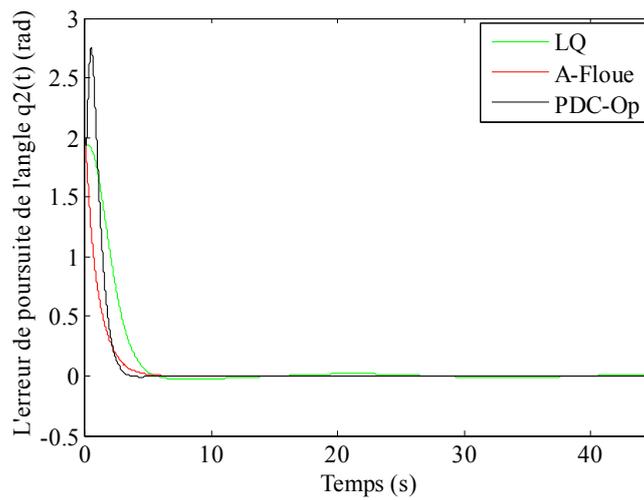


Figure III.27. L'erreur de poursuite  $e_2(t)$  de l'angle d'articulation  $x_2(t)$

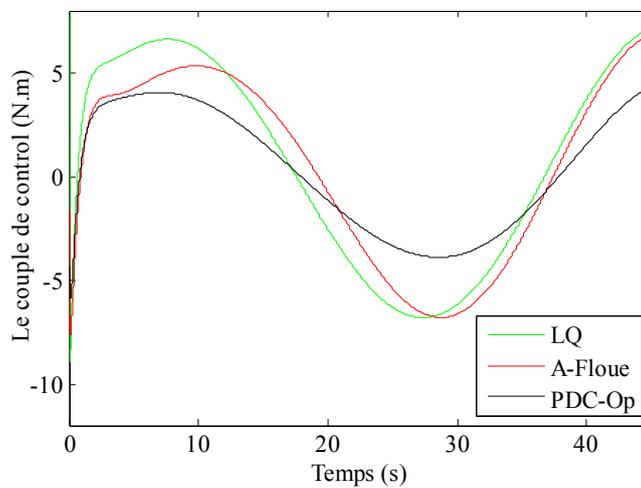


Figure III.28. La commande  $\tau_2(t)$ .

Tableau III.3. Critère IAE et commande maximale pour les trois types de commande

	Type de commande		
	Linéaire Quadratique (LQ)	Adaptative Floue	PDC-Optimale
IAE (rad)	2.758	2.2310	2.0538
$\tau_2(t)$ (N.m)	7.2402	7.1231	4.4012

D'après cette étude comparative on constate que la commande PDC optimale est la plus performante en termes d'erreur de poursuite et d'énergie de commande.

Pour situer les performances de la structure de commande PDC optimale développée, nous allons comparer les résultats obtenus avec ceux des deux références [49] et [58].

◆ *Pendule inversé*

- *Signal de référence*  $y_m(t)=0.1\sin(0.5t)+0.1\cos(t)$ .
- *Conditions initiales*  $x(0)=[-0.1,0]^T$ .

