République algérienne démocratique et populaire

Ministère de l'enseignement supérieur et de la recherche

Université de Yahia fares de Medea

Faculté des technologies

Département de génie mécanique

Projet de fin de cycle

En vue d'obtention du

Diplôme de Master

Filière : Génie mécanique Spécialité : Mécatronique

THEME

Synthèse d'une loi de commande adaptative hybride pour la Stabilisation des systèmes incertains non linéaires

Encadré par :

Prof. ABUDURA Salam

Dr. KHETTAB Khatir

Présenté par :

Mr. BRAHIMI Bilal

Année universitaire : 2016/2017

REMERCIEMENT

Tout d'abord je remercie (DIEU) le tout puissant pour la volonté et la patience qu'il m'a donné tout au long de mes études.

Je tiens à remercier vivement monsieur **Khatir KHETTAB** , MCB à l'université de M'sila pour son esprit scientifique et compréhensif qui a consacré beaucoup de son temps à mon travail et surtout ses critiques objectives.

Je tiens également à remercier monsieur le Professeure . **Salem ABUDORA** pour son enthousiasme et son soutien sans faille durant la réalisation de ce mémoire.

Je tiens également à remercier les plus vifs s'adressent aussi aux messieurs le président et les membres de jury d'avoir accepté d'examiner et d'évaluer mon travail.

DÉDICACES

Je dédie ce modeste travail a

Mes parents

Ma sœur et mes frères

Son oublier mes amies et ceux qui mon aidé

Et toutes personnes je le connais

BILAL ...

Résumé -

On propose un nouvel algorithme de commande hybride (directe/indirecte) intelligente (adaptative, floue …) de telle sorte que non seulement la stabilité du système de commande floue adaptative soit garantie mais également l'influence de la perturbation externe et d'erreur d'approximation sur l'erreur de poursuite puisse être atténuée à un niveau prescrit par l'intermédiaire de la technique de la conception d'une commande robuste.

La méthode de conception proposée essaye de combiner la technique d'atténuation, la méthode d'approximation floue, et l'algorithme de commande adaptative pour la conception de commande de poursuite robuste dune classe de systèmes non linéaires avec une incertitude ou une variation inconnue des paramètres et des structures du système (incertains).

L'objectif de cet algorithme de commande intelligente est d'améliorer les performances des systèmes de commande de processus sur lesquelles peu d'informations sont disponibles.

Les mots clés : la fuzzification , défuzzification , un mécanisme d'inférence , commande floue adaptative (directe/indirecte) , commande hybride (directe/indirecte) intelligente

Abstract -

A new intelligent (direct / indirect) hybrid control algorithm (adaptive, fuzzy, etc.) is proposed so that not only the stability of the adaptive fuzzy control system is guaranteed, but also the influence of the external disturbance and error d the approximation on the trackingerrorcanbeattenuated to a prescribed level by means of the robust control design technique.

The proposed design methodattempts to combine the attenuation technique, the fuzzy approximation method, and the adaptive control algorithm for the robusttracking control design of a class of nonlinearsystemswithuncertainty or unknown variation of parameters and system structures (uncertain).

The objective of this intelligent control algorithmis to improve the performance of process control systems on whichlittle information isavailable

Key words - Fuzzification, defuzzification, inferencemechanism, adaptive fuzzy control (direct / indirect), intelligent hybrid (direct / indirect) control.

ملخص -

ويقترح خوارزمية تحكم هجين ذكي (مباشر / غير مباشر) (تكيفي، غامض، وما إلى ذلك) بحيث لا يضمن استقرار نظام التحكم الضبابي التكيفي فحسب، بل أيضا تأثير الاضطراب والخطأ الخارجيين d يمكن تقريب التقريب على خطأ التتبع إلى مستوى محدد من خلال تقنية تصميم التحكم القوية.

وتحاول طريقة التصميم المقترحة الجمع بين تقنية التوهين وطريقة التقريب الضبابي وخوارزمية التحكم التكيفية لتصميم التحكم في التتبع المتين لفئة من الأنظمة غير الخطية مع عدم التيقن أو الاختلاف المجهول المعلمات وهياكل النظام (غير مؤكد).

والهدف من هذه الخوارزمية التحكم الذكي هو تحسين أداء أنظمة التحكم في العملية التي تتوفر القليل من المعلومات.

الكلمات المفتاحية : فزيفيكاتيون، ديفوزيفيكاتيون، آلية الاستدلال، التحكم غامض التكيف (مباشر / غير مباشر)، التحكم الهجين ذكي (مباشر / غير مباشر)

Liste des figures

Figure 1.1 Evaluation de la température de l'eau en logique classique
Figure 1.2 Evaluation de la température de l'eau en logique floue.
Figure 1.3complimentaire
Figure 1.4 Intersection
Figure.1.5 Union
Figure.1.6"La théorie des ensembles classiques n'est qu'un sous-ensemble de la théorie des ensembles flous"
Figure 1.7 Fonction d'appartenance caractérisant le sous-ensemble 'bon' de la qualité du service.
Figure.1.8comparaison entre les deux fonctions
Figure 1.9 Propriétés d'un ensemble flou
Figure 1.10 Variable linguistique 'qualité du service'
Figure 1.11 Variable linguistique 'qualité de la nourriture'
Figure 1.12 Application de la règle 'Si (la qualité de la nourriture est délicieuse)
Figure 1.13 Exemple d'application de l'ensemble des 3 règles de notre matrice des décisions à l'entrée (service = 7,83 ; nourriture = 7,32)
Figure 1.14 Défuzzification avec la méthode moyenne des maxima (MM)
Figure 1.15 Défuzzification avec la méthode centre de gravité(COG)
Figure 1.16 Aperçu synoptique d'un système flou
Figure 2.1 changement de la représentation du système
Figure 2.2 représentation entrée-sortie du système 2.7
Figure 2.3algorithme linguistique
Figure 2.4 système floue
Figure 2.5 Boucle de régulation
Figure 2.6 les étapes de fuzzification
Figure 2.7.1 commande adaptative avec modèle de référence explicite (directe)
Figure 2.7.2commande adaptative auto-ajustable (indirect)
Figure 2.8 schéma bloc du système de commande adaptative à gains pré-programmes
Figure 2.9 schéma de commande adaptative à modèle de référence
Figure 2.10 modèle d'erreur
Figure 2.11 plan d'état

Figure 2.12schéma bloc du système de commande adaptative avec régulateur auto ajustable
Figure 3.1 Schéma général de commande adaptative floue.
Figure 3.2 Système du pendule inversé.
Figure 3.3 Résultats de simulation sans perturbation.
Figure 3.4 Résultats de simulation avec perturbation

Table des matières

Introduction générale 1

Chapitre 1 Eléments de théorie de la logique floue

1.1 Introduction 3

1.2. Historique et définition de la logique floue 4

1.3. Concepts fondamentaux 5

1.3.1. Définition de sous-ensemble flou 5

1.3.2. Opérations de base sur les sous-ensembles flous. 6

1.3.2.1. Complémentaire de A par la fonction d'appartenance 6

1.3.2.2 Sous-ensemble A ou B, AÈB, par la fonction d'appartenance 8

1.4. Règles floues 8

1.4.1. Règle de Mamdani 8

1.4.2.Règle de Takagi-Sugeno 9

1.4.3. Etapes de conception d'un système flou 9

1.5 Théorie des possibilités 10

1.6 Sous-ensembles flous 10

1.6.1 DEFINITION 11

1.7 Variables linguistiques 14

1.8. Raisonnement en logique floue 15

1.9. Défuzzification 17

1.10 Conclusion 20

Chapitre 2 Théorie de la commande Adaptative Hybride directe/indirecte

2.1 Introduction 21

2.2 Représentation entrée-sortie 22

2.3 Commande par rétroaction d'un système non linéaire 24

2.4 La commande adaptative floue 25

2.5. Commande Floue 26

2.5.1. Introduction 26
2.5.2. Structure d'une Commande Floue 26
2.7. Applications de la logique et de la commande floue 29
2.7.1. L'utilisation de la commande floue 29
2.8. Système de commande adaptative 30
2.8.1. Définition de la commande adaptative 31
2.8.2. Objectifs de la commande adaptative 31
2.8.3. Commande adaptative directe et indirecte 31
2.8.4. Classifiction des systemes de commande adattive 32
2.9 Conclusion 40

Chapitre 3 Commande hybride adaptative floue avec superviseur

3.1 Introduction 41
3.2 Commande adaptative floue avec une direction de commande connue 42
3.2.1Position du problème 42
3.2.2 Conception de commande hybride adaptative floue 44
3.2.3 Résultats de simulation 51
3.2.3.1 Exemple d'un pendule inversé 52
3.3 Conclusion 58

Conclusion générale 59

Introduction générale

Depuis quelques années, les recherches sur la commande et l'automatisation industrielle se sont multipliées. Différentes structures de linéarisation ont été proposées pour résoudre le problème de la commande des systèmes non-linéaires. Contrairement à l'automatique linéaire, l'automatique non linéaire ne dispose pas de solutions universelles ni pour l'analyse des systèmes ni pour la conception de leurs contrôleurs. La plupart des approches de commande non linéaires exigent la connaissance d'un modèle mathématique du système. Les performances assurées seront directement liées à l'exactitude du modèle utilisé. Les techniques de la linéarisation du premier ordre sont utilisées pour linéariser les systèmes non linéaires autour d'un point d'opération. Etant donné que c'est la première technique qui a été introduite pour la commande des systèmes non linéaire, elle a trouvé une large acceptation dans le monde industriel. Par ailleurs cette solution reste, cependant valable uniquement autour de ce point d'opération, car il est difficile de prédire l'effet de la non linéarités si on s'éloigne de ce point. Sur le plan théorique, ce problème a été résolu par le développement de techniques de commande non linéaire basées sur la théorie de la géométrie différentielle. L'idée principale derrière cette théorie est de trouver une transformation non linéaire des états du système par lesquelles une classe des systèmes non linéaires peut être transformée en systèmes linéaires à travers le retour d'état, connu sous l'appellation anglophone **feedback linearization**.

Cette approche a souvent été employée pour résoudre des problèmes pratiques de commande mais elle impose que le vecteur d'état soit mesuré et que le modèle du procédé à commander soit précis. De plus, les propriétés de robustesse ne sont pas garanties face aux incertitudes paramétriques du modèle. En effet, cette technique est basée sur l'annulation exacte des termes non linéaires. Par conséquent, la présence d'incertitudes de modélisation sur les termes non linéaires rend l'annulation inexacte et l'équation entrée-sortie résultante non linéaire.

A ce stade, et afin de pallier à ces problèmes décrits précédemment, la commande adaptative floue des systèmes non linéaires trouve un grand essor et elle est la voie de recherche de plusieurs travaux.

L'idée principale est d'utiliser la logique floue pour approcher la dynamique du système ou pour approximer le contrôleur.

Un grand nombre de travaux sont disponibles sur la commande adaptative pour les systèmes affines en la commande, contrairement aux systèmes non-affines c'est-à-dire des systèmes caractérisés par des entrées de commande qui n'apparaissent pas linéairement dans l'équation d'état du système.

La conception de commande adaptative floue pour les systèmes non linéaires est basée sur le théorème de fonction implicite pour montrer l'existence d'une loi de commande idéale. Afin de simplifier le développement de loi de commande qui approxime la loi idéale. Le travail présenté dans ce mémoire est consacré au développement de lois de commande adaptative floue pour les systèmes non linéaires incertains dont le modèle est sous la forme générale d'un système non linéaire. Les deux types des systèmes monovariables et multivariables sont traités dans le développement de lois de commande. Les techniques développées dans ce mémoire assurent la stabilité et la robustesse des boucles de commande. Chaque technique proposée est testées par exemple de simulation pour confirmer ses performances. Le mémoire est organisé en trois chapitres : **Dans le premier chapitre**, une description de la logique floue utilisée est détaillée en donnant ses notions essentielles avec la formulation mathématique adéquate à son exploitation pour la commande adaptative. **Le deuxième chapitre** présente la théorie de la commande adaptative directe et indirecte pour une classe de système non linéaire monovariable sous la forme générale. Et puis deuxapproches distinctes se sont formulées pour la commande adaptative floue, après transformation, selon l'approche directe et l'approche indirecte. Dans l'approche directe, les systèmes sont utilisés pour approximer un contrôleur idéal inconnu. En revanche la commande indirecte exploite les systèmes flous pour estimer la dynamique inconnue du système non linéaire à commander afin de synthétiser des lois de commande basées sur ces estimations. Le problème de la singularité engendré par l'approche indirecte est résolu par l'utilisation d'un algorithme de projection. **Le troisième chapitre** présente la commande adaptative floue indirecte pour une classe de systèmes non linéaires monovariables SISO. Pour ce faire, on exploité en premier lieu le théorème de la fonction implicite pour montrer l'existence du contrôleur idéal inconnu. Ensuite, pour des raisons des incertitudes sur le modèle liées aux problèmes de transformations et modélisation, un système flou est utilisé pour approximer la loi de commande idéale. Les lois d'adaptation des paramètres ajustables sont déduites à partir de l'étude de stabilité au sens de Lyapunov. La robustesse étant assurée par l'ajout d'un terme de commande de supervision.

Chapitre 1

Eléments de théorie de la logique floue

Chapitre 1

Eléments de théorie de la logique floue

1.1 Introduction

La logique floue est une extension de la logique booléenne créée par Lotfi Zadeh en 1965 en se basant sur sa théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la vérification d'une condition, nous permettons à une condition d'être dans un autre état que vrai ou faux. La logique floue confère ainsi une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions et des incertitudes.

Un des intérêts de la logique floue pour formaliser le raisonnement humain est que les règles sont énoncées en langage naturel. Voici par exemple quelques règles de conduite qu'un conducteur suit, en supposant qu'il tienne à son permis

Si le feu est rouge...	si ma vitesse est élevée...	et si le feu est proche...	alors je freine fort.
Si le feu est rouge...	si ma vitesse est faible...	et si le feu est loin...	alors je maintiens ma vitesse.
Si le feu est orange...	si ma vitesse est moyenne...	et si le feu est loin...	alors je freine doucement.
Si le feu est vert...	si ma vitesse est faible...	et si le feu est proche...	alors j'accélère.

Intuitivement, il semble donc que les variables d'entrée à l'instar de cet exemple sont appréciées par le cerveau de manière approximative, correspondant ainsi au degré de vérification d'une condition de la logique floue. Un système complet de règles basées sur la logique floue et permettant de prendre des décisions est appelé un **système d'inférence flou**.

Afin d'illustrer chacune des définitions, nous allons concevoir au fil de ce cours un système d'inférence flou. L'objectif sera de décider du pourboire à donner à la fin d'un repas au restaurant en fonction de la qualité du service ainsi que de la qualité de la nourriture

Dans la logique classique, les variables gérées sont booléennes. C'est à dire qu'elles ne prennent que deux valeurs 0 ou 1.La logique floue à, elle, pour but de raisonner à partir de connaissances imparfaites qui opposent résistance à la logique classique. Pour cela la logique floue se propose de remplacer les variables booléennes par des variables flous.

Un contrôleur adaptatif est un contrôleur de structure classique ou de structure plus complexe muni de coefficients ajustables dont l'ajustement à l'aide d'un algorithme convenable permet d'étendre le domaine de fonctionnement. De manière plus générale, on peut également définir

un contrôleur adaptatif comme toute méthode de commande utilisant une mise à jour en temps réel du modèle mathématique du système à régler.

1.2. Historique et définition de la logique floue

Depuis longtemps l'homme recherche à maîtriser les incertitudes (par exemple celles qui sont obtenues avec des instruments d'observation qui produisent des erreurs de mesure‟ comme le cas d'un plan sur lequel on mesure grossièrement des distances : ‘’environ 200m’’ ″) et les imperfections inhérentes à sa nature. La première réelle manifestation de la volonté de formaliser la prise en compte des connaissances incertaines fut le développement de la théorie des probabilités à partir du XVII siècle. Mais les probabilités ne peuvent maîtriser les incertitudes psychologiques et linguistiques. On a donc assisté aux développements des théories de probabilité subjective (dans les années 50) puis de l'évidence (dans les années 60). [1,2,3]

Puis la Logique Floue est apparue en 1965 à Berkeley dans le laboratoire de Lotfi Zadeh avec la théorie des sous-ensembles flous puis en 1978 avec la théorie des possibilités. Ces deux théories constituent aujourd'hui ce que l'on appelle logique floue. La logique floue permet la formalisation des imprécisions dues à une connaissance globale d'un système très complexe et l'expression du comportement d'un système par des mots.

Elle permet donc la standardisation de la description d'un système et du traitement de données aussi bien numériques que l'exprimées symboliquement par des qualifications linguistiques.

Dans ce travail, notre objectif est de résoudre le problème de commande des systèmes non linéaires à l'aide de la logique floue. Cet intérêt est dû principalement au succès remarquable qu'a connu la logique floue et qui présente les avantages suivants : [2,3]

- La non nécessité d'une modélisation.
- La possibilité d'implémenter les connaissances d'experts.
- La simplicité de la relation.
- La robustesse face aux perturbations.

Pour ces raisons on va étudier dans ce chapitre les concepts de la théorie des ensembles flous :

Les ensembles flous, « les fonctions d'appartenances et les opérateurs de la logique floue ».

Les systèmes, qui sont les outils principaux de la commande floue, seront étudiés en détail ainsi que toutes leurs composantes, à savoir les règles floues, la fuzzification, la défuzzufication et le mécanisme d'inférence

1.3. Concepts fondamentaux

1.3.1. Définition de sous-ensemble flou

Le concept de sous-ensemble flou a été introduit pour éviter les passages brusques d'une classe à une autre (par exemple, le passage de la classe rouge à la classe jaune) et donnée la possibilité aux éléments d'appartenir partiellement à chacune des classes (par exemple l'orange à un fort degré d'appartenance a la classe rouge et un faible degré à la classe jaune) ou de n'appartenir complètement ni à l'une ni à l'autre [4].

Un sous-ensemble flou F est défini sur un ensemble de valeur, le référentiel U. Il est caractérisé par une fonction d'appartenance :

$$\mu : x \in U \rightarrow \mu(x) \in [0,1]. \qquad (1.1)$$

Qui quantifie le degré d'appartenance de chaque élément de U à F.

Exemple

La figure suivante montre l'évaluation de la température de l'eau d'un récipient par les mots

Froide : F Tiède : T Chaude : C

Dans la logique classique

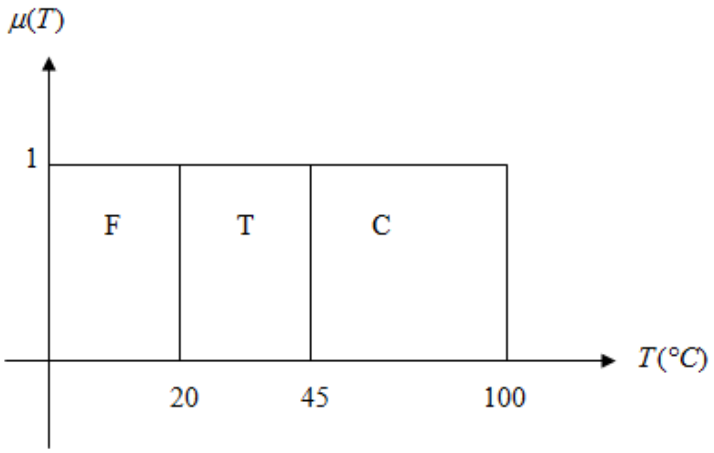


Figure 1.1. Evaluation de la température de l'eau en logique classique

On voit que la logique classique ne peut utiliser que le 0 et le 1 ainsi l'eau est d'abord totalement froide puis tiède et enfin chaude. En dessous nous pouvons observer la représentation graphique de trois fonctions d'appartenance ''Froid'', ''Tiède'' et ''Chaud''.

Ces fonctions nous permettent de superposer sur des plages de température données les« qualificatifs froid et tiède » ainsi « que tiède et chaud ». On se rapproche donc du raisonnement humain.

