

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS SETIF-1

FACULTE DE TECHNOLOGIE

THESE

Présentée au Département d'Electronique

Pour l'obtention du diplôme de

DOCTORAT EN SCIENCES

Filière : Electronique

Option : Electronique

Par

GUESSOUM ABDELHAKIM

THEME

**Conception d'un Contrôleur Intelligent utilisant les
Circuits Modernes**

Soutenue le 24/12/2020 devant le Jury:

FERHAT Hamida Abdelhak	Professeur	Univ. F. Abbas Sétif 1	Président
HASSAM Abdelouahab	Professeur	Univ. F. Abbas Sétif 1	Directeur de thèse
SRAIRI Kamel	Professeur	Univ. M.Khider Biskra	Examineur
HOCINI Abdeslam	Professeur	Univ. M. Boudiaf Msila	Examineur

Introduction

Contrôler un système, c'est être capable d'effectuer les manipulations adéquates sur ses entrées afin d'avoir des sorties dans un état désiré. Le contrôle est un domaine situé au carrefour de l'automatique, des mathématiques, de l'informatique et de l'ingénierie. La tendance actuelle, en particulier dans l'industrie, est de commencer par construire un modèle mathématique précis du système à contrôler, puis de s'en servir pour calculer les actions à entreprendre en fonction des consignes.

Devant la difficulté et le coût de l'élaboration (puis du paramétrage) d'un modèle, une alternative souvent adoptée est de l'apprendre. Doter un contrôleur de capacités d'apprentissage, afin qu'il se construise lui-même un modèle du système contrôlé pour ensuite l'exploiter et calculer les actions à entreprendre, est une solution séduisante. Cependant, elle montre ses limites sur les systèmes complexes. En effet, la complexité d'un modèle, notamment la non-linéarité, rend le calcul du contrôle bien trop coûteux, voire tout quasiment impossible. Améliorer le système de contrôle revient donc à concevoir un système capable d'apprendre à contrôler sans posséder de connaissances préalables sur le système auquel il s'applique. L'apprentissage doit se faire en temps réel, à partir de l'observation des entrées et des sorties du système contrôlé.

Dans le domaine de contrôle, on trouve que les régulateurs PID répondant à plus de 90% des besoins industriels. Malheureusement, malgré l'expérience acquise au fil des années, les valeurs choisies pour les paramètres P, I et D ne sont pas toujours satisfaisantes, ni adaptées au processus sous contrôle. Aujourd'hui, les PID se trouvent sous forme de modules électroniques ajustables, à travers l'exploitation des techniques de réglage intelligentes, on les appelle les PID intelligents.

L'objectif de cette thèse, est donc de concevoir des méthodes de commandes et de développer des techniques intelligentes pour améliorer la robustesse des PID numériques. Ces techniques de commande sont appliquées au contrôle d'un robot mobile de type voiture, afin de suivre une trajectoire de référence. Nous nous sommes donc intéressés à ce problème car La commande des robots mobiles a fait l'objet de nombreuses recherches ces dernières années, en raison de l'utilisation de plus en plus fréquente en milieu dangereux ou inaccessible où l'être humain peut difficilement intervenir. A titre d'exemples, nous pouvons citer l'intervention sur des sites nucléaires dangereux ou l'exploration spatiale. Les modèles utilisés dans ce contexte sont non linéaires et la propriété de non holonomie qui

caractérise un certain nombre de ces robots mobiles conduit à l'utilisation des techniques de commande non linéaire pour les piloter. La problématique générale se réduit dans la plupart des cas à faire déplacer le robot dans un environnement connu ou inconnu, pour réaliser une tâche prescrite.

Organisation du document

Le premier chapitre de ce document présente un bref historique du contrôle de systèmes, on y aborde les notions élémentaires. Il met le point sur la nécessité d'apprendre le contrôle. Nous présentons des généralités sur les PID conventionnels, ainsi que des techniques d'apprentissage artificiel, le contrôle abordé dans ce chapitre qui demeure important, se concentre dans un contrôleur de l'intelligence artificielle, et en particulier de l'apprentissage automatique. Ainsi Les améliorations apportées par l'intelligence artificielle concernent généralement la mise à jour, l'amélioration ou même le remplacement par un mécanisme d'apprentissage du rôle joué par le modèle du procédé. Elles s'intéressent également au paramétrage des contrôleurs, notamment dans le cas des PID.

Le deuxième chapitre de ce mémoire propose un état de l'art sur la robotique mobile en présentant dans un premier temps les deux modèles les plus fréquemment rencontrés en robotique mobile, nous exposons seulement les modélisations cinématiques puis quelques techniques de planification locales et globales pour la génération de trajectoire d'un robot mobile non holonome. Enfin on détaille les méthodes utilisées pour le suivi de trajectoire.

Dans le troisième chapitre, on expose les systèmes d'inférence flous (SIF). Nous commençons par énoncer les fondements théoriques des sous-ensembles flous et de la logique floue, suivi par la description de la structure générale d'un contrôleur flou, en montrant leurs types. Dans la deuxième partie, nous présentons une étude sur les réseaux de neurones artificiels (RNA) et l'apprentissage par renforcement (AR) ainsi que leur principe de fonctionnement. Une architecture d'un contrôleur neuro-PID qui permet d'ajuster les paramètres PID est donnée à la fin du chapitre.

Dans le chapitre quatre nous fournissons un bref aperçu concernant le contrôleur PID conventionnel et une étude plus détaillée sur le PID numérique de type (PI+D). Leur configuration, leurs méthodes de conception et leur analyse de stabilité y est aussi exposées. Les techniques développées pour le réglage des paramètres PID et autres concepts seront

également étudiés tels que : les flou PID, RBF PID, (flou PI+D), neuro (PI+D) flou-neuro (PI+D). Ces méthodes sont appliquées au contrôle de suivi de trajectoire d'un robot mobile, où on présentera les résultats de simulation.

On termine la thèse avec une conclusion qui résume et discute les résultats des travaux réalisés et quelques perspectives.

Chapitre1

Le contrôle des systèmes complexes

Ce chapitre présente une brève introduction et un aperçu des systèmes de contrôle. Certains des termes couramment utilisés pour décrire le fonctionnement, l'analyse et la conception des systèmes de contrôle sont présentés.

1.1 Le contrôle

Un système de contrôle dans un domaine interdisciplinaire couvrant de nombreux domaines de l'ingénierie et les sciences. Les systèmes s'expriment de bien des manières de contrôle et existent dans de nombreux systèmes d'ingénierie. Certains types de systèmes de contrôle affectent la plupart des aspects de nos activités quotidiennes.

1.1.1 De la révolution industrielle à nos jours

C'est au XIX siècle, lors de la révolution industrielle, que l'utilisation massive des machines donna aux systèmes de contrôle automatique une importance cruciale. Réguler température, pression, et autres niveaux ne pouvait effectivement pas se faire manuellement sur des systèmes tels que les machines à vapeur. Toutefois, les contrôleurs de l'époque manquaient encore de recul théorique et étaient majoritairement élaborés à partir de l'expérience et de l'intuition des ingénieurs.

Quelques percées mathématiques eurent tout de même lieu et conduisirent plus tard à la théorie du contrôle. Par exemple, le mathématicien et astronome George Biddell Airy utilisa en 1840 des équations différentielles pour caractériser l'instabilité d'un système de contrôle de télescope. La publication en 1868 d'un article de James Clerk Maxwell sur les régulateurs à boules (un système mécanique de régulation de la vitesse de rotation des machines à vapeur) est aujourd'hui considérée comme le début de la théorie du contrôle.

Jusqu'au début du XXe siècle, la notion de système était absente. Les concepts d'entrée et de sortie firent leur apparition à cette époque avec l'arrivée de la théorie des systèmes. Ce siècle fut marqué par les deux guerres mondiales ainsi que par l'essor des télécommunications. Les guerres appuyèrent les recherches sur le guidage d'engins et sur la visée de missiles. Les contrôleurs de type PID, une technique toujours d'actualité, furent notamment, utilisés pour la première fois en 1911 pour le contrôle de la direction de navires, avant d'être analysés théoriquement en 1922 [1].

Les technologies de communication à distance firent quant à naître des problématiques d'amplification du signal et donnèrent lieu à la notion de rétroaction négative, découverte par Harold Stephen Black en 1927. Elles entraînèrent également l'application des travaux de mathématiciens du siècle précédent, comme Augustin-Louis Cauchy ou Joseph Fourier, ouvrant la voie du contrôle par domaine fréquentiel.

Les travaux publiés jusqu'à cette époque constituent ce que l'on appelle aujourd'hui la théorie classique du contrôle. Suffisante dans beaucoup de cas, cette théorie est néanmoins en difficulté face au système non-linéaire et possédant plusieurs entrées et sorties. L'intérêt se porta alors sur le contrôle optimal, qui vise à réduire une certaine mesure de coût de fonctionnement d'un système, et sur l'utilisation d'équations différentielles pour analyser les systèmes et leur contrôle. C'est à ce moment que la discipline du contrôle rencontra celle de l'informatique, alors en plein développement. La faculté de cette dernière à calculer les solutions d'équations complexes a favorisé la mise en équation des systèmes de contrôle, ainsi que la modélisation mathématique des systèmes contrôlés. Cette tendance se poursuit depuis les années 70 et est progressivement complétée par l'utilisation de techniques issues de l'intelligence artificielle, en particulier les algorithmes d'apprentissage.

Ce bref historique appuie sur le lien entre la théorie du contrôle et les besoins représentatifs des époques et des technologies dans lesquelles elle s'inscrit. Les méthodes de contrôle, créées bien souvent par des spécialistes du domaine d'application, ont des objectifs variés (régulation, optimisation, maîtrise, etc) qui trouvent tous une solution en la manipulation adéquate des entrées d'un système.

1.1.2 Définition et concepts fondamentaux

Le contrôle de systèmes est au carrefour de plusieurs disciplines : automatique, mathématique et aujourd'hui informatique, et s'étend à de nombreux domaines d'application (biologie, physique, chimie, etc) apportant chacun leur vision du problème, leurs exigences et leurs contraintes. L'étude des systèmes et de leur contrôle met tout de même à jours des traits communs, faisant apparaître des concepts propres au contrôle, présentés dans cette partie.

1.1.2.1 La notion de système

La notion de contrôle ne saurait se départir de celle de système. Cette dernière prend différentes formes selon la discipline concernée. Mathématiques, informatique, physique ou encore automatique considèrent un "système" sous différents angles. Deux aspects complémentaires peuvent être mis en avant. Un système peut être décrit selon son aspect

structurel. Il est alors présenté comme un ensemble d'éléments constitutifs, leurs relations et par une frontière déterminant l'interface entre le système et son environnement. Mais un système peut également être abordé selon son aspect fonctionnel. On s'intéresse alors aux flux d'information. L'accent est mis sur les entrées et les sorties (du système ou de ses composants) et sur le processus de transformation qui lie les premières aux secondes.

Un système ne possédant qu'une entrée et qu'une sortie est dit SISO (Simple-Input and Simple-Output), il est qualifié de MIMO lorsqu'il en possède plusieurs (Multiple-Input and Multiple-Output).

La théorie des systèmes dynamiques exprime l'évolution d'un système au cours du temps comme le résultat d'une fonction prenant en argument son état (la valeur de ses entrées et sorties) ainsi que les actions appliquées sur ses entrées :

$$\frac{dx}{dt} = f(x, u) \quad (1.1)$$

Où x est un vecteur comprenant les entrées et les sorties du système, u un vecteur comprenant les modifications effectuées sur les entrées et f représente le processus de transformation qui caractérise le système. La linéarité de cette fonction définit celle du système.

Un système est ainsi dit linéaire si son état évolue linéairement par rapport à son état précédent et aux actions sur ses entrées. Cela en fait un système relativement aisé à contrôler, en raison de l'absence de minimums locaux sur f .

La classe des systèmes affines en contrôle est un peu plus difficile à gérer. Ces systèmes évoluent linéairement par rapport aux contrôles appliqués et non-linéairement par rapport à leur état courant[2] :

$$\frac{dx}{dt} = f(x) + g(x)u \quad (1.2)$$

Où f et g sont des fonctions quelconques, potentiellement non-linéaires. La composante linéaire de ces systèmes permet toutefois l'application de méthodes mathématiques de linéarisation facilitant par la suite le contrôle.

Enfin, les systèmes non-linéaires sont les systèmes dont l'évolution dépend de manière non-linéaire de son état comme des actions sur les entrées.

La non-linéarité fait partie des critères définissant les systèmes complexes. La complexité découle également de la difficulté, voire de l'impossibilité, d'identifier tous les mécanismes en jeu dans le système, et donc de prévoir son comportement. Certains des éléments et des relations du système, ou de son environnement, peuvent être incertains ou inaccessibles. Cela peut apparaître lorsqu'un très grand nombre d'éléments sont en jeu, que le graphe des relations n'est pas trivial (c'est-à-dire qu'il présente des cycles et que certains liens sont privilégiés), ou que certaines variables sont trop dynamiques ou physiquement non mesurables. En outre, ce type de systèmes présente fréquemment plusieurs niveaux : les éléments interagissant localement sont eux-mêmes des systèmes plus ou moins complexes, compliquant considérablement l'étude du système global.

1.1.2.2 La notion de contrôle

Contrôler signifie réglementer, diriger, commander ou gouverner. Un système est une collection, un ensemble ou arrangement des éléments (sous-systèmes). Un système de contrôle est une interconnexion de composants forment une configuration système qui fournira une réponse système souhaitée. Par conséquent, un système de contrôle est un agencement de composants physiques connectés ou liés de manière à commander, réglementer, diriger, se gouverner ou gouverner un autre système.

Afin d'identifier, délimiter ou définir un système de contrôle, nous introduisons deux termes: entrée et sortie ici. L'entrée est le stimulus, l'excitation ou la commande appliquée à un système de contrôle, et la sortie est la réponse réelle résultant d'un système de contrôle. La sortie peut ou ne peut pas être égale à la réponse spécifiée implicite par l'entrée. Les entrées peuvent être des variables physiques ou des valeurs abstraites telles que référence, point de consigne ou valeurs souhaitées pour la sortie du système de contrôle. Les systèmes de contrôle peuvent avoir plus d'une entrée ou sortie. L'entrée et la sortie représentent la réponse souhaitée et la réponse réelle, respectivement. Un système de contrôle fournit une sortie ou la réponse pour une entrée ou un stimulus donné, comme le montre la figure 1.1.

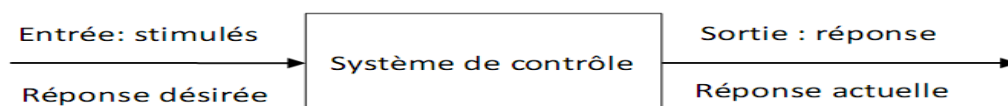


Figure 1.1 – Contrôle en boucle ouverte.

La sortie peut ne pas être égale à la réponse spécifiée impliquée par l'entrée. Si la sortie et l'entrée sont fournies, il est possible d'identifier ou de définir la nature des composants du système. D'une manière générale, il existe trois types de base de systèmes de contrôle:

- Systèmes de contrôle artificiels
- Systèmes de contrôle biologique
- Systèmes de contrôle dont les composants sont à la fois naturels et fabriqués par l'homme.

Il existe deux configurations de système de contrôle: le système de contrôle en boucle ouverte et le système de contrôle en boucle fermée.

1. bloc : le bloc est un ensemble d'éléments qui peuvent être regroupés, avec des caractéristiques globales décrites par une relation entrée / sortie comme illustré sur la figure 1.2. Un schéma de principe est une représentation graphique simplifiée de la relation de cause à effet entre les entrées et les sorties d'un système physique.

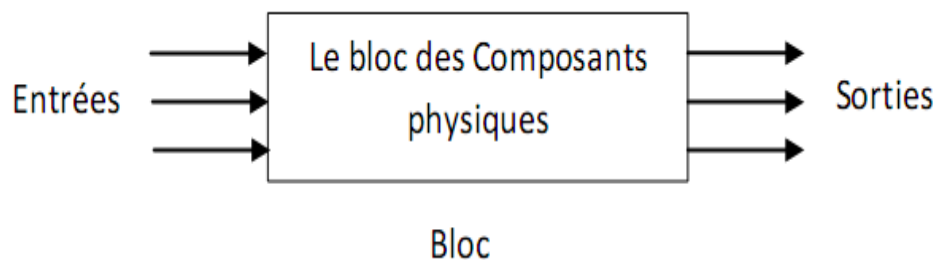


Figure. 1.2 Schéma fonctionnel

2. Fonction de transfert : La fonction de transfert est une propriété des éléments du système uniquement, et ne dépend pas de l'excitation et des conditions initiales. La fonction de transfert d'un système (ou un bloc) est défini comme le rapport entre la sortie et l'entrée, comme indiqué sur la figure.1.3.

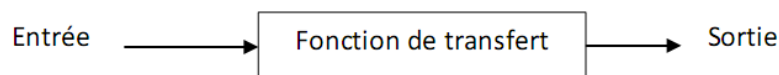


Figure. 1.3 Fonction de transfert

Fonction de transfert = $\text{Sortie} / \text{entrée}$

Les fonctions de transfert sont généralement utilisées pour représenter un modèle mathématique de chaque bloc dans la représentation du diagramme.

Par exemple, l'entrée de référence de la fonction temporelle est $r(t)$, et sa fonction de transfert est $R(s)$ où (t) est le temps et (s) est la variable transformée de Laplace ou la fréquence complexe. Les fonctions de transfert peuvent être utilisées pour représenter des systèmes en boucle fermée ainsi que des systèmes en boucle ouverte.

3. Système de contrôle en boucle ouverte : Les systèmes de contrôle en boucle ouverte représentent la forme la plus simple des dispositifs de contrôle. Un schéma de principe général du système en boucle ouverte est présenté à la figure.1.4.

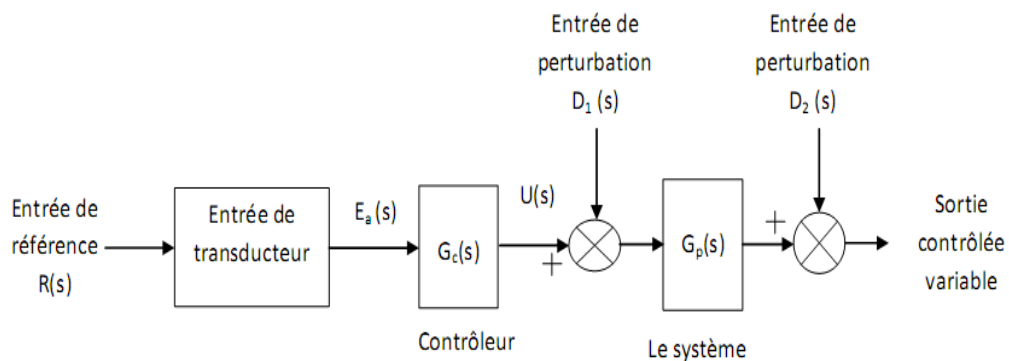


Figure. 1.4 Schéma général du système de contrôle en boucle ouverte

4. Système en boucle fermée (contrôle de retour). Les systèmes de contrôle en boucle fermée tirent leur précieuse reproduction fidèle des données de comparaison. L'architecture générale La figure 1.5 illustre la configuration d'un système de contrôle en boucle fermée. Un système avec un ou plusieurs commentaires chemins est appelé un système en boucle fermée.

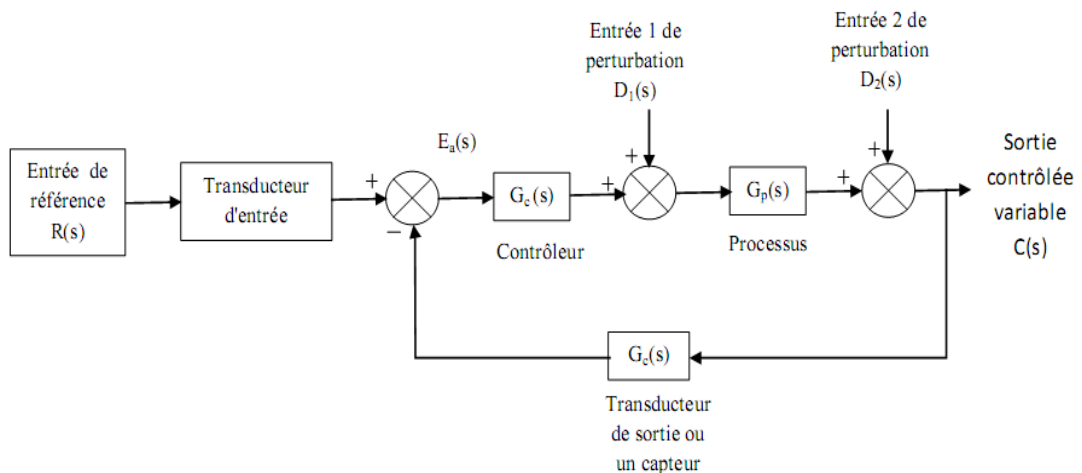


Figure. 1.5 Schéma général du système de contrôle en boucle fermée

1.2 Contrôle de systèmes complexes et apprentissage artificiel

L'objectif de cette partie est de donner une vue d'ensemble à la fois des techniques actuelles de contrôle de systèmes complexes et des méthodes d'apprentissage artificiel.

Les techniques de contrôle se répartissent en trois grandes familles. Nous commencerons par les approches classiques : les contrôleurs PID, qui sont les plus répandus dans l'industrie, et les contrôleurs adaptatifs, plus efficaces sur les systèmes complexes. Les contrôleurs intelligents, faisant appel à des techniques d'intelligence artificielle (IA) pour améliorer leurs performances, seront introduits après avoir fait un tour d'horizon des méthodes d'apprentissage automatique dont ils usent. Ces approches de contrôle seront évaluées selon leur capacité à être appliquées à de nombreux domaines, leur facilité d'interpréter à un procédé particulier, leur capacité à suivre l'évolution dans le temps du système contrôlé (adaptativité) ainsi que leur faculté à apprendre le fonctionnement du procédé ou au contraire leur besoin de connaissances a priori (apprentissage).

1.2.1 Les classiques du contrôle des systèmes

Le contrôle de systèmes étant un domaine étudié depuis de nombreuses décennies, les méthodes proposées sont très diverses et se sont largement ramifiées, raffinées et combinées au cours du temps. Celles présentées dans cette section, même si elles font toujours l'objet de travaux de recherche, sont bien implantées dans l'industrie et sont considérées comme classiques.

1.2.1.1 Contrôleurs PID

Les contrôleurs PID sont très largement répandus dans l'industrie. Ils tirent leur nom des trois termes qu'ils calculent à partir du signal de rétroaction afin de le transformer en correction à appliquer sur la variable contrôlée : Proportionnelle, Intégrale et Dérivée [3].

– Le terme Proportionnel, aussi appelé gain, permet de tenir compte de l'erreur actuelle entre la valeur observée du signal de rétroaction et la consigne.

$$p = K_p e(t) \quad (1.3)$$

Où K_p est une constante, e l'erreur actuelle et t le temps. Augmenter l'importance de ce terme permet une convergence plus rapide vers la consigne mais dégrade rapidement la stabilité du système.