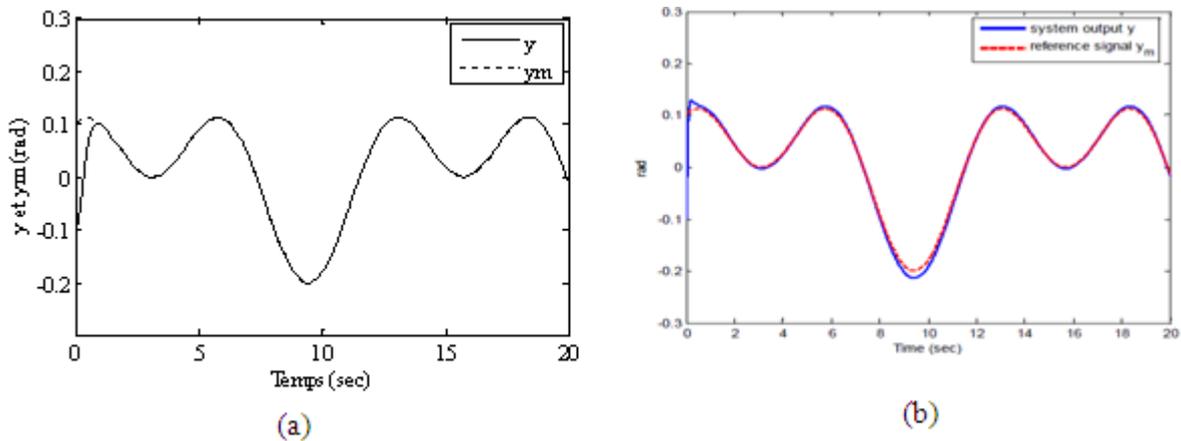


Figure III.29. Evolution de l'état  $x_1=y$ : (a) résultats obtenus, (b) résultats de [58]

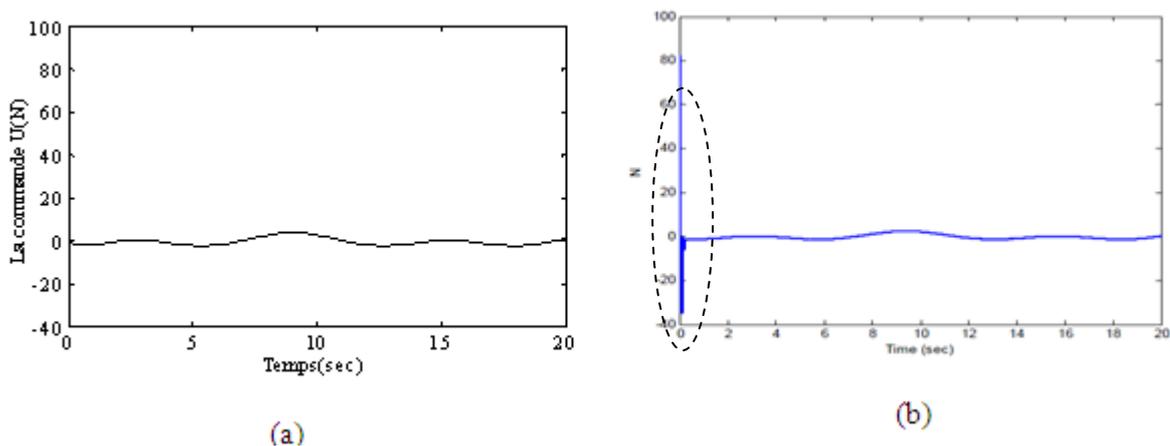


Figure III.30. Evolution de la commande: (a) résultats obtenus, (b) résultats de [58]

◆ Robot manipulateur

- Signaux de références  $y_{m1}=\sin(t)$  et  $y_{m2}=\cos(t)$ .
- Conditions initiales  $x(0)=[0.25,0.5,0,0]^T$ .

La sortie  $q_1$ :