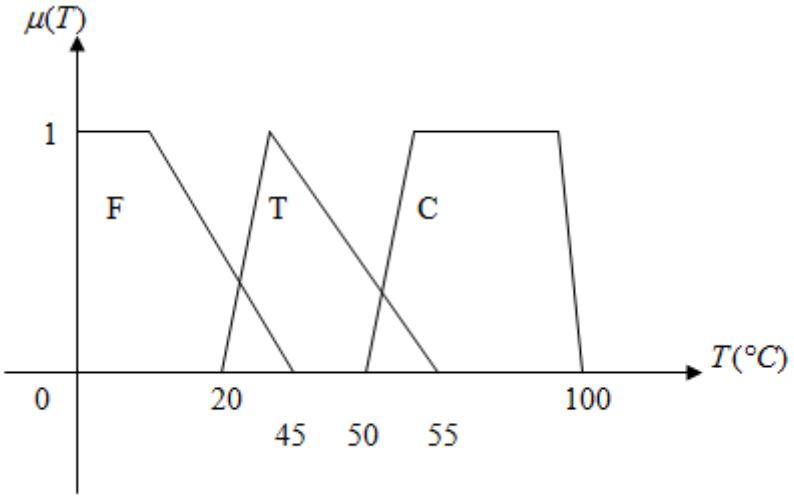


Figure 1.2 Evaluation de la température de l'eau en logique floue.

1.3.2.Opérations de base sur les sous-ensembles flous.

La théorie mathématique sur les sous-ensembles flous définit de nombreuses opérations sur ces sous-ensembles et sur les fonctions d'appartenances qui rendent ces notions utilisables. On ne présente ici que les opérations de base de cette théorie.

Si A et B sont deux sous-ensembles flous et $\mu(A)$ *et* $\mu(B)$ leur fonction d'appartenance, on définit :

1.3.2.1.Complémentaire de A par la fonction d'appartenance :

Le complément A^c d'un sous-ensemble flou A de U est défini comme le sous-ensemble flou de U de la fonction d'appartenance :

$$\forall\ x \in U;\ \mu_{A^c}(x) = 1 - \mu_A(x). \tag{1.2}$$

$$\mu(A \cap B) = \min(\ \mu(A), \mu(B)). \tag{I.3}$$

$$\mu(A \cap B) = \min(\ \mu(A), \mu(B)). \tag{I.3}$$

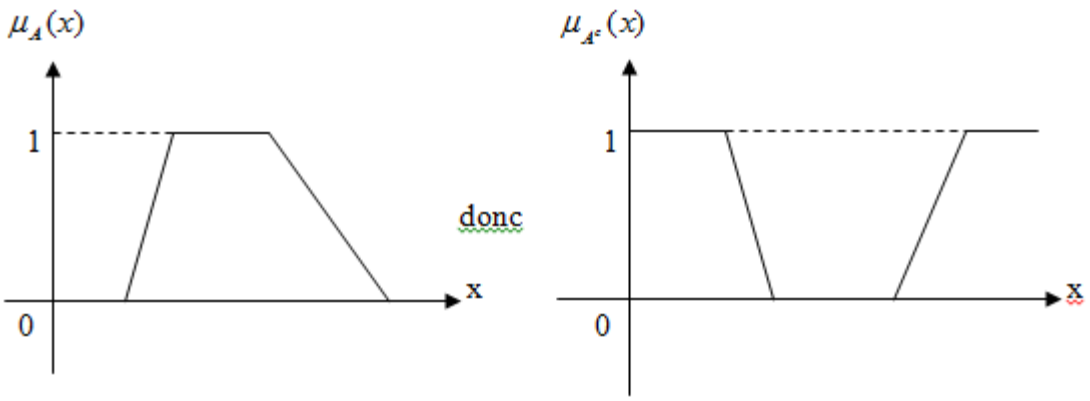


Figure 1.3. Complimentaire

$$\mu(A \cap B) = \min(\mu(A), \mu(B)) . \qquad (1.3)$$

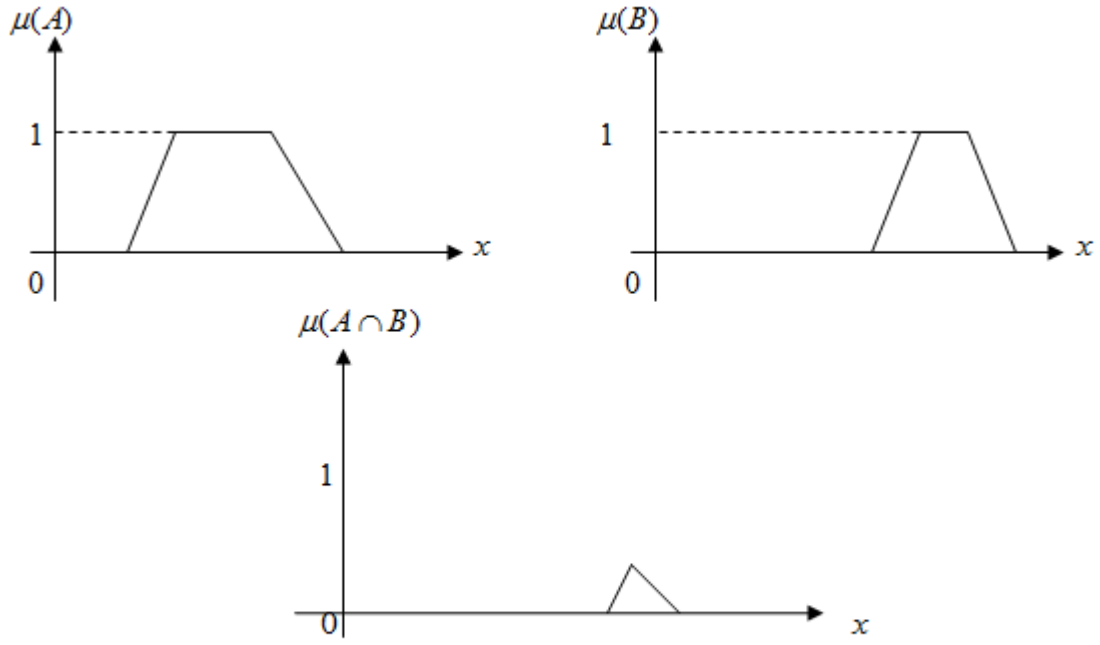


Figure 1.4. Intersection

1.3.2.2Sous-ensemble A ou B, AÈB, par la fonction d'appartenance

$$\mu(A \cup B) = Max(\mu(A), \mu(B)) . \qquad (1.4)$$

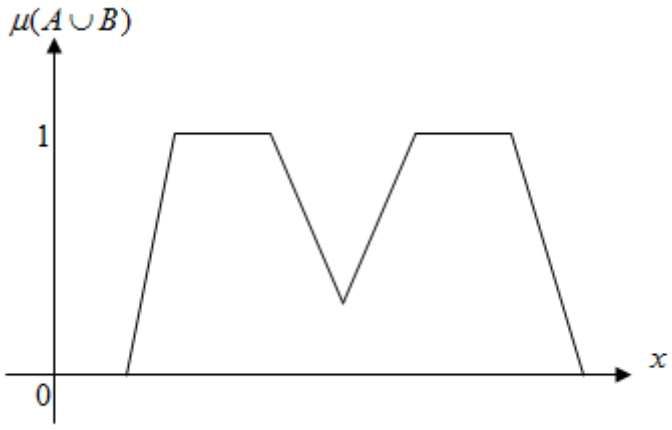


Figure.1.5. Union

Ces définitions sont celles les plus communément utilisées mais parfois, pour certains cas, d'autres sont plus appropriées. Par exemple l'intersection peut être définie par le produit des fonctions d'appartenance et l'union par la moyenne arithmétique des fonctions d'appartenance. Ces différentes techniques de calcul engendrent une énorme capacité d'adaptation des raisonnements flous.

1.4. Règles floues

Les règles floues représentent une connaissance humaine, exprimée en langage naturel, à l'aide de mots vagues, mal définis, flous. Dans la théorie des systèmes flous il existe deux types de règles :

1.4.1.Règle de Mamdani

Mamdani fut le premier à utiliser la logique floue pour la synthèse de commandes. Il utilise le minimum comme opérateurs de conjonction et d'implication. [5,6]

La règle utilisée par Mamdani est de la forme :

IF x_1 *is* A_1 and x_2 *is* A_2 and... x_n *is* A_n then y is B. (1.5)

Où A_i : i = 1…n et B sont des ensembles flous caractérisant la partie promise et la partie conséquence, définis par la fonction d'appartenance $\mu_{A_i}(x_i)$ *et* $\mu_B(y)$ .La règle peut être aussi écrite comme :

IF x is A then y is B. (1.6)

Où A est l'ensemble flou définit par le produit cartésien. La règle floue est interprétée comme une implication floue de la forme :

$$A \rightarrow B. \tag{1.7}$$

I.4.2.Règle de Takagi-Sugeno

La règle de TakagiSugeno présente à la différence de celle de Mamdani une conséquence qui n'est pas un ensemble flou mais une fonction des entrées.

Soit :

$$\text{IF } x_1 \text{ is } A_1 \quad \text{and } x_2 \text{ is } A_2 \quad \text{and... } x_n \text{ is } A_n \quad \text{then } y = B(x). \tag{1.8}$$

Où B(x) est une fonction analytique des entrées x. De la même manière, la règle de TS peut être vue comme une implication flou de la forme :

$$A \rightarrow B(x). \tag{1.9}$$

I.4.3. Etapes de conception d'un système flou

La conception d'un FLC nécessite le choix des paramètres suivants :

a) Définition du système en termes de ses variables d'entrées et de sortie :

Généralement les variables d'entrées sont l'erreur (e) et le changement de l'erreur ($\dot{e}$) de la variable d'état du processus.

b) Choix de la partition floue :

On associe un ensemble de terme linguistique caractérisés par des fonctions d'appartenances définis sur le même univers de discoure pour chaque variable d'E/S.

En fait, le choix de la partition floue consiste à déterminer le nombre de termes dans cet ensemble.

Généralement ce nombre est impair et compris entre **3** et **9**. Ce choix dépend du problème à traiter s'il exige la précision ou la robustesse du système.

c) Choix des fonctions d'appartenance :

Les fonctions d'appartenances les plus utilisées sont les fonctions :

Trapézoïdal et triangulaire.

Car elles sont prouvées d'être de bon compensateur entre l'efficacité et la facilité d'implémentation (programmation).Le chevauchement de la fonction d'appartenance avec ses voisines est compris entre **10%** et **50** du support de la fonction voisine.

d) Construction de la base de règle [19]:

De nombreuses méthodes ont été proposées, ils peuvent être groupés selon leur principe en deux catégories :

- Méthodes d'extraction naturelle.
- Méthode d'extraction automatique.

1.5. Théorie des possibilités

Cette théorie est le deuxième volet de la logique floue. Elle est basée sur les fonctions de possibilités comme l'autre l'est sur les fonctions d'appartenance.

La fonction de possibilités Π associe à chaque événement d'un univers Ω une valeur entre 0 et 1 qui définit le dégrée de possibilité de l'événement :

$$\Pi : u \in \Omega \rightarrow \Pi(u) \in [0 ; 1]. \qquad (1.10)$$

Il ne faut pas confondre possibilités et probabilités car, même si les deux notions traitent des incertitudes événementielles, certaines différences très importantes séparent ces deux concepts. Par exemple, si Å est l'événement contraire de A, on a P (Å)=1-P (A) mais il n'y a aucune raison que Π (Å) soit égal à 1-Π (A).

Ou encore, la somme des probabilités des événements d'un univers est égale à 1 mais celle des possibilités est presque toujours supérieure à 1.

On ne rentrera pas dans les détails de cette théorie car elle est beaucoup moins utile à la commande floue.

1.6. Sous-ensembles flous

La logique floue repose sur la théorie des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. Dire que la théorie des ensembles flous est une généralisation de la théorie des ensembles classiques signifie que cette dernière n'est qu'un cas particulier la théorie des ensembles flous. Pour faire une métaphore en langage ensembliste, la théorie des ensembles classiques n'est qu'un sous-ensemble de la théorie des ensembles flous

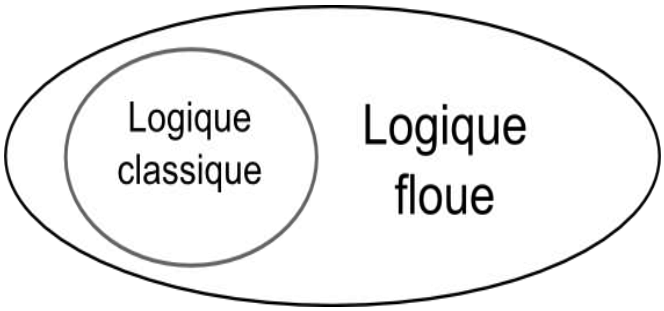


Figure.1.6."La théorie des ensembles classiques n'est qu'un sous-ensemble de la théorie des ensembles flous"

Par abus de langage, suivant les us de la littérature, nous utiliserons indifféremment les termes sous-ensembles flous et ensembles flous. Les ensembles classiques sont également appelés ensembles nets, par opposition à flous, et de même la logique classique est également appelée logique booléenne ou binaire.

Voici une figure montrant la fonction d'appartenance choisie pour caractériser le sous-ensemble 'bon' de la qualité du service :

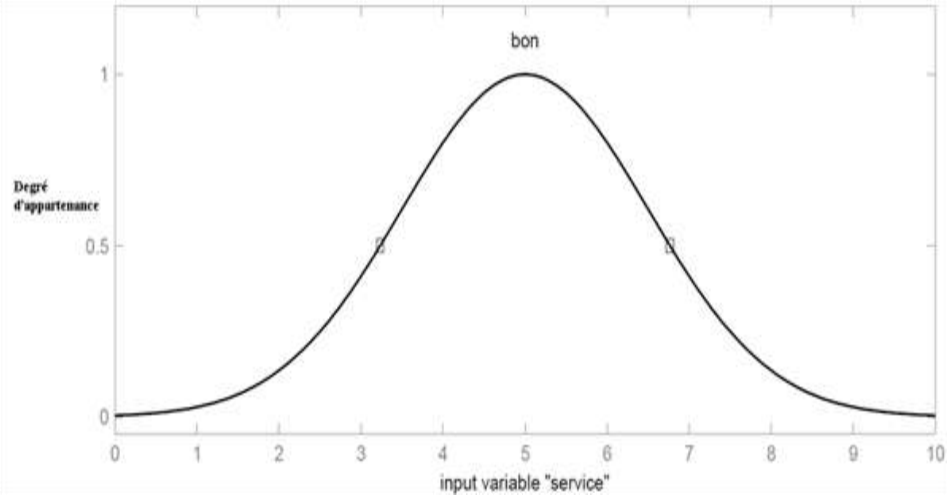


Figure 1.7. Fonction d'appartenance caractérisant le sous-ensemble 'bon' de la qualité du service.

1.6.1. **DEFINITION** :

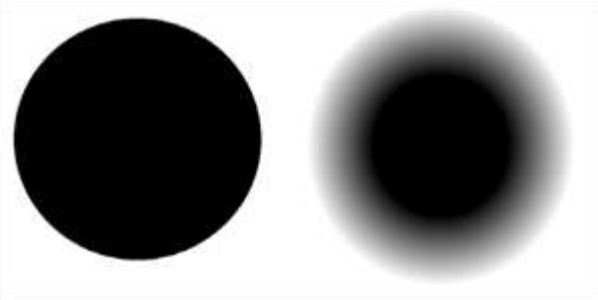
Cette notation veut simplement dire que quel que soit l'entrée X donnée à la fonction *fa*sa sortie est un réel entre 0 et 1. En théorie, il est possible que la sortie soit supérieure à 1, mais en pratique cela n'est quasiment jamais utilisé. Note : cette fonction d'appartenance est l'équivalent de la fonction caractéristique d'un ensemble classique.

Dans notre exemple du pourboire, il nous faudra redéfinir des fonctions d'appartenance pour chaque sous-ensemble flou de chacune de nos trois variables :

- Input 1 : qualité du service. Sous-ensembles : mauvais, bon et excellent.
- Input 2 : qualité de la nourriture. Sous-ensembles : exécrable et délicieux.
- Output : montant du pourboire. Sous-ensembles : faible, moyen et élevé.

La forme de la fonction d'appartenance est choisie arbitrairement en suivant les conseils de l'expert ou en faisant des études statistiques : formes sigmoïde, tangente hyperbolique, exponentielle, gaussienne ou de toute autre nature sont utilisables.

Cette figure montre graphiquement la différence entre un ensemble classique et l'ensemble flou correspondant à une nourriture délicieuse :



Cette figure compare les deux fonctions d'appartenance correspondant aux ensembles précédents :

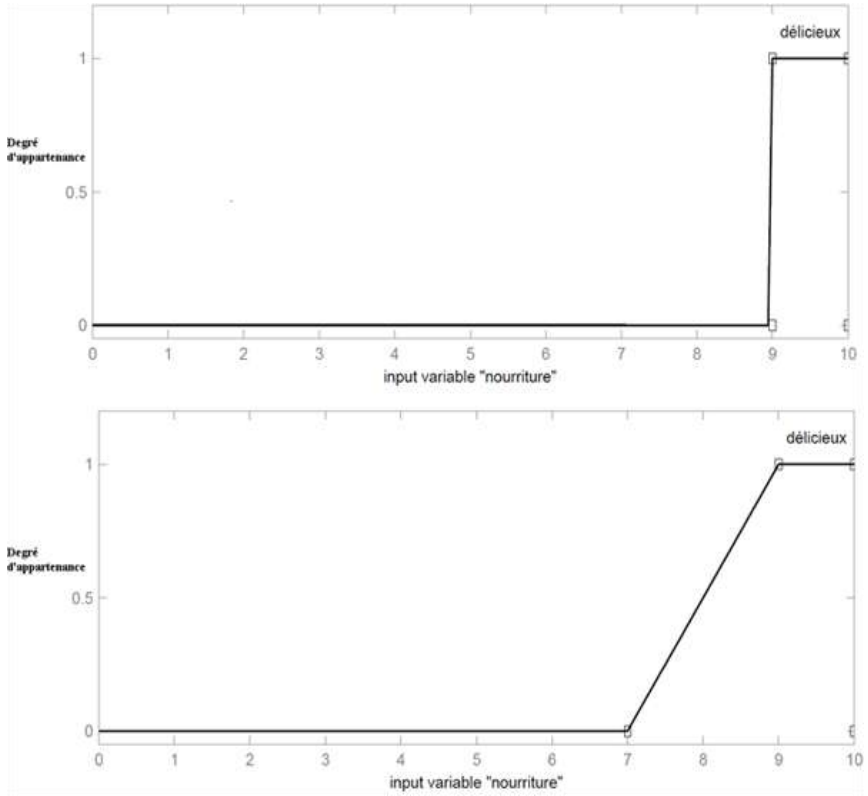


Figure.1.8.comparaison entre les deux fonctions

Comparaison entre fonction caractéristique d'un ensemble classique (graphique du haut) et fonction d'appartenance d'un ensemble flou (graphique du bas)

Pour pouvoir définir les caractéristiques des ensembles flous, nous redéfinissons et étendons les caractéristiques usuelles des ensembles classiques.

Les ensembles flous comportent un certain nombre de propriétés. Voici les définitions des propriétés les plus importantes, elles ne sont néanmoins énoncées qu'à titre informatif car non nécessaire pour la compréhension du cours. Si vous le souhaitez, vous pouvez donc passer dès maintenant à la section suivante.

Soit X un ensemble, A un sous-ensemble flou de X et μAla fonction d'appartenance le caractérisant.

DEFINITION : La **hauteur** de A, notée $h(A)$, correspond à la borne supérieure de l'ensemble d'arrivée de sa fonction d'appartenance : $h(A)=\sup\{\mu A(x)|x\in X\}$.

DEFINITION : A est dit **normalisé** si et seulement si $h(A)=1$. En pratique, il est extrêmement rare de travailler sur des ensembles flous non normalisés.