– Le terme Intégral sert à rendre le contrôle plus précis en prenant en compte l'erreur sur la durée.

$$I = K_i \int_0^t e(\tau) d\tau \quad (1.4)$$

Où K_i est une constante, e l'erreur, t le temps, et τ la variable d'intégration. Augmenter l'importance de ce terme tend à éliminer l'erreur statique, mais ralentit fortement le temps d'établissement d'un régime stationnaire car cela provoque un dépassement et des oscillations.

– Enfin, le terme Dérivée permet de considérer le taux de variation de l'erreur et donc d'avoir une estimation de sa future valeur. Ainsi, le contrôleur peut par exemple diminuer l'amplitude de sa correction lorsque l'erreur diminue et inversement.

$$D = K_d \frac{de(t)}{dt} \quad (1.5)$$

Où K_d est une constante, e l'erreur et t le temps. Augmenter l'importance de ce terme tend donc à diminuer le dépassement et les oscillations, mais fait perdre en précision. En effet, l'amplitude de la correction diminuant avec l'erreur, il perdure une différence entre la réponse et la consigne, que l'on appelle erreur statique.

Ces trois termes sont ensuite combinés selon une méthode propre à l'instance de PID considérée. La plus classique est la somme, la formule générale de la commande $u(t)$ d'un PID est alors exprimée comme :

$$u(t) = k \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (1.6)$$

Où K , T_i , T_d sont des constantes appelées respectivement gain, temps d'intégration et temps de dérivation. Notons que selon les cas les fonctions I ou D peuvent être omises, on parle alors de contrôleurs P , PI ou PD .

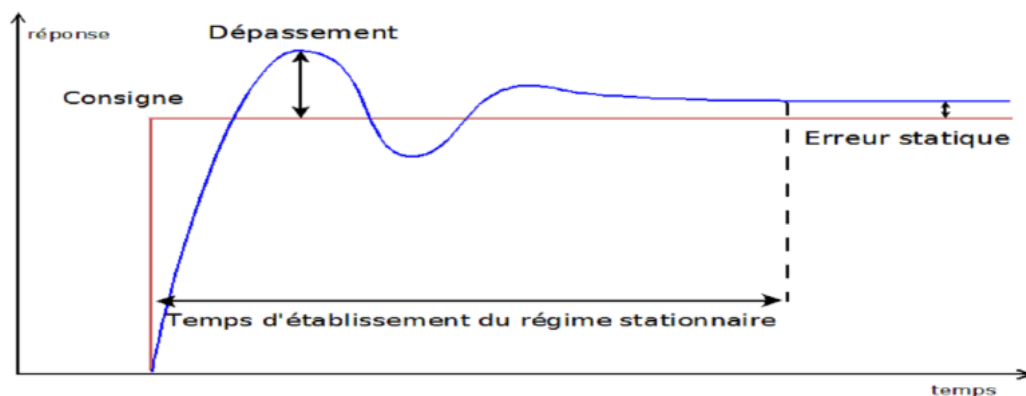


Figure 1.6 réponse typique d'un PID

La figure 1.6 montre une réponse typique d'un PID (mal réglé) sur un procédé stable. Aussi, pour fonctionner correctement sur un système particulier, un contrôleur PID nécessite d'être paramétré avec justesse. Trouver les valeurs des constantes qui permettent un bon compromis entre rapidité et précision est un problème compliqué qui est parfois la raison principale de la désactivation d'une des fonctions. De nombreuses méthodes ont été développées pour faciliter le paramétrage des PID, la plus connue étant certainement la méthode de Ziegler-Nichols.

1.2.1.1.1 Méthode de Ziegler-Nichols

Ziegler et Nichols ont étudié l'influence des paramètres K , T_i , T_d de l'équation 2.1 afin de fournir une technique plus simple de paramétrage [4]. La méthode consiste à désactiver les fonctions Intégrale et Dérivée du PID, par exemple en définissant $T_i = +\infty$ et $T_d = 0$. À consigne fixe, le gain K est ensuite mis à zéro puis progressivement augmenté, jusqu'à ce qu'il atteigne une valeur critique K_{crit} pour laquelle la réponse oscille de manière périodique autour d'une valeur stable. On note T la période de ces oscillations. Le tableau 1.1 donne les valeurs à appliquer aux paramètres selon le type de contrôleur (P, PI ou PID).

Table 1.1 – Ajustement d'un contrôleur PID selon la méthode de Ziegler-Nichols

Contrôle	K	T_i	T_d
P	$K_{crit}/2$		
PI	$K_{crit}/2.2$	$T/1.2$	
PID	$K_{crit}/1.7$	$T/2$	$T/8$

Cette méthode donne des résultats globalement bons mais n'est pas toujours optimale. En outre, l'exploration des valeurs de K peut amener le procédé vers des états instables et donc rendre l'application de la méthode impossible à certains systèmes. De nombreuses variations ont été développées [5], ou d'autres améliorant les résultats obtenus grâce à l'utilisation de la logique floue [6].

1.2.1.1.2 limite des PID

Étant appliqués à de nombreux domaines, la généralité des PID n'est plus à démontrer. Cependant, elle est obtenue au prix d'un lourd travail de paramétrage demandant une connaissance approfondie aussi bien du fonctionnement du contrôleur que du procédé contrôlé. En outre, la fiabilité du matériel est importante puisque les PID réagissent mal aux imprécisions sur les mesures ainsi qu'à l'évolution au cours du temps du procédé contrôlé.

Pour répondre à ce problème, des méthodes de paramétrage dynamique, permettant de mettre à jour les paramètres à la volée, ont été étudiées mais restent limitées à certaines classes de procédés [7] ce qui en limite fortement le caractère générique.

Utilisés seuls, les contrôleurs PID suffisent pour les systèmes dont la dynamique est inférieure au second ordre et où les exigences sur les performances du contrôle sont peu importantes. Sinon, il est nécessaire de leur adjoindre des mécanismes permettant d'optimiser leur comportement, parmi lesquels on note la logique floue [8], la mise en cascade de PID [9], l'implémentation sous forme de réseau de neurones artificiels [10], ou encore le principe du feed-forward, c'est-à-dire l'utilisation de connaissances extérieures relatives au procédé contrôlé [11]. Ces raffinements font pour la plupart partie du "contrôle intelligent" qui sera abordé plus tard.

1.2.1.1.3 Bilan des PID

En conclusion, l'algorithme de base des PID est générique mais limité aux systèmes SISO, de faible dynamique et stables dans le temps. Ses raffinements lui permettent de s'étendre à d'autres classes de problèmes mais souffrent en contrepartie de la nécessité d'un lourd travail pour être appliqués à un procédé particulier. En outre cette instanciation demande de la part de l'utilisateur de bonnes connaissances sur le système contrôlé. Ces remarques sont récapitulées par le tableau 1.2.

Table 1.2 – Bilan des PID.

Critère	PID
Généricité	+
Instanciation	--
Adaptativité	--
Apprentissage	Aucun (la connaissance du procédé est implicitement contenue dans le paramétrage)

Les PID constituent souvent la couche basse de l'implémentation de systèmes de contrôle plus évolués. Ces systèmes, gérant plusieurs variables et une dynamique plus complexe, produisent des consignes envoyées aux PID qui contrôlent chacun une seule variable spécifique

1.2.1.2 Contrôle adaptatif

Les caractéristiques de certains procédés évoluent au cours du temps, leur contrôle nécessite alors d'être réajusté. Un exemple courant est la température d'un moteur à

combustion, augmentant avec le temps lorsque le moteur est en fonctionnement et influant grandement sur son comportement. Les méthodes s'attaquant à ce problème sont regroupées sous le terme de contrôle adaptatif. Nous abordons ici les principales approches à savoir l'utilisation d'un modèle de référence, l'identification de modèle, le contrôle dual et le contrôle par apprentissage itératif.

1.2.1.2.1 Contrôle avec modèle de référence

Les systèmes de contrôle adaptatif à modèle de référence (Model Reference Adaptive Controller, ou MRAC, en anglais) comprennent trois briques : un modèle mathématique du procédé contrôlé, un mécanisme d'ajustement de paramètres et le contrôleur à proprement parler [12].

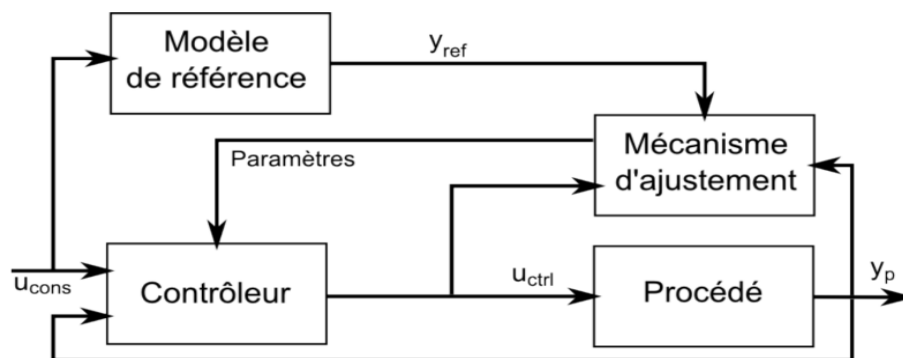


Figure 1.7 Schéma d'un système MRAC.

Le schéma général d'un système MRAC est donné par la figure 1.8. L'idée de base est d'ajuster les paramètres du contrôleur (et donc modifier les commandes u_{ctrl} qu'il produit) afin que la sortie du procédé y_p se comporte comme celle de son modèle "idéal" y_{ref} . Autrement dit, l'ajustement des paramètres du contrôleur compense l'évolution du procédé. La consigne u_{cons} n'a donc pas besoin d'être modifiée. C'est là une utilisation assez inhabituelle d'un modèle puisqu'on ne cherche pas à l'ajuster pour qu'il reste fidèle au procédé, mais on ajuste le contrôle du procédé pour qu'il reste fidèle au modèle.

Le mécanisme d'ajustement est généralement basé sur un processus de type descente de gradient comme la MIT rule (du nom du célèbre institut), minimisant l'erreur entre y_{ref} et y_p [13]. Il doit évidemment être choisi en adéquation avec le type de contrôleur utilisé, qui peut par exemple être un PID sous réserve de quelques ajustements [14].

1.2.1.2.2 Contrôle avec identification de système

Plutôt que de se servir d'un modèle comme d'une référence à suivre, les systèmes de contrôle à identification de modèle (Model Identification Adaptive Controllers, ou MIAC, en anglais) cherchent à ajuster la représentation qu'ils ont du système contrôlé pour pouvoir paramétrer dynamiquement le contrôleur afin qu'il demeure adapté au procédé [15]. Les briques qui les composent sont semblables à celles des systèmes MRAC mais ne jouent pas exactement le même rôle figure 1.8.

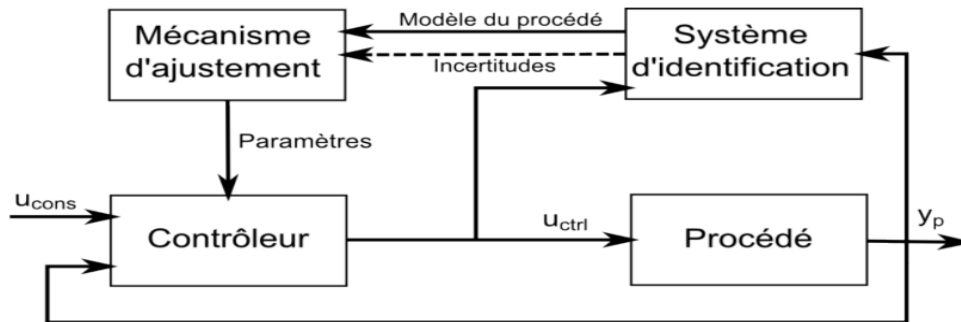


Figure 1.8 Schéma d'un système MIAC.

Le mécanisme d'ajustement se base sur la connaissance préalable de la relation entre les paramètres du modèle identifié et ceux du contrôleur. Il ajuste dynamiquement les paramètres du contrôleur en fonction des informations que lui donne le système d'identification.

Branché sur les entrées et les sorties du procédé, le système d'identification a pour rôle d'en édifier et d'en maintenir un modèle. Ce modèle peut-être construit avec ou sans apport de connaissances préalables (on parle alors respectivement d'identification de boîte grise ou de boîte noire). L'identification de systèmes est une discipline à part entière qui, bien que fortement liée au domaine du contrôle, fait l'objet de travaux qui lui sont propres [16]. On y retrouve des méthodes de régression linéaire comme les moindres carrés [17], de filtrage [18] ou encore des méthodes dites non-paramétriques basées sur l'analyse des entrées et sorties du procédé (analyse temporelle, analyse fréquentielle, etc) [19], ainsi que des techniques d'intelligence artificielle comme la logique floue ou les réseaux de neurones [20]. Du choix du système d'identification dépendra le mécanisme d'ajustement de paramètres, et donc indirectement, la loi de contrôle implémentée.

1.2.1.2.3 Commande prédictive

Une variante largement répandue d'un système de contrôle avec modèle est la commande prédictive (Model Predictive Command, MPC, en anglais). Ici, la fonction du

modèle est de prévoir les futures réactions possibles du système contrôlé. En explorant, grâce au modèle, l'espace d'états du procédé sur un horizon de temps fini, un algorithme d'optimisation calcule les meilleurs prochains contrôles à appliquer. L'algorithme général est représenté par la figure 1.9 [21].

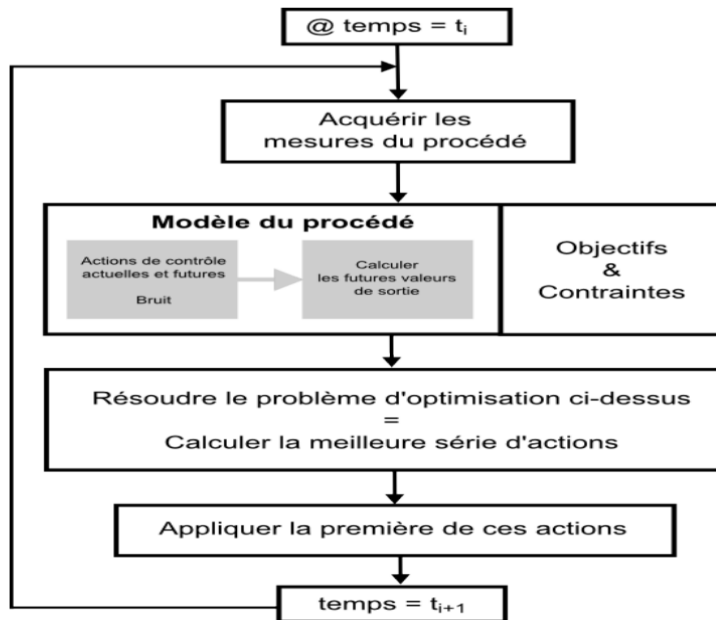


Figure 1.9 Algorithme général de la commande prédictive.

La commande prédictive est une technique de commande avancée de l'automatique. Elle a pour objectif de commander des systèmes industriels complexes. Le principe de cette technique est d'utiliser un modèle dynamique du processus à l'intérieur du contrôleur en temps réel afin d'anticiper le futur comportement du procédé

Cette méthode permet de gérer plusieurs entrées et sorties, et de prendre en compte des contraintes de manière explicite tout en maintenant des objectifs à plus ou moins long terme.

Ses performances dépendent fortement de l'algorithme d'optimisation choisi. Celui-ci peut par exemple utiliser un modèle inverse du procédé pour calculer les entrées à appliquer à partir des sorties souhaitées.

La commande prédictive fonctionne de manière satisfaisante lorsque le modèle utilisé est linéaire, mais cela signifie souvent se limiter à une plage de fonctionnement réduite du procédé sur laquelle son comportement est correctement approximé par un modèle linéaire. Contrôler le système sur l'ensemble de ses plages de fonctionnement demande de passer à un modèle non-linéaire. On parle, dans ce cas, de Non-linear Model Predictive Commade

(NMPC). En faisant apparaître des minimums locaux dans le problème d'optimisation, la non-linéarité du modèle complique considérablement son exploitation. Les principales solutions à cette difficulté consistent à reformuler le problème d'optimisation en linéarisant le signal de rétroaction reçu, souvent à l'aide d'un changement de variable [22]. Cependant cette technique n'est utilisable que sur un sous-ensemble des systèmes non-linéaires : les systèmes "affines en contrôle", c'est-à-dire dont l'évolution de l'état dépend linéairement des contrôles appliqués et non-linéairement de l'état précédent. En outre, elle complique significativement la prise en compte de contraintes [23].

Enfin, La commande prédictive se différencie des autres techniques de commande par le fait que doit être résolu en ligne, elle consiste à optimiser, à partir des entrées/sorties d'un système, le comportement futur prédit du système considéré. La prédiction est faite à partir d'un modèle interne du système sur un intervalle de temps fini appelé horizon de prédiction. La solution du problème d'optimisation est un vecteur de commande dont la première entrée de la séquence optimale est injectée au système. Le problème est à nouveau résolu sur l'intervalle de temps suivant en utilisant les données du système mises à jour. Certaines stabilité du système contrôlé [24]. Le découpage en sous-parties et la définition des contrôleurs locaux demeure un problème ouvert dans le cas général.

1.2.1.2.4 Contrôle dual

Le contrôle dual (Dual Control Theory) propose des bases pour le contrôle d'un système inconnu au départ et repose sur l'utilisation de deux types d'action de contrôle [25] :

- Les actions de contrôle effectives, ayant pour but d'amener le procédé dans l'état désiré, et basées sur les connaissances actuelles du contrôleur.
- Les actions "sondes", dont les conséquences sont analysées par le contrôleur pour affiner ses connaissances sur le système contrôlé.

Ici, et contrairement aux MIAC, le système de contrôle ne se base pas sur l'erreur entre les sorties d'un modèle et les observations passives du procédé pour acquérir de nouvelles informations sur le système contrôlé. Il entreprend directement des actions pour en extraire de l'information. La structure du système contrôlé est supposée connue, c'est-à-dire que le contrôleur dispose d'une fonction f (un modèle) représentant le procédé et calculant la sortie y à l'instant $k + 1$ à partir du contrôle u effectué à l'instant k , de l'historique γ de toutes les actions et observations jusqu'à l'instant k , d'un vecteur de paramètres inconnus θ et d'un processus stochastique ξ connu représentant l'évolution (des paramètres) du procédé.

$$y(k + 1) = f(u(k), \gamma_k, \theta(k), \xi(k)) \quad (1.7)$$

Il reste alors au contrôleur à estimer correctement les paramètres de ce modèle grâce à des actions d'exploration.

$$V(\xi(k), k) = \min E\{(y(k) - y_r(k))^2 + V(\xi(k + 1), k + 1) | \gamma_{k=1}\} u(k = 1) \quad (1.8)$$

$V(\xi(k), k)$ peut être interprétée comme la perte minimum attendue à partir de l'instant k et pour la suite du contrôle entre la sortie observée $y(k)$ et la consigne $y_r(k)$, étant donné les observations acquises jusqu'à maintenant γ_{k-1} . E dénote l'espérance mathématique prise sur la distribution de ξ . Résoudre cette équation signifie trouver la meilleure action de contrôle $u(k - 1)$ celle qui assurera une perte minimum pour la suite du contrôle. Cette action ayant une influence immédiate sur la sortie du procédé et différée sur l'estimation future des paramètres, la solution de l'équation est un choix entre une perturbation (une action "sonde") qui minimisera le terme $V(\xi(k), k + 1)$ (c'est-à-dire la prévision moyenne des prochaines pertes) ou une action visant à améliorer la perte immédiatement (c'est-à-dire minimiser le terme $(y(k) - y_r(k))^2$). Ceci s'avère très difficile dans la plupart des cas concrets, notamment en raison de la grande dimension de ξ .

Malgré tout, la recherche d'une corrélation entre les variations sur les entrées et conséquences observables sur les sorties après l'application d'une action est une idée très intéressante qui mérite d'être conservée.

1.2.1.2.5 Contrôle par apprentissage itératif

Introduit en anglais pour la première fois par [Arimoto, Kawamura et Miyazaki 1984][26], le contrôle par apprentissage itératif (Iterative Learning Control, ILC) a d'abord été motivé par la maîtrise de bras robotisés industriels, répétant indéfiniment la même tâche. Chaque nouvelle passe améliore le contrôle en s'appuyant sur la précédente.

$$u_{k+1}(t) = u_k(t) + [ke_k](t) \quad (1.9)$$

Le nouveau contrôle appliqué u_{k+1} est donc fonction du précédent u_k , de la dernière erreur e_k entre la sortie et la consigne, et d'un "opérateur de gain" K propre à l'implémentation. Cet opérateur fait le plus souvent appel à un modèle du système contrôlé, représenté sous forme matricielle [Owens et Daley 2008][27].

Cette méthode est réservée aux procédés présentant un comportement périodique. En cela elle est comparable au contrôle répétitif (Repetitive Control, RC) et au contrôle run-to-run (R2R) [28].

1.2.1.2.6 Bilan du contrôle adaptatif

L'utilisation de modèles donne aux méthodes de contrôle adaptatif l'avantage de gérer naturellement plusieurs entrées et sorties. En outre, une fois couplés à des algorithmes d'optimisation (éventuellement dynamiques), ces modèles permettent de mettre à jour les paramètres d'un contrôleur (MRAC et MIAC), ou bien interviennent directement dans le calcul du meilleur contrôle à appliquer (MPC, Dual Control Theory, ILC). Ainsi, le contrôleur est capable de suivre l'évolution du procédé au cours du temps et d'adapter ses actions en conséquence, ce qui lui confère une certaine robustesse.

Mis à part l'apprentissage itératif qui n'est utilisable qu'avec un système répétitif, les approches de contrôle adaptatif sont suffisamment génériques pour être appliquées dans de nombreux domaines. On retrouve, par exemple, l'architecture MIAC dans des domaines aussi variés que le contrôle de l'énergie dans un bâtiment [29] ou celui d'un robot en interaction avec un humain [30]. Cependant cette généralité est à modérer selon la disponibilité d'un modèle adéquat du système à contrôler.

En effet, si la modélisation joue un rôle clé dans les contrôleurs adaptatifs. Dans le cas d'un système à contrôler totalement inconnu au départ, établir un modèle peut demander des années d'études ainsi que de lourds moyens technologiques. Une fois le modèle obtenu, il faut le paramétrer pour qu'il corresponde parfaitement à l'instance considérée du procédé modélisé. Cette étape s'appelle la calibration. Dans un système de contrôle, cette calibration peut non seulement concerner le modèle utilisé, mais aussi les différents mécanismes d'ajustement et d'optimisation. Cette tâche est d'autant plus difficile lorsque le procédé et son modèle ne sont pas linéaires. Néanmoins, cette limite ne remet pas en cause les principes fondamentaux du contrôle adaptatif. Le tableau 2.3 synthétise les paragraphes précédents.

Table 1.3 – Bilan du contrôle adaptatif.