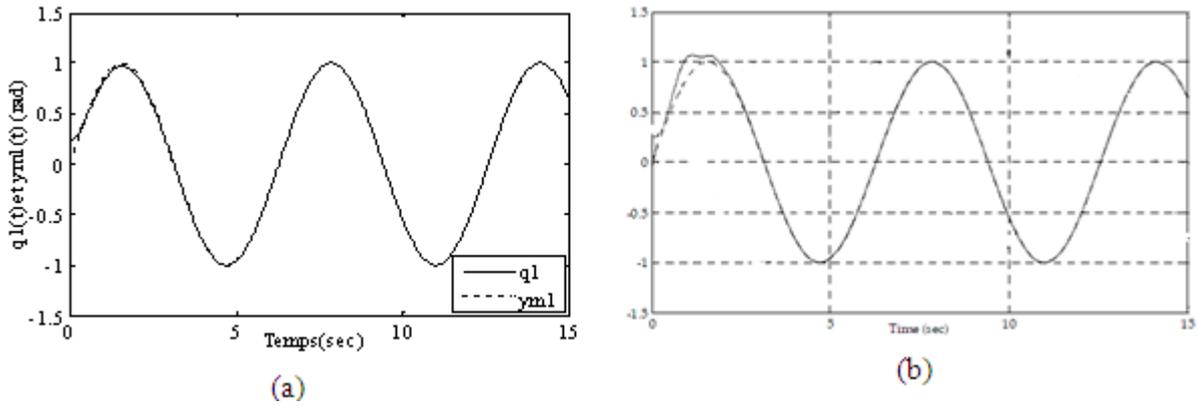


Figure III.31. Evolution de l'état  $x_1=q_1$ : (a) résultats obtenus, (b) résultats de [49]

La sortie  $q_2$ :

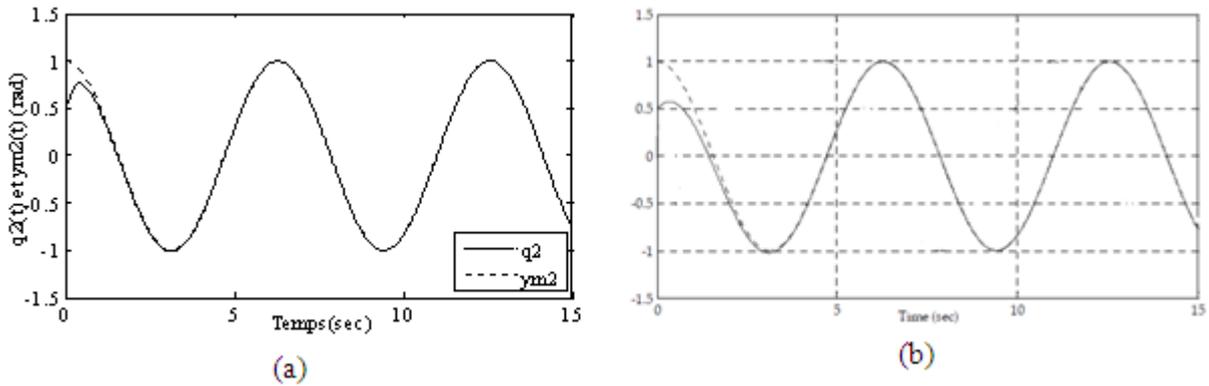


Figure III.32. Evolution de l'état  $x_2=q_2$ : (a) résultats obtenus, (b) résultats de [49]

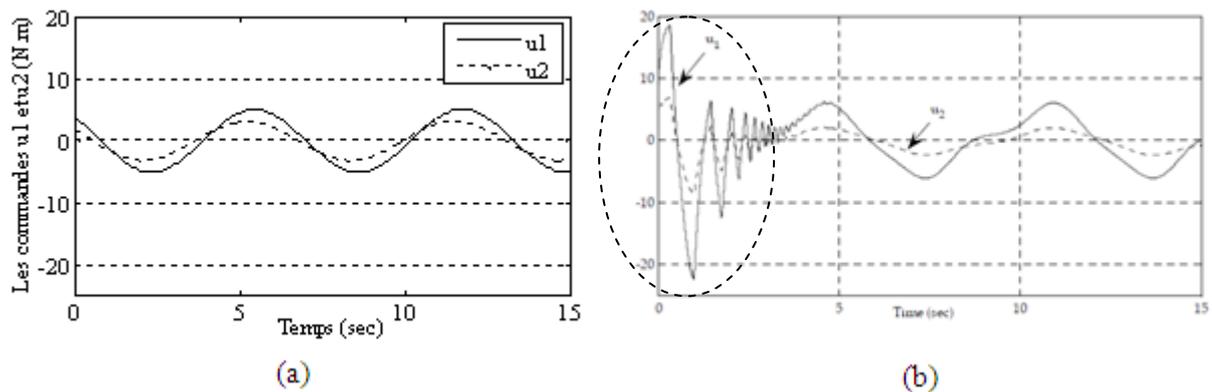


Figure III.33. Evolution du couple de contrôle: (a) résultats obtenus, (b) résultats de [49]

En comparant les résultats obtenus par la structure de commande PDC optimale développée, avec ceux des deux références [49] et [58] on constate une convergence plus rapide et un régime transitoire des signaux de commande moins brutal.