DEFINITION : Le **support** de A est l'ensemble des éléments de X appartenant au moins un peu à A. Autrement dit, c'est l'ensemble $supp(A)=\{x\in X|\mu A(x)>0\}$.

DEFINITION : Le **noyau** de A est l'ensemble des éléments de X appartenant totalement à A. Autrement dit, c'est l'ensemble $noy(A)=\{x\in X|\mu A(x)=1\}$. Par construction, $noy(A)\subseteq supp(A)$.

DEFINITION : Une ***α*-coupe** de A est le sous-ensemble classique des éléments ayant un degré d'appartenance supérieur ou égal à α : α-coupe$(A)=\{x\in X|\mu A(x)\geqslant\alpha\}$

Voici une autre fonction d'appartenance pour un pourboire moyen sur lequel nous avons fait figurer les propriétés précédentes :

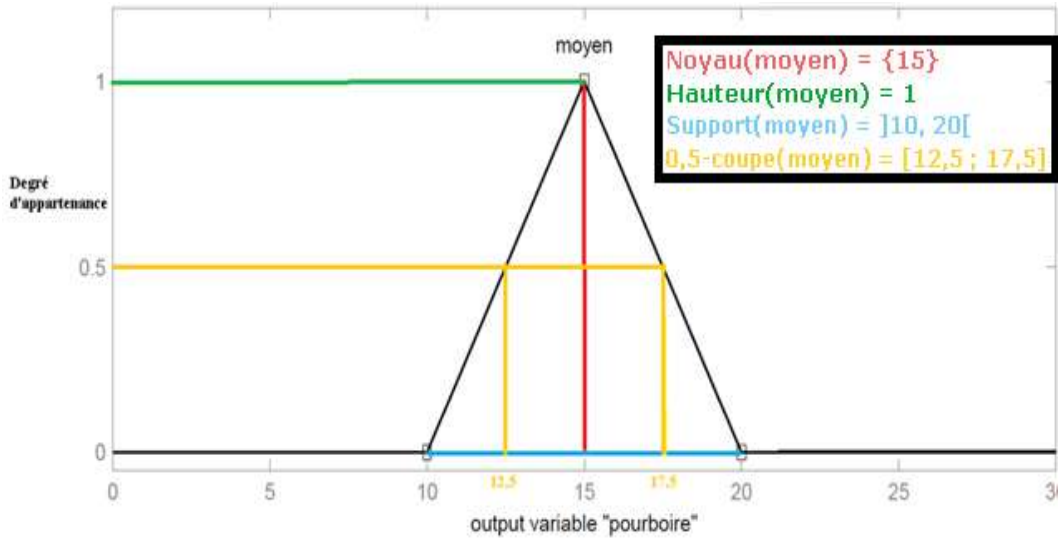


Figure 1.9. Propriétés d'un ensemble flou

Nous remarquons que si A était un ensemble classique, nous aurions simplement $supp(A)=noy(A)$ et $h(A)=1$ (ou $h(A)=0$ si $A=\emptyset$).

Nos définitions permettent donc bien de retrouver les propriétés usuelles des ensembles classiques. Nous ne parlerons pas de la cardinalité (c'est-à-dire le nombre d'éléments qu'un ensemble contient) car nous n'utiliserons pas cette notion dans la suite de ce cours.

1.7. Variables linguistiques

Le concept de fonction d'appartenance vu précédemment nous permettra de définir des systèmes flous en langage naturel, la fonction d'appartenance faisant le lien entre logique floue et variable linguistique que nous allons définir à présent.

DEFINITION : Soit V une variable (qualité du service, montant du pourboire, etc.), X la plage de valeurs de la variable (par exemple, entre 0 et 30€ pour le pourboire) et *TV*

un ensemble fini ou infini de sous-ensembles flous. Une **variable linguistique** correspond au triplU+Het (*V*,*X*,*TV*)

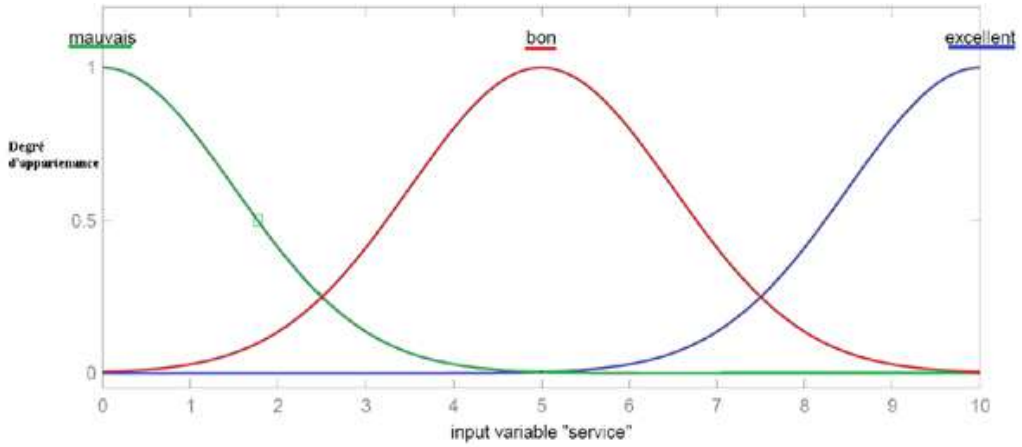


Figure 1.10. Variable linguistique 'qualité du service'

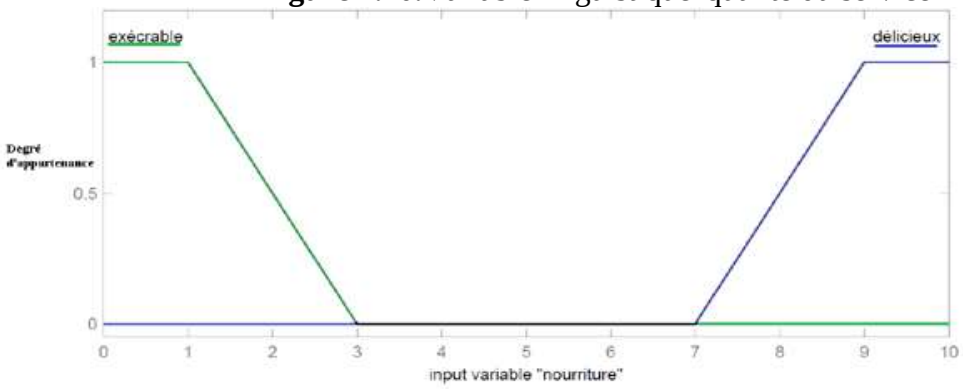


Figure 1.11. Variable linguistique 'qualité de la nourriture'

Variable linguistique 'montant du pourboireLorsque nous définissons les sous-ensembles flous d'une variable linguistique, l'objectif n'est pas de définir exhaustivement la variable linguistique. Au contraire, nous définirons seulement les sous-ensembles flous qui nous seront utiles plus tard dans la définition des règles que nous appliquerons dessus. C'est par exemple la raison pour laquelle nous n'avons pas défini de sous-ensemble "moyen" pour la qualité de la nourriture. En effet, ce sous-ensemble ne nous sera pas utile dans nos règles ; cela peut être considéré comme étant le fait qu'une nourriture moyenne est considérée comme normale, par conséquent qu'il n'y a rien à signaler en particulier concernant ce fait. De même, c'est également la raison pour laquelle (par exemple) 30 est un pourboire plus élevé que 25, alors que 25 appartient pourtant davantage au sous-ensemble flou "élevé" que 30 : cela est dû au fait que 30 est considéré non pas comme élevé mais très élevé (ou exorbitant si l'on veut changer d'adjectif). Néanmoins, nous n'avons pas créé de sous-ensemble flou "très élevé" car nous n'en avons pas besoin dans nos règles.

1.8. Raisonnement en logique floue

Attention, c'est la sous-partie du tuto la plus importante ! C'est l'essence de la logique floue. Un conseil : relisez-la plusieurs fois

En logique classique, les raisonnements sont de la forme :

$$\begin{cases} si\ p\ alors\ q \\ p\ vrai\ alors\ q\ vrai \end{cases}$$

En logique floue, le raisonnement flou, également appelé raisonnement approximatif, se base sur des **règles floues** qui sont exprimées en langage naturel en utilisant les variables linguistiques dont nous avons donné la définition précédemment. En logique, une règle est sous la forme :

Si [prémisses] alors [conclusion].

En logique floue, les règles auront des formes telles que : $Six \in A$

$ety \in B\ alors\ z \in CSix \in A\ alors\ z \in CSix \in A\ ou\ y \in B\ alors\ z \in C$

avec A, B et C des ensembles flous.

Dans notre exemple, cela donnerait : *'Si (la qualité de la nourriture est délicieuse), alors (le pourboire sera élevé)'*. (ce qui correspond à la forme $Si\ x \in Aalorsz \in C$)

La variable 'pourboire' appartient à l'ensemble flou 'élevé' à un degré égal au degré de validité de la prémisse, autrement dit en l'occurrence au degré d'appartenance de la variable 'qualité de la nourriture' à l'ensemble flou 'délicieux'. L'idée sous-jacente est que plus les propositions en prémisse sont vérifiées, plus l'action préconisée pour les sorties doit être respectée. Voici ce que nous obtenons pour la règle floue *'Si (la qualité de la nourriture est délicieuse), alors (le pourboire sera élevé)'* lorsque la qualité de la nourriture est notée 8,31 sur 10 :

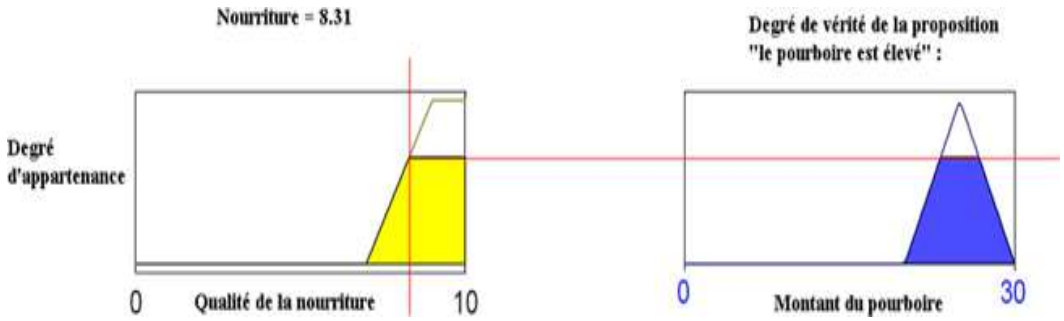


Figure 1.12. Application de la règle 'Si (la qualité de la nourriture est délicieuse), alors (le pourboire sera élevé)' avec nourriture = 8,31.

Le résultat de l'application d'une règle floue dépend donc de deux facteurs :

- la définition de la fonction d'appartenance de l'ensemble flou de la proposition située en conclusion de la règle floue ;
- le degré de validité des propositions situées en prémisse.

En lisant que la qualité de la nourriture est notée 8,31 sur 10, vous vous êtes peut-être demandé : comment avons-nous défini la valeur des variables ? Il faut garder en tête que nous nous plaçons dans le cas où le système est capable de mesurer précisément les inputs. Dans la réalité, la précision des inputs dépend bien sûr de la précision des inputs : dans le cas du pourboire, on pourrait imaginer qu'une personne lamba donne une note à l'unité près (par exemple 8/10 pour la qualité de la nourriture) alors qu'un expert du guide Michelin donne une note à deux décimales près (par exemple 8,31/10 pour la qualité de la nourriture). N'oubliez pas que les systèmes flous sont des systèmes artificiels : ce n'est pas une description du cerveau des humains. Par conséquent, en pratique, il faut s'imaginer par exemple un FPS ou un robot qui reçoit des inputs de façon classique, inputs qui sont le plus souvent sous forme numérique (par exemple, position d'un joueur dans un FPS ou luminosité constatée par les capteurs du robot).

Comme nous avons défini les opérateurs flous ET, OU et NON, la prémisse d'une règle floue peut très bien être formée d'une conjonction (ET) ou disjonction (OU) de propositions floues. L'ensemble des règles d'un système flou est appelé la **matrice des décisions**. Voici celui de notre exemple du pourboire :

Si le service est mauvais ou la nourriture est exécrable	alors le pourboire est faible.
Si le service est bon	alors le pourboire est moyen.
Si le service est excellent ou la nourriture est délicieuse	alors le pourboire est élevé.

Nous allons maintenant appliquer l'ensemble des 3 règles de notre matrice des décisions. Pour chacune des règles, nous obtenons un résultat intermédiaire (à droite sur le schéma suivant). Pour les régles utilisant l'opérateur OU (règles 1 et 3), nous utiliserons l'opération MAX (cela correspond aux opérateurs de Zadeh MIN/MAX que nous avons vu dans la sous-partie précédente "les opérateurs flous"). Nous agrégerons ces trois résultats intermédiaires en un résultat final en prenant simplement le maximum. Nous voyons le résultat final en bas à droite

sur le schéma suivant :

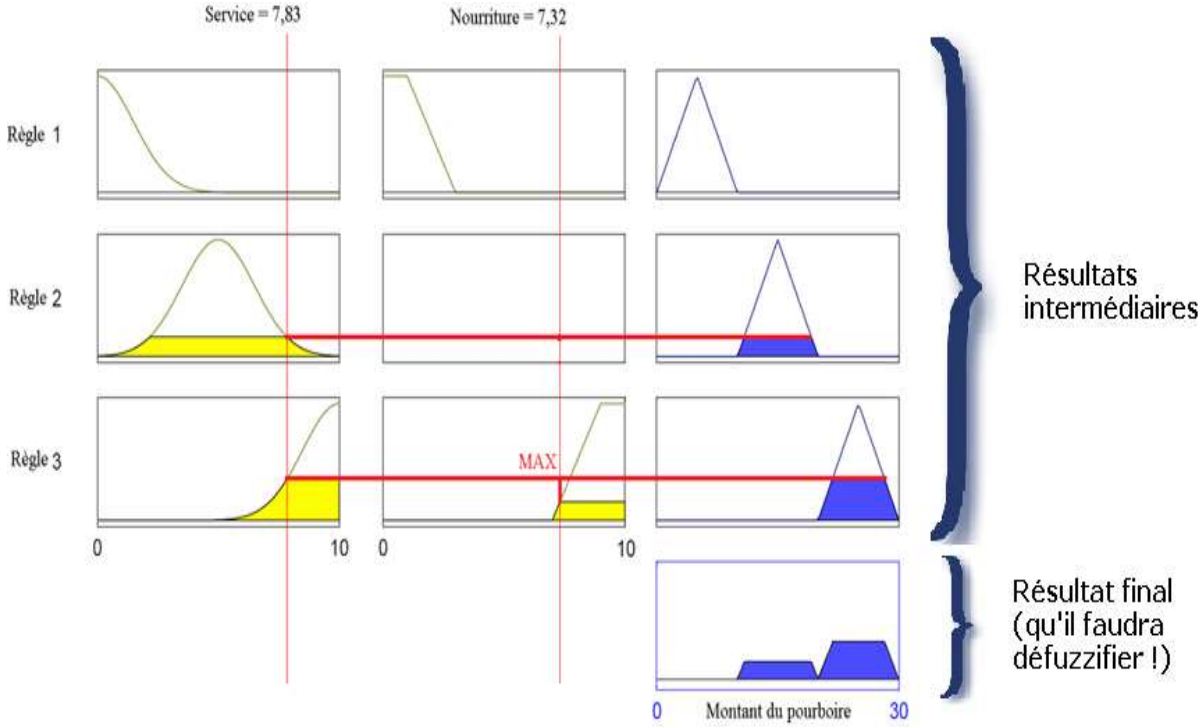


Figure 1.13. Exemple d'application de l'ensemble des 3 règles de notre matrice des décisions à l'entrée (service = 7,83 ; nourriture = 7,32)

Comme nous le voyons, il ne nous reste plus qu'à prendre la décision finale, à savoir quel pourboire nous allons réellement donner, sachant que la qualité du service est notée 7,83 sur 10 et la qualité de la nourriture 7,32 sur 10. Cette étape finale, qui permet de passer de l'ensemble flou issu de l'agrégation des conclusions à une décision unique, s'appelle la **défuzzification**. Courage, c'est bientôt fini

1.9. Défuzzification

Comme pour tous les opérateurs flous, le concepteur du système flou doit choisir parmi plusieurs définitions possibles de défuzzification. Nous allons présenter brièvement les deux principales méthodes de défuzzification : la méthode moyenne des maxima (MM) et la méthode du centre de gravité (COG).

La défuzzification MM définit la sortie (décision du montant du pourboire) comme étant la moyenne des abscisses des maxima de l'ensemble flou issu de l'agrégation des conclusions.

Pour ceux qui aiment les formules de maths compliquées (pour les autres, vous pouvez les zapper), la moyenne des abscisses des maxima se définit ainsi : $Decision=\int sy \cdot dy \int sdy$

où $S=\{y_m \in R, \mu(y_m)=SUP_{y \in R}(\mu(y))\}$

et R est l'ensemble flou issu de l'agrégation des conclusions.

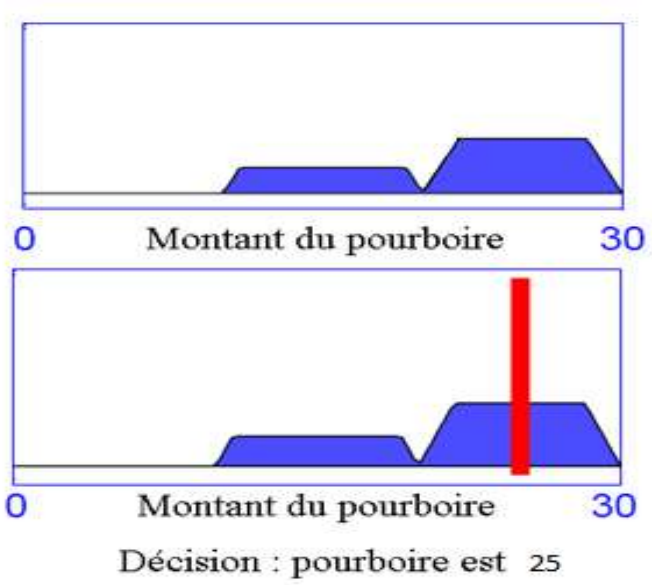


Figure 1.14.Défuzzification avec la méthode moyenne des maxima (MM)

La défuzzification COG est plus couramment utilisée. Elle définit la sortie comme correspondant à l'abscisse du centre de gravité de la surface de la fonction d'appartenance caractérisant l'ensemble flou issu de l'agrégation des conclusions.Pour les cracks des maths curieux, l'abscisse du centre de gravité de la surface se calcule ainsi :

$$Decision = \int_S y \cdot \mu(u) \cdot dy \Big/ \int_S \mu(u) \cdot dy$$

(c'est joli non ?)

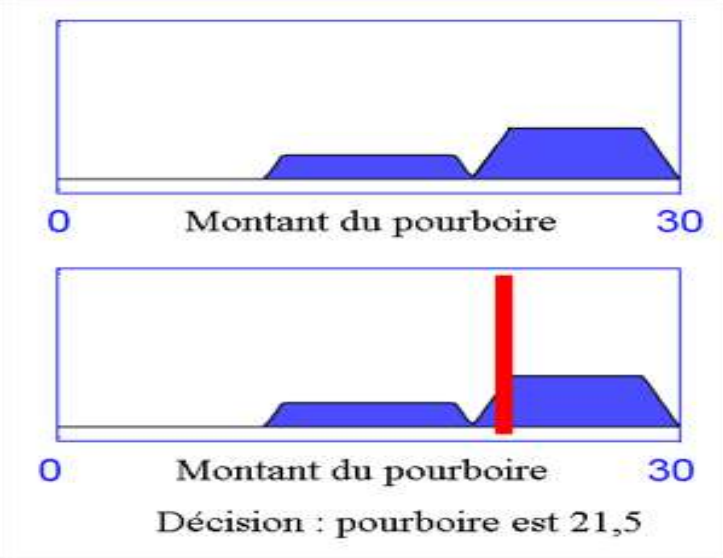


Figure 1.15.Défuzzification avec la méthode centre de gravité (COG)

Cette définition de la défuzzification COG permet d'éviter les discontinuités qui pouvaient apparaître dans la défuzzification MM, mais est plus complexe et demande des calculs plus

importants. Certains travaux cherchent à améliorer les performances en cherchant d'autres méthodes aussi efficaces mais avec une complexité algorithmique moindre. Comme nous le voyons sur les 2 figures montrant les méthodes de défuzzification MM et COG appliquées à notre exemple, le choix de cette méthode a un effet important sur la décision finale.