Critère	Contrôle adaptatif
Généricité	+
Instanciation	- -
Adaptativité	+
Apprentissage	Limité (ajustement de paramètres d'une structure fixe)

1.2.1.3 Bilan des approches classiques de contrôle

Nous remarquons que leur difficile instanciation constitue la limite commune des approches présentées jusqu'ici. Pour contourner les difficultés de la modélisation et de la calibration (qui sont les tâches principales à réaliser pour instancier un système de contrôle à son procédé) les travaux les plus récents incorporent au contrôle des techniques issues de l'intelligence artificielle. L'objectif est souvent de faire apprendre le modèle, ses paramètres ou encore sa propre calibration au contrôleur de manière automatique et avec le minimum de connaissances a priori. Dans d'autres cas, il peut aussi s'agir d'apprendre directement une loi de contrôle.

1.2.2 L'apprentissage artificiel

On parle d'apprentissage artificiel (ou apprentissage automatique) lorsqu'un programme a la capacité d'améliorer ses performances à partir de données acquises en cours de fonctionnement, c'est-à-dire à partir de son expérience [31]. Ce type de système est utilisé pour résoudre des tâches trop complexes pour les algorithmes classiques. L'apprentissage étant un sujet central depuis les débuts de l'intelligence artificielle, il en existe de nombreuses approches. Elles sont généralement regroupées en trois familles qui se différencient par le type d'information dont dispose le système pour apprendre et le protocole avec lequel il interagit avec son environnement : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement [32].

1.2.2.1 Apprentissage supervisé

L'apprentissage supervisé désigne les techniques se basant sur un oracle pour guider le système apprenant. Lors d'une première étape, l'oracle fournit des exemples étiquetés, c'est-à-dire un ensemble de m couples comprenant une donnée x et la sortie attendue pour cette donnée u . On appelle cet ensemble l'échantillon d'apprentissage et on le note S .

$$S = \{x_i, u_i\}_{1 \leq i \leq m} = \{x_i, f(x_i)\}_{1 \leq i \leq m} \quad (1.10)$$

La fonction f , parfois appelée fonction cible, n'est connue que de l'oracle et est l'objet de l'apprentissage du système. En exploitant cet échantillon, l'apprenant doit trouver (ou s'approcher de) la bonne sortie $u_n = f(x_n)$ correspondant à une donnée d'entrée u_n qui n'appartient pas nécessairement à S . On appelle la fonction ainsi estimée par le programme apprenant une hypothèse.

Un algorithme d'apprentissage supervisé cherche dans l'espace des hypothèses possibles celle qui est la plus adéquate en regard de l'échantillon d'apprentissage. Cependant,

il existe une infinité de fonctions qui passent par un ensemble de points donnés. Et aucune de ces fonctions n'est a priori meilleure qu'une autre pour décrire l'ensemble des points de S et interpoler les résultats attendus. Comment choisir celle qu'il faut retenir est un des problèmes fondamentaux de l'apprentissage artificiel. Il faut s'assurer que l'apprenant dispose d'un moyen pour le contraindre à converger vers une hypothèse qui conviendra aux données sur lesquelles il sera appliqué après apprentissage. Il s'agit par exemple de la marge maximale dans le cas des machines à vecteurs de support. De manière plus générale le principe du rasoir d'Occam, qui privilégie toujours la solution la plus simple, est une réponse possible. En fait, il s'agit de faire un compromis entre les hypothèses complexes qui correspondent mieux aux données et celles plus simples qui sont meilleures pour généraliser [33]. Si l'hypothèse trouvée est trop complexe, le système est en sur-apprentissage : il ne fait plus d'erreur sur les données d'apprentissage, mais il en commet de plus en plus sur les exemples nouveaux à identifier, à l'image d'un étudiant qui apprend par cœur son cours sans arriver à généraliser.

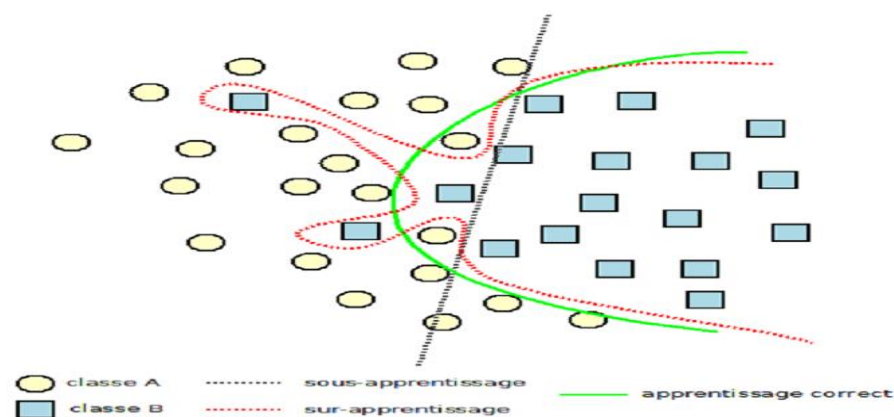


Figure 1.10 Différents apprentissages de la séparation de 2 classes en 2 dimensions, sur un échantillon bruité.

1.2.2.1.2 Réseaux de neurones artificiels

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau.

Les réseaux de neurones biologiques réalisent facilement un certain nombre d'applications telles que la reconnaissance de formes, le traitement du signal, l'apprentissage par l'exemple, la mémorisation, la généralisation. Ces applications sont pourtant, malgré tous les efforts déployés en algorithmique et en intelligence artificielle, à la limite des possibilités

actuelles. C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau que les réseaux de neurones artificiels se sont développés. Les réseaux de neurones artificiels sont des modèles, à ce titre ils peuvent être décrits par leurs composants, leurs variables descriptives et les interactions des composants.

La figure 1.10 montre la structure d'un neurone artificiel. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associée un poids w abréviation de weight (poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associé un poids.

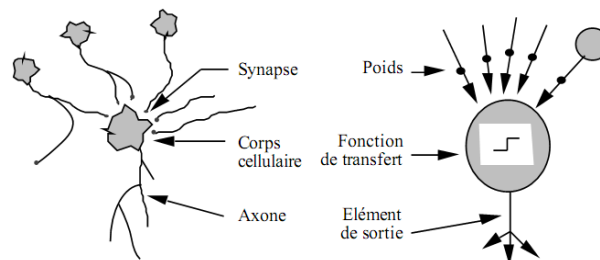


Figure 1.11 Mise en correspondance neurone biologique / neurone artificiel

1.2.2.2 Apprentissage non-supervisé

À la différence des méthodes présentées jusque là, celles d'apprentissage non supervisé ne bénéficient pas d'un oracle pour les guider. Elles ne reposent ni sur une fonction d'évaluation, ni sur l'étiquetage d'un échantillon d'exemples. Leur but est d'apprendre quelles sont les séparations, les corrélations ou encore les composantes naturelles au sein de données non étiquetées. Les algorithmes les plus classiques sont ceux des k-moyennes ainsi que les analyses par composantes, mais de nombreuses autres techniques utilisent les réseaux de neurones. Nous présentons parmi celles-ci les cartes de Kohonen, les réseaux de Hopfield et les machines de Boltzmann.

Pour exemple, la mesure de distance la plus connue est probablement la distance euclidienne définie par l'équation 2.4 pour deux instances x_1 et x_2 .

$$dist(x_1, x_2) = \sqrt{\sum_{i=1}^d (x_1^{(i)} - x_2^{(i)})^2} \quad (1.11)$$

1.2.2.3 Apprentissage actif

Dans le cas d'un apprentissage supervisé, le but est d'apprendre un modèle qui lie les instances à leurs classes respectives, en utilisant un échantillon aléatoire de données d'apprentissage étiquetées. Dans ce cas, l'algorithme d'apprentissage apprend un modèle de classification en fonction des données dont il dispose déjà, autrement dit, de données imposées. Afin d'enrichir la qualité des données utilisées pour l'apprentissage, il est possible d'autoriser des interactions entre l'algorithme d'apprentissage et l'opérateur qui effectue l'étiquetage des données. Il s'agit alors d'un apprentissage actif (à l'opposé d'un apprentissage passif).

1.2.3 Le contrôle intelligent

Nous présentons dans cette partie, le contrôle intelligent qui est le plus important de notre travail. Le contrôle intelligent se concentre sur l'incorporation dans un contrôleur de méthodes issues de l'intelligence artificielle (IA), et en particulier de l'apprentissage automatique.

Les améliorations apportées par l'IA concernent généralement la mise à jour, l'amélioration ou même le remplacement par un mécanisme d'apprentissage du rôle joué par le modèle du procédé. Elles s'intéressent également au paramétrage des contrôleurs, notamment dans le cas des PID.

Le contrôle intelligent décrit la discipline dans laquelle sont développées des méthodes de contrôle imitant des caractéristiques importantes de l'intelligence humaine. Ces caractéristiques incluent l'adaptation et l'apprentissage, la planification dans une grande incertitude et la gestion d'un grand nombre de données.

Les concepts d'intelligence et de contrôle sont étroitement liés et le terme "contrôle intelligent" a une signification unique et distincte. Un système intelligent doit définir et utiliser des objectifs. Un contrôle est ensuite nécessaire pour que le système atteigne ces objectifs et définisse ces objectifs. Par conséquent, tout système intelligent sera un système de contrôle. Inversement, l'intelligence est nécessaire pour assurer le fonctionnement souhaitable des systèmes dans des conditions changeantes, et il est nécessaire pour atteindre un degré élevé de comportement autonome dans un système de contrôle. Le terme "systèmes de contrôle intelligents" est parfois utilisé dans la littérature technique au lieu de "systèmes intelligents" ou "intelligent machines". L'expression "système de contrôle intelligent" souligne simplement l'aspect contrôle du système intelligent.

Les apports de l'intelligence artificielle au contrôle concerne les mécanismes d'apprentissage, il en est deux qui n'en sont pas mais demeurent assez largement répandus. Les systèmes experts sont le couplage d'une base de connaissances et d'un moteur d'inférence tandis que la logique floue permet de raisonner dans l'incertain. Notre travail est basé sur l'application de la logique floue appliquée au réglage des PID.

1.2.3.1 le PID intelligent

Les contrôleurs PID fractionnaires sont une variante des PID classiques dans laquelle au moins un des termes I ou D est fractionnaire (c'est-à-dire n'est pas une dérivée ou une intégrale d'ordre entier). Cela permet aux PID d'être plus robustes au retard. L'équation générale d'un PID fractionnaire est alors de la forme :

$$u(t) = K_p e(t) + K_i D_t^{-\alpha} e(t) + K_d D_t^{\beta} e(t) \quad (1.12)$$

Où $e(t)$ est l'erreur, K_p , K_i et K_d sont les gains (respectivement proportionnel, intégral et dérivé), I est l'intégrale fractionnaire d'ordre a et D est la dérivée fractionnaire d'ordre b . Dans la pratique, D [L'implémentation de Jesus et Barbosa 2013] intègre à un contrôleur PD+I (un PID dont le terme D est fractionnaire d'ordre b) une base de règles de logique floue afin de combiner les termes P et D. Son équation s'exprime donc de la manière suivante :

$$u(t) = \left[f \left(K_p e(t) + K_c D^{\beta} e(t) \right) + K_i \int e(\tau) d\tau \right] K_u \quad (1.13)$$

Où f est la fonction réalisée par la base de règles floues et K_p , K_c , K_i et K_u les différents gains à régler. La figure 1.12 schématise ce contrôleur.

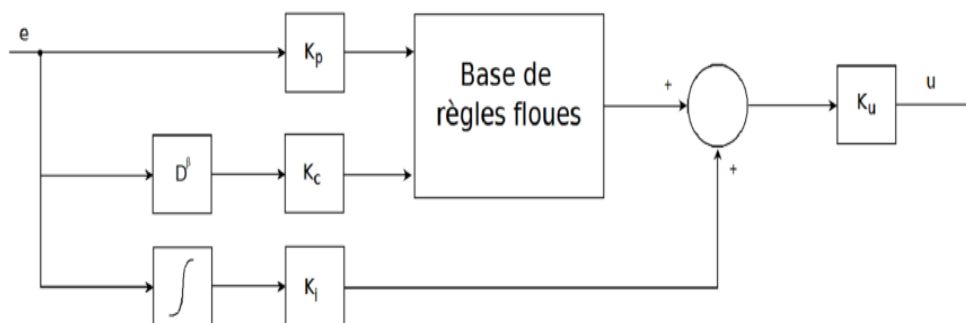


Figure 1.12 Schéma d'un contrôleur PD +I fluou.

Les contrôleurs ainsi générés et paramétrés sont plus robustes et précis que les PID classiques et sont paramétrés automatiquement.

1.3 Conclusion

Dans ce chapitre nous avons exposé le principe de contrôle et leur utilité dans le domaine de l'automatisations. Aussi les différents types de contrôles pour arriver au contrôle intelligent qui est le plus important dans notre travail.

On note une complexité des contrôleurs intelligents, qui combinent de plus en plus de techniques et les affinent. Chaque fois que des paramètres sont à estimer, on pile une nouvelle technique qui va s'en charger. Mais celle-ci apporte souvent son propre de paramètres à définir, et ainsi de suite. Cela semble en accord avec la loi de la variété qui se puisque les procédés visés sont eux aussi de plus en plus complexes.

Donc à travers l'application de la technique de l'intelligence artificielle, différents types de conceptions de contrôleurs seront étudiés dans l'état de l'art. Donc l'objectif de les appliquer au traitement d'un problème important comme par exemple dans le domaine de robot mobile c'est le suivi de trajectoire.

Le chapitre suivant sera consacré à présenter un aperçu détaillé sur le robot mobile.

Chapitre 2

2 Le robot mobile

L'objectif de ce chapitre est de donner un bref état de l'art de la robotique mobile.

2.1 Introduction

Ces dernières années, une pléthore de recherches a été menée sur le problème de contrôle des systèmes robotiques mobiles autonomes. Ceci est principalement dû à l'application croissante de ces systèmes dans les environnements industriels et de service. Quelques applications typiques de tels systèmes sont par exemple des robots de prise de commande dans des entrepôts automatisés, des robots dans des immeubles de bureaux et des robots d'exploration en profondeur. Différents types de robots peuvent être utilisés dans ces applications [1][34]. Sur les terrains difficiles, les robots marcheurs sont généralement préférés. Sur des surfaces plus lisses, les robots mobiles à roues ont l'avantage, car ils sont beaucoup plus rapides et plus agiles. D'autres types de systèmes tels que les véhicules aériens sans pilote ou véhicules non habités sous l'eau sont utilisés lorsque nous avons besoin de manœuvrer en trois dimensions les espaces. Les robots mobiles à roues sont le type de systèmes robotiques mobiles que nous sommes compte tenu de cette thèse car ils sont les plus largement utilisés dans la classe des téléphones mobiles des robots. Cela est dû à leurs manœuvres rapides, à leurs contrôleurs simples et à leurs économies d'énergie caractéristiques [2][35].

Il existe différents problèmes de commande pour le robot mobile :

- le suivi de chemin où l'objectif est qu'un point lié au robot suive une courbe prédéterminée en imposant au robot une vitesse donnée ;
- la stabilisation de trajectoires consistant à prendre en compte la dimension temporelle ;
- la trajectoire de référence dépend du temps et la vitesse du robot n'est plus fixée à l'avance ;
- la stabilisation de configurations fixes : il s'agit de stabiliser asymptotiquement le
- système dans une position d'équilibre donnée.

Dans ce travail, nous nous intéressons uniquement au problème de la poursuite de trajectoire pour un robot mobile non holonome de type voiture.

2.2 Modélisation et propriétés structurelles

Il existe de nombreux modèles de robots mobiles selon leur structure, le type de roues utilisées, etc. Nous ne présentons ici que les deux classes de modèles les plus fréquemment rencontrés : les robots de type unicycle et ceux de type voiture.

2.2.1 Définition

On note $R = (O, \vec{x}, \vec{y}, \vec{z})$ un repère fixe quelconque, dont l'axe \vec{z} est vertical et $R' = (O', \vec{x}', \vec{y}', \vec{z}')$ un repère mobile lié au robot. On choisit généralement pour O' un point remarquable de la plate-forme, typiquement le centre de l'axe des roues motrices s'il existe, comme illustre à la figure 2.1.

Par analogie avec la manipulation, on appelle situation [Fourquet 99] ou souvent posture [Campion 96] [36] du robot le vecteur:

$$\xi = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (2.1)$$

Où x et y sont respectivement l'abscisse et l'ordonnée du point (\vec{x}', \vec{y}') . La situation du robot est donc définie sur un espace M comparable à l'espace opérationnel d'un manipulateur plan.

La configuration [Neimark 72][37] d'un système mécanique est connue quand la position de tous ses points dans un repère donne est connue. Alors que pour un bras manipulateur cette notion est définie sans ambiguïté par les positions angulaires des différentes articulations, on peut, dans le cas d'un robot mobile, donner une vision plus ou moins fine de la configuration, comme on le verra par la suite. Dans tous les cas, on définira La configuration du robot mobile par un vecteur :

$$q = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ \cdot \\ q_n \end{pmatrix} \quad (2.2)$$

de n coordonnées appelées coordonnées généralisées. La configuration est ainsi définie sur un espace N de dimension n , appelée l'espace des La configuration.

2.2.2 Roulement sans glissement et contraintes non holonomes

2.2.2.1 Roulement sans glissement

La locomotion à l'aide de roues exploite la friction au contact entre roue et sol. Pour cela, la nature du contact (régularité, matériaux en contact) a une forte influence sur les propriétés du mouvement relatif de la roue par rapport au sol. Dans de bonnes conditions, il y a roulement sans glissement (r.s.g.) de la roue sur le sol, c'est-à-dire que la vitesse relative de la roue par rapport au sol au point de contact est nulle. Théoriquement, pour vérifier cette condition, il faut réunir les hypothèses suivantes:

- le contact entre la roue et le sol est ponctuel ;
- les roues sont indéformables, de rayon r .

En pratique le contact se fait sur une surface, ce qui engendre bien évidemment de légers glissements. De même, alors qu'il est raisonnable de dire que des roues pleines sont indéformables, cette hypothèse est largement fautive avec des roues équipées de pneus. Malgré cela, on supposera toujours qu'il y a r.s.g. et, par ailleurs, que le sol est parfaitement plan.

Mathématiquement, on peut traduire la condition de r.s.g. sur une roue. Soit P le centre de la roue, Q le point de contact de la roue avec le sol, φ l'angle de rotation propre de la roue et θ l'angle entre le plan de la roue et le plan (Q, \vec{x}, \vec{z}) comme indique à la figure 2.1. La nullité de la vitesse relative v_q roue/sol au point de contact permet d'obtenir une relation vectorielle entre la vitesse v_p du centre P de la roue et le vecteur vitesse de rotation ω de la roue:

$$\vec{v}_Q = \vec{v}_P + \vec{\omega} \wedge \vec{PQ} = 0$$

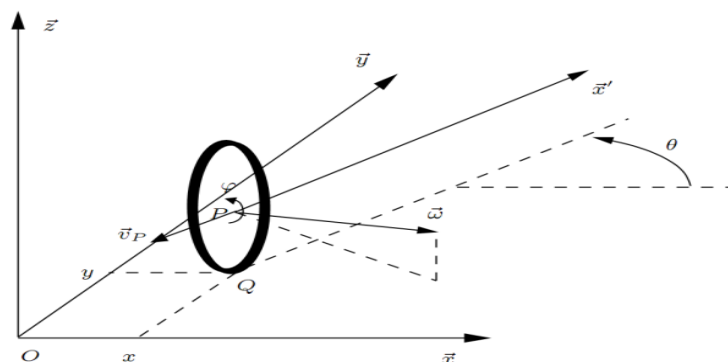


Figure 2.1 Caractérisation du roulement sans glissement

Les points P et Q ont pour coordonnées respectives $(x \ y \ r)^T$ et $(x \ y \ 0)^T$. Il vient alors :

$$\begin{aligned} \dot{x}\vec{x} + \dot{y}\vec{y} + \left(\dot{\theta}\vec{z} + \dot{\phi}(\sin\theta\vec{x} - \cos\theta\vec{y}) \right) \wedge (-r\vec{z}) &= 0, \\ (\dot{x} + r\dot{\phi}\cos\theta)\vec{x} + (\dot{y} + r\dot{\phi}\sin\theta)\vec{y} &= 0 \end{aligned} \quad (2.3)$$

Ceci nous donne le système de contraintes scalaires :

$$\begin{aligned} \dot{x} + r\dot{\phi}\cos\theta &= 0 \\ \dot{y} + r\dot{\phi}\sin\theta &= 0 \end{aligned} \quad (2.4)$$

que l'on peut transformer pour faire apparaître les composantes de vitesse dans le plan de la roue d'une part et perpendiculairement à la roue d'autre part :

$$\begin{aligned} -\dot{x}\sin\theta + \dot{y}\cos\theta &= 0, \\ \dot{x}\cos\theta + \dot{y}\sin\theta &= -r\dot{\phi}. \end{aligned} \quad (2.4)$$

Ces contraintes traduisent le fait que le vecteur \vec{v}_p soit dans le plan de la roue et ait pour module $r\dot{\phi}$.

2.2.2.2 Contraintes non holonomes

Les équations précédentes, caractérisant le r.s.g. d'une roue sur le sol, sont des contraintes non holonomes. Nous nous proposons dans ce paragraphe de préciser ce que recouvre ce terme et de caractériser les systèmes non holonomes.

Soit un système de configuration q soumis à des contraintes indépendantes sur les vitesses, regroupées sous la forme (pfaffienne) $A^T(q)\dot{q} = 0$. S'il n'est pas possible d'intégrer l'une de ces contraintes, elle est dite non intégrable ou non holonome. De manière concrète l'existence de contraintes non holonomes implique que le système ne peut pas effectuer certains mouvements instantanément. Par exemple, dans le cas de la roue, il ne peut y avoir de translation instantanée parallèlement à l'axe de la roue. Un tel déplacement nécessitera des manœuvres. De même, comme on le sait bien, une voiture ne peut se garer facilement sans effectuer de créneaux.

Il n'est pas évident de dire a priori si une contrainte est intégrable ou non. Pour cela, on a recours à l'application du théorème de Frobenius, dont une version complète pourra être trouvée dans un ouvrage de référence de géométrie différentielle [Warner 83] ou de commande non-linéaire [Nijmeijer 90][38]. Seule la connaissance du crochet de Lie est nécessaire à notre étude. Pour deux vecteurs $b_i(q)$ et $b_j(q)$, cet opérateur est défini par :

$$[b_i(q), b_j(q)] = \frac{\partial b_j}{\partial q} b_i - \frac{\partial b_i}{\partial q} b_j \quad (2.4)$$

Théorème 2.1 Soit un système de configuration q , de dimension n , soumis à un ensemble de contraintes indépendantes s'écrivant sous la forme $A^T(q)\dot{q} = 0$. Soit $B(q) = (b_1(q) \ b_2(q) \ \dots \ b_m(q))$ une matrice de rang plein m , orthogonale à $A(q)$ sur tout l'espace des configurations. Soit enfin l'algèbre de Lie de dimension p , avec $m \leq p \leq n$, engendrée par l'ensemble des colonnes de $B(q)$, auxquelles s'ajoutent les crochets de Lie successifs formés à partir de ces colonnes, à condition qu'ils augmentent la dimension de l'algèbre.