### III.6 Conclusion

Le problème de suivi de trajectoire dans ce chapitre, a été traité avec une approche dite PDC. Elle permet de faire une synthèse de commande minimisant une fonction de coût de type quadratique. La loi de commande optimale floue basée sur l'approche PDC a été synthétisée par deux techniques. La première technique dite sous-optimale consiste à calculer une loi de commande optimale pour chaque sous-système, et la loi de commande globale obtenue n'est pas nécessairement optimale. La deuxième technique dite optimale consiste à calculer des lois de commande locales optimales et une loi de commande globale optimale en minimisant la borne supérieure du critère de performance quadratique. Les conditions de stabilité sont transformées et formulées en termes d' LMIs pour une résolution numérique exploitant les outils de l'optimisation convexe.

Les exemples illustratifs de simulation, sous l'environnement Matlab, montrent l'efficacité des deux techniques de conception de contrôleurs PDC proposées pour prendre en charge une classe de systèmes non linéaires monovariables et multivariables.

L'étude comparative faite dans ce chapitre a mis en évidence les performances de la commande PDC optimale proposée en termes d'erreur de poursuite et d'énergie de commande.

La comparaison des résultats obtenus avec ceux des deux références [49] et [58], a permis de prouver et confirmer l'efficacité de la structure de commande PDC optimale développée à travers la rapidité de convergence et l'amélioration du régime transitoire des signaux de commande.

---

*CONCLUSION GENERALE*

---

### CONCLUSION GENERALE

Dans ce travail, nous avons abordé le problème de poursuite de trajectoires des systèmes non linéaires mono-variables et multi-variables. Pour traiter ce problème nous avons élaboré trois approches de commande, la première approche est la commande optimale linéaire quadratique, où on a introduit l'optimisation par essaim particulaire pour choisir les matrices de pondération. La deuxième approche est la commande adaptative floue avec ses deux variantes direct et indirect. La troisième approche concerne la commande PDC.

La première partie a été allouée à la commande linéaire quadratique et l'optimisation par essaim particulaire où nous avons mené une étude détaillée de la commande optimale linéaire quadratique et nous avons utilisé l'algorithme de l'OEP pour réaliser un bon choix de matrices de pondération durant la synthèse du régulateur LQR.

Dans la deuxième partie nous avons présenté l'approche adaptative floue, où les définitions de base de la commande adaptative et la commande floue sont données. Ensuite nous avons développé deux régulateurs adaptatifs flous, l'un indirect pour commander un système non linéaire mono-entrée mono-sortie et l'autre direct pour prendre en charge un système non linéaire multi-entrée multi-sortie. La stabilité des systèmes en boucle fermée est étudiée au sens de Lyapunov.

La troisième partie a été consacrée à la combinaison de la commande floue et la commande optimale afin de synthétiser une commande optimale floue pour des systèmes non linéaires décrits par des modèles flous de type T-S. À cet égard nous avons proposé deux techniques de commande basées sur l'approche PDC minimisant un critère de performance quadratique. La première technique connue sous le nom PDC sous-optimale, car la loi de commande n'est optimale que pour chacun des sous modèles linéaires. La deuxième technique est basée sur la minimisation de la borne supérieure du critère de performance. La conception du contrôleur flou optimal exploite l'approche des inégalités matricielles linéaires et les outils de l'optimisation convexe.

## CONCLUSION GENERALE

---

Les résultats de simulation obtenus pour les deux exemples (SISO et MIMO) montrent l'efficacité de la structure de commande PDC optimale proposée en termes de performance de poursuite et de caractéristiques du signal de commande (forme, amplitude).