1.10. Conclusion

Au cours des définitions, nous avons vu que le concepteur d'un système flou doit faire un nombre de choix important. Ces choix se basent essentiellement sur les conseils de l'expert ou sur l'analyse statistique des données passées, en particulier pour définir les fonctions d'appartenance et la matrice des décisions.

Voici un aperçu synoptique d'un système flou :

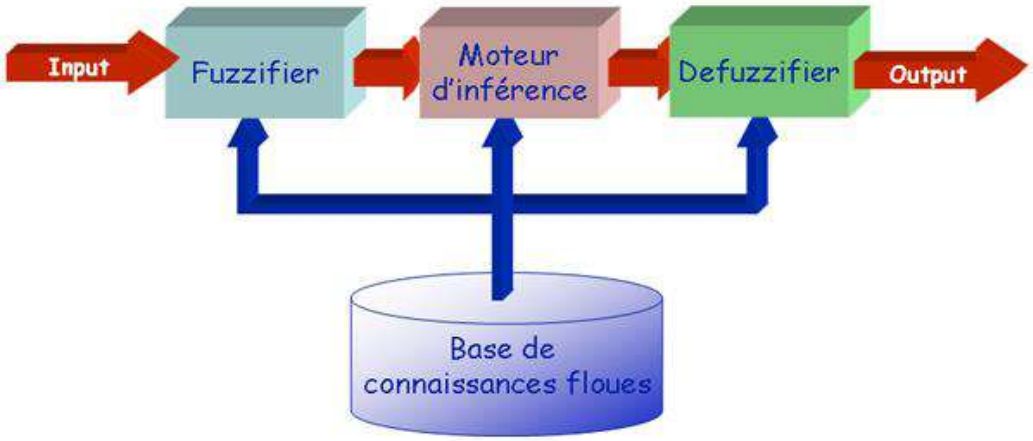


Figure 1.17. Aperçu synoptique d'un système flou

Dans notre exemple :

- l'**input** est 'la qualité du service est notée 7,83 sur 10 et la qualité de la nourriture 7,32 sur 10' ;
- le **fuzzifier** correspond aux 3 variables linguistiques 'qualité du service', 'qualité de la nourriture' et 'montant du pourboire' ;
- le **moteur d'inférence** est constitué du choix des opérateurs flous ;
- la **base de connaissances** floues est l'ensemble des règles floues ;
- le **defuzzifier** est la partie où entre en jeu la méthode de défuzzification ;
- l'**output** correspond à la décision finale : 'le montant du pourboire est 25'.

Il est intéressant de voir l'ensemble des décisions en fonction de chacune des variables avec notre système d'inférence flou par rapport au type d'ensemble de décisions que nous obtiendrions en utilisant la logique classique :

Ensemble des décisions d'un système se basant sur la logique classique, qui ne peut générer et que des surfaces linéaires Ainsi, toute la puissance de la logique floue est de rendre possible la mise en place de systèmes d'inférence dont les décisions sont sans discontinuité, flexibles et non linéaires, plus proches du comportement humain que ne l'est la logique classique. De plus, les règles de la matrice des décisions sont exprimées en langage naturel. Cela comporte de nombreux avantages, comme inclure des connaissances d'un expert non-informaticien au cœur d'un système décisionnel ou encore modéliser plus finement certains aspects du langage naturel.

Chapitre 2

Théorie de la commande adaptative hybride directe/indirecte

Chapitre 2

Théorie de la commande adaptative hybride directe/indirecte

2.1 Introduction

Dans les travaux antérieurs de la commande non linéaire, plusieurs approches ont été utilisées: commande par apprentissage, commande à structure variable,…etc. Dans ce contexte, la technique de la linéarisation entrée-sortie est spécialement utilisée dans le cadre des systèmes non linéaires [2, 14].

Une classe de systèmes non linéaires peut être transformée en une classe de systèmes linéaires à travers la technique de la linéarisation entrée-sortie. Dans ce cas, le système linéaire transformé peut être commandé par des méthodes classiques de l'automatique linéaire (placement de pôles, synthèse quadratique, synthèse robuste, …etc) [11, 16].

Normalement, une telle transformation n'existe pas toujours, mais lorsqu'elle existe, elle nécessite la connaissance parfaite du model du système pour que la linéarisation soit correcte, pour ces raisons on voit bien que l'utilisation d'une commande adaptative par les systèmes flous peut contourner le problème de méconnaissance du model et la variation dans le temps des différents paramètres qui constituent le système à commander. Plusieurs schémas de commande adaptative floue ont été présentés dans la littérature.

Depuis sa naissance la commande adaptative floue est basée essentiellement sur deux approches : La première dite directe, [6,7,8, 17], propose d'utiliser un système flou pour l'approximation d'une loi de commande linéarisante à condition de disposer du signe du gain de commande. Plus tard, plusieurs solutions ont été proposées pour éliminer l'hypothèse qui exige la connaissance du signe gain de commande.

La deuxième dite approche indirecte [1,5,7,16], dans laquelle le système obtenu par la technique de linéarisation entrée-sortie est approximé par deux systèmes adaptatifs flous. En utilisant le principe des certitudes équivalentes, ces approximations sont utilisées pour la synthèse d'une loi de commande par rétroaction.

Dans ces deux approches, les lois d'adaptations des paramètres ajustables (conclusion des règles floues) sont extraites de l'étude de la stabilité en utilisant la méthode de Lyapunov. Dans ce chapitre, notre objectif est de présenter le principe de la linéarisation entrée-sortie qui permet d'aboutir à une relation directe entre l'entrée du système et sa sortie. À partir de cette relation, une loi de commande par rétroaction « feedback linéarisation »est synthétisée. Cependant, cette loi de commande nécessite la connaissance parfaite du modèle dynamique du système à commander. Motivé par les travaux développés sur la commande adaptative floue [6,8,20], trois lois de commande adaptative floue sont proposées. Une loi adaptative

directe et deux lois adaptatives indirectes. Des résultats de simulation sont présentés pour montrer les performances des lois proposées.

En somme, nous avons essayé de résumer le présent chapitre dans un schéma illustré dans la figure 2.1 et ce pour simplifier et de permettre aux lecteurs de bien comprendre notre mémoire.

2.2 Représentation entrée-sortie

On considère une classe de systèmes non linéaires mono entré mono sortie (SISO) affines dans la commande, représenté par les équations différentielles suivantes:

$$\begin{cases} \dot{x} = \underline{f_s}(x) + \underline{g_s}(x)u \\ y = h\,(x) \end{cases} \tag{2.1}$$

Où $\underline{f}_s(x) \in R^n$,$\underline{g}_s(x) \in R^n$ et $h(x) \in R$ sont des fonctions non linéaires continues , $u \in R$ et $y \in R$représentent respectivement l'entrée et sortie du système. Le vecteur d'état étant : $x= [x_1,x_2\ldots..x_n] \in R^n$

À partir de la structure du système (2.1) on ne peut pas avoir une relation directe entre l'entré et la sortie du système, ce qui rend la mise en œuvre d'une loi de commande en poursuite d'unetrajectoire de référence difficile. Pour pallier ce problème plusieurs solutions ont été proposées dansles travaux de [3], une de ces solutions consiste à transformer le système (2.1)sous forme canonique ou forme dite normale, en utilisant la représentation entrée sortie.

On dit, **[Isidori-89]**, qu'un système est de degré relatif r si :

$$\begin{cases} L_{\underline{g_s}} L_{\underline{f_s}}^{(k)} h(x) = 0\,; k < r-1 \\ L_{\underline{g_s}} L_{\underline{f_s}}^{(r-1)} h(x) \neq 0 \end{cases} \tag{2.2}$$

où la dérivé de Lie $L_a^{(k)}$ b(x) représentent la dérivée $k^{éme}$ de b(x) dans la direction de a. Pour exprimer explicitement la relation liant l'entrée et la sortie du système (2.1), on dérive successivement cette dernière jusqu'à l'apparition de l'entrée u.

Pour un système de la forme (2.1) de degré relatif 2, on peut écrire :

$$y = h(x)$$

$$y^{(1)} = \frac{\partial h}{\partial x}\frac{dx}{dt} = \frac{\partial h}{\partial x}\left(\underline{f_s}(x) + \underline{g_s}(x)u\right)$$

$$= L_{\underline{f_s}} h(x) + L_{\underline{g_s}} h(x)u$$

or par définition, on a :$L_{gs}h(X) = 0$, alors $y^{(1)}=L_{\underline{f}s}\, h(x)$

$$y^2 = \frac{\partial L_{\underline{f_s}} h}{\partial x}\frac{dx}{dt} = \frac{\partial L_{\underline{f_s}} h}{\partial x}\left(\underline{f_s}(x) + \underline{g_s}(x)u\right)$$
$$= L^2_{\underline{f_s}} h(x) + L_{\underline{g_s}} L_{\underline{f_s}} h(x)u \qquad (2.3)$$

d'où :

Pour un système de degré relatif r, on peut aisément montrer que :

$$\begin{cases} y^{(k)} = L^{(k)}_{\underline{f_s}} h(x); k < r-1 \\ y^{(k)} = L^{(r)}_{\underline{f_s}} h(x) + L_{\underline{g_s}} L^{(r-1)}_{\underline{f_s}} h(x)u \end{cases} \qquad (2.4)$$

Avec cette équation on peut avoir une relation directe entre l'entrée du système et la sortie, dans lecas où r = n, l'équation (2.4) peut être réécrite sous la forme :

$$\begin{aligned} y^{(1)} &= L^{(1)}_{\underline{f_s}} h(x) \\ y^{(2)} &= L^{(2)}_{\underline{f_s}} h(x) \\ &\vdots \\ y^{(n)} &= L^{(n)}_{\underline{f_s}} h(x) + L_{\underline{g_s}} L^{(n-1)}_{\underline{f_s}} h(x)u \end{aligned} \qquad (2.5)$$

Ou

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\dots \\ \dot{x}_n &= f(x_1, \dots, x_n) + g(x_1, \dots, x_n)u \\ y &= x_1 \end{aligned} \qquad (2.6)$$

Avec f(x) = $L_{\underline{fs}}^{(n)}$h(x) et (x) = L_{gs} $L_{\underline{fs}}^{(n-1)}$ h(x)

Cette structure représente la forme normale d'un système non linéaire SISO affine dans la commande qu'on peut reformuler comme suit :

$$\begin{cases} x^{(n)} = f(x) + g(x)u \\ y = x_1 \end{cases} \qquad (2.7)$$

Où x= $[x_1, \dots, x_n]^T$= $[x, \dots, x^{(n-1)}]^T$ représente le vecteur d'état du système constitué des dérivées successives de la sortie. Les figures 2.2 et 2.3 représentent respectivement les structures des systèmes (2.5) et (2.6) **[19]**.

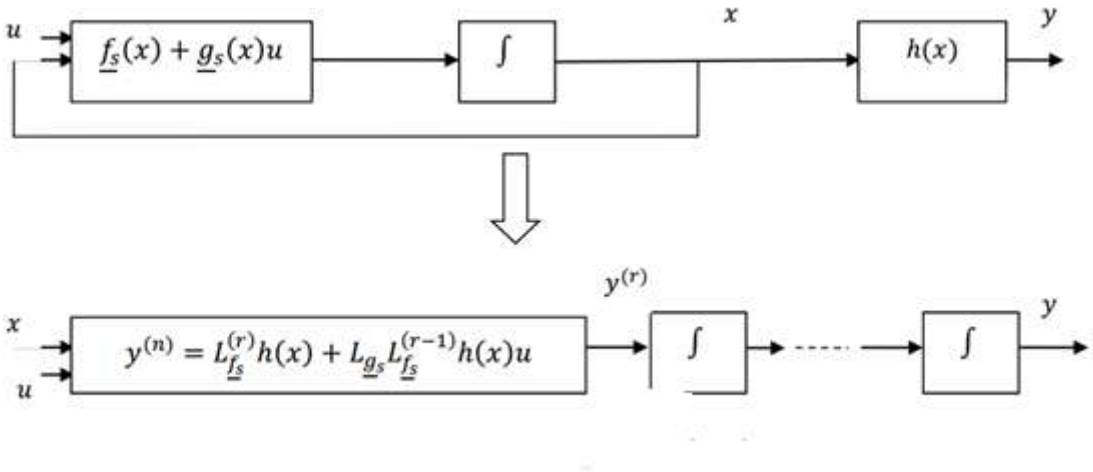


Figure 2.1 changement de la représentation du système

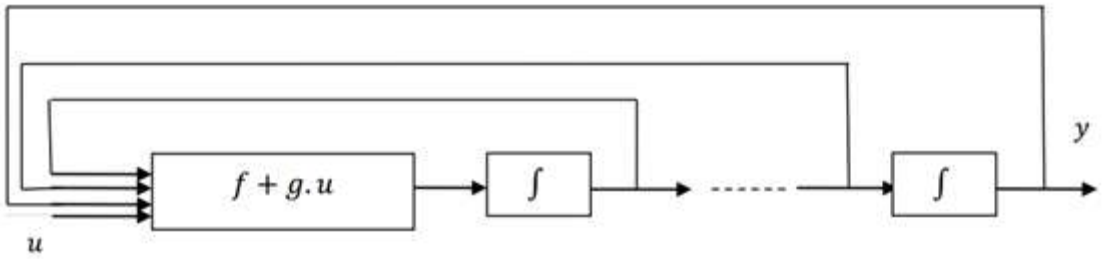


Figure 2.2 représentation entrée-sortie du système 2.7

2.3 Commande par rétroaction d'un système non linéaire

La relation définie précédemment par l'équation (2.7) nous a permis de déterminer un lien direct entre l'entrée et la sortie du système, en se basera sur cette relation pour concevoir une loi de commande capable de forcer la sortie du système à suivre une trajectoire de référence $y_d(t)$ donnée, tout en assurant que les variables d'états restent bornées le long de celle-ci. Définissons l'erreur de poursuite par

$$e = y_d(t) - y(t) \tag{2.8}$$

Le fait d'assurer la convergence asymptotique équivaut à satisfaire l'équation suivante :

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e \tag{2.9}$$

Où le coefficient $k_i$sont calcules de telle sorte que le polynôme

$H(s) = s^n + k_1 s^{(n-1)} + \ldots + k_n$,ait des racines a partie réelle négative

Alors la dérivée n^{éme} de l'erreur de poursuite est donnée par :

$$e^{(n)} = y_d^{(n)} - y^{(n)} = y_d^{(n)} - f(x) - g(x)u \tag{2.10}$$

En substituant (2.10) dans (2.9), on obtient

$$y_d^{(n)} - f(x) - g(x)u + k_1 e^{(n-1)} + \cdots + k_n e = 0 \tag{2.11}$$

$$y_d^{(n)} - f(x) - g(x)u + k^T e = 0 \tag{2.12}$$

Avec k = [k_n….k_1] et e=[e,ė,……$e^{(n-1)}$]

D'où l'on peut déduire la loi de commande :

$$u = \frac{1}{g(x)}(-f(x) + y_d^{(n)} + k^T e) \tag{2.13}$$

La loi de commande (2.13) ne peut pas être mise en œuvre que dans le cas où le système est complétement connu, dans le cas contraire une façon pour contourner ce problème serait d'utiliser la commande adaptative floue que l'on présentera comme suit.

2.4 La commande adaptative floue

Dans la commande adaptative par les systèmes flous on distingue deux types de commandes :

- La commande adaptative floue directe : Dans ce type, la loi de commande est directement approximée par un ou plusieurs systèmes flous à conclusion constante, les lois d'adaptations sont choisies de telles sortes que le système atteint les performances désirées et ils sont déduits de l'étude de stabilité via la méthode de Lyapunov.
- La commande adaptative floue indirecte : Dans ce type, pour calculer la loi de commande, on approxime d'abord le model du processus par deux systèmes flous adaptatifs. Les lois d'adaptation sont déduites directement à travers l'étude de la stabilité via la méthode de Lyapunov.

2.5.Commande Floue

2.5.1.Introduction [10, 17,19,22,24]

La commande floue est l'application la plus utilisée de la logique floue.Elle consiste à remplacer les algorithmes de réglage conventionnels par des règles linguistiques du type : SI la voiture qui précède est proche ET que je roule vite ALORS il faut freiner rapidement.

Ainsi on obtient un algorithme linguistique qui se prête mieux que les méthodes traditionnelles à la commande d'un processus.

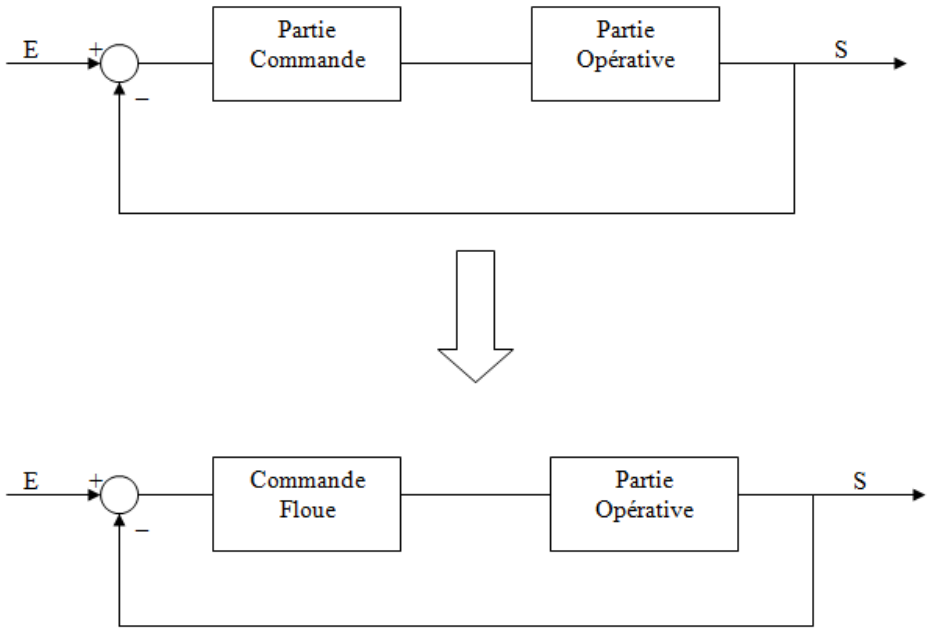


Figure 2.3. algorithme linguistique

2.5.2.Structure d'une Commande Floue

Le schéma général d'une Commande Floue est celui-ci :

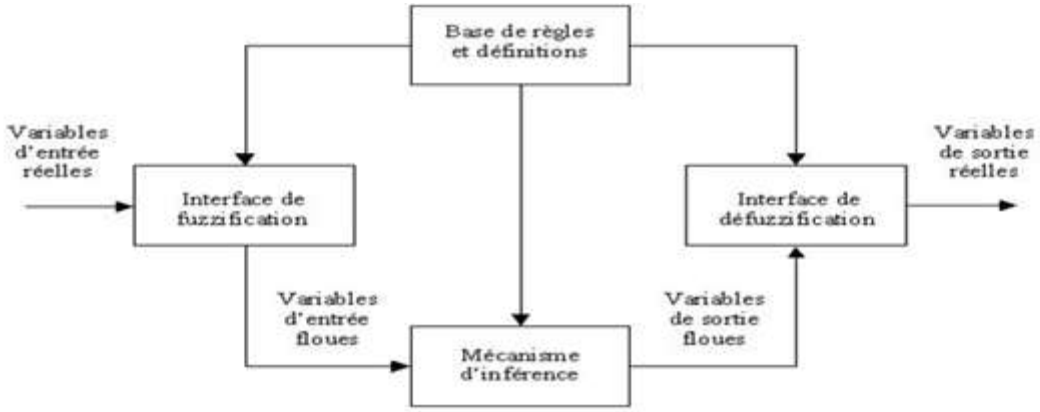


Figure 2.4.système floue

On procède tout d'abord à la partition en sous-ensembles flous des différents univers de discours (ou référentiels) que le système impose. Ensuite on détermine **la base de règles** qui va caractériser le fonctionnement désiré du système.