Alors, parmi les contraintes auxquelles est soumis le système, n prenons l'exemple de la roue dont on a écrit le modèle précédemment. Sa configuration est entièrement définie par sa configuration $q = (x \ y \ \theta \ \varphi)^T$. D'après (2.3) et (2.4) on peut déterminer :

$$A(q) = \begin{pmatrix} -\sin \theta & \cos \theta \\ \cos \theta & \sin \theta \\ 0 & 0 \\ 0 & r \end{pmatrix}$$

et en déduire :

$$B(q) = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \\ -\frac{1}{r} & 0 \end{pmatrix}$$

matrice formée par les colonnes indépendantes $b_1 = (\cos \theta \ \sin \theta \ 0 \ -\frac{1}{r})^T$ et $b_2 = (0 \ 0 \ 1 \ 0)^T$ et qui vérifié bien $A^T(q)B(q) = 0$ pour tout q . On calcule alors les crochets de Lie successifs à partir de $b_1(q)$ et $b_2(q)$:

$$\begin{aligned} [b_1(q), b_2(q)] &= (\sin \theta \ -\cos \theta \ 0 \ 0)^T, \\ [b_1(q), [b_1(q), b_2(q)]] &= 0, \\ [b_2(q), [b_1(q), b_2(q)]] &= (\cos \theta \ \sin \theta \ 0 \ 0)^T. \end{aligned} \quad (2.5)$$

On constate que $b_1(q)$; $b_2(q)$; $[b_1(q); b_2(q)]$ et $[b_2(q); [b_1(q); b_2(q)]]$ engendrent une algèbre de Lie de dimension 4, égale à celle de q . On arrête donc le calcul des crochets et on peut conclure qu'il n'existe pas de contrainte intégrable. Les contraintes (2.3) et (2.4) sont donc des contraintes non holonomes.

2.2.3 Les grandes classes de robots mobiles et leurs modèles

2.2.3.1 Disposition des roues et centre instantané de rotation

Nous pouvons classer le robot mobile à roues en deux catégories en fonction de leur type de roue: les roues conventionnelles et roues suédoises.

Pour la roue conventionnelle, le contact entre la roue et le sol est supposé satisfaire le roulement pur sans contrainte de glissement. Cela implique que la vitesse de la Le point de contact est égal à zéro et les composantes de cette vitesse sont parallèles et orthogonales. Au plan de roue sont égaux à zéro. Une forme schématique d'une roue conventionnelle est illustrée à la figure 2.2:

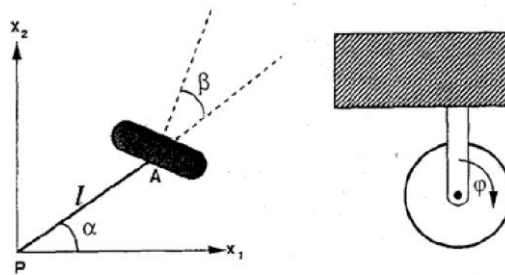


Figure 2.2 la forme de roue conventionnelle

Il existe trois types de roues conventionnelles: roues fixes, roues orientables centrées et des roues orientables excentrées. Ces trois types de roues conventionnelles sont illustrés dans la figure 2.3:

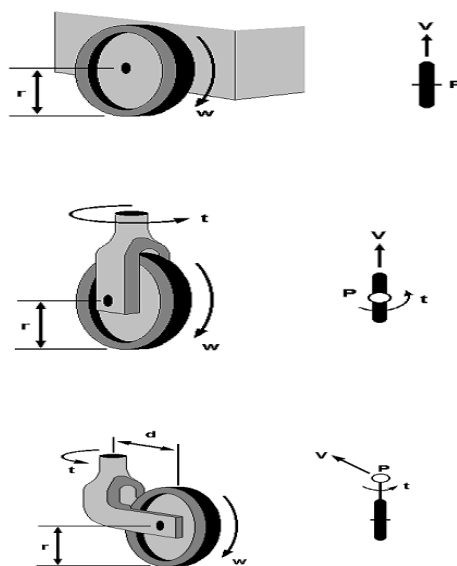


Figure 2.3 composante de vitesse des roues convontionnels

Comme on peut le voir sur la figure ci-dessus, la composante de vitesse qui est orthogonale au plan de roue est nul dans tous les types ci-dessus des roues conventionnelles.

Pour une roue suédoise, une seule composante de la vitesse du point de contact de la roue avec le sol est supposée être égale à zéro le long du mouvement. La direction de cette composante zéro est arbitraire mais est fixe par rapport à l'orientation de la roue comme peut être vu dans la figure 2.4:

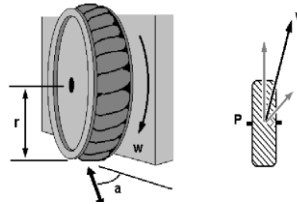


Figure 2.4: la forme schématique d'une roue suédoise

Différents types de roues peuvent être utilisés en fonction de l'application et des types de environnement. L'autre problème important concernant les robots mobiles à roues est leur type de locomotion. Les robots mobiles à roues peuvent avoir les cinq types de système de locomotion suivants:

- Entraînement différentiel
- Roues directrices: tricycle, vélos et wagon
- Entraînement synchrone
- Omni-directionnel
- Voiture (direction Ackerman)

Les différents types de systèmes de locomotion ci-dessus pour robots mobiles à roues sont illustrés ci-dessous figure 2.5:

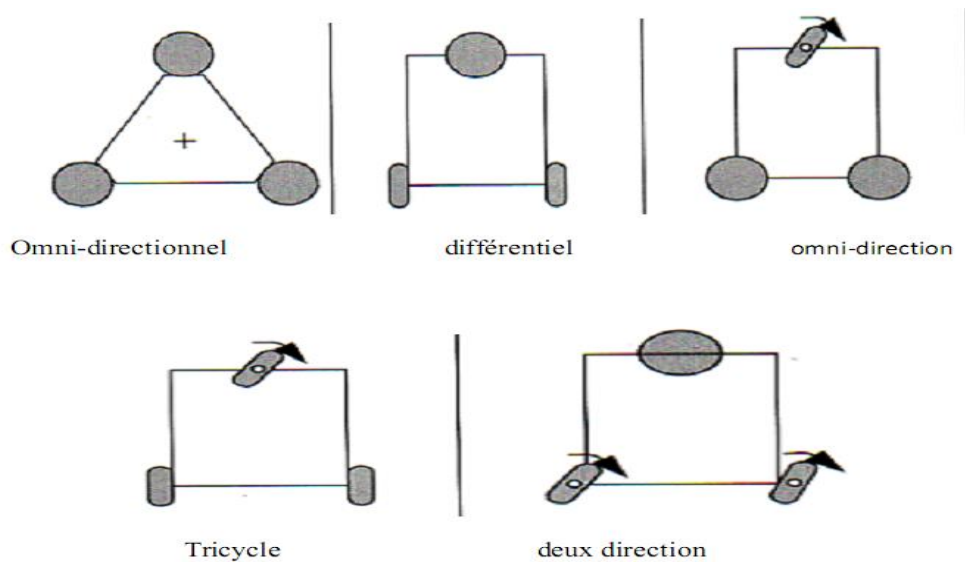


Figure 2.5 différents types de locomotion pour les robots mobiles à roues

Le robot mobile à roues à entraînement différentiel est le mobile le plus simple et le plus populaire robot qui a le mécanisme d'entraînement le plus simple. C'est une plate-forme équipée d'une façade roulette et une paire de roues motrices coaxiales arrière. Chacune de ces roues motrices sont entraînée indépendamment par un moteur à courant continu qui est à son tour alimenté par une tension de commande.

Il est connu que la stabilisation de robots mobiles non holonomiques à roues avec des restrictions la mobilité à un état d'équilibre est généralement difficile à quitter. Un travail bien connu de Brockett [3] [39] identifie les systèmes non holonomiques comme une classe de systèmes qui ne peuvent pas être stabilisé via un retour d'état en douceur. Cela implique que les problèmes de contrôle systèmes non holonomiques ne peuvent pas être appliqués aux méthodes de la théorie du contrôle linéaire, et ils ne sont pas transformables en problèmes de contrôle linéaires. En raison de leur richesse et de leur richesse dureté, de tels problèmes de contrôle non linéaires ont motivé un grand nombre de chercheurs impliquant diverses techniques de commande automatique [4][40].

Bien évidemment, pour un ensemble de roues donné, toute disposition ne conduit pas à une solution viable. Un mauvais choix peut limiter la mobilité du robot ou occasionner d'éventuels blocages. Par exemple, un robot équipé de deux roues fixes non parallèles ne pourrait pas aller en ligne droite ! Pour qu'une disposition de roues soit viable et n'entraîne pas de glissement des roues sur le sol, il faut qu'il existe pour toutes ces roues un unique point de vitesse nulle autour duquel tourne le robot de façon instantanée. Ce point, lorsqu'il existe,

est appelé centre instantané de rotation (CIR). Les points de vitesse nulle liés aux roues se trouvant sur leur axe de rotation, il est donc nécessaire que le point d'intersection des axes de rotation des différentes roues soit unique. Pour cette raison, il existe en pratique trois principales catégories de robots mobiles à roues, que l'on va présenter maintenant.

2.2.3.2 Robots mobiles de type tricycle et de type voiture

Généralement pour la commande de robots mobiles, un modèle de commande en vitesse est utilisé plutôt qu'un modèle de commande en couple. Les principales raisons de ce choix sont les suivantes :

- Le calcul de la commande est plus simple pour un modèle cinématique que pour un modèle dynamique.
- Il n'y a pas de paramètres géométriques ou inertiels compliqués à identifier pour un modèle cinématique.

Pour le robot utilisé en simulation, nous ne considérons que des modèles cinématiques en prenant en compte les hypothèses simplificatrices suivantes :

- le robot mobile est considéré comme un véhicule rigide évoluant dans un plan horizontal ;
- Les roues conventionnelles sont supposées indéformables, de rayon notée ci-dessous r ;
- chaque contact roue/sol est réduit à un point ;
- les roues roulent sans glisser sur le sol.

2.2.3.2.1 Présentation des robots de type voiture

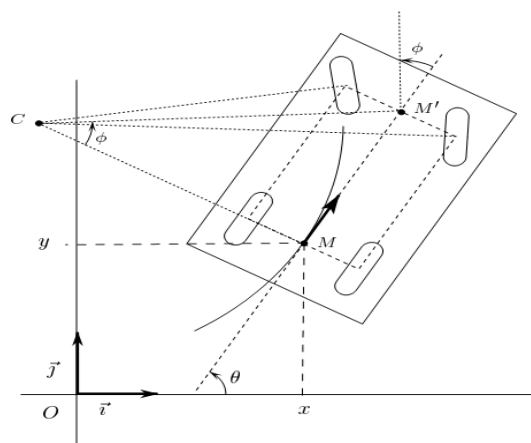


Figure 2.6 robot de type voiture

Nous présentons maintenant quelques systèmes dynamiques modélisant une voiture. Nous voulons seulement rendre compte des propriétés géométriques des trajectoires faisables par un robot de type voiture. Dans ce cadre, pour une modélisation relativement fine d'une voiture, il est naturel de considérer que le conducteur contrôle l'accélération linéaire des roues ainsi que la vitesse angulaire des roues. Un vecteur d'état du véhicule est alors, a priori, défini par (x, y, θ, v, ϕ) où x et y sont les coordonnées du milieu de l'essieu arrière pris comme point de référence du robot, θ détermine la direction du véhicule et, informellement, v représente sa vitesse linéaire et ϕ l'angle que font les roues avant avec l'axe du véhicule (voir figure 1.1). La contrainte sur le rayon de giration du véhicule se traduit par $|\phi| \leq \phi_0$. Si α et β sont les valeurs de l'accélération linéaire des roues et de la vitesse angulaire des roues, à un instant donné, les contraintes cinématiques s'écrivent :

$$\begin{cases} \dot{x} = v \cos \phi \cos \theta \\ \dot{y} = v \cos \phi \sin \theta \\ \dot{\theta} = v \sin \phi \\ \dot{v} = \alpha \\ \dot{\phi} = \beta \end{cases} \quad (2.6)$$

La contrainte cinématique correspondant à un roulement sans glissement se traduit par le fait que la vitesse du robot est tangente à l'axe du robot c'est à dire :

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0$$

Cette équation n'est pas intégrable et caractérise la non-holonomie du système (voir [Lat91]).

Un robot satisfaisant aux contraintes cinématiques du système (1.1) (avec $|\phi| \leq \phi_0$) modélise bien une voiture, cependant ce modèle s'avère encore trop détaillé. En effet, la caractérisation des plus courts chemins pour un tel système n'a pu être effectuée, à l'heure actuelle. Certains systèmes simplifiés ont cependant été étudiés : considérons le système dynamique formé uniquement par les trois premières équations du système (1.1). En posant $u_1 = v \cos \phi$ et $u_2 = v \sin \phi$, nous obtenons :

$$\begin{cases} \dot{x} = u_1 \cos \theta \\ \dot{y} = u_1 \sin \theta \\ \dot{\theta} = u_2 \end{cases} \quad (2.7)$$

Où u_1 et u_2 sont des lois de commande. Il est raisonnable de supposer que la borne -0 sur le rayon maximal de giration est inférieure à $\pi/4$ ce qui implique $|u_2| \leq |u_1|$. D'autre part, dans la mesure où nous ne considérons que des contraintes cinématiques et aucune contrainte dynamique dans la modélisation du robot, nous pouvons borner la vitesse linéaire v par une constante arbitraire, ceci sans aucune perte de généralité. Nous pouvons donc supposer $|u_2| \leq |u_1| \leq 1$.

On appelle robot de type voiture ou unicycle un robot dont les trajectoires satisfont le système (1.2) et dont les lois de commande u_1 et u_2 sont à valeur dans $\{(u_1; u_2) \in \mathbb{R}^2 / |u_2| \leq |u_1| \leq 1\}$. Notons que la courbure d'une trajectoire satisfaisant au système (1.2) est, en un point où elle est définie, $\frac{u_2}{u_1} \leq 1$. Le système (1.2) est appelé système de Dubins si $u_1 = 1$ et $|u_2| \leq 1$, il est appelé système de Reeds et Shepp si $u_1 \in \{-1, 1\}$ et $|u_2| \leq 1$.

On Appelle robot de Dubins ou robot de Reeds et Shepp un robot satisfaisant aux contraintes cinématiques respectivement décrites par le système de Dubins ou le système de Reeds et Shepp. La figure 2.7 montre les domaines de commande pour les robots de Dubins, de Reeds et Shepp et les robots de type voiture.

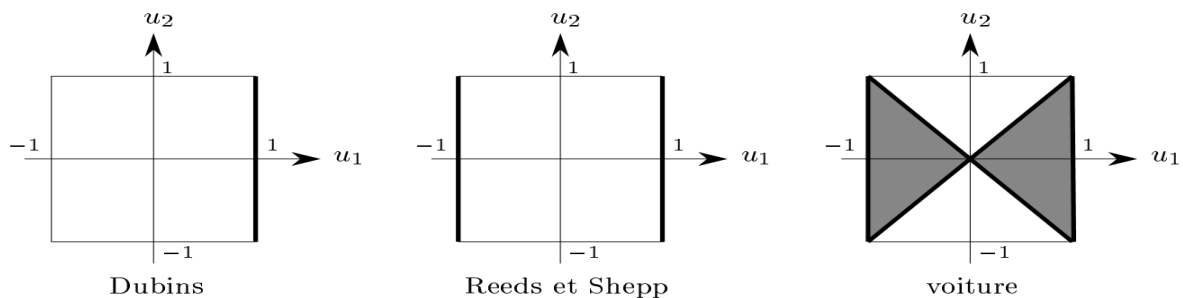


Figure 2.7 Domaines de commande correspondant aux différents systèmes

Pour ces deux systèmes, un triplet (x, y, θ) est appelé configuration. L'espace des configurations est $\mathcal{R}^2 * S^1$. Une conguration peut être vue soit comme un point de $\mathcal{R}^2 * S^1$, soit comme un point orienté du plan, c'est-à-dire un point du plan muni d'une orientation.

Dans la suite nous nous intéresserons tout particulièrement au système de Dubins

$$\begin{cases} \dot{x} = \cos \theta \\ \dot{y} = \sin \theta \\ \dot{\theta} = u_2 \quad |u_2| \leq 1 \end{cases} \quad (2.8)$$

Une courbe dans $\mathcal{R}^2 * S^1$, joignant deux configurations ω_1 et ω_2 , et satisfaisant aux équations du système (1.3) sera souvent confondue avec sa projection sur \mathcal{R}^2 , c'est-à-dire avec la courbe décrite par le point de référence du robot entre les deux points orientés du plan ω_1 et ω_2 . Remarquons qu'une telle courbe est une courbe C^1 de courbure $|u_2|$ majorée par 1 en tout point où la courbure est définie. Une telle courbe est également une courbe dont la courbure moyenne est majorée par 1 sur tout intervalle, c'est-à-dire une courbe différentiable dont la dérivée est une fonction 1-lipschitzienne lorsque la courbe est paramétrée par l'abscisse curviligne. Réciproquement, une courbe de courbure moyenne majorée par 1, de même qu'une courbe C^1 de courbure majorée par 1 en tout point où elle est définie, satisfait aux contraintes du système de Dubins. Fréquemment dans la suite, de telles courbes seront simplement appelées courbes de courbure bornée. Lorsque l'on parle de la longueur d'une trajectoire, il s'agit toujours de sa longueur dans \mathcal{R}^2 , c'est-à-dire de la distance parcourue par le point de référence de la voiture.

Nous avons défini la borne sur la courbure par $|u_2| \leq 1$. Dans certains ouvrages, la courbure est supposée bornée par une constante $1/R$ (R est le rayon de courbure minimal le long d'une trajectoire). Remarquons qu'il est équivalent, à une homothétie près, de considérer la courbure bornée par 1 ou par une constante positive $1/R$.

2.2.3.2 Robots mobiles de type unicycle

C'est un robot actionné par deux roues indépendantes et possédant éventuellement un certain nombre de roues folles assurant sa stabilité.

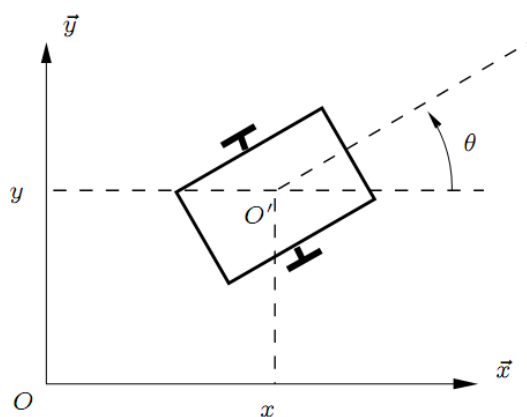


Figure 2.8 Robot mobile de type unicycle

Ce type de robot est très répandu en raison de sa simplicité de construction et de propriétés cinématiques intéressantes.

Centre instantané de rotation, Les roues motrices ayant même axe de rotation, le CIR du robot est un point de cet axe. Soit le rayon de courbure de la trajectoire du robot, c'est-à-dire la distance du CIR au point 0 (voir figure 2.6). Soit L l'entre-axe et ω la vitesse de rotation du robot autour du CIR. Alors les vitesses des roues droites et gauche, respectivement notées v_d et v_g et définies à la figure 2.6, vérifient :

$$\begin{aligned} v_d &= -r\dot{\phi}_d = (\rho + L)\omega \\ v_g &= r\dot{\phi}_g = (\rho - L)\omega \end{aligned} \quad (2.9)$$

Ce qui permet de déterminer ρ et ω à partir des vitesses des roues :

$$\begin{aligned} \rho &= L \frac{\dot{\phi}_d - \dot{\phi}_g}{\dot{\phi}_d + \dot{\phi}_g} \\ \omega &= -\frac{r(\dot{\phi}_d + \dot{\phi}_g)}{2L} \end{aligned} \quad (2.10)$$

L'équation (2.9) permet de situer le CIR sur l'axe des roues. Par ailleurs ces équations expliquent deux propriétés particulières du mouvement des robots de type unicycle : $\dot{\phi}_d = -\dot{\phi}_g$, le robot se déplace en ligne droite ; si $\dot{\phi}_d = \dot{\phi}_g$, alors le robot effectue une rotation sur lui-même. L'utilisation de ces deux seuls modes de locomotion, bien que limitée, permet de découpler les mouvements et de fournir une solution simple pour amener le robot d'une posture à une autre. C'est sans doute à une des raisons du succès de ce type de robots. Pour élaborer une stratégie plus fine de déplacement, il est cependant intéressant de savoir comment la posture du robot est reliée à la commande de ses roues.

2.3 Planification de trajectoire

On considère un robot mobile A se déplaçant dans un espace de travail W , l'objectif est de trouver les chemins qui relient la position du départ du robot P_i à sa position finale P_f . Ces chemins doivent être réalisables par le robot, c'est-à-dire, respecter les contraintes cinématiques du robot /LaValle 2006/[41], /Laumond 1998/[42], /Dombre et Khalil 2007/[43], /Scheuer 1998/[44]. Les chemins doivent aussi éviter des obstacles fixes/ Defoort 2007-a/[45] ou mobiles /Sharir 1989/[46]. Dans cette section, les contraintes cinématiques à prendre en compte dans la planification sont présentées, puis les principales approches pour définir des trajectoires acceptables sont développées.

2.3.1 Approches polynomiales pour la génération de trajectoires

Ces méthodes permettent en particulier de générer des trajectoires à courbure continue pour réaliser diverses manœuvres (parking,...). Dans tous les cas, la trajectoire est conçue en prenant en compte les contraintes cinématiques du robot mobile ainsi que la position des différents obstacles statiques à éviter/Haddad 2007/[47] /De Luca 1991/ [48]ou B-spline /Gomez-Bravo 2008/[49] /Wang 1988/[50]. Une trajectoire générée par une courbe B-spline peut-être représentée à titre d'exemple comme sur la figure 2.9.

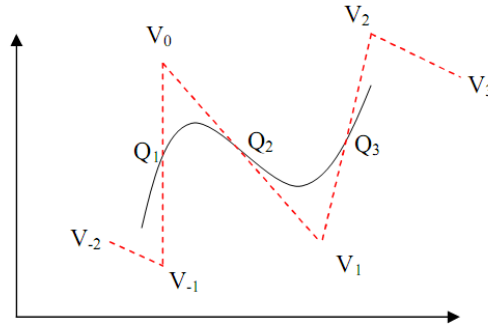


Figure 2.9 Courbe B-spline

Sur cette figure, Q1, Q2 et Q3 représentent les points de passage de la courbe et V-2, V-1, V0, V1, V2 et V3 sont les points qui forment le polygone de contrôle de la courbe. En pratique pour les robots mobiles de type unicycle ou véhicule, les B-splines cubiques s'avèrent être un choix suffisant pour définir une trajectoire vérifiant les contraintes cinématiques /Munoz 1992/[51]. Les coefficients du polynôme permettant de construire une courbe B-spline cubique sont obtenus à l'aide de l'expression suivante :

$$\begin{bmatrix} b_{-2}(u) \\ b_{-1}(u) \\ b_0(u) \\ b_1(u) \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{6} \\ \frac{2}{3} & 0 & -1 & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{6} \end{bmatrix} \times \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}$$

et les coordonnées $Q_i(u)$ du i -ième point de la courbe pour V_{i+r} avec $r = -2, -1, 0, 1$ sont données par la formule suivante :

$$Q_i(u) = \sum_{r=-2}^1 b_r(u)V_{i+r}, \quad i = 2, 3, \dots$$

avec le paramètre $u \in [0, 1]$. La trajectoire est définie en précisant la dépendance de u avec le temps (loi $u=u(t)$).