Pour situer les résultats obtenus par rapport à ceux des références du domaine, la comparaison réalisée avec [49] et [58] a montré que la commande PDC optimale développée est capable d'assurer de bonnes performances de poursuite avec un régime transitoire amélioré.

Comme perspectives à ce travail, il serait intéressant d'utiliser la commande linéaire quadratique Gaussienne (LQG) et une autre technique d'optimisation pour le choix des matrices de pondération connue sous le nom « optimisation par colonies d'abeilles », en anglais « ABC: Artificial Bee Colony optimization ».

---

*BIBLIOGRAPHIE*

---

### BIBLIOGRAPHIE

- [1] Thomas Kailath, "linear systems," Englewood Cliffs, NJ, Prentice-Hall, 1980.
- [2] Anderson, Moore, "Optimal Control: Linear Quadratic Methods" Englewood Cliffs, NJ, Prentice-Hall, 1989.
- [3] Ragnar Eide, Per Magne Egelid, Alexander Stams, Hamid Reza Karimi, " LQR Control Design for Balancing an Inverted Pendulum Mobile Robot," Intelligent Control and Automation, vol. 2, pp. 160-166, May 2011.
- [4] Laroche E, Notes de cours de master Automatique et robotique: Commande Optimale, Ecole Nationale supérieure de physique, Strasbourg, 2010.
- [5] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," In Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948, Perth, Australia, 1995.
- [6] J. Kennedy, R. Eberhart, "A New Optimizer Using Particles Swarm Theory," Proc. Sixth International Symposium on Micro Machine and Human Science, IEEE Service Center, Pkcataway, NJ, pp. 39-43, Nagoya, Japan, 1995.
- [7] H. Khalil, "Nonlinear systems," Prentice hall, Inc. second edition, USA, 1996.
- [8] S. Sastry, "Nonlinear systems: analysis, stability, and control," Springer-Verlag, New York, 1999.
- [9] Wang. L. X, "Adaptive fuzzy systems and control: design and stability analysis," Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [10] Y. Oulmas, "Commande Adaptative Floue de la Position et de l'Orientation des Robots Manipulateurs'', Ecole Nationale Polytechnique Laboratoire de Commande des Processus, 2006.
- [11] L. Foulloy, S. Galichet, A. Titli, "Commande floue 1 de la stabilisation à la supervision", Lavoisier, Paris, 2003.
- [12] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its application to modeling and control". IEEE Trans. Syst., Man and Cyber, Vol. 1115, pp. 116-132, 1985.

## BIBLIOGRAPHIE

---

- [13] L. X. Wang, “stable adaptive fuzzy control of nonlinear system,” IEEE Trans. Fuzzy Syst., vol.1, pp. 146-155, 1993.
- [14] L. X. Wang, “stable adaptive fuzzy controllers with to application to inverted pendulum tracking”. IEEE Trans. on Syst., Man and cybernetic-part B, vol. 26, N°. 5, pp.677-691, October 1996.
- [15] Y. S. Yang, C.J. Zhou, “Robust adaptive fuzzy tracking control for a class of perturbed strict-feedback nonlinear systems via small-gain approach”, Information science, Vol.170, pp.211-234, 2005.
- [16] N. Golea, A. Golea, K. Benmahammed, “Stable indirect fuzzy adaptive control ”, Fuzzy Sets and Systems, Vol 137, pp. 353–366, 2003.
- [17] J. Yoneyama, “Robust stability and stabilization for uncertain Takagi–Sugeno fuzzy time-delay systems”, Fuzzy Sets and Systems, Vol 158, pp. 115–134, 2007.
- [18] H.K. Lam, F.H. F. Leung, P.K.S. Tam, “Stable and Robust Fuzzy Control for Uncertain Nonlinear Systems”, IEEE transactions on fuzzy systems, vol. 30, N°. 6, November 2000.
- [19] J. R. Panah, V.J. Majd, “A robust multi-objective DPDC for uncertain T–S fuzzy systems”, Fuzzy Sets and Systems, Vol 159, pp. 2749 – 2762, 2008.
- [20] K. Tanaka, “Robust Stabilization of a Class of uncertain nonlinear systems via fuzzy control: quadratic stabilizability, control theory, and linear matrix inequalities”, IEEE transactions on fuzzy systems, Vol. 4, N°. 1, February 1996.
- [21] T. Taniguchi, K. Tanaka, K. Yamafuji, H. O. Wan, “A new PDC for fuzzy reference models,” IEEE International Fuzzy Systems Conference Proceedings, Seoul, Korea, August 22-25, 1999.
- [22] R.A. Freeman, P.V. Kokotovic, “Robust non linear control design state-space and Lyapunov technique”, Birkhauser Boston, London, 1996.
- [23] W. H. Ho and J.H. Chou, “Design of optimal controller for Takagi–Sugeno fuzzy model based systems,” IEEE Trans. Syst., Man and Cybern. A, Syst. Humans, Vol. 37, N°. 3, pp. 329–339, May 2007.