Puis il faut transformer les variables réelles, c'est à dire celles qui ont une réalité physique, en variables floues. On appelle cette étape **la fuzzification** (de fuzzy=flou en anglais).

On utilise alors ces variables floues dans **unmécanisme d'inférence** qui crée et détermine les variables floues de sortie en utilisant les opérations sur les fonctions d'appartenance.

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

2.6.Exemple de commande d'un système du premier ordre

On considère ici un circuit électrique du premier ordre suivant :

$$\frac{\partial i(t)}{\partial t} + \frac{R}{L} i(t) = \frac{1}{L} v(t) \quad (2.11)$$

tel que son transformer de Laplace est :

$$(P + \frac{R}{L}) I(P) = \frac{1}{L} V(P) \quad (2.12)$$

les conditions initial sont nul.

On à alors :

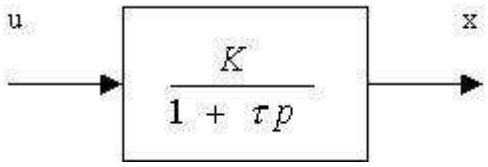
$$\frac{I(P)}{V(p)} = \frac{1/L}{P+R/L} = \frac{1/R}{1+(L/R)P} \quad (2.13)$$

On pose :

$K = 1/R$ Et $\tau = L/R$; remplaçons « i » et « v» par

L'entrée « u » et la sortie « v » respectivement.

On obtient finalement le processus suivant :



Le schéma bloc de la boucle de régulation est :

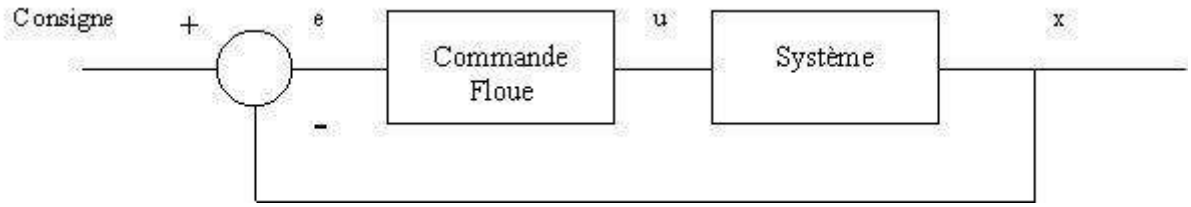


Figure 2.5. Boucle de régulation

On nomme e=Consigne-x l'erreur et Δe=e (t)-e (t-T) où T est une constante de temps fixée par la nature de la boucle de retour et la commande. La commande est Δu. Il faut tout d'abord établir la base de règles:

Règle n°	SI e est	ET Δe est	ALORS Δu est
1	-	-	-
2	-	+	0
3	+	-	0
4	+	+	+

L'étape de fuzzification établie les trois sous-ensembles nécessaires à la commande :

Erreur (e) ; Variation de l'erreur (Δe) ; Commande (Δu)

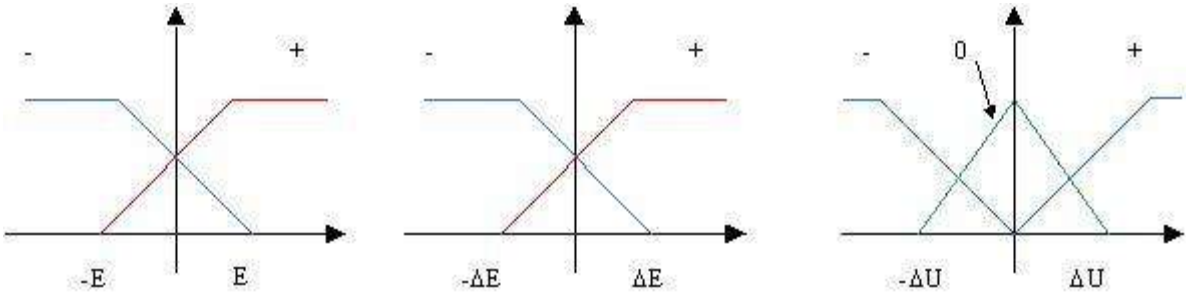


Figure 2.6. Etapes de fuzzification

Pour les mécanismes d'inférence on choisit les opérations de base :

- intersection par min
- union par max
- Fuzzification : singleton
- Défuzzification : barycentrique

Il faut alors procéder à l'élaboration de la fonction d'appartenance de commande en appliquant les règles de base. Pour chaque règle on procède de la même façon :

- On fait apparaître les fonctions d'appartenance concernées par la règle.
- On place les fonctions d'appartenance singletons de e_0 et Δe_0.
- On procède à une suite d'intersections :

Entre e_0 et la fonction d'appartenance de e.

Entre Δe_0 et la fonction d'appartenance de Δe.

Entre les deux valeurs obtenues

Entre la valeur précédentes et la fonction d'appartenance de Δu .

On obtient ainsi une fonction d'appartenance de sortie partielle.

2.7. Applications de la logique et de la commande floue

2.7.1. L'utilisation de la commande floue

La commande floue a montré de plusieurs années sa fiabilité et sa capacité d'adaptation aux problèmes industriels concrets. Les exemples d'utilisation concluante de la logique floue sont très nombreux au Japon, le premier pays à avoir utilisé la logique floue en masse dés les années 88-89. On peut citer une usine d'incinération de déchets à Nishigaya ou l'élaboration par Monsieur Takahashi d'une boite de vitesse automatique entièrement contrôlée par la logique floue.

Tableau .2.1 Domaines d'applications page de Bernard Mantel

Domaine d'applications	Exemples
Automatisme	ateliers de production, automates programmables, pilotes d'avions, caméras
Robotique	commande de robots très divers : aide au mouvement, à la détection des formes
Informatique	systèmes experts, base de données relationnelles, langage de programmation flou
Engineering, Gestion, Prise de décision	contrôle de circulation (aérienne ou automobile), télécommunications, théorie des jeux, gestion de personnel d'investissement
Environnement	Météorologie (prévisions, évaluation des variations climatiques), évaluation des risques sismiques
Médecine	système expert d'aide aux diagnostics comme par exemple le MYCIN en 1974
Mathématiques appliquées	statistique, recherche opérationnelle
Sciences humaines	sociologie, psychologie

2.8. Système de commande adaptative :

Les recherches concernant la commande adaptative ont débuté vers le début des années 1950 et ont été motivées par le besoin de fournir une aide de haute performance pour le pilotage d'avions. Après une première période de relatif succès, les premiers résultats théoriques fondamentaux sont apparus vers 1960 et ont permis au domaine de réaliser de grandes avancées. De plus, les progrès rapides de la micro-électronique ont rendu possible la réalisation de tels contrôleurs de manière simple et peu coûteuse.

Les différentes méthodes de commande adaptative se différencient par la manière choisie pour réaliser l'ajustement des coefficients. On distingue principalement :

> La commande adaptative à paramètres pré-programmés.

> La commande adaptative à modèle de référence.

> La commande adaptative avec identification du système à régler.

2.8.1. Définition de la commande adaptative

Parmi les différentes définitions données, nous présentons celle de Dion (1990) : la commande adaptative ajuste de manière automatique en temps réel, les paramètres de régulateurs afin d'obtenir un certain niveau de performances quand le comportement dynamique du procédé est inconnu, mal connu, et/ou variable dans le temps.

L'asservissement en boucle fermée d'un système par une commande adaptative est formé généralement par deux boucles, une boucle interne normale contenant le régulateur et le processus, une boucle externe ajuste les paramètres du régulateur dans le but de satisfaire un indice de performance choisi.

En commande adaptative, on assume que la vitesse de variation des paramètres de processus à régler est plus faible que la vitesse d'adaptation des variables du régulateur. La boucle d'ajustement est généralement non linéaire.

2.8.2. Objectifs de la commande adaptative

Vu les limites des commandes linéaires, la commande adaptative vient prendre place dans le cas de leurs incapabilités et lors de la dégradation des performances.

La commande adaptative trouve son application dans les cas suivants :

- Système à paramètres inconnus.
- Système à paramètre mal connus.
- Système à paramètre variable.
- Système non linéaire.
- Présence de bruits perturbateurs.

Deux approches principales existent pour la Commande adaptative (des processus à paramètres inconnus ou variables dans le temps) :

- la Commande adaptative directe dont laquelle les paramètres du régulateur sont ajustés directement et en temps réel à partir d'une comparaison entre performance réelle et performance désirée (c'est le cas en particulier de la Commande adaptative à modèle de référence (*Model Reference Adaptative control))* (fig.2.7.1).

- la Commande adaptative indirecte, qui suppose une estimation des paramètres du processus par une procédure d'identification (c'est le cas des régulateur auto-ajustables (*Self-tuning control STC)*) (fig. 2.7.2).

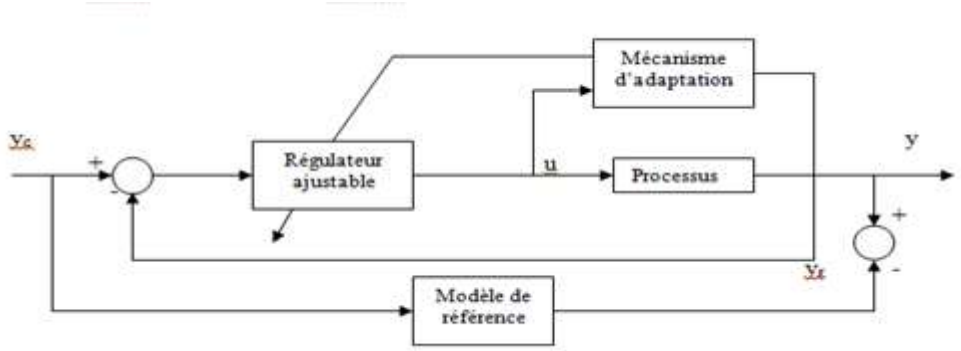


Figure 2.7.1commande adaptative avec modèle de référence explicite (directe)

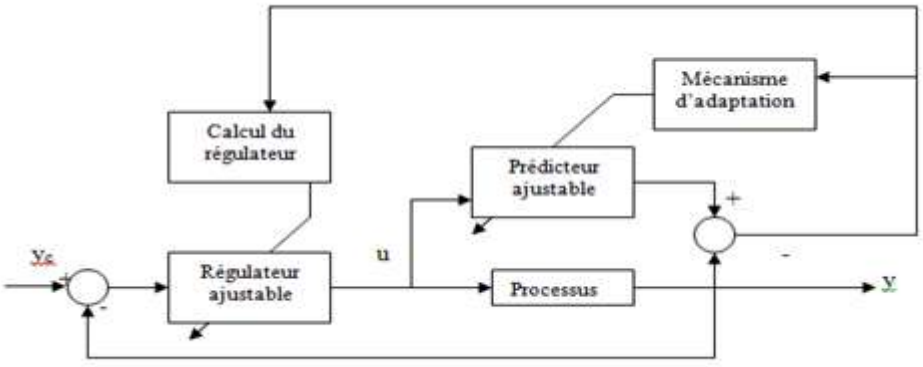


Figure 2.7.2.commande adaptative auto-ajustable (indirect)

2.8.4. CLASSIFICTION DES SYSTEMES DE COMMANDE ADATTIVE

Selon les caractères des systèmes dynamiques, on distingue les systèmes de commande adaptative suivant :

- ❖ A gains pré-programmés (gain-scheduling).
- ❖ A modèle de référence.
- ❖ Avec un régulateur auto-ajustable.

A). Système de commande adaptative à gains pré-programmés

Dans ces systèmes, les adaptations des caractéristiques dynamiques du processus sont déterminées en fonction de l'information à priori disponible sur les conditions externes et internes de fonctionnement du système.

La dépendance programmée des gains est déterminé à l'avance pour un état du système de commande correspondant à un régime optimale calculé, soit pour certaines conditions

externes et internes les plus probables, soit pour toute l'étendue de leurs variations pour laquelle l'information peut être disponible.

La commande adaptative à gains pré-programmés présente les inconvénients suivants :

a) la commande en boucle ouverte qui ne compense pas les effets des perturbations dues aux imprécisions et aux changements des caractéristiques dynamiques du processus.

b) le régulateur à gains pré-programmés est assez coûteux couse de l'étude approfondie des relations liant les mesures des perturbations et les propriétés dynamique du processus. De plus il est nécessite la mise en place des capteurs complémentaires.

Cette approche de commande adaptative à été utilise dans plusieurs applications industrielles, notamment dans l'aéronautique.

B). Commande adaptative à modèle de référence (MRAC)

Pour pallier à certains inconvénients de la commande précédente, Whitaker M.F (en 1985) a proposé un système de commande à modèle de référence (Figure II.2) largement développé par plusieurs spécialistes. De tels systèmes sont composés par deux boucles fermées : une boucle interne (principale) et l'autre externe.

La boucle interne comprend le système à commander et le régulateur dont les paramètres sont ajustés selon un algorithme établi au niveau de la boucle externe de façon que la déférence entre la sortie du modèle et celle du processus soit minimale.

L'étape la plus importante dans la synthèse des systèmes de commande adaptative à modèle de référence est l'élaboration de l'algorithme du mécanisme d'ajustement destiné en fonction de l'indice (critère) de performance adopté.

La commande adaptative MRAC est classifiée comme une commande adaptative directe, car les paramètres du contrôleur sont directement ajustés pour réduire une certaine norme de l'erreur de sortie entre le système et le modèle de référence.

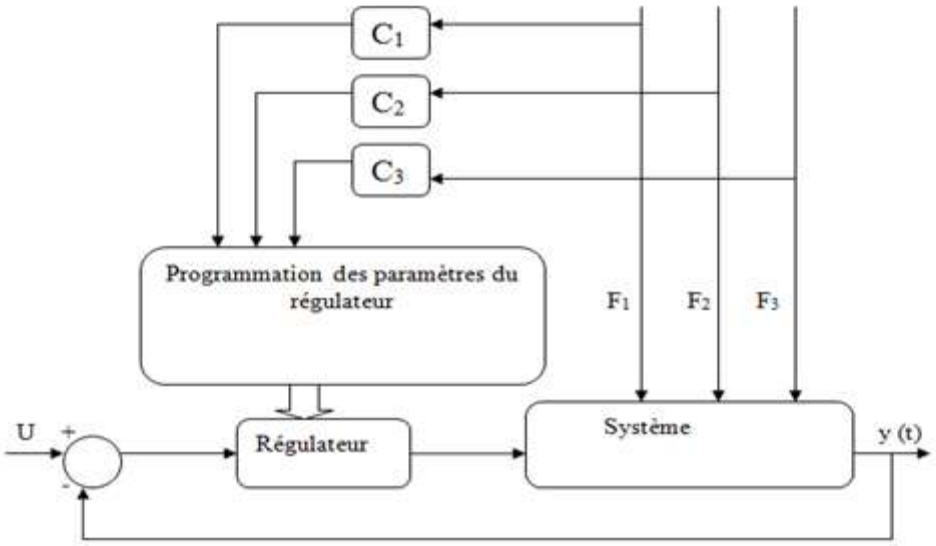


Figure 2.8schéma bloc du système de commande adaptative à gains pré-programmes

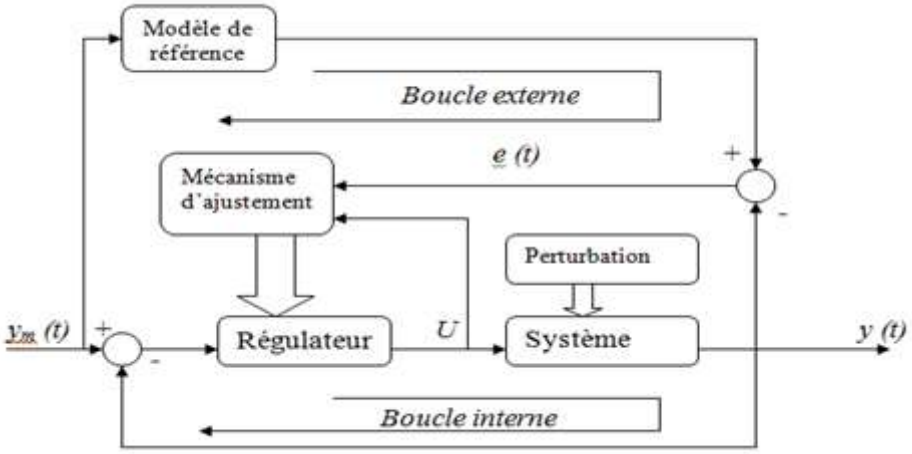


Figure 2.9 schéma de commande adaptative à modèle de référence

Approches pour l'analyse des MRAC :

La construction d'un système adaptatif à modèle de référence MRAC nécessite :

- Un choix d'un modèle de référence qui fixe les performances désirées.
- Un choix de la commande.
- Une détermination de la loi d'adaptation.

Si le choix du modèle de référence et de la loi de commande est lié aux performances désirées, la loi d'adaptation doit assurer la stabilité du système et la tendance de l'erreur vers zéro. Parmi les approches utilisées pour la détermination des loi d'adaptations, on trouve : l'approche du gradient, la fonction de Lyapunov et l'hyperstabilité de Popov.

- **Méthode du gradient**

En utilisant la règle du gradient pour la synthèse du mécanisme d'ajustement. Cette approche et basée sur la minimisation de l'indice de performance sur la base du gradient des paramètres. Cette règle ne nécessite pas la stabilité globale du système, mais il s'agit de trouver un compromis entre la stabilité et la vitesse de réponse lors de simulation.

Règle MIT :

Soit « e » l'erreur entre $y_m$et y (e= y_m- y) et « θ » le vecteur des paramètres à ajuster, un critère à minimiser est propos comme :

$$J(\theta) = \tfrac{1}{2} e^2 \qquad (2.14)$$

Par conséquent pour que *J* soit petit, il est raisonnable de changer les paramètres dans le sens négatif du gradient de *J*.

$$\frac{\partial \theta}{\partial t} = -\gamma \frac{\partial J}{\partial \theta} \quad (2.15)$$

$$\frac{\partial \theta}{\partial t} = -\gamma . e . \frac{\partial e}{\partial \theta} \quad (2.16)$$

S'il est supposé que les paramètres « θ » changent plus lentement que les autres variables du système, alors on peut calculer $\frac{\partial e}{\partial \theta}$ dans ce cas en supposant θ constant.

$\frac{\partial e}{\partial \theta}$ représente la sensibilité du système, γ détermine la vitesse d'adaptation des paramètres. Le schéma de la figure suivante représente le modèle d'erreur.

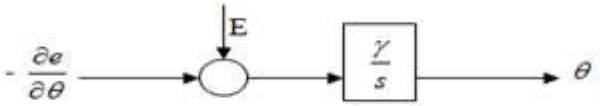


Figure 2.10 modèle d'erreur

Le choix du critère est arbitraire, si on pose $J = |e|$ alors $\frac{\partial e}{\partial \theta} = -\gamma \frac{\partial e}{\partial \theta} \sin g(e)$.

La règle MIT est performante si le gain d'adaptation γ est petit, mais sa valeur peut dépendre de l'amplitude du signal de référence et du gain du processus.Par conséquent, il n'est pas possible de donner à γ des limites fixées qui assurent la stabilité globale du système.