Une autre famille de trajectoires fréquemment utilisées pour la planification des robots mobiles non-holonomes est basée sur l'utilisation des courbes de Bézier /Demengel 1998/[52]. Un exemple de courbe de Bézier cubique est représenté sur la figure 2.10.

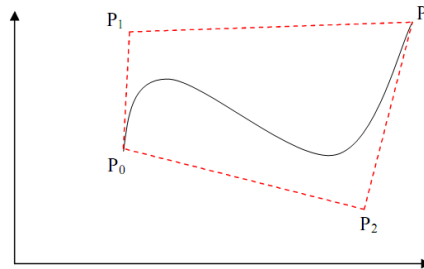


Figure 2.10 Courbe de Bézier cubique

Les points P_0, P_1, P_2 et P_3 constituent le polygone de contrôle de la courbe. Elle est obtenue à partir de la formule suivante :

$$O\vec{M}(u) = \sum_{i=0}^n B_n^i(u) O\vec{P}_i \quad (2.11)$$

Où $B_n^i(u) = C_n^i(1-u)^{n-i}$ est un polynôme de Bernstein, les P_i (pour $i= 1, 2, \dots$) sont les points de contrôle de la courbe et le paramètre $u \in [0, 1]$.

2.3.2 Décomposition en cellules

Cette méthode consiste à décomposer l'espace W d'évolution du robot (espace de configuration) en plusieurs régions adjacentes qui correspondent à une représentation discrète de l'environnement /Latombe 1991/[53]. A partir de la décomposition obtenue, les barycentres de chaque cellule sont calculés (figure 2.11 (a)) et, en utilisant la théorie de graphe, la génération d'un graphe reliant les barycentres compris entre les configurations initiales et finales du robot (figure 2.12 (b)) est réalisée.

Les points P_i et P_f sont respectivement les configurations initiale et finale du robot. Le graphe déterminé est alors exploré pour calculer le chemin permettant de minimiser un critère donné. Généralement dans le cas de robots mobiles, le critère peut-être le temps de parcours ou la détermination du plus court chemin par l'utilisation d'algorithmes d'optimisation : Dijkstra /Tournassoud 1992/[54], A^* /Stuart 1994/ ou D^* /Stentz 1995/[55].

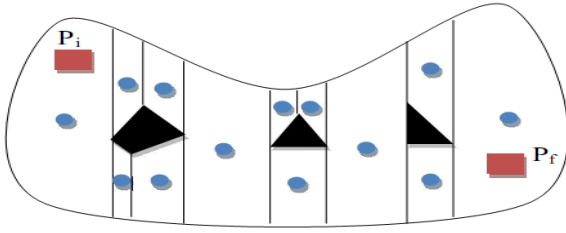


Figure 2.11 (a) : Décomposition en cellules

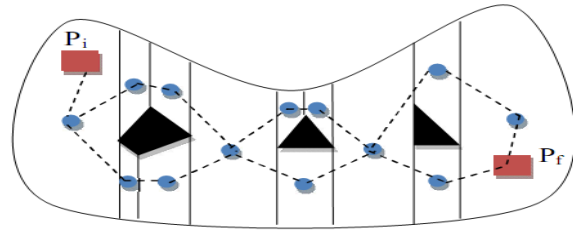


Figure 2.12 (b) : Génération du graphe

2.3.3 Approche par champs de potentiel

Dans cette approche, on considère que le robot roule dans un champ de forces virtuelles. Ce dernier est composé de deux champs : un champ de potentiel répulsif autour des obstacles et un champ de potentiel attractif dont le point d'attraction est la configuration finale du robot (figure 2.13).

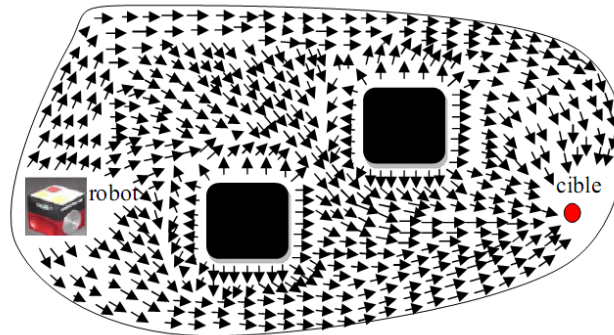


Figure 2.13 Champs de potentiel répulsif et attractif.

Sachant que q est la configuration courante du robot, la commande du robot mobile permettant d'éviter un obstacle noté 'O' et d'atteindre la configuration désirée (cible) q_d est donnée par le champ de potentiel artificiel composé du champ de potentiel attractif et répulsif /Khatib 1986/[56] /Khalil 1999/[57]. Dans le cas d'évitement de plusieurs obstacles 'O_i' ($i = 1, \dots, n$) n est le nombre des obstacles, la commande à envoyer au robot est composée de la force résultante du champ de potentiel attractif et de la somme des forces qui résultent des champs de potentiel répulsifs des différents obstacles.

Cette technique se met en œuvre relativement facilement et peut être adaptée au cas d'obstacles mobiles. Par contre, le robot peut rester bloqué en un minimum local ou osciller lors du franchissement d'un passage étroit entre des obstacles. Plusieurs adaptations de cette technique ont été proposées. Par exemple, pour éviter que le robot ne soit bloqué (minima locaux), les auteurs /Koren 1991/[58], /Barraquand 1992/[59] et

/Kavraki 1996/ [60] proposent d'ajouter un point intermédiaire aléatoire afin de pouvoir sortir le robot de sa trappe.

2.3.4 Utilisation des roadmaps

Les méthodes de planification nommées 'roadmaps' sont basées sur l'utilisation de la théorie des graphes. Le principe est de construire un graphe reliant la configuration initiale à la configuration finale tout en évitant les obstacles. Ces techniques sont parmi les plus utilisées et les plus développées ces dernières années de part le fait que ce sont des planificateurs globaux. Deux types de méthodes de cette famille sont développés dans cette section.

2.3.4.1 Diagramme de Voronoï

Les diagrammes de Voronoï sont des structures de données fondamentales largement étudiés en géométrie /Wormser 2008/[61]. Dans le cadre de la planification de trajectoire, le diagramme de Voronoï est construit, en traçant des lignes d'égale distance aux obstacles. Une fois que le graphe est construit, l'objectif est de déterminer le chemin reliant la configuration initiale du robot à la configuration finale dans ce graphe en minimisant un critère de distance (le plus court chemin). Plusieurs adaptations pour généraliser le graphe de Voronoï ont été proposées par /Nagatani 1999/[62] et /Choset 2000/[63] dans le cadre de planification de trajectoire dans des environnements inconnus.

2.3.4.2 Graphe de visibilité

Le graphe de visibilité d'un ensemble d'objets est un graphe où pour chaque sommet d'objet, on trace une ligne droite entre celui-ci et les sommets des autres objets que l'on peut voir /Srinivasaraghavan 1994/[64]. Le graphe de visibilité est utilisé dans plusieurs domaines notamment dans les études de marquage, d'évitement d'obstacles, de calcul du plus court chemin, etc. Un algorithme pour la construction d'un graphe de visibilité réduite est proposé par /Sridharan 2004/[65] dans le contexte particulier d'un environnement polygonal convexe. Depuis la configuration initiale, on considère les droites reliant le point de départ aux sommets visibles des obstacles, puis l'opération est réitérée entre les sommets des obstacles et les sommets visibles des suivants jusqu'au point d'arrivée (figure 2.14).

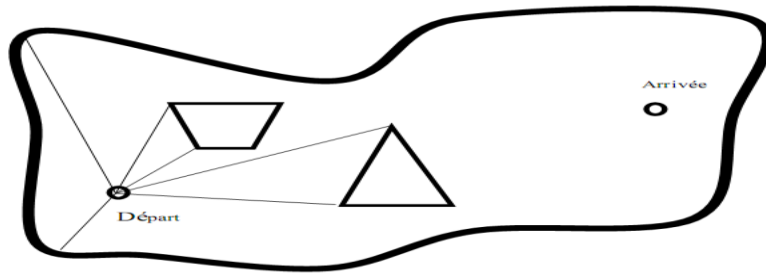


Figure 2.14 Graphe de visibilité d'un environnement polygonale

Une fois que le graphe de visibilité est construit, le plus court chemin reliant la position de départ et la position d'arrivée dans l'espace morcelé en cellules est calculé à l'aide d'algorithmes d'optimisation tels que, par exemple, l'algorithme de Dijkstra.

2.4 Poursuite de trajectoire

Le problème de suivi d'une trajectoire de référence pour un robot mobile non-holonome est apparu comme un problème de premier ordre pour la communauté roboticienne dans ces dernières années. En effet, la forte utilisation des robots mobiles dans les domaines où l'être humain ne peut pas être présent, notamment dans les sites nucléaires à haut risque ou dans le cas de l'exploration spatiale, nécessite la mise en œuvre de lois de commande autonomes et performantes pour assurer les tâches assignées aux robots. Plusieurs travaux concernant la poursuite de trajectoire ont été développés dans ce contexte.

2.4.1 Formulation du problème

Considérons un robot mobile représenté par un modèle non linéaire de la forme :

$$\dot{q} = f(q, u) \quad (2.12)$$

où $q \in R^n$ est l'état du système, $u \in R^n$ la commande du système et $f: R^n \times R^n \rightarrow R^n$ une application infiniment différentiable.

Etant donnée une trajectoire de référence admissible r_q , l'objectif de la poursuite de trajectoire est de trouver la commande u du système (1.11) en fonction de q , q_r et du temps t telle que l'erreur de poursuite $e = r_q - q$ converge asymptotiquement vers zéro. Dans le cadre d'un robot mobile non holonome de type unicycle, le principe de la poursuite de trajectoire est illustré sur la figure 2.15

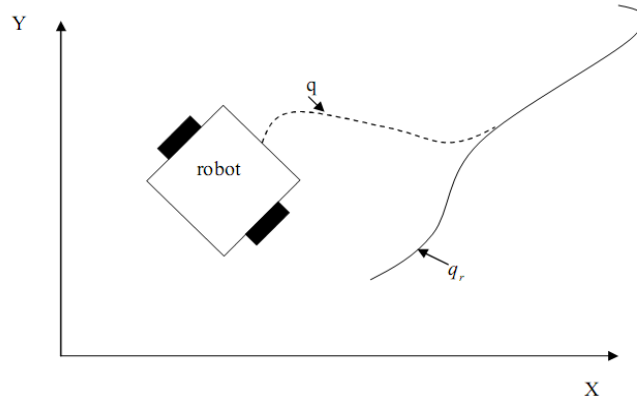


Figure 1.15 Poursuite de trajectoires admissibles

Une propriété des modèles de robot mobile non-holonyme (de type voiture ou unicycle) est qu'ils ne sont pas stabilisables par un retour d'état continu et stationnaire. Ce corollaire est le résultat des travaux de /Brockett 1983/[66].

Théorème 1.1 /Brockett 1983/ [67]: Considérons le système non linéaire (1.11) où f est suffisamment différentiable et $f(0, 0) = 0$. Une condition nécessaire pour qu'il existe un retour d'état $u(q)$ continu qui rende l'origine du système bouclé localement stable est que l'image par f de tout voisinage de l'origine $R^{n \times m}$ soit un voisinage de 0 dans R^n .

La condition nécessaire énoncée dans ce théorème n'est pas satisfaite dans le cas des robots mobiles de type unicycle ou voiture dont les modèles ont été donnés en (1.3) et (1.5). plusieurs travaux ont été développés pour résoudre ce problème en proposant des commandes avec un retour d'état non stationnaire continu /Morin 2008[68], Oriolo 2002[69] et Samson 1991/ [70].

2.4.2 Stabilisation par un retour d'état non stationnaire continu

2.4.2 .1 Commande non linéaire non stationnaire

Une autre façon de considérer le problème de suivi de trajectoire est de développer des lois de commande non linéaires non stationnaires. On suppose que la trajectoire de référence est réalisable par le robot mobile. Dans les travaux de /Kanayama 1991/[71], une loi de commande stabilisante basée sur le modèle de l'erreur de suivi de trajectoire suivant est proposée et est détaillée dans le chapitre 2 :

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \cos(e_\theta) & 0 \\ \sin(e_\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (2.13)$$

avec $e_p = [e_x, e_y, e_\theta]^T$, l'erreur de posture du robot par rapport à la trajectoire de référence, v_r et w_r sont respectivement les vitesses linéaire et angulaire de la trajectoire de référence, et v et w sont les commandes du robot, respectivement les vitesses linéaire et angulaire. Les commandes proposées pour les vitesses linéaire et angulaire du robot mobile sont données par la formule suivante :

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} v_r \cos(e_\theta) + k_x e_x \\ w_r + v_r(k_y + k_\theta e_\theta) \end{bmatrix} \quad (2.14)$$

Où, k_x, k_y et k_θ sont des coefficients positifs. La solution nulle du système (1.13) utilisant la commande (1.14) est alors stable /Kanayama 1991/ [72]/Maalouf 2006/[73]. La preuve est basée sur le choix de la fonction de Lyapunov $V = \frac{1}{2}(e_x^2 + e_y^2) + (1 - \cos e_\theta)/k_y$. On montre alors que la dérivée de V le long des trajectoires du système bouclé est négative (mais pas définie négative), prouvant la stabilité de l'équilibre considéré. Moyennant certaines hypothèses sur la trajectoire de référence (existence d'une vitesse minimale v_r non nulle et accélérations linéaire et angulaire suffisamment faibles), il est possible de montrer que la solution $e_p = 0$ est localement asymptotiquement stable.

La commande du modèle (1.13) peut être aussi traitée dans le cas discret. Pour cela un modèle linéarisé du (1.13) autour du point d'équilibre $e_x = e_y = e_\theta = 0, u_B = 0$ est calculé et il est donné par l'équation

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} 0 & w_r & 0 \\ -w_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_B \quad (2.15)$$

En notant T_e le pas d'échantillonnage, le modèle discret du modèle (1.15) est donné par :

$$\begin{bmatrix} e_x(k+1) \\ e_y(k+1) \\ e_\theta(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_e w_r & 0 \\ -T_e w_r & 1 & T_e v_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_x(k) \\ e_y(k) \\ e_\theta(k) \end{bmatrix} + T_e \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_B(k) \quad (2.16)$$

La commande par retour d'état pour le modèle discret linéarisé (1.16) proposée dans /Klancar et Skrjanc, 2007/ est la suivante :

$$u_b(k) = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & \text{sign}(v_r(k))k_2 & k_3 \end{bmatrix} e(k)$$

Pour le calcul des gains, les auteurs utilisent une technique de placement de pôles. Soient $Z_1, Z_2, et Z_3$ les pôles désirés, alors les gains $k_1, k_2 et k_3$ sont données par les expressions :

$$k_1 = k_3 = \frac{z_1 + z_2 + z_3 + 3}{2T_e}$$

$$k_2 = \frac{-1/4(z_{123} + 3)^2 - T_e^2 \omega_r^2 + 2z_{123} + 3 + z_1 z_2 + z_3 z_1 + z_3 z_2}{T_e^2 \nu_r}$$

avec $Z_{123} = Z_1 + Z_2 + Z_3$.

Le système (1.16) étant non stationnaire, la stabilité asymptotique de $e = 0$ n'est pas prouvée.

2.4.2.2 Commande de système du robot de type unicycle sous forme chaînée

Dans le plan, on considère une courbe C et trois repères : $\Gamma_0 = \{O, i, j\}$ un repère fixe, $\Gamma_m = \{P_m, i_m, j_m\}$ un repère attaché au robot et $\Gamma_s = \{P_s, i_s, j_s\}$ un repère attaché à l'abscisse curviligne 's' de la courbe C (figure 1.12). On définit :

- 's' l'abscisse curviligne au point P_s obtenu par la projection du point P_m (barycentre du robot) orthogonalement sur la courbe C .
- 'd' l'ordonnée du point P_m dans le repère Γ_s et elle est aussi la distance entre le point P_m et la courbe C .
- $\theta_e = \theta - \theta_s$ l'orientation du robot, selon le repère $s\Gamma_s$

Sachant que $c(s) = \frac{\partial \theta_s}{\partial s}$ est la courbure de C au point P_s et en prenant en compte les

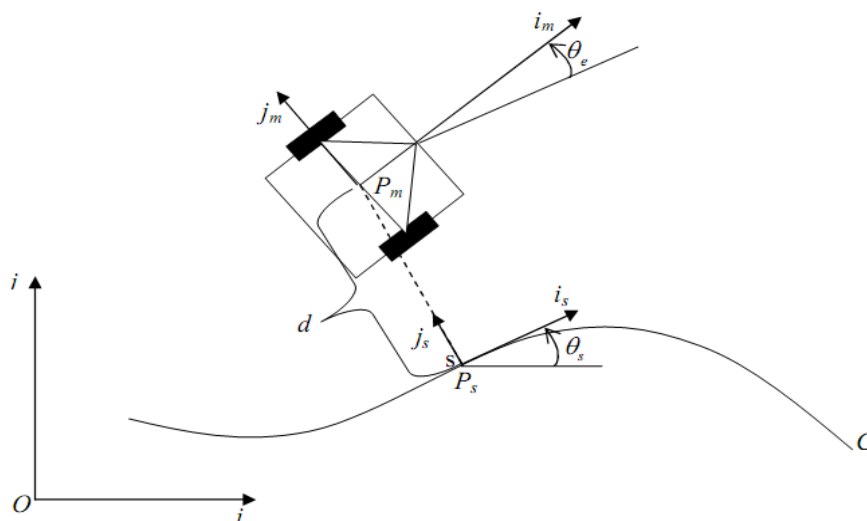


Figure 2.16 le modèle du robot dans le repère de Frénet

définitions ci-dessus, le modèle du robot mobile de type unicycle dans le repère de Frénet /Morin et Samson, 2008/ est donné par la formule suivante :

$$\begin{cases} \dot{s} = \frac{v}{1-dc(s)} \cos(\theta_c) \\ \dot{d} = v \sin(\theta_c) \\ \dot{\theta}_e = w - \dot{s}c(s) \end{cases} \quad (2.17)$$

Pour l'existence et l'unicité du 'Ps', il faut que le point P_m soit toujours à une distance de C, inférieure à son rayon de courbure $\rho(s)$ en 'Ps'. C'est-à-dire que $|d| \leq |\rho(s)|$, $\forall s \in [0, 1]$ ou $|dc(s)| < \rho(s)$, $\forall s \in [0, 1]$.

En faisant un changement de variables $(s, d, \theta_e, v, w) \rightarrow (z_1, z_2, z_3, u_2)$, le modèle (1.20) peut être représenté sous forme d'un système chaîné /Floquet et al, 2003/ comme suit :

$$\begin{aligned} \dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_1 z_3 \\ \dot{z}_3 &= u_2 \end{aligned}$$

Avec : $u_1 = \dot{s} = \frac{v}{1-dc(s)} \cos(\theta_e)$ et $z_3 = (1 - dc(s)) \tan \theta_e$

En utilisant le retour d'état proportionnel :

$$u_2 = \dot{z}_3 = -u_1 k_2 z_2 - |u_1| k_3 z_3$$

où k_2, k_3 sont deux constantes strictement positives, alors le système chaîné (1.21) est stable en $z_2 = z_3 = 0$ donc en $d = \theta_e = 0$.

2.4.2.3 Approche par fonction transverse

La commande par fonction transverse, introduite par /Morin 2001-a/ [74], assure une poursuite de trajectoires quelconques, y compris des trajectoires non admissibles, c'est-à-dire des trajectoires ne respectant pas la contrainte de non holonomie du robot. C'est une technique qui garantit la stabilisation pratique des erreurs de suivi. La première consiste en la détermination de la fonction transverse qui caractérise la dynamique souhaitée. Cette fonction dépend de paramètres qui fixent la taille du voisinage. Lors de la seconde étape, la loi de commande est calculée afin de garantir que le système suive la dynamique issue de la fonction transverse. L'intérêt de cette loi de commande est sa continuité et sa robustesse par rapport aux erreurs de modélisation.

2.4.3 Stabilisation par un retour d'état discontinu

Une autre manière de s'affranchir du fait que le modèle de robot mobile ne puisse pas être stabilisé par un retour d'état continu et stationnaire est d'utiliser des lois de commande discontinues. A ce titre, dans cette partie, nous abordons la commande à mode glissant pour le suivi de trajectoire. Ce type de commande est approprié pour les systèmes qui ont des incertitudes fortement non linéaires, ainsi que des perturbations bornées /Young 1988[75]. Plusieurs travaux ont été réalisés pour la stabilisation du modèle du robot mobile non holonome mais la plupart d'entre eux ne considère pas de perturbations. Plus récemment, l'utilisation du principe de la commande par modes glissants avec action a été introduite par /Defoort 2007-b et Defoort 2005/[76] pour la poursuite de trajectoire d'un robot mobile non holonome en présence de perturbations.

Considérons le modèle du robot mobile non holonome donné sous la forme suivante :

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} + p(q, t) \quad (2.18)$$

Telle que $p(q, t)$ est une perturbation due aux erreurs de modélisation, de mesure, ou encore, à des variations paramétriques et $q = [x(t) \ y(t) \ \theta(t)]^T$.

La perturbation $p(q, t)$ vérifie la condition de recouvrement /Defoort 2007-a/[77], c-à-d $p(q, t)$ peut s'écrire sous la forme suivante :

$$p(q, t) = \sigma_v(q, t)[\cos \theta(t) \ \sin \theta(t) \ 0]^T + \sigma_w(q, t)[0 \ 0 \ 1]^T$$

avec $\sigma_v(q, t)$ et $\sigma_w(q, t) \in j$. Les fonctions $\sigma_v(q, t)$ et $\sigma_w(q, t)$ sont supposées bornées par des fonctions positives connues, respectivement $a_v(q)$ et $a_w(q)$.

Les vitesses linéaire et angulaire de la trajectoire de référence v_r et w_r sont bornées et différentes de zéro pour tout $t \geq 0$.