## BIBLIOGRAPHIE

---

- [24] C.W.Tao, J.S.Taur, Y.C.Chen, "Design of a parallel distributed fuzzy LQR controller for the twin rotor multi-input multi- output system," *Fuzzy Sets and Systems*, Vol. 161, pp. 2081–2103, 2010.
- [25] Russell C.Eberhart, Yuhui Shi, "Particle Swarm Optimization: Developments, Applications and Resources," *Proc. Congress on Evolutionary Computation*, Seoul, Korea. Piscataway, NJ: IEEE Service Center. 2001.
- [26] R. Mendes, J. Kennedy, J. Neves, "The Fully Informed Particle Swarm: Simpler, Maybe Better," *IEEE Transactions on evolutionary computation*, Vol. 8, N°. 3, pp. 204-210, June 2004.
- [27] M. Clerc, J. Kennedy, "The Particle Swarm Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transaction on Evolutionary Computation*, Vol. 6, N°. 1, pp. 58-73, February 2002.
- [28] Tansel Yucelen, "A study on PDC of nonlinear systems by T–S Fuzzy Model with comparing pole placement method and linear quadratic regulator," 9<sup>th</sup> International Conference on Automatic Control, Modeling & Simulation, May 27-29, 2007.
- [29] Hee-sang Ko, Juri Jatskevich, Guy Dumont, Gi-Gap Yoon, "An advanced LMI-based-LQR design for voltage control of grid-connected wind farm," *Electric Power Systems Research*, vol. 78, pp. 539-549, 2008.
- [30] B. Mansori, "Contribution a la synthèse de lois de commandes en poursuite de trajectoire pour les systèmes flous e type Takagi-Sugeno incertain," *Thèse de doctorat*, université de Reims Champagne Ardenne, France, 9 décembre 2005.
- [31] C. M. Lin, Y. J. Mon, "A fuzzy-PDC-based control for robotic systems," *Information science*, vol. 137, pp. 135-155, 2001.
- [32] Chen-shen Ting, "Stable adaptive control for multivariable nonlinear system via Takagi-Sugeno fuzzy model," *National University Formosa*, vol. 2, pp. 55-66, 31 December 2007.
- [33] A. L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning - I," *Information Science*, vol. 8, pp. 199- 249, 1975.
- [34] J. J. E. Slotine, W. Li, "Applied nonlinear control," Englewood Cliffs, NJ, Prentice-Hall, 1991.