La règle MIT peut alors donner un système en boucle fermée instable. Il est possible d'obtenir des procédures modifiées du gradient dans lesquelles le taux d'ajustement ne dépend pas de

l'amplitude du signal de commande. Une possibilité est de faire une normalisation et de remplacer la règle de MIT par :

$$\frac{\partial e}{\partial t} = -\gamma \frac{e.\frac{\partial e}{\partial \theta}}{\alpha + (\frac{\partial e}{\partial \theta})^T (\frac{\partial e}{\partial \theta})} \tag{2.17}$$

Le paramètre $\alpha>0$ a été introduit pour éviter une division possible par zéro.

Remarquer qu'il est raisonnable d'avoir le taux d'ajustement des paramètres dépendant de l'amplitude de commande pour de petits niveaux à cause du bruit.

Pour garantir que le taux d'ajustement des paramètres soit toujours au dessous de la limite donnée, on utilise la saturation, alors la règle d'ajustement devient :

$$\frac{\partial e}{\partial t} = -\gamma \, \text{Sat}\left[\frac{e.\frac{\partial e}{\partial \theta}}{\alpha + (\frac{\partial e}{\partial \theta})^T (\frac{\partial e}{\partial \theta})}, \beta \right] \tag{2.18}$$

Où :

$$\text{Sat}(x,\beta) = \begin{cases} -\beta & x < -\beta \\ x & |x| \le \beta \\ \beta & x > \beta \end{cases}$$

- **Méthode de Lyapunov :**

D'une manière très générale, la synthèse de lyapunov consiste à sélectionner la candidature d'une fonction positive *V* puis à choisir des lois de commande ou d'adaptation assurant sa décroissance, c'est-à-dire:

$\begin{cases} V(0) = 0 \\ V > 0 \end{cases}$ et $\dot{V} \le 0$,

pour des systèmes continus et $\Delta V(k) = V(k+1) - V(k) \le 0$ pour des systèmes discrets.

Si ces conditions sont vérifiées, la fonction V est dite fonction de Lyapunov et assure la stabilité. De plus, la méthode de Lyapunov peut servir dans l'ensemble à la synthèse de lois de commande stabilisantes.

La synthèse de Lyapunov est une condition suffisante pour la stabilité. La difficulté de cette approche repose sur la détermination d'une fonction de Lyapunov qui conduit, soit à choisir une fonction particulière, soit à essayer de résoudre des inéquations aux dérivées partielles. Dans ce travail, la sélection des fonctions de Lyapunov est faite de manière intuitive. Généralement, ces dernières sont choisies comme des fonctions quadratiques.

On peut expliquer la méthode proposée par la figure suivante :

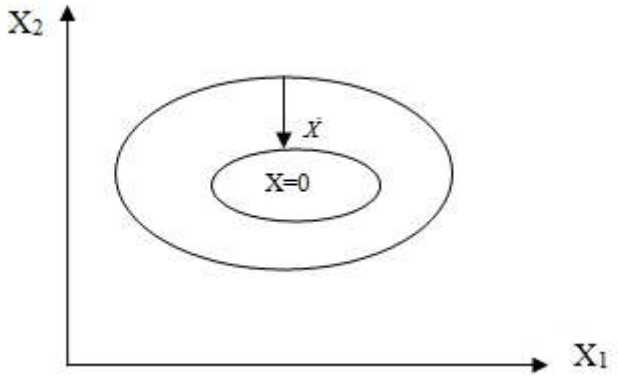


Figure 2.11.plan d'état

Un système est dit stable si l'on trouve une fonction réelle sur l'espace d'état, sa courbe contenant l'état d'équilibre, et la variation de la variable d'état s'achemine vers l'intérieur de cette courbe. [22,19,24]

- Théorème de la stabilité de Lyapunov

Considérons un système décrit par l'équation différentielle suivant :

$$\dot{x} = f(x),\ f(0) = 0 \quad (2.19)$$

Où : x représente le vecteur d'état de dimension n.

Soit une fonction V ; $R^{n+1} \longrightarrow R$ satisfaisant les conditions suivantes :

- V(0)=0
- V est différentiable en x.
- V est définie positive, c'est-à-dire $V(0) > 0$.

Pour une stabilité asymptotique uniforme du système (2.19), la condition suffisante est que :

$$\dot{V}(x) = \frac{\partial v^T}{\partial x} f(x) < 0, \text{pour } x \neq 0.$$

Cette condition permet d'avoir des lois d'adaptation assurant la stabilité et la tendance de l'erreur vers zéro.

- **Méthode d'hyper-stabilité**

L'objectif de l'approche basé sur les concepts d'hyperstabilité et de positivité consiste à assurer l'annulation de l'erreur tout an garantissant la stabilité du système, en dépit des incertitudes paramétriques et de perturbation externe. Ainsi, le problème consiste à déterminer un mécanisme d'adaptation qui assure à la fois la stabilité et la poursuite du système à commander ‘’modèle de référence. Pour la synthèse de ce mécanisme, il faut connaître le

modèle du système à commander. Cependant, quelque soit le modèle, il faut respecter certains règles pour que la structure du mécanisme d'adaptation soit réalisable. Par exemple, en ce qui concerne le signale de commande, il ne doit pas atteindre des valeurs excessives conduisant à un fonctionnement non linéaire du système.

C). Commande adaptative à régulateur auto-ajustable

Ce type de commande adaptative consiste à insérer des régulateurs auto-ajustables dans la chaîne d'action de commande, proposés initialement par Kalman. La conception des systèmes et basée sur le principe d'équivalence certaine qui consiste à séparer les problèmes d'identification et de commande.[24]

La commande adaptative à régulateur auto-ajustable c'est une commande indirecte, où les paramètres du système sont estimés et le contrôleur est choisi de façon que ces paramètres représentent les vraies valeurs des paramètres du système.

Cette dernière est calculée en supposant que les paramètres du système à régler sont connus.

Pour réaliser cette commande, on distingue trois étapes :

a) Estimation des paramètres du processus ;
b) Calcul des paramètres du régulateur auto-ajustable ;
c) Commande selon une loi linéaire dans la boucle interne.

Le schéma bloc du système de commande adaptative avec régulateur auto-ajustable est montré par la figure (II.8). Ce système se compose de deux boucles interne et externe, la première est constituée du système à régler et du régulateur auto-ajustable de type linéaire. La boucle externe comprend un identificateur (estimateur) des paramètres du système à régler qui permet de calculer en même temps la grandeur de prédiction $y_p(t)$, de la grandeur à régler y(t) et les paramètres du régulateur auto-ajustable.

Parmi les méthodes d'estimation des paramètres, on trouve :

- Les approximations stochastiques
- Les moindres carrés
- Les moindres carrés étendus et généralisés.

Parmi les méthodes d'ajustement des paramètres, on trouve :

- La variance minimale
- Le placement des pôles
- Le suivi de modèle

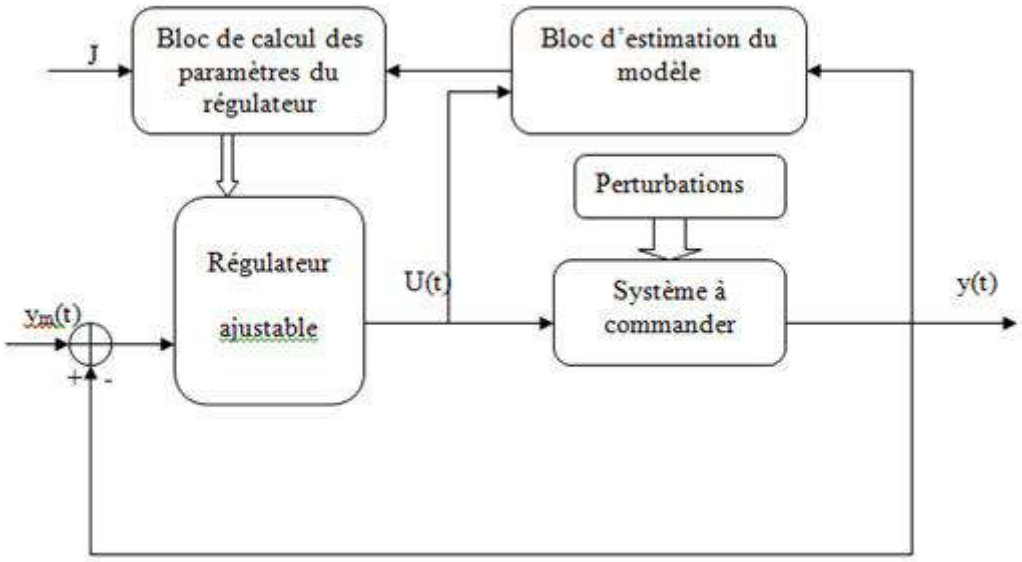


Figure 2.12.schéma bloc du système de commande adaptative avec régulateur auto ajustable

2.9 Conclusion

Les systèmes flous ont connus un succès considérable dans la command des procédés industrielles complexes, du fait de leurs caractère approximatif et qualitatif inspirés de la pensé humaine. Leurs performances sont liées à deux facteurs importants :

1- la disponibilité de l'expertise

2- la validité de techniques d'acquisition de connaissances et la justesse des données acquises.

La logique floue est une théorie très puissante qui permet d'obtenir des conclusion et de générer des réponses à partir des informations incomplètes et imprécises, la ou le monde mathématique du système est inconnu ou difficile à extraire.

L'objectif est de concevoir ou de réaliser des systèmes qui peuvent prendre en charge certaines taches avec la même habilité que possède l'être humain. La phase d'acquisition des connaissances est la plus difficile dans des cas ou le domaine d'expertise n'est pas disponible.

Pour cette raison, des recherches très poussées ont conduit au développement des méthodes systématiques et optimales pour la conception des contrôleurs flous.

Parmi ces méthodes on trouve :

Les réseaux de neurones qui sont très puissants dans le domaine de l'apprentissage et de l'optimisation, et cela en utilisant les fonctions qu'on veut optimiser comme poids a ajusté.

Lors de la dégradation des performances et dans le cas de l'incapabilités des commandes linéaires, la commande adaptative vient pour prendre la place et commence par gérer les systèmes inconnus, mal connus ou même les systèmes à paramètre variable et finalement les systèmes non linéaires qui permet le continûment du développent de ce domaine. Mais la commande adaptative ne traite pas les données requise par l'expert, pour cela nous allons la fusionné avec la logique floue pour que nous bénéficions de l'expérience et les remarques d'expert, la fusion nous donne ce que n'appelons « La commande adaptative floue et nous le traitent dans le prochain chapitre.

Chapitre 3
Commande hybride adaptative floue avec superviseur

Chapitre 3

Commande hybride adaptative floue avec superviseur

3.1 Introduction

Les systèmes flous en plus de leur efficacité dans les problèmes de commande, sont capables d'approximer n'importe quelle fonction continue non linéaire sur un ensemble compact comme il a été prouvé dans [22], [5], [16] Basés sur cette propriété importante, plusieurs schémas de commande adaptative floue indirecte (CAFI) ont été proposés pour les systèmes non linéaires inconnus ou mal connus [22], [5]. La supposition clé dans ces schémas de CAF est que la totalité de l'état est mesurable. La conception de telle commande quand il n'y a aucune connaissance à priori sur le signe de gain de commande, devient un vrai défi, comme expliqué dans [18], et généralement impossible via l'inversion si le système est non affines même si les non-linéarités du système sont parfaitement connues.

Dans ce chapitre, deux schémas de CAFI sont proposées. Les systèmes flous sont utilisés pour estimer en ligne la dynamique du système et la loi de commande est synthétisée en se basant sur ces estimées. Le théorème de la valeur moyenne est appliqué sur les systèmes non affines et les transformer en systèmes affines. Les lois de commande et les lois d'adaptation sont obtenues en se basant sur des fonctions de Lyapunov candidates.

Le premier schémade CAF proposé, est développé pour une classe des systèmes SISO non linéaires inconnus avec un signe de gain de commande connu.

Le second schéma de commande proposé, est développé pour une classe de systèmes SISO non linéaires inconnus avec un signe gain de commande inconnu. Une fonction de Nussbaum est utilisée pour estimer le signe de gain de commande [10].

Pour toutes les deux approches de commande développées, des résultats de simulation effectuée sur un pendule inversé qui est un système affine en la commande et sur un système non affine académique sont données pour montrer les performances de ces approches.

3.2 Commande adaptative floue avec une direction de commande connue

3.2.1Position du problème

Considéronsle système $n^{ème}$ ordre de la forme suivante

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_n &= f(x_1, x_2, ..., x_n) + g(x_1, x_2, ..., x_n)u \\
y &= x_1
\end{aligned} \tag{3.1}$$

où de façon équivalente, de la forme

$$\begin{aligned}
x^{(n)} &= f(x, \dot{x}, ..., x^{n-1}) + g(x, \dot{x}, ..., x^{n-1})u, \\
y &= x.
\end{aligned} \tag{3.2}$$

où, $\underline{x} = \left[x_1, x_2, ..., x_n\right]^T = \left[x, \dot{x}, ..., x^{(n-1)}\right]^T \in R^n$ est le vecteur d'état de ce système, $u \in R$ et $y \in R$ sont l'entrée et la sortie du système, respectivement, $f(\underline{x})$ et $g(\underline{x})$ sont des fonctions non linéaires continuesincertaines.

Les systèmes sous la forme (2.2) sont à la forme normale, affine en la commande,qui est un cas particulier des systèmes non affinedonné par la forme :

$$\begin{cases}
x^{(n)} = f(\underline{x}, u), \\
y = x,
\end{cases} \tag{3.3}$$

ou l'entrée du système u n'apparait pas linéairement. En général, même si la fonction non affine $f(\underline{x}, u)$ est complètement connue, la conception d'une commande par un bouclage linéarisant est loin d'être facile. Quand la structure de $f(\underline{x}, u)$ est inconnue, la construction d'une telle commande devient encore plus difficile voire impossible [10].

Afin de surmonter ce problème, et d'après, le théorème de la valeur moyenne donné dans *la section 1.4.1* est utilisé pour transformer le systèmenon affine (1.3)en système affine de forme (2.2),avec $f(\underline{x}, u) = f(\underline{x}) + g(\underline{x}, u^*)u$ .Ce que rend par la suite l'analyse et la conception de lacommande possible.

Dans la littérature, il existe d'autres méthodes permettant de traiter le problème des systèmes non linéaires et non affines :

- ✓ L'utilisation du développement en série de Taylor afin d'obtenir une forme affine en la commande [2].
- ✓ L'utilisation du théorème des fonctions implicites [14,18].

L'objectif de la commande est de forcer y de suivre un signal de référence borné $y_r(t)$, sous la contrainte quetous les signauximpliquésdoiventêtrebornés.

Comme dans [5,17], les hypothèses suivantes sont considérées.

Hypothèse 3.1 : Le gain de commande $g(\underline{x})$ est différent de zéro pour tout x et de signe connu. Sans perte de généralité, il est supposé que $g(\underline{x})>0$ et $g(\underline{x})=\dfrac{f(\underline{x},u)}{u}>0$ pour les systèmes non affines.

Hypothèse 3.2 : Le vecteur d'état $\underline{x}$ est mesurable.

Hypothèse 3.3 : La trajectoire désirée $y_r(t)$ et ses dérivées jusqu'à l'ordre n sontconnues et bornées.

Remarque 3.1: L'hypothèse 2.1 peut être supposé dans le cas du système non affine, ou $g(\underline{x})=\dfrac{\partial f(\underline{x},u)}{\partial u}$.

Soit $\underline{e}=\left[e,\dot{e},.....,e^{(n-1)}\right]$ et $k=\left[k_n,.....,k_1\right]^T\in R^n$ telles que toutes les racines du polynôme $h(S)=S^n+k_1S^{n-1}+.....+k_n$ sont dans le demi-plan gauche.

Si les fonctions $f(\underline{x})$ et $g(\underline{x})$ sont connues, alors la loi de commande est,

$$u=\frac{1}{g(\underline{x})}\left[-f(\underline{x})+y_r^{(n)}+k^Te\right] \qquad (3.4)$$

La commande (3.4) appliqué à (2.2),résulte que

$$e^{(n)}+k_1e^{(n-1)}+.....+k_ne=0 \qquad (3.5)$$

Ce qui donne, $\lim_{t\to 0}e(t)=0$, qui est l'objectif principal de la commande.

La loi de commande idéale (2.4) du système (2.1) peut être facilement implantée si $f(\underline{x})$ et $g(\underline{x})$ sont parfaitement connues. Ces fonctions étant généralement incertaines, le but estalors de les approcher par des systèmes flous.Dans le reste de ce chapitre 3, un contrôleur adaptatif floua été proposé pour réaliser les objectifs de la commande.

3.2.2 Conception de commande hybride adaptative floue

Puisque $f(\underline{x})$ et $g(\underline{x})$ sont inconnues, les systèmes flous sont utilisés pour les approcher.A partir des approximations floues, des lois de commande sont considérés avec leurs lois d'adaptation pour atteindre les objectifsde commande et garantir la bornitude de tous les signaux du système bouclé.

Les systèmes flous de la forme de (1.20) sont utilisés pour identifier en ligne lesfonctions $f(\underline{x})$ et $g(\underline{x})$. Ces systèmes sont définis comme suit :

$$\tilde{f}(\underline{x}, \theta_f) = \theta_f^T \xi(\underline{x}) \qquad (3.6)$$

$$\hat{g}(\underline{x}, \theta_g) = \theta_g^T \xi(\underline{x}) \qquad (3.7)$$

Avec $\xi(\underline{x})$ estun vecteur de fonctions floues de base supposé convenablement fixé en avance par l'utilisateur, θ_f et θ_g sont les vecteurs de paramètresajustés.

La loi de commande résultante est

$$u_c = \frac{1}{\hat{g}(\underline{x}, \theta_g)}\left[-\hat{f}(\underline{x}, \theta_f) + y_r^{(n)} + k^T e\right] \qquad (3.8)$$

en appliquant (2.8) à (2.2), l'équation de l'erreur suivante est obtenue,

$$e^{(n)} = -k^T e + \left[\hat{f}(\underline{x}, \theta_f) - f(\underline{x})\right] + \left[\hat{g}(\underline{x}, \theta_g) - g(\underline{x})\right] u_c \qquad (3.9)$$

qui est équivalente à,

$$\dot{e} = A_c e + B_c\left[\left(\hat{f}(\underline{x}, \theta_f) - f(\underline{x})\right) + \left(\hat{g}(\underline{x}, \theta_g) - g(\underline{x})\right) u_c\right] \qquad (3.10)$$

où

$$A_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & \cdots & \cdots & -k_1 \end{bmatrix}, \qquad B_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \qquad (3.11)$$

A_c est une matrice stable ($\left|SI - A_c\right| = S^{(n)} + k_1 S^{(n-1)} + ... + k_n$ est stable), alors, il existe une matrice unique symétrique définie positive $P(n \times n)$ qui satisfait l'équation de Lyapunov[17].

$$A_c^T P + P A_c = -Q \qquad (3.12)$$

Où Q est une matrice symétrique arbitraire définie positive de dimensions $(n \times n)$.

Soitla fonction de Lyapunov suivante,

$$V_e = \frac{1}{2} e^T P e \qquad (3.13)$$

La dérivée de la fonction de Lyapunov(3.13) est :

$$\dot{V}_e = \frac{1}{2} \dot{e}^T P e + \frac{1}{2} e^T P \dot{e} \qquad (3.14)$$

En utilisant (2.10) et (2.12)dans (2.14)

$$\dot{V}_e = \frac{1}{2} e^T Q e + e^T P B_c \left[\left(\hat{f}(\underline{x}, \theta_f) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}, \theta_g) - g(\underline{x}) \right) u_c \right] \qquad (3.15)$$

Pour que $x_i = y_r^{(i-1)} - e^{(i-1)}$ soit borné, V_e doit être limité, ce qui signifie que $\dot{V}_e \leq 0$ lorsque V_e est supérieur à une constante$\bar{V}$. Cependant, de (2.15)il est très difficile de concevoir u_c tel que le dernier terme de (2.15) soit négatif. Pour résoudre ce problème, un autre terme de commande u_s a été rajoute à u_c [18].