En considérant l'erreur de position du robot $[e_x, e_y, e_\theta] = [x - x_r, y - y_r, \theta - \theta_r]$, et le changement de coordonnées :

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} -\cos \theta & -\sin \theta & 0 \\ \sin \theta & -\cos \theta & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}$$

alors, la dynamique de l'erreur de poursuite peut s'écrire sous la forme suivante :

$$\dot{e} = g_1(e, t) + g_2(e)(u + \sigma(q, t))$$

Avec :

$$\begin{cases} e = [e_1, e_2, e_3]^T \\ g_1(e, t) = [v_r \cos e_3, v_r \sin e_3, w_r]^T \\ g_2(e) = \begin{bmatrix} -1 & 0 & 0 \\ e_2 & -e_1 & -1 \end{bmatrix}^T \\ \sigma(q, t) = [\sigma_v(q, t), \sigma_w(q, t)]^T \\ u = [v \quad w]^T \end{cases}$$

Supposons que les hypothèses citées ci-dessus sont vérifiées, alors la solution nulle du système (1.25) est globalement asymptotiquement stable par la commande discontinue suivante:

$$u = u_{nom} + u_{disc}(e, e_{aux})$$

Telle que

$$u_{nom} = \begin{bmatrix} v_{nom} \\ w_{nom} \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + \mu_3 \tanh e_1 \\ w_r + \frac{\mu_1 v_r e_2}{1 + e_1^2 + e_2^2} \frac{\sin e_3}{e_3} + \mu_2 \tanh e_3 \end{bmatrix}$$

Et

$$u_{disc}(e, e_{aux}) = \begin{bmatrix} -G_1(e) \text{sign}(\sigma_1) \\ -G_2(e) \text{sign}(-e_2 \sigma_1 + \sigma_2) \end{bmatrix}$$

avec la variable de glissement $\sigma = [\sigma_0(e) \sigma_2]^T$ définie par :

$$\begin{cases} \sigma = \sigma_0(e) + e_{aux} \\ \sigma_0(e) = [-e_1, -e_3]^T \\ \dot{e}_{aux} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} (g_1(e, t) + g_2(e)u_{nom}(e)) \\ e_{aux}(0) = [e_1(0), e_3(0)]^T \end{cases} \quad (2.19)$$

les coefficients μ_1, μ_2, μ_3 sont des constantes réelles positives choisies de manière à assurer une certaine robustesse vis-à-vis des perturbations. Les fonctions g_1 , g_2 et g , quant à elles, vérifient les relations suivantes :

$$\begin{cases} g_1(e) \geq a_v(q) + \eta \\ g_2(e) \geq a_w(q) + \eta \end{cases} \quad (2.20)$$

Avec $\eta > 0$

2.5 Conclusion

Dans ce chapitre, nous avons présenté et détaillé différents modèles existants, les robots mobiles non holonomes et ces modélisations cinématiques, ainsi que les différentes techniques de planification de trajectoire, et méthodes de poursuite de trajectoire.

Chapitre 3

La logique floue et la régulation intelligente

Afin de pouvoir appliquer la technique de la logique floue à la régulation des PID et le principe de l'application au contrôle du robot, nous présentons dans ce chapitre les concepts de la logique floue et nous exposons tous les aspects méthodologiques nécessaires à la compréhension de cette méthode. Nous étudions ensuite la conception des contrôleurs flous pour la commande robot mobile.

3. 1 Logique floue

3.1.1 Relations floues et logique floue

Dans le présent chapitre, des sujets importants de la théorie des ensembles flous, comme Fonction d'appartenance; les attributions de valeurs de fuzzification et d'appartenance sont discutées. Les relations Floues et leurs graphiques sont étudiés en détail. Cependant, notre accent est mis sur la logique floue qui est une méthode pour formaliser la capacité humaine de raisonnement imprécis ou raisonnement approximatif. Un tel raisonnement représente la capacité humaine à raisonner approximativement et juger une incertitude ambiguë. Dans la logique floue, toutes les vérités sont partielles ou approximatives. Dans ce sens ce raisonnement a également été appelé raisonnement interpolatif et le processus de interpoler entre les extrêmes binaires de vrai et de faux est représenté par la capacité de logique floue pour encapsuler des vérités partielles. Dans la dernière partie du chapitre, les concepts de classique la logique floue et le raisonnement approximatif sont expliqués en détail.

3.1.2 Principes et définitions de la logique floue

Une des caractéristiques du raisonnement humain est qu'il est généralement fondé sur des données imprécises ou même incomplètes. En effet, les connaissances dont nous disposons sur un système quelconque sont généralement incertaines ou vagues, soit parce que nous avons un doute sur leur validité ou alors nous éprouvons une difficulté à les exprimer clairement.

Il est donc nécessaire de penser et de développer un nouveau type de raisonnement, le raisonnement approché, qui permettra de traiter mathématiquement l'imprécis et l'incertain.

Le premier à avoir souligné ces possibilités de développement est Lotfi A. Zadeh qui, dès 1965, introduit la théorie de la logique floue [78].

C'est une technique pour le traitement de connaissances imprécises et incertaines. Elle permet de prendre en considération des variables linguistiques dont les valeurs sont des mots ou des expressions du langage naturel, telle que faible, élevée, rapide, lent, grand, petit, etc....

A côté d'un formalisme mathématique fort développé, nous préférons aborder la présentation de manière intuitive. Nous présentons un exemple simple pour comprendre l'intérêt de la logique floue sur la logique classique:

- **Exemple :** Dans la logique classique, une vitesse peut être qualifiée par le terme «faible» ou « élevée». Dans la logique floue, des échelons d'appréciation intermédiaires de la variable vitesse sont possibles. La «vitesse» devient une variable linguistique dont les valeurs sont par exemple :«très faible»,«faible»,«moyenne»,«élevée»,«très élevée». Une vitesse de 1000tr/mn par exemple dans la logique classique correspond au seul ensemble «moyenne», alors que dans le modèle flou, elle appartient à la fois aux ensembles «faible» et «moyenne».

3.1.3 Ensembles flous

La notion d'ensemble flou a pour but de permettre des gradations dans l'appartenance d'un élément à une classe, c'est à dire d'autoriser un élément à appartenir plus ou moins fortement à cette classe.

Afin de mettre en évidence cette notion, on introduit les définitions suivantes :

Soit un ensemble de référence X continu ou discret d'objets dénotés $\{x\}$.

Un ensemble classique A de X est défini sur un univers de discours par une fonction Caractéristique μ_A qui prend la valeur 0 pour les éléments de X n'appartenant pas à A et la Valeur 1 pour ceux qui appartiennent à A (figure 3.1). L'univers de discours est l'ensemble des Valeurs réelles que peut prendre la variable floue X .

$$\mu_A : X \rightarrow \{0,1\}$$

Plus généralement, le domaine de définition de $\mu_A(x)$ peut être réduit à un sous-ensemble de X .

Un ensemble flou A peut être représenté comme un ensemble de paires (élément générique, degré d'appartenance) ordonnées:

$$A = \{(x, \mu_A(x)) / x \in X\}$$

On adopte souvent la notation suivante pour représenter l'ensemble A , qui indique pour tout élément x de X son degré $\mu_A(x)$ d'appartenance à A :

$$A = \sum_{x \in X} \mu_A(x) / x, \text{ si } X \text{ est discret}$$

$$A = \int_X \mu_A(x) / x$$



Figure 3.1 exemple de fonctions d'appartenance (a) logique classique (b) logique floue

3.1.4 Variable linguistique

Une variable linguistique est représentée par un triplet (V, X, T_V) , dans lequel V est une Variable (la vitesse, la température ...) définie sur un ensemble de référence X (ses valeurs Peuvent être n'importe quel élément de X). On note $T_V(A_1, A_2, \dots)$ un ensemble, fini ou infini, de sous-ensembles flous de X , qui sont utilisés pour caractériser V . Afin de permettre un traitement numérique, il est indispensable de les soumettre à une définition à l'aide de fonctions d'appartenance.

Par exemple, si la vitesse est interprétée comme une variable linguistique, alors son ensemble de termes $T_{vitesse} = \{lente, moyenne, rapide.. \}$ où chaque terme est caractérisé par un ensemble flou.

Ces termes peuvent être définis comme des ensembles flous dont les fonctions d'appartenance sont montrées sur la figure 3.2.

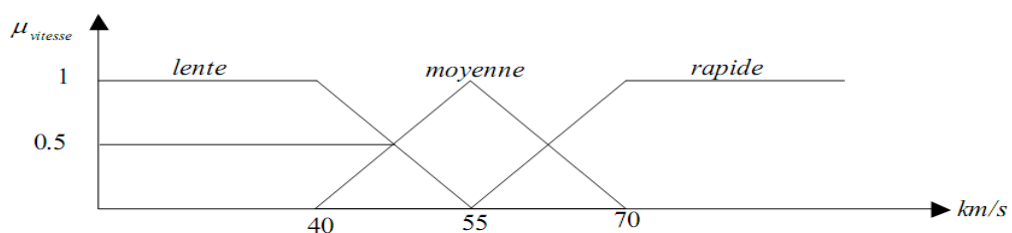


Figure 3.2 : représentation graphique des termes linguistiques

3.1.5 Fonctions d'appartenance

Une définition des variables linguistiques à l'aide des fonctions d'appartenance est nécessaire dans le but de traiter des déductions floues par ordinateur. Dans ce contexte, est attribuée à chaque valeur de la variable linguistique une fonction d'appartenance μ , dont la valeur varie entre 0 et 1.

3.1.5.1 Caractéristique de la fonction d'appartenance

La fonctionnalité de la fonction d'appartenance est définie par trois propriétés:

- Le noyau
- Le support
- Les Frontières

La figure 3.3 ci-dessous définit les propriétés énumérées ci-dessus.

3.1.5.1.1 Le noyau

Si la région de l'univers est caractérisée par l'appartenance à part entière 1 à l'ensemble A, cela donne le noyau de la fonction d'appartenance de flou en A. Les éléments qui ont la composition fonctionnent comme 1 sont les éléments du noyau, à savoir, i.e., $\mu_A(x) = 1$

3.1.5.1.2 Le support

Si la région de l'univers est caractérisée par une appartenance non nulle à l'ensemble A, cela définit le support d'une fonction d'appartenance pour l'ensemble flou A. le support a les éléments dont la composition est supérieure à zéro, $\mu_A(x) \Rightarrow 0$

3.1.5.1.3 Les Frontières

Si la région de l'univers a une appartenance non nulle mais pas une adhésion complète, cela définit le limite d'une fonction d'appartenance; ceci définit les limites d'une fonction d'appartenance pour ensemble flou A.

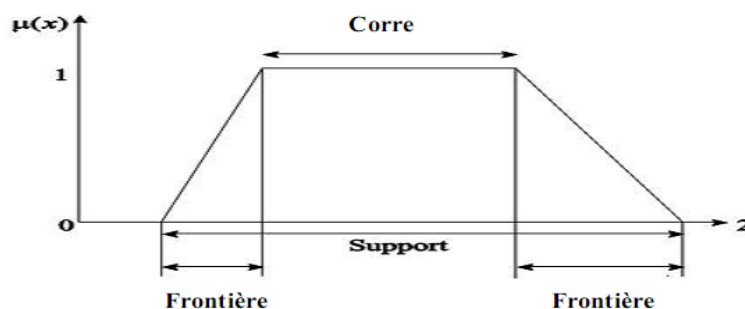


Figure 3.3 Caractéristiques de la fonction d'appartenance

Si la région de l'univers a une appartenance non nulle mais pas une adhésion complète, cela définit le limite d'une fonction d'appartenance; ceci définit les limites d'une fonction d'appartenance pour ensemble flou A.

Les limites ont les éléments dont les membres sont compris entre 0 et 1, i.e $0 < \mu_A(x) < 1$.

Ce sont les régions standards définies dans les fonctions d'appartenance. Ensuite, nous définissons deux termes importants.

- **Point de croisement**

Le point de croisement d'une fonction d'appartenance est l'élément de l'univers dont la valeur d'appartenance égale à 0.5. i.e $\mu_A(x) = 0.5$.

- **La hauteur**

La hauteur de l'ensemble flou A est la valeur maximale de la fonction d'appartenance, i.e $\max(\mu_A(x)) = 0.5$.

Les fonctions d'appartenance peuvent être symétriques ou asymétriques.

Le plus souvent, nous utilisons des fonctions d'appartenance de type triangulaire ou trapézoïdale (figure 3.4).

Fonction triangulaire : L'allure est complètement définie par trois paramètres $\{a, b, c\}$.

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-a}{c-b}\right), 0\right) \quad (3.1)$$

Fonction trapézoïdale : Elle est définie par quatre paramètres $\{a, b, c, d\}$.

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{c-a}{c-b}\right), 0\right) \quad (3.2)$$



Figure 3.4 Formes usuelles des fonctions d'appartenance.

3.1.6 Opérations sur les ensembles flous

Puisque les fonctions caractéristiques des ensembles flous sont graduelles, l'extension de certaines opérations définies sur les ensembles classiques au cas des ensembles flous pose certaines difficultés. Il n'est donc pas possible d'appliquer directement la logique classique qui manipule des ensembles à fonctions caractéristiques binaires (0 ou 1). Pour pallier à ce problème, les définitions d'outils nécessaires au traitement des ensembles flous sont introduites.

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

Ces relations sont traduites par les opérateurs « Et », « Ou » et « Non ».

Les opérateurs les plus utilisés en logique floue sont donc :

- l'opérateur « Et » correspond à l'intersection de deux ensembles A et B . Il peut être réalisé par :
- La fonction « min » : $\forall x \in X \quad \mu_D = \min(\mu_A(x), \mu_B(x))$.
- La fonction arithmétique « produit » : $\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x)$.
- L'opérateur « Ou » correspond à l'union de deux ensembles A et B . Il peut être réalisé par :
- La fonction « max » : $\forall x \in X \quad \mu_C(x) = \max(\mu_A(x), \mu_B(x))$
- La fonction arithmétique « somme » : $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x)$.
- L'opérateur « Non » est réalisé par : $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

3.1.7 Propositions floues

3.1.7.1 Propositions floues élémentaires

Une proposition floue élémentaire est définie à partir d'une variable linguistique (V, X, T_V) , par la qualification « V est A », pour une caractérisation floue A , appartenant à T_V .

3.1.7.2 Propositions floues générales

Une proposition floue générale est obtenue par la composition de propositions floues élémentaires « V est A », « W est B » pour des variables linguistiques V, W, \dots

Plus généralement, on peut construire des propositions floues par conjonction, disjonction ou implication, par exemple de la forme « si V est A et W est B alors U est C » (si la taille est moyenne et le prix est peu cher, alors l'achat est conseillé).

3.1.8 Raisonnement en logique floue

Avec l'unique schéma de raisonnement (Si les conditions sont remplies, Alors la conclusion est validée) et les trois opérateurs Et, Ou et Non, nous pouvons déjà prendre un grand nombre de décisions logiques. Nous produisons aussi une nouvelle information (une décision) à partir d'informations anciennes.

Le raisonnement flou fait appel à trois notions et étapes fondamentales :

- l'implication floue,
- l'inférence floue,
- l'agrégation des règles.

3.1.8.1 Implication floue

L'implication floue donne une information sur le degré de vérité d'une règle floue. En d'autres termes, on quantifie la force de véracité entre la prémisse et la conclusion. Considérons par exemple les deux propositions floues.

"x est A"

"y est B" Où x et y sont des variables floues et A et B des ensembles flous de l'univers des discours U.

Ainsi que la règle floue : Si " x est A " Alors " y est B".

L'implication floue donne alors le degré de vérité de la règle floue précédente à partir des

Degrés d'appartenance de x à A (prémisse) et de y à B (conclusion). On notera implication : opérateur imp (équivalent à l'opérateur Alors). Les normes d'implication les plus utilisées sont :

- La norme Mamdani $\text{imp}(\mu_A(x), \mu_B(y)) = \min(\mu_A(x), \mu_B(y))$
- La norme Larsen $\text{imp}(\mu_A(x), \mu_B(y)) = (\mu_A(x), \mu_B(y))$

3.1.8.2 Inférence floue

Le problème tel qu'il se pose en pratique n'est généralement pas de mesurer le degré de véracité d'une implication mais bien de déduire, à l'aide de faits et de diverses règles

implicatives, des évènements potentiels. En logique classique, un tel raisonnement porte le nom de Modus Ponens (raisonnement par l'affirmation).

Si $p \Rightarrow q$ vrai

Alors q vrai

Et p vrai

De façon générale, les conditions d'utilisation du Modus Ponens Généralisé sont les suivantes :

	prémisse	conclusion
Règle floue :	si x est A	alors y est B
Fait observé :	si x est A'	
consequence :		y est B'

A' et B' sont les ensembles flous constatés dans le cas que l'on traite et ne sont pas nécessairement strictement égaux à A et B. B' est l'ensemble flou résultant de A' par l'application de l'implication.

Les informations disponibles pour déterminer la conséquence sont donc d'une part celles relatives aux règles, quantifiées par l'implication floue $\mu_{\frac{B}{A}}(x, y)$ d'autres part celles relatives au fait observé, quantifiées par la fonction d'appartenance μ_A .

- **Par matrice d'inférence**

Elle rassemble toutes les règles d'inférences sous forme de tableau. Dans le cas d'un tableau à deux dimensions, les entrées du tableau représentent les ensembles flous des variable d'entrées (température : T et vitesse : V). L'intersection d'une colonne et d'une ligne donne l'ensemble flou de la variable de sortie définie par la règle. Il y a autant de cases que de règles tableau 3.1.

U		T		
		F	M	E
V	F	Z	P	GP
	E	Z	Z	P

Les règles que décrit ce tableau sont (sous forme symbolique) :

Si T est F Et V est F Alors U=Z Ou

Si T est M Et V est F Alors U = P Ou

Si Test E Et V est F Alors U=GP Ou

Si T est F Et V est E Alors U = Z Ou

Si T est M Et V est E Alors U = Z Ou

Si T est E Et V est E Alors U=P Ou

Dans l'exemple ci-dessus, on a représenté les règles qui sont activées à un instant donné par des cases sombres :

Si (T est M Et V est F) Alors U = P Ou

Si (T est E Et V est F) Alors U = GP

Il s'agit maintenant de définir les degrés d'appartenance de la variable de sortie à ses sous-ensembles flous. Nous allons présenter les méthodes d'inférence qui permettent d'y arriver. Ces méthodes se différencient essentiellement par la manière dont vont être réalisés les opérateurs (ici « Et » et « Ou ») utilisés dans les règles d'inférence. Les trois méthodes d'inférence les plus usuelles sont :Max–min, Max-produit et Somme-produit.

3.1.8.3 Agrégation des règles

Lorsque la base de connaissance comporte plusieurs règles, l'ensemble flou inféré B' est Obtenu après une opération appelée agrégation des règles. En d'autres termes, l'agrégation des règles utilise la contribution de toutes les règles activées pour en déduire une action de commande floue [79]. Généralement, les règles sont activées en parallèle et sont liées par l'opérateur « Ou ».

Nous pouvons considérer que chaque règle donne un avis sur la valeur à attribuer au signal de commande, le poids de chaque avis dépend du degré de vérité de la conclusion.

3.1.9 Fuzzification

La fuzzification est un concept important dans la théorie de la logique floue. La fuzzification est le processus où les quantités croustillantes sont converties en flou (de vives à floues). En identifiant certains des incertitudes présentes dans les valeurs croquantes, nous

formons les valeurs floues. La conversion des valeurs floues sont représenté par les fonctions d'appartenance.

Dans toutes les applications pratiques dans les industries, mesure de la tension, du courant, de la température, etc., il pourrait y avoir une erreur négligeable. Cela provoque une imprécision dans les données. Cette imprécision peut être représentée par les fonctions d'appartenance. Ainsi, la fuzzification est effectuée.

Affectation les valeurs d'appartenance

Il existe différentes méthodes pour affecter les valeurs d'appartenance ou les fonctions d'appartenance à des variables floues. L'attribution peut être faite simplement par intuition ou en utilisant des algorithmes ou procédures logiques. Les méthodes d'affectation des valeurs d'appartenance sont répertoriées comme suit [80] :

- I. Intuition
- II. Inférence
- III. Le Classement
- IV. Ensembles angulaires flous
- V. Les réseaux de neurones
- VI. Algorithmes génétiques
- VII. Assaisonnement inductif

3.1.9.1 Intuition

L'intuition est basée sur l'intelligence et la compréhension de l'humain pour développer les fonctions d'adhésion. La connaissance approfondie du problème et de la variable linguistique devrait également être connu.

3.1.9.2 Inférence

Cette méthode implique la connaissance nécessaire pour effectuer un raisonnement déductif. Les membres la fonction est formée à partir des faits connus ou des connaissances.

3.1.9.3 Le classement

Le concept d'interrogation est utilisé pour attribuer des valeurs d'appartenance par processus de classement. Les préférences sont indiquées ci-dessus pour les comparaisons par paires et, partant, l'ordre d'adhésion est fait.

3.1.9.4 Les réseaux de neurones

Les réseaux de neurones sont utilisés pour simuler le réseau de travail des neurones dans le cerveau humain. Le concept de cerveau humain est utilisé pour effectuer des calculs sur des ordinateurs.

Dans ce cas, la fonction d'appartenance floue peut être créée pour les classes floues d'un jeu de données d'entrée. La procédure est le nombre de valeurs de données d'entrée sélectionnées. Ensuite, il est divisé en ensemble de données d'apprentissage et en ensemble de données de test. L'ensemble de données d'apprentissage peut être utilisé pour former le réseau.

Les générations de la fonction d'appartenance du réseau de neurones sont illustrées à la Fig.

La figure (a) montre l'ensemble de données d'entraînement. Ceci est passé par un réseau de neurones montré dans la figure (b). Les points de données de la figure 2.7 (a) sont divisés en trois régions comme R^1 , R^2 et R^3 comme dans la figure 2.7 (c). Selon les points de données, les régions sont classées. Si le point de données se trouve dans la région 1, nous affectons une adhésion complète à la région 1 et une appartenance nulle aux régions 2 et 3. De la même manière, si les points de données se trouvent dans la région 2, le nombre de membres est complet dans la région et nul dans les régions, etc.