## BIBLIOGRAPHIE

---

- [35] K. J. Aström, B. Wittenmark, "Adaptive Control," Addison-wesely, 1989.
- [36] T. Procyk, E. Mamdani, "A linguistic self-organizing process controller," *Automatica*, Vol. 15, pp. 15-30, 1979.
- [37] D. L. Tsang, H. Y. Chang, C. J. lee, "The adaptive control of nonlinear systems using the Sugeno-type of fuzzy logic," *IEEE Trans. Fuzzy Syst.*, Vol. 7, N°. 2, pp. 225-229, April 1999.
- [38] N. Goléa, "Identification et commande adaptative floue," Thèse de doctorat d'état en électronique, université de Batna, 2001.
- [39] M. Hojati, S. Gazor, "Hybride adaptive fuzzy identification and control of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, Vol. 10, N°. 2, pp. 198-210, April 2002.
- [40] R. Boukezzoula, "Commande floue d'une classe de systèmes non linéaires: application au problème de suivi de trajectoire," Thèse de doctorat, l'université de Savoie, 31 mars 2000.
- [41] J. Jantzen, "Foundations of Fuzzy Control," John Wiley & Sons Ltd, 2007.
- [42] A.M. Ibrahim, "Fuzzy logic for embedded systems applications," Elsevier Science, 2004.
- [43] R. Babuška, "Fuzzy Modeling for Control," Kluwer Academic Publishers, Mass., USA, 1998.
- [44] A. L. Zadeh, "Outline Of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. on Systems Man and Cybernetics*, SMC 3, pp. 28-44, 1973.
- [45] E. Mamdani, "Application of Fuzzy Logic to Approximate Reasoning using Linguistic Synthetis," *IEEE Transactions on computers*, vol. C26, N°. 8, pp. 1182-1191, 1977.
- [46] R. Babuška, B. H. Verbruggen, "A new identification method for linguistic fuzzy model," In *Proc. of Fuzzy IEEE'95*, Yokohama, Japan, pp. 905-912, 1995.
- [47] L. X. Wang, "Adaptive fuzzy systems and control: design and stability analysis," Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [48] F. Khaber, A. Hamzoui, " Commande adaptative floue pour les systèmes non linéaires incertains," *Courrier du Savoir*, N°. 02, pp. 35-40, Juin 2002.
- [49] S. Labiod, T. M. Guerra, " Direct and Indirect Adaptive Fuzzy Control for a Class of MIMO Nonlinear Systems" *Advances in Robot Manipulators*, InTech, 01 April 2010.

## BIBLIOGRAPHIE

---

- [50] P. A. Ioannou, J. Sun, "Robust adaptive control," Prentice Hall, NJ. 1996.
- [51] F. Khaber, A. Hamzoui, "Commande et stabilité des systèmes linéaires flous de Takagi-Sugeno," *Courrier du Savoir*, N°. 02, pp. 23-28, Juin 2002.
- [52] P. Gahinet, A. Nemirovski, A.J. Laub, M. Chilali, *LMI Control Toolbox*, The Math Works Inc., 1995.
- [53] A. M. Farahmand, M. J. Yazdanpanah, "Locally Optimal Takagi-Sugeno Fuzzy Controllers," Control and Intelligent Processing Center of Excellence, 28 February 2005.
- [54] R. Kahani, B. Moaveni, "Control of Two-Wheels Inverted Pendulum Using Parallel Distributed Compensation and Fuzzy Linear Quadratic Regulator," 3rd International Conference on Computer Modeling and Simulation, Vol. 2, pp. 312-317, 2011.
- [55] A. Khaled, K. Farid, "Optimal Fuzzy Controller Design via LMI Approach" International Conference on Electrical Engineering and Automatic Control, Setif, 24-26 November 2013.
- [56] A. Jadbabaie, M. Jamshidi, J. Jadbabaie, *Guaranteed-cost design of Takagi-Sugeno controllers via LMIs: Continuous-Time case*, NASA Center for Autonomous Control Engineering, New Mexico, 1997.
- [57] S. Boyd, L. E. Ghaoui, E. Feron, V. Balakrishnan, "Linear Matrix Inequalities in System and Control Theory," Society for Industrial and Applied Mathematics (SIAM), 1994.
- [58] Yi-Hsing Chien, Wei-Yen Wang, I-Hsum Li, Kuang-Yow Lian, and Tsu-Tian Lee, "Hybrid Intelligent Output-Feedback Control for Trajectory Tracking of Uncertain Nonlinear Multivariable Dynamical Systems" *International Journal of Fuzzy Systems*, Vol. 14, No. 1, March 2012.