La commande finale sera,

$$u = u_c + u_s \tag{3.16}$$

leterme de commande u_s est appelé la commande de superviseurdéterminé de tel sorte que $\dot{V}_e \le 0 \text{quand} V_e > \bar{V}$.

En remplaçant (2.16) dans (2.2), la nouvelle équation de l'erreur est,

$$\dot{e} = A_c e + B_c \left[\left(\hat{f}(\underline{x}, \theta_f) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}, \theta_g) - g(\underline{x}) \right) u_c - g(\underline{x}) u_s \right] \tag{3.17}$$

Utilisant (2.17) et (2.12) dans (2.14), il est obtenu que

$$\dot{V}_e = -\frac{1}{2} e^T Q e + e^T P B_c \left[\left(\hat{f}(\underline{x}, \theta_f) - f(\underline{x}) \right) + \left(\hat{g}(\underline{x}, \theta_g) - g(\underline{x}) \right) u_c - g(\underline{x}) u_s \right]$$

$$\le -\frac{1}{2} e^T Q e + \left| e^T P b_c \right| \left[\left(\left| \hat{f}(\underline{x}, \theta_f) \right| + \left| f(\underline{x}) \right| \right) + \left| \hat{g}(\underline{x}, \theta_g) u_c \right| + \left| g(\underline{x}) u_c \right| \right] - e^T P B_c g(\underline{x}) u_s \tag{3.18}$$

Pour concevoir u_s tel que le dernier terme de (2.18) soit négatif,il faut connaître les limites des fonctions f et g , alors, l'hypothèse suivante est considérée:

Hypothèse 3.4 : $f^U(\underline{x})$, $g^U(\underline{x})$ et $g_L(\underline{x})$ sont des fonctions déterminées de telle sorte que $\left| f(\underline{x}) \right| \le f^U(\underline{x})$ et $g_L(\underline{x}) \le g(\underline{x}) \le g^U(\underline{x})$ pour $x \in U_c$,où $f^U(\underline{x}) < \infty$, $g^U(\underline{x}) < \infty$ et $g_L(\underline{x}) > 0$ pour $x \in U_c$.

de (3.18) et basé sur les bornes $f^U(\underline{x})$, $g^U(\underline{x})$ et $g_L(\underline{x})$, la commande de supervision u_s est choisiecomme suit :

$$u_s = I_1^* sign(e^T P b_c) \frac{1}{g_L(\underline{x})} \left[\left| \hat{f}(\underline{x}, \theta_f) \right| + f^U(\underline{x}) + \left| \hat{g}(\underline{x}, \theta_g) u_c \right| + \left| g^U(\underline{x}) u_c \right| \right] \tag{3.19}$$

où

$$\begin{cases} I_1^* = 1 & si \quad V_e > \bar{V} \\ I_1^* = 0 & si \quad V_e \le \bar{V} \end{cases}$$

et

$$\begin{cases} sign(y) = 1 & si \quad y > 0 \\ sign(y) = -1 & si \quad y < 0 \end{cases}$$

avec$\bar{V}$ constant indiqué par le concepteur,

En remplaçant (3.19) dans (3.18), aveclecasoù$V_e > \bar{V}$, l'inégalité suivante est obtenue

$$\dot{V_e} \le \frac{1}{2} e^T Q e + \left| e^T P b_c \right| \left[\left| \hat{f} \right| + |f| + |\hat{g} u_c| + |g u_c| - \frac{g}{g_L} \left(\left| \hat{f} \right| + \left| f^U \right| + |\hat{g} u_c| + \left| g^U u_c \right| \right) \right]$$

$$\le -\frac{1}{2} e^T Q e \le 0 \qquad (3.20)$$

Alors, en utilisant la commande (3.16) avec u_c donnée en (3.8) et u_s donnée en (3.19), $V_e \le \bar{V} < \infty$est garanti.

Puisque P est définie positive, la borgnitude de V_e implique la borgnitude de *e* qui à son tour implique que x est bornée.

La tâche suivante, consiste à développer une loi d'adaptation pour ajuster les paramètres θ_f et θ_g dans le but de forcer l'erreur de suivi de convergé vers zéro.

Soient les paramètres optimaux suivants,

$$\theta_f^* = \arg\min_{\theta_f \in \Omega_f} \left[\sup_{x \in U_c} \left| \hat{f}(\underline{x}, \theta_f) - f(\underline{x}) \right| \right] \qquad (3.21)$$

$$\theta_g^* = \arg\min_{\theta_g \in \Omega_g} \left[\sup_{x \in U_c} \left| \hat{g}(\underline{x}, \theta_g) - g(\underline{x}) \right| \right] \qquad (3.22)$$

OùΩ_f et Ω_g sont des ensembles de contraintes pourθ_f etθ_g respectivement, spécifiés par le concepteur.

Pour Ω_f , il est exigé que θ_f soit borné comme suivant :

$$\Omega_f = \{\theta_f : |\theta_f| \leq M_f\} \tag{3.23}$$

Pour Ω_g , en plus des contrainte semblable à (2.23), il est exigé aussi que $\hat{g}(\underline{x}, \theta_g)$ soit positive (car $g(\underline{x})$ est positive) :

$$\Omega_g = \{\theta_g : |\theta_g| \leq M_g\} \tag{3.24}$$

Où M_g et M_f sont des constantes positives spécifiées par le concepteur.

L'erreur d'approximation minimale est définiecomme suit,

$$\omega = \left(\hat{f}(\underline{x}, \theta_f^*) - f(\underline{x})\right) + \left(\hat{g}(\underline{x}, \theta_g^*) - g(\underline{x})\right) u_c \tag{3.25}$$

L'équation de l'erreur (2.16) peut être réécrite comme,

$$\dot{e} = A_c e - B_c g(\underline{x}) u_s + B_c \left[\left(\hat{f}(\underline{x}, \theta_f) - \hat{f}(\underline{x}, \theta_f^*)\right) + \left(\hat{g}(\underline{x}, \theta_g) - \hat{g}(\underline{x}, \theta_g^*)\right) u_c + \omega\right] \tag{3.26}$$

Si $\hat{f}$ et $\hat{g}$ sont remplacés par(3.6) et (3.7), respectivement, (3.26) peut être réécrite de la façon suivante :

$$\dot{e} = A_c e - B_c g(\underline{x}) u_s + B_c \omega + B_c \left[\phi_f^T \xi(\underline{x}) + \phi_g^T \xi(\underline{x}) u_c\right] \tag{3.27}$$

avec, $\phi_f = \theta_f - \theta_f^*$, $\phi_g = \theta_g - \theta_g^*$ et $\xi(\underline{x})$ est la fonction floue de base.

Soit la fonction candidate de Lyapunov suivante :

$$V = \frac{1}{2} e^T P e + \frac{1}{2\gamma_1} \phi_f^T \phi_f + \frac{1}{2\gamma_2} \phi_g^T \phi_g \tag{3.28}$$

Où, γ_1 et γ_2 sont des constantes positives.

En utilisant (3.27) et (3.12), la dérivée de V par rapport au temps est:

$$\dot{V} = -\frac{1}{2}e^T Qe - g(\underline{x})e^T PB_c u_s + e^T PB_c \omega + \frac{1}{\gamma_1}\phi_f^T \left(\dot{\theta}_f + \gamma_1 e^T PB_c \xi(\underline{x})\right)$$
$$+\frac{1}{\gamma_2}\phi_g^T \left(\dot{\theta}_g + \gamma_2 e^T PB_c \xi(\underline{x})u_c\right) \qquad (3.29)$$

Où $\dot{\phi}_f = \dot{\theta}_f$, $\dot{\phi}_g = \dot{\theta}_g$.

De (3.19) et $g(\underline{x}) > 0$, implique que $g(\underline{x})e^T PB_c u_s \geq 0$, les lois d'adaptation sont choisi comme suite :

$$\dot{\theta}_f = -\gamma_1 e^T PB_c \xi(\underline{x}) \qquad (3.30)$$

$$\dot{\theta}_g = -\gamma_2 e^T PB_c \xi(\underline{x})u_c \qquad (3.31)$$

De (2.29) il est obtenu que,

$$\dot{V} \leq -\frac{1}{2}e^T Qe + e^T PB_c \omega \qquad (3.32)$$

Le terme $e^T PB_c \omega$ est de l'ordre de l'erreur minimale de l'approximation. Puisque les systèmes flous sont des approximations universelles, ω est petit, sinon égale à zéro. Ce qui permet de conclure que la loi de commande synthétisée garantit la stabilité du système au sens de Lyapunov ($\dot{V} \leq 0$) .

Le problème final est de savoir comment contraindre θ_f et θ_g dans les ensembles Ω_f (2.23) et Ω_g (2.24) respectivement. Si $\theta_f \in \Omega_f$ et $\theta_g \in \Omega_g$ sont gardés, alors u_c (2.8) et u_s (2.19) seront bornées parce que dans ce cas $\hat{f}$ est bornée et $\hat{g} > 0$, et rappelant que e est bornée grâce à la commande de supervision u_s .

Il est clair que les lois d'adaptation (2.30) et (2.31) ne peuvent pas garantir que $\theta_f \in \Omega_f$ et $\theta_g \in \Omega_g$, pour résoudre ce problème, l'algorithme de la projection de paramètres est utilisée [4], [9].

Les lois d'adaptation modifiées sont données comme suit :

- pour ajuster le vecteur de paramètres θ_f :

$$\dot{\theta}_f = \begin{cases} -\gamma_1 e^T PB_c \xi(\underline{x}) & si \ \left(|\theta_f| < M_f\right), \\ & ou \ \left(|\theta_f| = M_f \ et \ e^T PB_c \theta_f^T \xi(\underline{x}) \geq 0\right) \\ P\left\{-\gamma_1 e^T PB_c \xi(\underline{x})\right\} & si \ \left(|\theta_f| = M_f \ et \ e^T PB_c \theta_f^T \xi(\underline{x}) < 0\right) \end{cases} \quad (3.33)$$

où l'opérateur de projection $P\{*\}$ est défini comme [4], [9],

$$P\left\{-\gamma_1 e^T PB_c \xi(\underline{x})\right\} = -\gamma_1 e^T PB_c \xi(\underline{x}) + \gamma_1 e^T PB_c \frac{\theta_f \theta_f^T \xi(\underline{x})}{|\theta_f|^2} \quad (3.34)$$

- pour ajuster le vecteur de paramètres θ_g :
  - chaque fois qu'un élément θ_{g_i} de $\theta_g = \varepsilon$, utiliser :

$$\dot{\theta}_{g_i} = \begin{cases} -\gamma_2 e^T PB_c \xi_i(\underline{x}) u_c & si \ \ e^T PB_c \xi_i(\underline{x}) u_c < 0 \\ 0 & si \ \ e^T PB_c \xi_i(\underline{x}) u_c \geq 0 \end{cases} \quad (3.35)$$

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

  - Sinon, utiliser

$$\dot{\theta}_g = \begin{cases} -\gamma_2 e^T PB_c \xi(\underline{x}) u_c & si \ \left(|\theta_g| < M_g\right) \\ & ou \ \left(|\theta_g| = M_g \ et \ e^T PB_c \theta_g^T \xi(\underline{x}) u_c \geq 0\right) \\ P\left\{-\gamma_2 e^T PB_c \xi(\underline{x}) u_c\right\} & si \ \left(|\theta_g| = M_g \ \ et \ e^T PB_c \theta_g^T \xi(\underline{x}) u_c < 0\right) \end{cases} \quad (3.36)$$

où,

$$P\left\{-\gamma_2 e^T PB_c \xi(\underline{x}) u_c\right\} = -\gamma_2 e^T PB_c \xi(\underline{x}) u_c + \gamma_2 e^T PB_c \frac{\theta_g \theta_g^T \xi(\underline{x})}{|\theta_g|^2} u_c \quad (3.37)$$

Le schéma de la commande proposée est donné par la **Figure3.1**

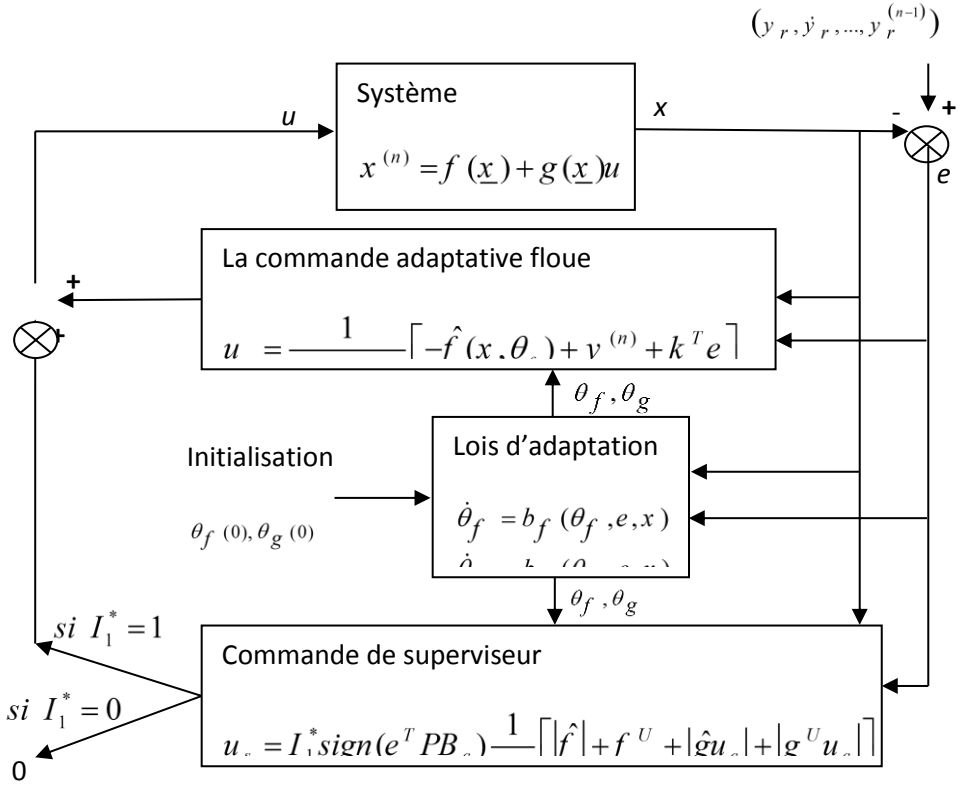


Figure 3.1. *Schéma général de commande adaptative floue.*

3.2.3 Résultats de simulation

Pour illustrer les performances de cette approche de commande proposée, nousconsidérons la commande en poursuite d'un système non linéaire du pendule inversé.

3.2.3.1 Exemple d'un pendule inversé :

Soit $x_1 = \theta$ est la position angulaire du pendule inversé et $x_2 = \dot{\theta}$ sa vitesse angulaire. La figure.2.2 représente le système de pendule inversé .Son modèle dynamique est donné par [05]:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{mlx_2 \sin x_1 \cos x_1 - (M+m) g_a \sin x_1}{ml \cos^2 x_1 - \dfrac{4}{3} l (M+m)} + \dfrac{-\cos x_1}{ml \cos^2 x_1 - \dfrac{4}{3} l (M+m)} u \\ y = x_1 \end{cases} \quad (3.38)$$

où g_a est l'accélération due à la pesanteur, M est la masse du chariot, m est la masse du pendule, l est la demi-longueur du pendule et u est la force appliquée.

Les valeurs numériques de pendule sont choisissons comme suite: $M = 1kg$, $m = 0.1kg$, $l = 0.5m$ et $g_a = 9.8 m/s^2$.

L'équation (2.38) peut être écrite comme suit:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (f(x) + g(x)u + d(t)),$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.39)$$

où

$$f(\underline{x}) = \frac{mlx_2 \sin x_1 \cos x_1 - (M+m) g_a \sin x_1}{ml \cos^2 x_1 - \frac{4}{3} l (M+m)},$$
$$g(\underline{x}) = \frac{-\cos x_1}{ml \cos^2 x_1 - \frac{4}{3} l (M+m)}.$$

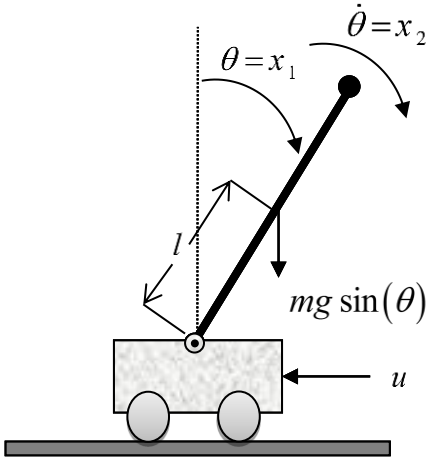


Figure 3.2.*Système du pendule inversé.*

L'objectif de la commande est de forcer la sortie du système y à suivre la trajectoire $y_r = \frac{\pi}{30}\sin(t)$ (autres choix sont possibles).Les fonctions $f(\underline{x})$ et $g(\underline{x})$ sont totalement inconnues par le contrôleur.

Le vecteur de gain par retour d'état est choisi comme suit : $k = [2,1]$ (de sorte que $s^2 + k_1 s + k_2$ soit stable) et $Q = diag(10,10)$, alors la solution de l'équation de Lyapunov est $P = [15\ 5; 5\ 5]$.

Les limitesdes fonctions $f(\underline{x})$ et $g(\underline{x})$ sont : $f^U(\underline{x}) = 15.78 + 0.036x_2^2$, $g^U(\underline{x}) = 1.46$, $g_L(\underline{x}) = 1.12$.

Chacune des fonctions $f(\underline{x})$ et $g(\underline{x})$ est représentée par un système flou, et chaque système flou a pour entrer x_1 et x_2, bien que $g(\underline{x})$ est fonction seulement de x_1, le système flou est utilisé avec les entrée x_1 et x_2 car la forme de la fonction est inconnue.Pour chaque variable d'entrée, on définit cinq fonctions d'appartenance de type gaussienne uniformément distribuée sur les intervalles $[-\pi/6, \pi/6]$,

$\mu_{F_i^l}(x_i) = \exp\left[-\left(\frac{x_i - \bar{x}}{\pi/24}\right)^2\right]$, $i = 1,2$, $l = 1,..,5$, $\bar{x}$ choisi de l'intervalle $[-\pi/6, \pi/6]$.

D'où, chaque système flou a une base de règle de 25 règles.

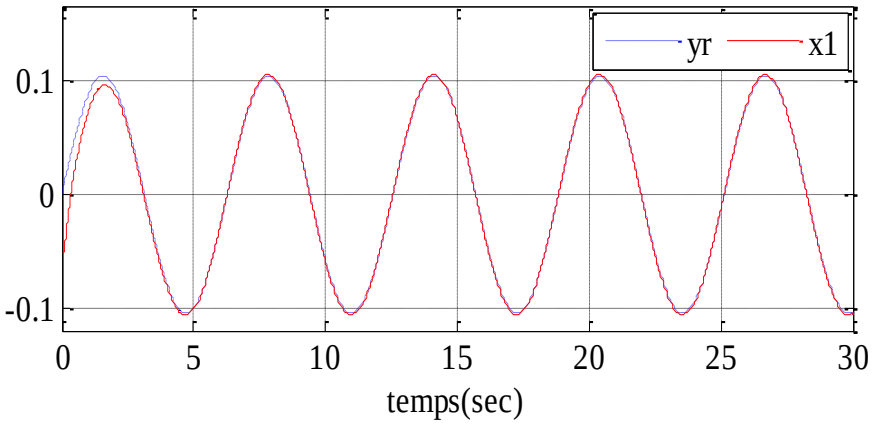
Les conditions initiales sont choisies comme suit :

$\underline{x}(0) = [x_1(0), x_2(0)]^T = [-\pi/60, 0]^T$ et $\theta_i(0) = 0$.