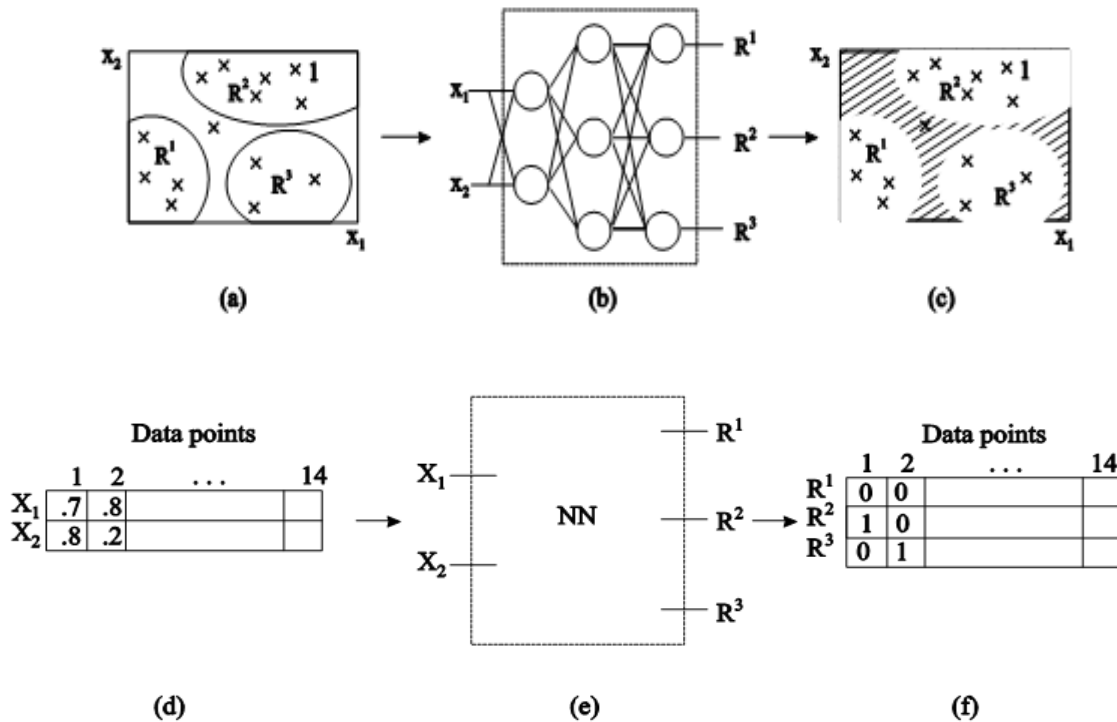


Figure 3.5 Génération de la fonction d'appartenance à l'aide de réseaux de neurones

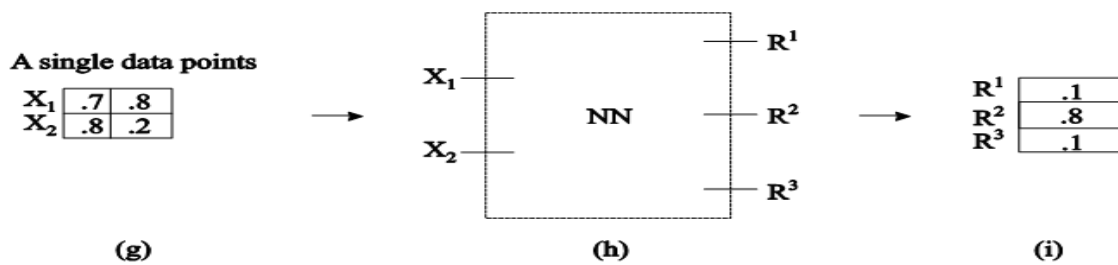


Figure 3.5 Génération de la fonction d'appartenance à l'aide de réseaux de neurones

Le réseau de neurones est ensuite créé, à partir duquel la formation est effectuée entre les valeurs d'appartenance dans différentes classes, pour simuler la relation entre la coordonnée des lieux et les valeurs des membres. Le réseau de neurones utilise l'ensemble de valeurs de données et comme le montre la figure 2.7 (d). Ce processus de formation est continué jusqu'à ce que le réseau de neurones puisse simuler l'ensemble des données d'entrée et de sortie valeur.

Une fois le réseau est formé, ses performances peuvent être vérifiées à l'aide des données de test. Après une formation complète et le processus de test est terminé, le réseau de neurones est prêt et peut être utilisé pour déterminer les valeurs d'appartenance de toutes les données d'entrée dans les différentes régions. Ceux-ci sont tous montrés dans Figure 2.7 (g) – (i).

La cartographie complète de l'appartenance de différents points de données dans différentes classes floues peut être déterminée en utilisant une approche de réseau neuronal.

3.1.9.5 Algorithmes génétiques

Les étapes impliquées dans le calcul des fonctions d'adhésion à l'aide de GA sont les suivantes :

- Pour la cartographie fonctionnelle donnée d'un système, certaines fonctions d'appartenance et leurs formes sont supposées pour diverses variables floues à définir.
- Ces fonctions d'appartenance sont ensuite codées sous forme de chaînes de bits.
- Ces chaînes de bits sont ensuite concaténées (jointes).
- Semblable à la fonction d'activation dans les réseaux de neurones, GA a une fonction de remise en forme.
- Semblable à la fonction d'activation dans les réseaux de neurones, GA a une fonction de remise en forme.

- (f) Cette fonction de mise en forme permet d'évaluer l'aptitude de chaque ensemble de fonctions d'appartenance.
- (g) Ces fonctions d'appartenance sont les paramètres qui définissent le mappage fonctionnel du système.

Ainsi, GA peut être utilisé pour déterminer les fonctions d'appartenance.

3.1.9.6 Assaisonnement inductif

L'appartenance peut également être générée par les caractéristiques du raisonnement inductif. L'induction est réalisée selon le principe de minimisation d'entropie, qui regroupe les paramètres correspondant aux classes de sortie. Pour la méthode de raisonnement inductif, il devrait exister une base de données bien définie pour les relations entrée-sortie. Cette méthode peut convenir aux systèmes complexes où les données sont abondantes et statiques. Lorsque les données sont dynamiques, cette méthode est utilisée. Ne convient pas, car les fonctions d'appartenance changent continuellement avec le temps.

Il existe trois lois de l'induction.

- (1) Étant donné un ensemble de résultats irréductibles d'une expérience, les probabilités induites sont ces probabilités cohérentes avec toutes les informations disponibles qui maximisent l'entropie de l'ensemble.
- (2) La probabilité induite d'un ensemble d'observations indépendantes est proportionnelle à la densité de probabilité de la probabilité induite d'une seule observation.
- (3) La règle induite est celle de règle cohérente avec toutes les informations disponibles dont l'entropie est minimale.

La troisième loi énoncée ici est la plus utilisée pour le développement de la fonction d'appartenance.

Les étapes à suivre pour générer des fonctions d'appartenance à l'aide d'un raisonnement inductif sont les suivantes :

- (a) Il est nécessaire d'établir un seuil flou entre les classes de données.
- (b) Premièrement, déterminez la ligne de seuil avec une méthode de filtrage de minimisation d'entropie.
- (c) Après cela, démarrez le processus de segmentation.
- (d) Le processus de segmentation aboutit dans deux classes.

- (e) En divisant encore une fois les deux premières classes, il existe trois classes différentes.
- (f) Le partitionnement est répété avec des calculs de valeur de seuil, ce qui nous conduit à partitionner le jeu de données en un certain nombre de classes ou de jeux flous.
- (g) Ensuite, en fonction de la forme, la fonction d'appartenance est déterminée.

Ainsi, la génération de la fonction d'appartenance est basée sur un concept de partitionnement ou de filtrage analogique. Ceci trace une ligne de seuil entre deux classes de données exemple. Le concept principal après avoir tracé la ligne de seuil, il faut classer les échantillons lors de la minimisation de l'entropie pour un partitionnement optimal.

3.1.10 Defuzzification

Il existe des méthodes de défuzzification utilisées pour convertir les quantités floues en quantités précises. La sortie d'un processus flou complet peut être l'union de deux ou plusieurs fonctions d'appartenance floue. Pour expliquer cela en détail, considérons une sortie floue constituée de deux parties, l'une étant de forme triangulaire (figure 3.6 (a)) et l'autre de forme trapézoïdale (figure 3.6 (b)). L'union de ces deux formes (figure 3.6 (c)), la partie externe enveloppe des deux formes.

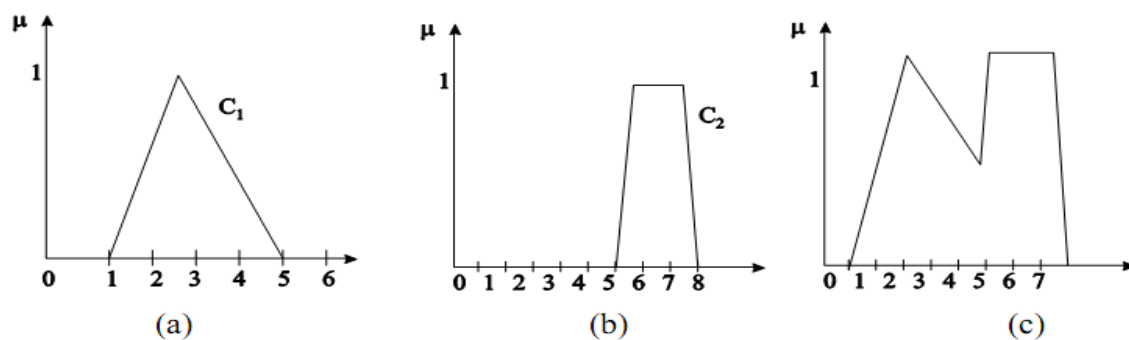


Figure 3.6 Sortie floue typique

Il existe sept méthodes utilisées pour défuzzifier les fonctions de sortie floues :

- I. Principe d'appartenance maximum
- II. Méthode centroïde
- III. Méthode de moyenne pondérée
- IV. moyenne-maximale d'appartenance
- V. Centre des sommes

- VI. Centre de la plus grande surface
- VII. Premier des maximum ou dernier des maximum

Ensuite, nous expliquons chaque méthode avec un graphique.

3.1.10.1 Principe d'appartenance maximum

Cette méthode est donnée par l'expression

$$\mu_c(z^*) \geq \mu_c(z), \text{ for all } z \in X' \tag{3.3}$$

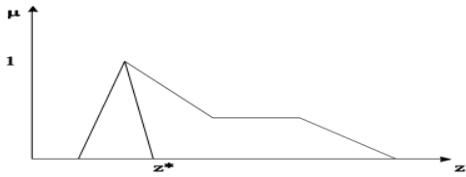


Figure 3.7 méthode d'appartenance maximum

3.1.10.2 Méthode centroïde

C'est la méthode la plus largement utilisée. Ceci peut être appelé méthode du centre de gravité ou du centre de la zone. Il peut être défini par l'expression algébrique.

$$z^* = \int \frac{\mu_c(z)zdx}{\mu_c(z)dx} \tag{3.4}$$

Où ∫ est utilisé pour l'intégration algébrique. La figure 3.3 représente graphiquement cette méthode.

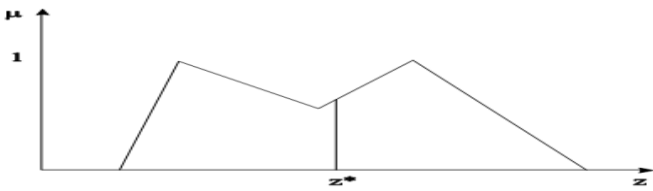


Figure 3.8 Méthode centroïde

3.1.10.3 Méthode de moyenne pondérée

Cette méthode ne peut pas être utilisée pour les fonctions d'appartenance de sortie asymétrique, elle ne peut être utilisée que pour les fonctions d'appartenance de sortie symétriques. Pondération de chaque fonction d'appartenance dans l'obtenu en sortie par sa plus grande valeur d'appartenance forme cette méthode. L'expression d'évaluation de cette méthode est

$$z^* = \frac{\sum \mu_c(\bar{z}) \bar{z}}{\sum \mu_c(\bar{z})} \quad (3.5)$$

Où \sum est utilisé pour la somme algébrique.

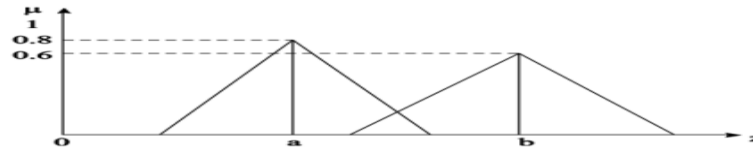


Figure .3.9. Méthode de moyenne pondérée

3.1.10.4 moyenne-maximale d'appartenance

Cette méthode est liée au principe de max-d'appartenance, mais la composition actuelle maximale n'a pas besoin d'être unique, c'est-à-dire que la composition maximale n'est pas nécessairement un point unique, elle peut être une plage. Cette méthode est également appelée méthode au milieu des maxima, l'expression est donnée sous la forme

$$z^* = \frac{a + b}{2} \quad (3.6)$$

Où $a \times b$ sont le point final de la plage d'appartenance maximale, comme indiqué sur la figure 3.10.

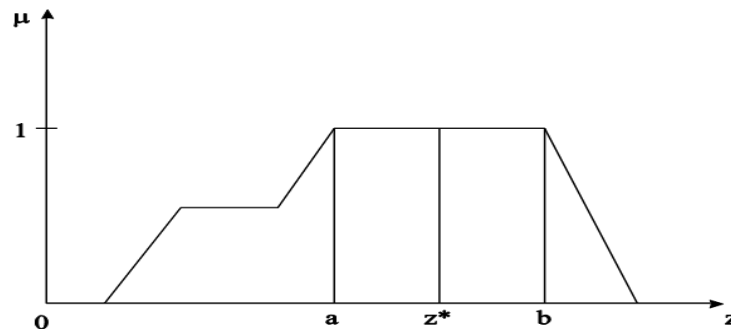


Figure3.10 moyenne-maximale d'appartenance

3.1.10.5 Centre des sommes

Cela implique la somme algébrique des ensembles flous de sortie individuels, par exemple c_1 et c_2 au lieu d'union. Avec cette méthode, il est noté que les zones d'intersection sont ajoutées deux fois. Cette méthode est similaire à la méthode de la moyenne pondérée, mais dans le centre des sommes, les pondérations correspondent aux zones des fonctions d'appartenance respectives, tandis que dans la méthode de la moyenne pondérée, les pondérations correspondent aux valeurs d'appartenance individuelles.

La valeur defuzzifiée z^* est donnée comme

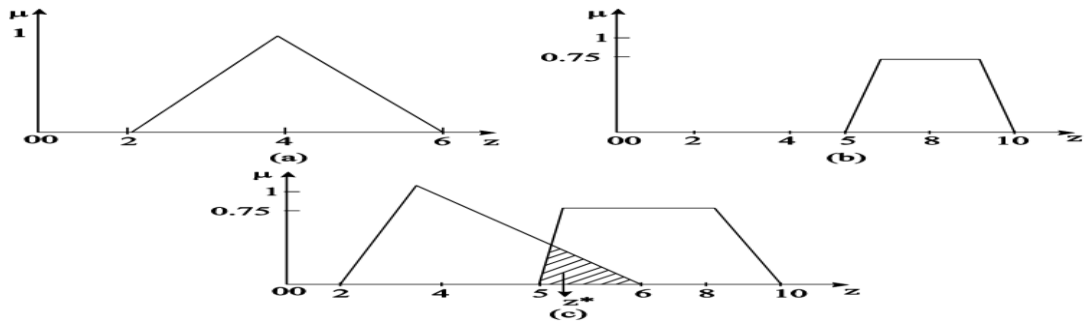


Figure 3.11 (a) Première appartenance (b) Deuxième appartenance (c) Étape de défuzzification

$$z^* = \frac{\int x \sum_{k=1}^n \mu_{c_k}(z) dx}{\int x \sum_{k=1}^n \mu_{c_k}(z) dz} \quad (3.7)$$

3.1.10.6 Centre de la plus grande surface

Si l'ensemble flou a deux sous-régions convexes, alors la totalité de la gravité de la sous-région convexe avec la plus grande surface peut être utilisée pour calculer la valeur de défuzzification (figure 3.12). L'équation est donnée comme

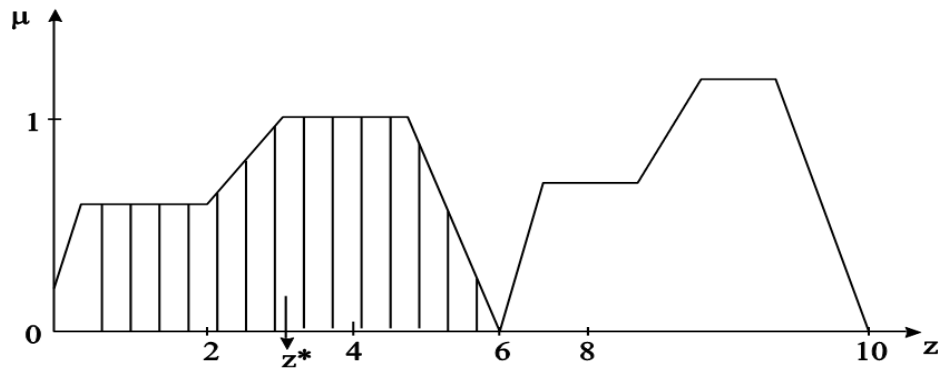


Figure 3.12 Centre de la plus grande surface

$$z^* = \frac{\int \mu_{c_m}(z) z dx}{\int \mu_{c_m}(z) dx} \quad (3.8)$$

Où c_m est la région convexe avec la plus grande surface. La valeur z^* est identique à la Valeur z^* obtenue par la méthode du centroïde. Cela peut être fait même pour les régions non convexes.

3.1.10.7 Premier des maximums ou dernier des maximums

Ici, la sortie calculée de tous les ensembles flous de sortie individuels c_k est utilisée pour déterminer la plus petite valeur, avec le degré d'appartenance maximisé dans c_m .

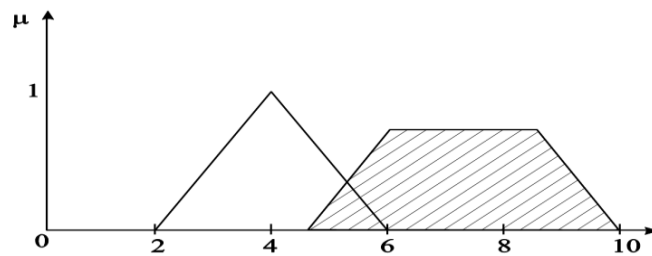


Figure 3.13 Premier des maximums ou dernier des maximums

Après avoir présenté un rappel précis sur la logique floue, nous devons pouvoir concevoir et simuler les techniques des systèmes de contrôle flou. Cela va faire beaucoup avancer vers l'implémentation de contrôleurs flous puisque nous fournissons des conseils sur la façon de surmonter certains problèmes pratiques rencontrés dans la conception et la mise en œuvre de systèmes de contrôle flou.

3.2 Conception d'un contrôleur flou

Après avoir énoncé les concepts de base et les termes linguistiques utilisés en logique floue, nous présentons la structure d'un contrôleur flou.

Le contrôle flou fournit une méthodologie formelle pour représenter, manipuler et implémenter les connaissances heuristiques d'un humain sur la manière de contrôler un système. Dans cette section nous cherchons à fournir une philosophie sur la manière d'aborder la conception de contrôleurs flous [81]. Le schéma synoptique du contrôleur flou est présenté à la figure 4.2, où nous montrons un contrôleur flou intégré à un système de contrôle en boucle fermée. Les sorties de l'installation sont indiquées par $y(t)$, ses entrées par $u(t)$ et l'entrée de référence du contrôleur flou par $r(t)$.

Un schéma synoptique d'un système de contrôle flou est présenté à la figure 3.14. Parfois, un contrôleur flou s'appelle un «contrôleur de logique floue» (FLC) ou même un «contrôleur linguistique flou», car il utilise la logique floue pour la quantification des descriptions linguistiques.

Le contrôleur flou est composé des quatre éléments suivants:

1. Une base de règles (un ensemble de règles If-Then), qui contient une quantification en logique floue de la description linguistique de l'expert sur la manière d'obtenir un bon contrôle.
2. Un mécanisme d'inférence (également appelé module «moteur d'inférence» ou «inférence floue»), qui émule le processus décisionnel de l'expert en matière d'interprétation et d'application. connaissances sur la meilleure façon de contrôler la plante....
3. Une interface de fuzzification, qui convertit les entrées du contrôleur en informations que le mécanisme d'inférence peut facilement utiliser pour activer et appliquer des règles.
4. L'interface defuzzification convertit les conclusions tirées par le mécanisme d'inférence en entrées de la centrale.

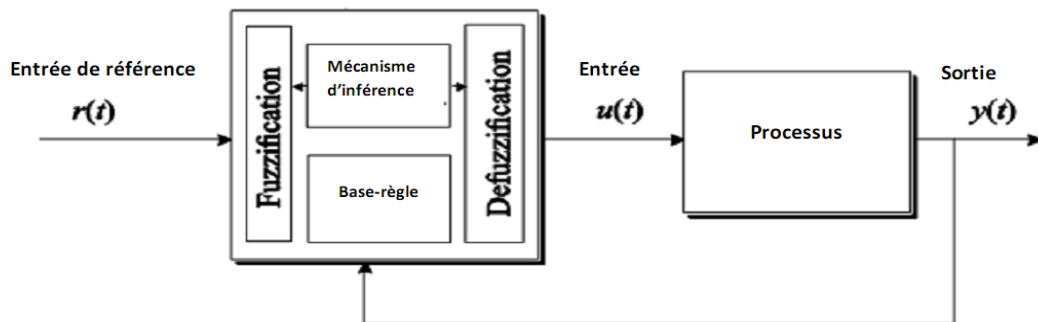


Figure 3.14 conception de contrôleurs flous

En général, un contrôleur flou est un système qui associe à tout vecteur d'entrée $X=[x_1, x_2, \dots, x_n]$ un vecteur de sortie $Y=[y_1, y_2, \dots, y_n]$ tel que $Y=F(X)$ où $F(X)$ est souvent une fonction non linéaire.

Le schéma de base d'un contrôleur flou repose sur la structure d'un régulateur classique à la différence que l'on va retenir la forme incrémentale. Cette dernière donne en sortie, non pas la grandeur de commande à appliquer au processus mais plutôt l'incrément de cette grandeur.

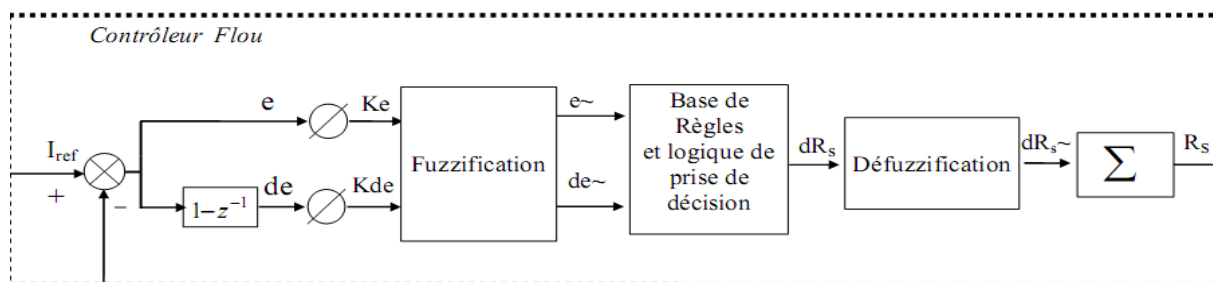


Figure 3.15 Structure interne d'un contrôleur flou

Dans le schéma ci-dessus comme dans ce qui suit, nous notons:

- e : l'erreur, elle est définie par la différence entre la consigne et la grandeur à réguler.

$$e(k) = e^*(k) - e(k) \quad (3.9)$$

- de : la dérivée de l'erreur, elle est approchée par

$$de(k) = \frac{e(k) - e(k-1)}{T_e} \quad (3.10)$$

La sortie du régulateur est donnée par

$$R_s(k) = R_s(k-1) + dR_s(k) \quad (3.11)$$

Des facteurs d'échelle des gains sont utilisés en entrée et en sortie du contrôleur flou. Ils permettent de changer la sensibilité du régulateur flou sans en changer sa structure.

Les règles d'inférences permettent de déterminer le comportement du contrôleur flou. Il doit donc inclure des étapes intermédiaires qui lui permettent de passer des grandeurs réelles vers les grandeurs floues et vice versa; ce sont les étapes de fuzzification et de defuzzification (figure3.4).

1. L'interface de fuzzification inclut les fonctions suivantes:

Les ensembles flous des variables d'entrée et leurs fonctions d'appartenance sont à définir en premier lieu.