---

## Résumé

---

Dans le cadre de ce mémoire nous avons proposé trois approches de commande en poursuite pour des systèmes non linéaires SISO et MIMO. La première approche est une commande optimale linéaire quadratique où l'optimisation des matrices de pondération a été faite par essaim particulière. La seconde approche est une commande adaptative floue avec ses deux variantes directe et indirecte. La troisième approche est une commande optimale floue à base de modèles flous de Takagi-Sugeno. Cette commande a été synthétisée en utilisant l'approche PDC selon deux techniques. La première dite PDC sous optimale consiste à minimiser un critère de performance quadratique. La seconde dite PDC optimale repose sur la minimisation de la borne supérieure du critère de performance quadratique en exploitant les la technique des inégalités matricielles linéaires (LMIs) et les outils de l'optimisation convexe. La stabilité des systèmes en boucle fermée a été étudiée au sens de Lyapunov. Les résultats de simulation obtenus, sous l'environnement Matlab, ont montré l'efficacité des structures de commande développées pour atteindre les performances souhaitées.

**Mots clés:** Commande linéaire quadratique, Optimisation par essaim particulière, Commande adaptative floue, modèles flous de T-S, LMI, Synthèse de Lyapunov, Systèmes non linéaires SISO et MIMO.

---

## Abstract

---

In this memory we have proposed three approaches to design a tracking control for SISO and MIMO nonlinear systems. The first approach is a linear quadratic control where the weighting matrices have been optimized by the particle swarm method. The second approach is an adaptive fuzzy control with its two variants direct and indirect. The third approach is an optimal fuzzy control based on the dynamic Takagi-Sugeno fuzzy models, where the controller was synthesized using the PDC approach by two techniques. The first known as the suboptimal PDC minimizes a linear quadratic performance measure. The second called optimal PDC is based on the minimization of an upper bound on the linear quadratic performance measure using the linear matrix inequality (LMI). The stability of closed-loop systems was studied using Lyapunov synthesis. The obtained simulation results, under the Matlab environment, showed the effectiveness and the performances of the proposed control techniques.

**Key words:** Linear quadratic control, Particle swarm optimization, adaptive fuzzy control, T-S fuzzy models, LMI, Lyapunov synthesis, SISO and MIMO nonlinear systems.

---

## ملخص

---

في هذه المذكرة قمنا بعرض ثلاث تقنيات للتحكم في الأنظمة اللاخطية ذات مدخل واحد ومخرج واحد و المتعددة المداخل والمخارج. التقنية الأولى هي عبارة عن تحكم خطي مثالي , أين تم اختيار مصفوفات الترجيح باستخدام طريقة الأسراب الجزئية. التقنية الثانية هي تحكم تأقلمي غامض بمتغيريه المباشر و الغير مباشر . التقنية الثالثة هي تحكم مثالي غامض يعتمد على النماذج الديناميكية الغامضة لتكاجي-سيجينو, حيث تم تشكيلها و صياغتها باستخدام نظام التوزيع التكميلي المتوازي بطريقتين. الأولى تركز على تحسين معيار الجودة في حين الثانية تعتمد على تحسين الحد الأعلى لمعيار الجودة باستعمال المصفوفات المتراحة الخطية. دراسة استقرار الأنظمة تمت باستعمال طريقة ليابونوف. نتائج المحاكاة المتحصل عليها باستخدام برنامج ماتلاب أثبتت جودة و فعالية تقنيات التحكم المقترحة.

**الكلمات المفتاحية:** التحكم المثالي, طريقة الأسراب الجزئية, تحكم تأقلمي غامض, النماذج الغامضة لتكاجي-سيجينو, المصفوفات المتراحة الخطية, الإستقرار حسب ليا بونوف, الأنظمة اللاخطية ذات مدخل واحد ومخرج واحد و المتعددة المداخل والمخارج .