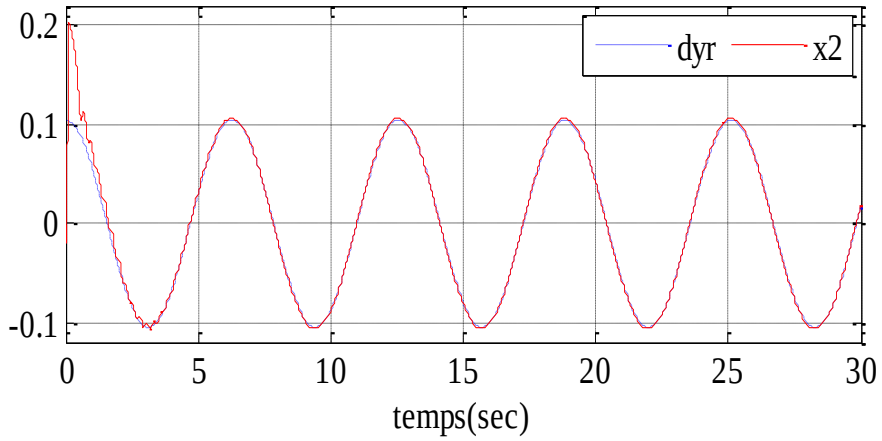
Les Figures 3.3-3.4illustrent respectivement les résultats de simulation de la commandeadaptative floue

- Sans perturbation *d(t)*=0.

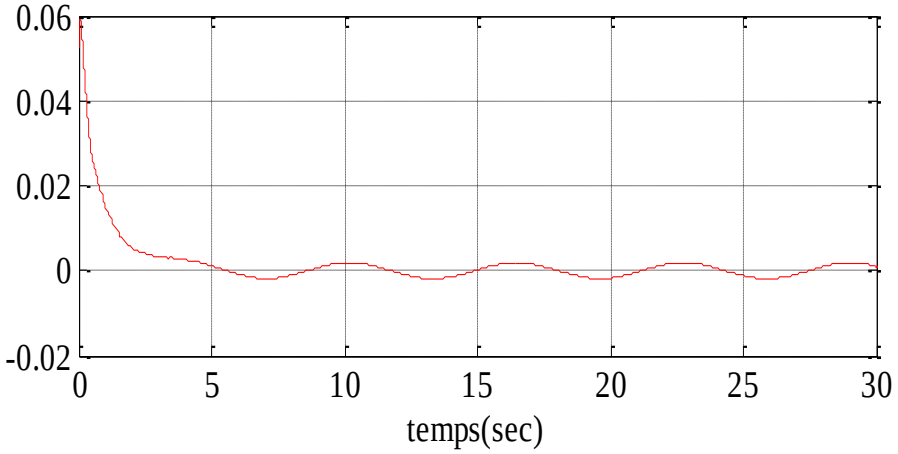
(a) la sortie du système yr et x1



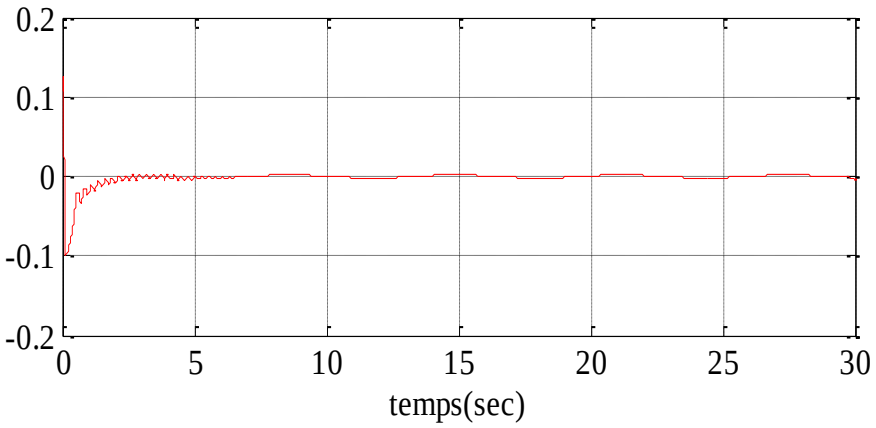
(b) Sortie du système dyr et x2



(c) Erreur de suivi e_1



(d) Erreur de suivi e_2



(e) Signal de commande

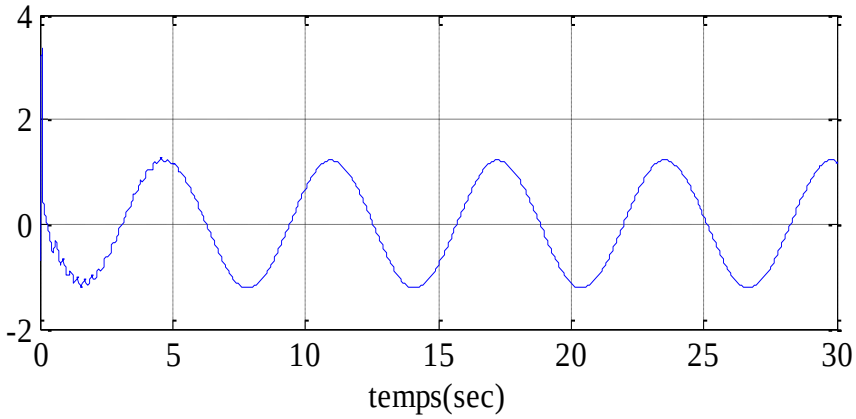
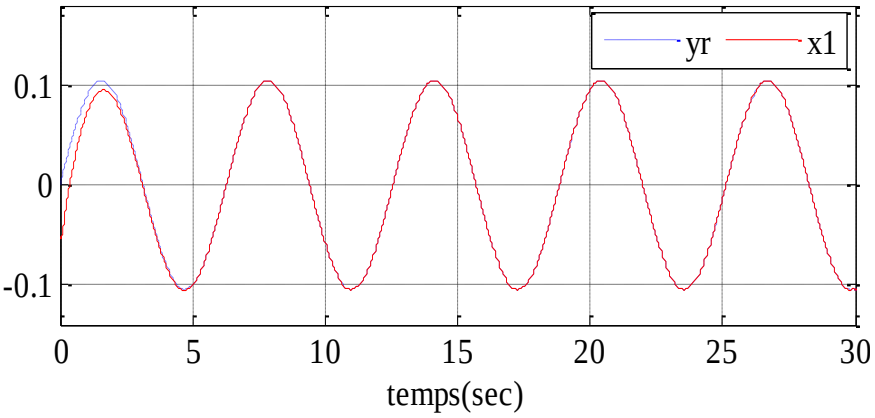


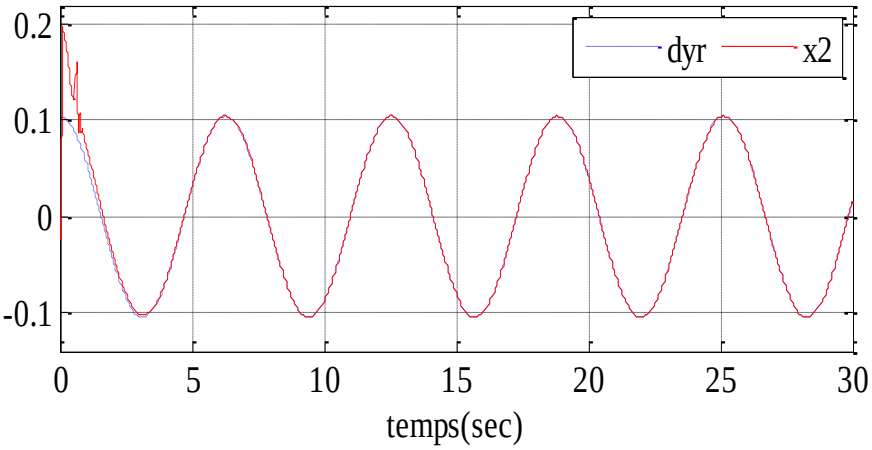
Figure 3.3 R*ésultats de simulation sans perturbation.*

- avec perturbation de $d(t) = 1.5\sin(2\pi t)$.

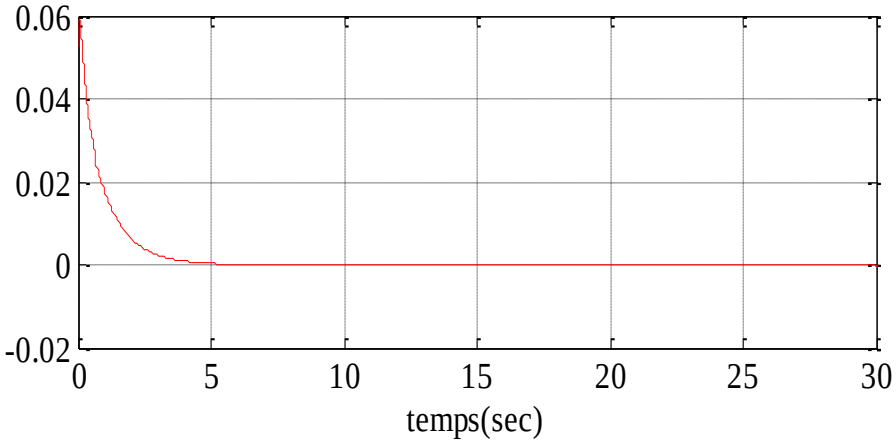
(a) Sortie du système yr et x1



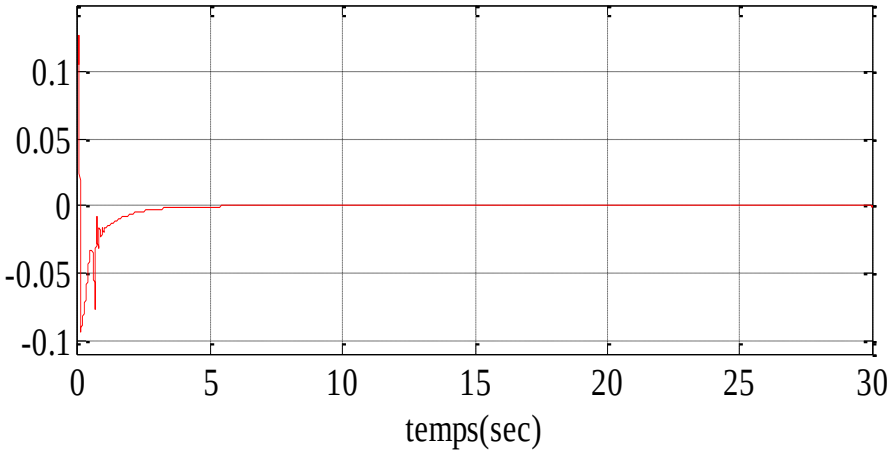
(a) Sortie du système dyr et x2



(c) Erreur de suivi e_1



(d) Erreur de suivi e_2



(e) Signal de commande

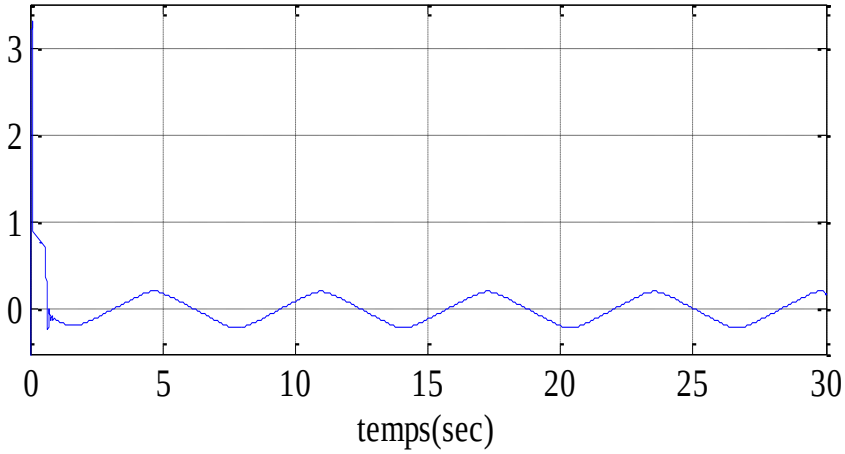


Figure 3.4. *Résultats de simulation avec perturbation*

Interprétation des résultats

Les résultats pour une poursuite de la consigne $y_r = \frac{\pi}{30}\sin(t)$ sont représentés dans les figures (1.17-1.21). Ces résultats révèlent une bonne performance de poursuite (fig1.17). L'oscillation de la commande est due à la variation rapide du gain d'entrée.

Dans cet exemplede simulation, le contrôleur adaptatif flou force la sortie du système à suivre un signal de référence sinusoïdal. La poursuite s'effectue en minimisant l'erreur d'approximation d'une part, et en assurant la stabilité globale du système d'autre part. Cette erreur estimée, peut être améliorée par l'affinement des procédures de calcul et un choix judicieux des lois d'adaptation.

3.3 Conclusion

Dans ce chapitre, deux schémas de CAFI sont proposé pour une classe de systèmes SISO non linéaires (pendule inversé). Le SLF a été utilisé pour approcher en ligne la dynamique inconnue englobant toutes les non linéarités du système.

Dans ce schéma, une loi de commande adaptative, avec ces lois d'adaptation paramétrique, augmentée d'un terme de commande robuste (superviseur) est proposée. Le terme sert à la compensation des erreurs de reconstruction. Ce schéma utilise un algorithme de projection. Ce dernier permet de forcer les paramètres estimés à rester dans une région prédéfinie par l'utilisateur. Cependant, en pratique, la détermination de telle région n'est pas une tâche facile, car, les valeurs des paramètres désirés sont inconnues. Notons que,la direction de

commande dans ce schéma est supposée connue, cette hypothèse n'est pas réaliste dans le cas général.

Les méthodes de commande développées n'exigent pas la connaissance du modèle dynamique du système, garantit la bornitude de tous les signaux de la boucle fermée et la convergence de l'erreur de poursuite. Les résultats de la simulation effectuée sur un pendule inversé montrent les bonnes performances de poursuite de ces méthodes.

Conclusion générale

D'une manière générale, l'analyse et la commande non linéaires sont des problèmes difficiles et la majorité des approches de la commande non linéaire exigent la disponibilité d'un modèle mathématique. Cette commande doit être robuste dans le sens où elle devra assurer une faible sensibilité aux incertitudes sur les paramètres, à leurs variations et aux perturbations. L'objectif principal de ce mémoire est de développer des lois de commande adaptatives floues stables et robustes pour réaliser la poursuite des certaines classes de systèmes non linéaires SISO. Dans la première partie de ce travail, nous avons commencé par présenter l'ensemble des définitions et des outils nécessaires à la mise au point de la stratégie adoptée pour le développement des lois de commande garantissant les performances souhaitées. En effet, un rappel sur les principes de la logique floue, et l'utilisation des systèmes flous comme éléments de base pour l'approximation et la commande des systèmes non linéaires est présenté. Dans la deuxième partie du travail et à base de ce résultat, nous avons proposé et développé des schémas de commande adaptative floue directe et indirecte pour une classe de systèmes non linéaires monovariables. Dans le schéma de commande adaptative directe floue, le but était l'approximation d'une loi de commande stabilisatrice inconnue. En revanche le schéma de commande adaptative indirecte exploite deux systèmes adaptatifs flous pour approximer la dynamique du système, et aussi évite le problème de singularité de la loi de commande par l'utilisation d'un algorithme de projection. Dans la dernière partie de ce travail, nous avons développé un contrôleur adaptatif direct flou pour une classe de systèmes non linéaires. Pour ce faire, nous avons exploité en premier lieu le théorème de la fonction implicite globale pour montrer l'existence du contrôleur idéal inconnu. Ensuite, nous avons utilisé le développement en série de Taylor autour du contrôleur idéal pour transforme le modèle non affine en un modèle affine. En fin, pour des raisons des incertitudes sur le modèle liées aux problèmes de transformations et modélisation, un système flou est utilisé pour approximer la loi de commande idéale. Les lois d'adaptation des paramètres ajustables sont déduites à partir de l'étude de stabilité au sens de Lyapunov des systèmes en boucle fermée. La robustesse étant assurée par l'ajout d'un terme de commande type mode glissant. La simulation, des systèmes non linéaires SISO, a permis de mettre en évidence l'apport des approches proposées.

Références bibliographiques

[1] R. Boukezzoula, S. Galichet et L. Foulloy, Fuzzy Nonlinear Adaptive Internal Model Control, Part I: Fuzzy Model Identification, *Proc. of the European Control Conference*, (ECC'99),Germany, 1999.

[2] Takagi T. et M. Sugeno, Fuzzy identification of systems and its applications to modelling and control, *IEEE Trans. on Man, and Cyber.*, pp. 116-132, 1985.

[3] S. Labiod, M.S. Boucherit et T.M. Guerra, Adaptive fuzzy control of a class of MIMO nonlinear systems, *Fuzzy Sets and Systems*, vol. 151, no. 1, pp. 59-77, 2005.

[4] A. Boulkroune, M. Tadjine, M. M'Saad et M. Farza, A unified approach for design of indirect adaptive output-feedback fuzzy controller, *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 1-2, pp. 83-103, 2008.

[5] R. Boukezzoula, *Commande floue d'une classe de systèmes non linéaires : application au problème de suivi de trajectoire*, Thèse de Doctorat, Université de Savoie, 2000.

[6] K. Khettab. Commande adaptative floue des systèmes non linéaires par la méthode de backstepping utilisant le mode glissant. Mémoire de Magister, Université de Ferhat Abbas - Sétif, 2005.

[7] A. Hamzaoui, N. Essounbouli et J. Zaytoon, Fuzzy Sliding Mode Control for Uncertain SISO Systems, *IFAC Conf. on Intelligent Control Systems and Signal Processing (ICONS'03)*, pp. 233-238, Faro (Portugal), 2003.

[8] H.K. Khalil, *Nonlinear Systems*, Prentice Hall, 1996.

[9] F. Liu, M. Farza, et M. M'Saad, Non-linear observers for state and unknown inputs estimation, *International Journal of Modelling, Identification and Control*, vol. 2, no. 1, pp. 33-48, 2007.

[10] A. Isodori, *Nonlinear control systems II*, Springer-verlag, 1999.

[11] V.I. Utkin, Variable structure systems with sliding modes, *IEEE Trans. Automatic Control*, vol. 22, pp. 212-222, 1977.

[12] J.-J. E. Slotine, Sliding controller design for nonlinear systems, *Int. J. Control*, vol. 40, no.2, pp. 421-434, 1984.

[13] W. Perruquettiet J. P. Barbot, *Sliding Mode Control in Engineering*, Marcel Dekker, 2002.

[14] J.-J. E. Slotineet W. Li, *Applied nonlinear control*, Prentice Hall, N.J., 1991.

[15] G. Bartolini, A Ferrara et E. Usai, Chattering Avoidance by Second Order Sliding Mode control. *IEEE transactions on Automatic Control,* vol. 43, no. 2, pp. 241-246, 1998.

[16] W.-S. Lin et C.-S. Chen, Robust adaptive sliding mode control using fuzzy modelling for aclass of uncertain MIMO nonlinear systems, IEE Proc. Control Theory and Applications, vol. 149, no. 3, pp. 193-201, 2002.

[17] L.-X. Wang, *A Course in Fuzzy Systems and Control*, Prentice Hall, 1997.

[18] B. Yoo et W. Ham, Adaptive fuzzy sliding mode control of nonlinear system, *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 2, pp. 315-321, 1998.

[19] L.X. Wang, *Adaptive Fuzzy Systems and Control Design and Stability: Analysis.* Englewood Cliffs, NJ: Prentice Hall, 1994.

[20] N. Manamanni, A. Hamzaoui et N. Essounbouli, Sliding mode control with adaptive fuzzy approximator for MIMO uncertain systems, *European Control Conference, ECC'03*, Cambridge (UK), 2003.

[21] Frédéric Nollet, *Lois de commande par modes glissants du moteur pas a pas*, Ecole centrale de Lille, 2008.

[23] C.F. Hsu, T.-T. Lee, C.-M. Lin et L.-Y. Chen, Robust neuro-fuzzy controller design via sliding-mode approach, *IEEE International Conference on Fuzzy Systems*, Budapest (Hongrie), vol.2, pp. 917- 922, 2004.

[24] T.M. Guerra, L. Vermeiren, Control laws for Takagi-Sugeno fuzzy models, *Fuzzy Sets and Systems*, vol. 120, no.1, 2001.