L'étape de fuzzification permet de fournir les degrés d'appartenance de la variable floue à ces ensembles flous en fonction de la valeur réelle de la variable d'entrée.

2. La base de connaissance comprend une connaissance du domaine d'application et les buts du contrôle prévu. Elle est composée:

- d'une base de données fournissant les définitions utilisées pour définir les règles de contrôle linguistique et la manipulation des données floues dans le contrôleur;
- d'une base de règles caractérisant les buts et la politique de contrôle des experts du domaine au moyen d'un ensemble de règles de contrôle linguistique.

Comme nous l'avons précédemment évoqué, nous allons nous baser sur une matrice ou Une table d'inférence pour cette étape. La construction d'une telle table d'inférence repose sur une Analyse qualitative du processus. L'inférence se fait donc sur la base des matrices que l'on vient de décrire. On commence par utiliser un opérateur pour définir la description symbolique associée à la prémisse de la règle ; c'est-à-dire réaliser le "Et ". On passe ensuite à l'inférence proprement dite qui consiste à caractériser la variable floue de sortie pour chaque règle. C'est l'étape de la conclusion "Alors ".

Enfin, la dernière étape de l'inférence, appelée agrégation des règles, permet de synthétiser ces résultats intermédiaires.

Comme nous l'avons vu, la manière de réaliser les opérateurs va donner lieu à des contrôleurs flous différents. Les régulateurs les plus courants sont ceux de:

- **Régulateur type Mamdani**
- **Régulateur type Sugeno**

Ils sont dits de type procédural. En effet, seule la prémisse est symbolique. La conclusion, qui correspond à la commande, est directement une constante réelle ou une expression polynomiale fonction des entrées.

L'établissement de la règle d'inférence est généralement basé sur un des points suivants:

- l'expérience de l'opérateur et/ou du savoir-faire de l'ingénieur en régulation et contrôle.
- Un modèle flou du processus pour lequel on souhaite synthétiser le régulateur.
- Les actions de l'opérateur; s'il n'arrive pas à exprimer linguistiquement les règles qu'il utilise implicitement.

- L'apprentissage ; c'est dire que la synthèse de règle se fait par un procédé automatique également appelé superviseur. Souvent, des réseaux neuronaux y sont associés.

L'évaluation des règles d'inférence étant une opération déterministe, il est tout à fait envisageable de mettre sous forme de tableau ce contrôleur.

Il reste, toutefois, intéressant dans certains systèmes complexes, de garder l'approche linguistique plutôt que d'avoir à faire à un nombre trop important de valeurs précises.

De plus, un algorithme linguistique peut être examiné et discuté directement par quelqu'un qui n'est pas l'opérateur mais qui possède de l'expérience sur le comportement du système.

La formulation linguistique de la sortie permet également d'utiliser le régulateur flou en boucle ouverte donnant ainsi à l'opérateur les consignes à adopter.

Si, après inférence, on se retrouve avec un ensemble flou de sortie caractérisé par l'apparition de plus d'un maximum, cela révèle l'existence d'au moins deux règles contradictoires. Une grande zone plate (figure3,16), moins grave de conséquence, indiquerait que les règles, Dans leur ensemble, sont faibles et mal formulées.

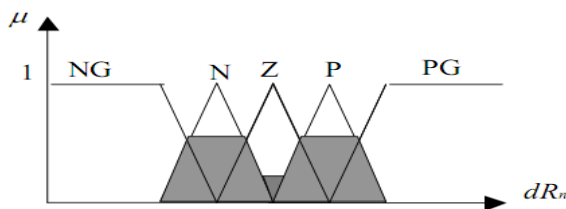


Figure3.16 Cas de règles floues Contradictoires

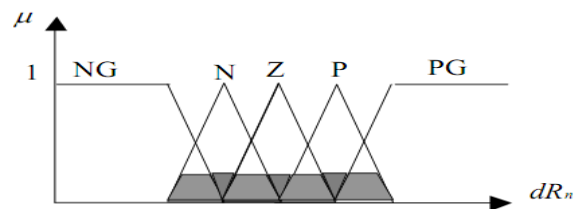


Figure 3.17: Cas des règles floues Mal formulées

3. La logique de prise de décision est le noyau du contrôleur flou, elle est capable de simuler la prise de décision de l'être humain en se basant sur les concepts flous et les règles d'inférences en logique floue.

4. L'inférence de défuzzification réalise les fonctions suivantes:

- Une cartographie d'échelle convertissant la plage des variables de sortie aux univers de discours appropriés;
- Une défuzzification fournissant une action de contrôle (physique) à partir d'une action de contrôle flou.

Pour pouvoir définir la loi de commande, le contrôleur flou doit être accompagné d'une procédure de défuzzification jouant le rôle de convertisseur de la commande floue en valeur physique nécessaire pour un tel état du processus. Il s'agit de calculer, à partir des degrés d'appartenance à tous les ensembles flous de la variable de sortie, l'abscisse qui correspond à la valeur de cette sortie.

3.2.1 Fusion de la logique floue et les PIDs

Un exemple typique est la conception conventionnelle du contrôleur PID (proportionnelle-intégrale-dérivée), dans laquelle la fonction de transfert d'installation linéaire du premier ou du second ordre doit d'abord être donnée. Ici, pour concevoir un contrôleur de logique floue, pas nécessairement de type PID, pour le suivi du point de consigne, nous supposons que la formulation mathématique de l'installation est complètement inconnue.

La plupart des recherches sur le contrôle flou se concentrent sur le problème de la régulation du point de consigne; par conséquent, le contrôle flou est souvent considéré comme une forme de contrôle PID non linéaire et les comparaisons entre contrôle flou et contrôle PID sont fréquentes dans la littérature.

Le contrôle PID conventionnel est cependant bien établi et peut satisfaire aux exigences de performance de la plupart des problèmes de régulation de point de consigne à un coût minimal.

L'amélioration des performances offerte par le contrôle flou ne permet souvent pas de compenser la complexité accrue des calculs et des réglages. Par conséquent, le contrôle PID flou est rare dans les applications commerciales; Les applications commerciales du contrôle flou sont largement axées sur le contrôle de haut niveau, orienté tâche, qui sort du domaine des méthodes de contrôle conventionnelles Chiu [8].

i) Supervision des contrôleurs conventionnels

Dans cette configuration, la stratégie de haut niveau est utilisée pour ajuster les paramètres des boucles de contrôle conventionnelles. Un problème commun aux contrôleurs PID linéaires utilisés pour le contrôle de processus hautement non linéaires est que l'ensemble de paramètres de contrôleur ne produit des performances satisfaisantes que lorsque le processus se situe dans une petite fenêtre opérationnelle. En dehors de cela

Différents paramètres du contrôleur PID sont nécessaires, et ces réglages peuvent être effectués automatiquement par un contrôleur de niveau supérieur (Fig. 3.18).

Le système de contrôle consiste en un contrôleur PID discret conventionnel dont les gains proportionnel, intégral et dérivé (K_P , K_I , K_D) sont modifiés par un superviseur flou à chaque temps d'échantillonnage (le contrôleur flou est donc considéré comme un programmeur de gain). Les entrées du superviseur pourraient être une erreur et le changement d'erreur Nauta Lemke et Krijgsman [9] Zhao et. al. [10] le superviseur est utilisé pour régler les facteurs de mise à l'échelle du contrôleur de PD flou sous-jacent en fonction du dépassement, de la durée et de l'amplitude d'oscillation.

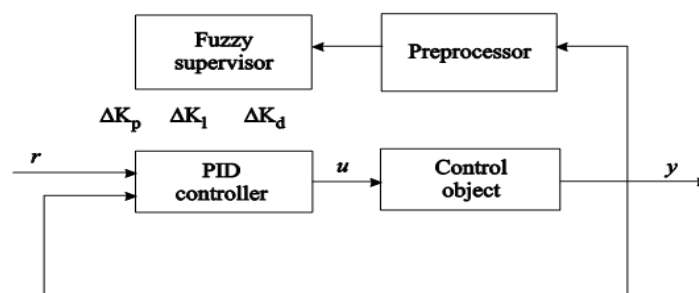


Figure 3.18 Système de contrôle pour le contrôle avancé du PID

Un schéma de programmation de gain alternatif peut être mis en œuvre via l'utilisation d'un système TS homogène du premier ordre avec le format de règle suivant:

$$\text{IF } e \text{ is } A_{1r} \text{ and } \Delta e \text{ is } A_{2r} \text{ Then } y_r = p_{1r} e + p_{2r} \Delta e$$

Chacune de ces règles représente un contrôleur PD local (avec la sélection de variable appropriée, d'autres lois de contrôle peuvent être mises en œuvre) qui sont combinées ensemble au moyen d'une logique floue. C'est donc Il est possible de spécifier une loi de commande distincte pour chaque région d'espace variable déterminée par la sélection des MF en entrée. Une mesure de faible transparence du contrôleur serait utile.

ii) Correction des contrôleurs conventionnels

Normalement, les systèmes de contrôle conventionnels, basés sur des contrôleurs PID, sont capables de contrôler le processus lorsque le fonctionnement est régulier et proche des conditions normales.

Cependant, si des changements soudains se produisent ou si le processus entre dans des situations anormales, une configuration capable de ramener le processus à un fonctionnement normal aussi rapidement que possible peut être utile. Cette idée peut être mise en œuvre en utilisant la connexion parallèle des contrôleurs flous et PID (figure 3.19). Pour un fonctionnement normal, l'ajout à la sortie du PID contrôleur est égal à zéro, alors que dans des situations anormales, le contrôleur flou développe une sortie non nulle qui restaure l'état normal.

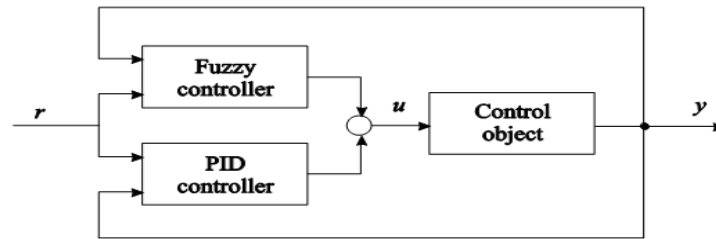


Figure 3.19 Connexion parallèle des commandes floues et PID

iii) Coordination des points de consigne de la boucle de commande

La logique floue peut être appliquée non seulement pour le calcul de la variable de contrôle (c'est-à-dire le contrôle direct), mais également pour la modification de la stratégie de contrôle en général, en utilisant un superviseur flou qui ajuste le point de consigne du contrôleur (figure 3.20).

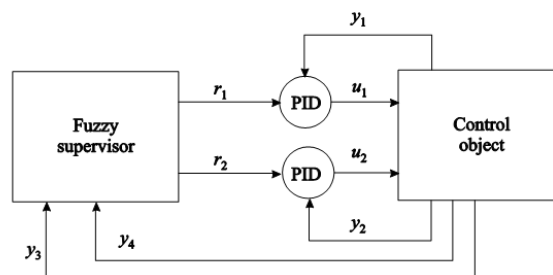


Figure 3.20 Système de contrôle pour la modification de la stratégie de contrôle

L'histoire de cette configuration de contrôle hiérarchique floue remonte à la première application industrielle du contrôle flou dans laquelle le superviseur humain du four à ciment a été remplacé. Une telle utilisation de la logique floue peut être très efficace, car la stratégie de contrôle de haut niveau est souvent formulée en langage naturel et la traduction dans le langage des systèmes flous peut être relativement simple. Le contrôle du point de consigne, en revanche, est souvent plus efficace s'il est implémenté avec des outils classiques pour les

raisons indiquées précédemment. Les applications décrites au chapitre 6 utilisent exclusivement ce type d'architecture de contrôle hiérarchique.

3.2.2 Inversion des systèmes flous

Le processus de conception des contrôleurs flous, qui utilise des informations heuristiques provenant d'experts humains, a de nombreuses applications réussies. L'apprentissage supervisé de contrôleurs flous a également trouvé une utilisation. Les deux approches présentent toutefois plusieurs inconvénients, comme cela a déjà été décrit dans l'introduction. Il peut être difficile d'effectuer la synthèse initiale du contrôleur ou de maintenir les performances requises au fil du temps (en raison de la dérive imprévisible des paramètres de la centrale ou présence de bruit ou de tout autre type de perturbation).

La solution consiste à utiliser des contrôleurs flous à auto-apprentissage capables de s'adapter à différentes conditions de la centrale, c'est-à-dire un contrôle flou adaptatif.

D'autre part, on peut postuler que le but ultime de la conception du contrôleur est de dériver le modèle inverse du processus. En théorie, l'utilisation d'un modèle inverse présente les avantages du contrôle en boucle ouverte, c'est-à-dire la stabilité inhérente et un contrôle parfait avec zéro erreur. La préoccupation majeure est que si la configuration inverse existe réellement ou si elle est physiquement réalisable. Inversion globale du système, où tous les états deviennent des sorties de l'inversé model et la sortie du système d'origine devient la variable d'état (figure 3.21, centre) a normalement une solution non unique et doit être fournie par une famille de solutions. En cas de partielle inversion, un seul des états du système d'origine devient une sortie du modèle inversé et les autres états ainsi que la sortie d'origine sont les entrées du modèle inversé (figure 3.21, à droite).

Un modèle partiellement inversé peut également être plus facilement intégré au système de contrôle que l'inversion globale.

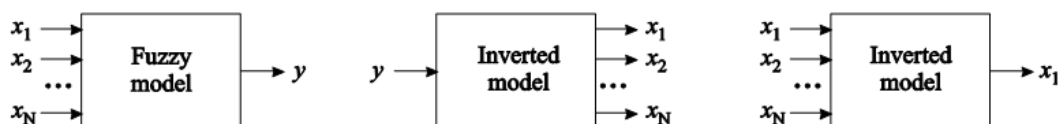


Figure. 3.21 Inversion globale (au centre) et partielle (à droite) du modèle flou d'origine (à gauche)

L'inversion partielle a souvent une solution unique mais pas nécessairement (le modèle d'origine doit être strictement monotone par rapport à l'état inversé pour être inversible).

3.2.3 Inversion numérique des systèmes flous

La méthode la plus intuitive pour obtenir l'inversion consiste à inverser les données d'entrée et de sortie et à former un modèle inverse d'un système ou d'un processus, comme indiqué dans la (figure 3.21, à gauche). Pour l'amour de simplicité, seuls deux états sont introduits avec $x_1 = u(k)$, $x_2 = y(k)$, $y = y(k+1)$. Ce type de formation est pratiqué depuis longtemps avec des réseaux de neurones. Cette approche présente deux inconvénients majeurs: d'abord, si plusieurs valeurs de u sont possibles pour une même sortie de Le processus (mappage de plusieurs à un, figure.3.32. à droite) et une approche des moindres carrés est utilisé, l'algorithme d'identification mappe y à la moyenne de tout u , ce qui peut conduire à un modèle inverse dénué de sens.

Deuxièmement, il est difficile de choisir le signal approprié u pour l'apprentissage inverse, comme on le suppose pour travailler sur une large gamme d'amplitudes de y et pour une large bande passante.

Le deuxième problème peut être résolu avec le schéma d'apprentissage en ligne décrit à la figure 3.34. Dont l'application a été rapportée avec succès dans Jang (1995)[82]. Ici, le contrôleur est la copie du modèle inverse et traite plutôt la sortie u ; nous devons choisir r qui est le point de consigne du processus contrôlé que nous connaissons habituellement.

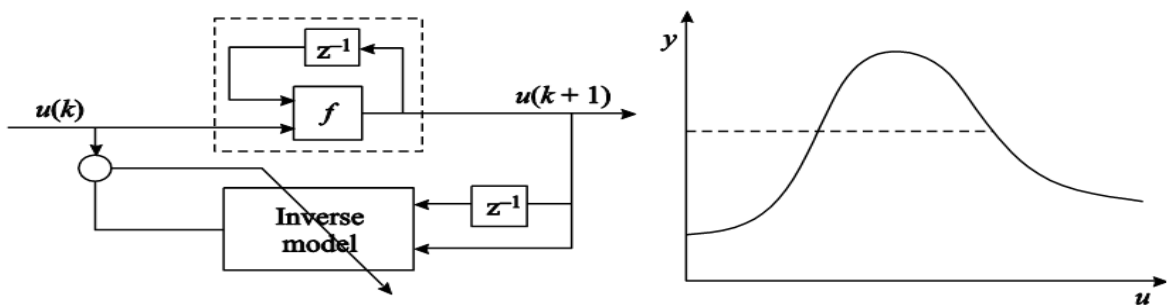


Figure 3.22 Apprentissage du modèle de processus inverse (gauche), fonction non inversible (droite)

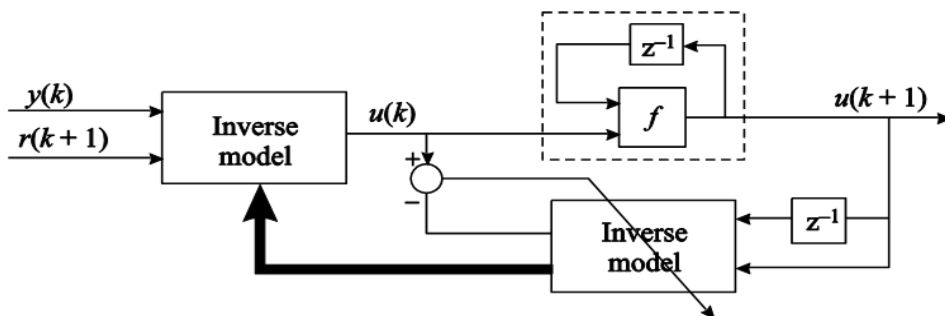


Figure 3.23 Apprentissage inverse en ligne

Une autre technique d'inversion numérique généralement associée aux réseaux de neurones est la propagation dans le temps (propagation dans le temps) qui a été inventée par plusieurs chercheurs simultanément dans les années 1980. L'une des applications les plus connues de cette technique est la simulation camion haut-porteur.

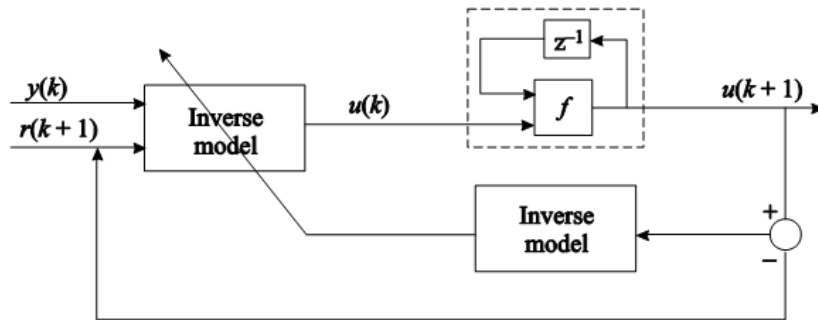


Figure 3.24 Back propagation dans le temps

Le but de la formation est de minimiser la fonction de coût (3.12) en modifiant le jeu de paramètres le contrôleur.

$$K[\theta(K)] = \frac{1}{2} (r(K) - y(K))^2 \quad (3.12)$$

Nous appliquons la méthode de descente de gradient en utilisant la règle de chaîne:

$$\frac{\partial J[\theta(K)]}{\partial \theta(K)} = \frac{\partial J[\theta(K)]}{\partial y(K)} \frac{\partial y(K)}{\partial u(K)} \frac{\partial u(K)}{\partial \theta(K)} \quad (3.13)$$

Notez que l'erreur est renvoyée à travers le modèle (bien que les paramètres du modèle ne soient pas mis à jour par descente de gradient). Le modèle du système peut être un émulateur neural, un système neuro-flou ou même un ensemble d'équations mathématiques, où la méthode mise en œuvre avec ANFIS est utilisée pour équilibrer le pendule inversé.

3.3 Conclusion

Au cours des dernières décennies, le contrôle flou a été appliqué avec succès à de nombreux problèmes de contrôle. Ses applications concernent une large gamme de produits allant des simples produits grand public aux systèmes de contrôle de systèmes technologiques complexes.

Les principales forces du contrôle de la logique floue ont sans doute été:

1. Il peut être utilisé dans des systèmes difficiles à modéliser mathématiquement. En particulier, les systèmes à réponses non linéaires difficiles à analyser peuvent répondre à une approche de contrôle flou.
2. En tant qu'approche de contrôle basée sur des règles, le contrôle flou peut être utilisé pour représenter efficacement les connaissances d'un expert sur un problème.
3. Les variables continues peuvent être représentées par des constructions linguistiques plus faciles à comprendre, ce qui facilite la mise en œuvre et la modification du contrôleur.
4. Les contrôleurs flous peuvent être moins sensibles au bruit du système et aux modifications de paramètres, ce qui les rend plus robustes.
5. Les processus complexes peuvent souvent être contrôlés par relativement peu de règles logiques, ce qui permet une conception de contrôleur plus compréhensible et un calcul plus rapide pour les applications en temps réel.

Un contrôleur flou est construit pour prendre des décisions sur ce à quoi les entrées de contrôle de l'installation doivent recevoir les versions traitées des sorties de l'usine et des entrées de référence. C'est une forme de système décisionnel artificiel (c'est-à-dire non biologique). Les systèmes décisionnels trouvent de nombreuses applications dans de nombreux domaines, en plus de ceux traditionnellement étudiés dans les systèmes de contrôle. Par exemple, l'étude de cas sur la planification des machines de la section précédente montre une application non traditionnelle du contrôle de rétroaction dans laquelle un système flou peut jouer un rôle utile en tant que système décisionnel.

Il existe de nombreux autres domaines dans lesquels des systèmes de prise de décision flous peuvent être utilisés, notamment:

- Fabrication: planification et planification du flux de matériaux, de l'allocation des ressources, du routage et de la conception des machines et des équipements.
- Systèmes de trafic: routage et commutation de signaux.
- Robotique: planification de chemin, planification des tâches, navigation et planification de mission.
- Ordinateurs: allocation de mémoire, planification des tâches et conception matérielle.

- Industries de traitement: surveillance, évaluation de la performance et diagnostic des défaillances.
- Science et médecine: systèmes de diagnostic médical, surveillance de la santé et interprétation automatisée des données expérimentales.
- Entreprises: finance, évaluation du crédit et analyse du marché boursier.

Cette liste n'est nullement exhaustive. Pratiquement tout système de prise de décision informatique au potentiel de tirer parti de l'application de la logique floue pour permettre des décisions «souples» lorsqu'il est nécessaire de prendre des décisions dans des conditions d'incertitude. Manque de moyens fiables pour l'analyse de l'optimalité et de la stabilité du contrôleur.

La plupart des contrôleurs à logique floue existants depuis l'application Mamdani, Mamdani et Assilian, sont des contrôleurs de point de consigne qui répondent aux termes proportionnel et dérivé (PD) ou proportionnel et intégral (PI).

Dans ce chapitre nous avons donné un aperçu globale concernant les techniques d'intelligence artificielle. Aussi nous avons présente quelques exemples de réglages intelligentes des contrôleurs PIDs.

Dans le chapitres quatre, nous proposons d'autres outils pour réaliser un suivi de trajectoire en se basant sur une classe appelés modèles de type Takagi-Sugeno (TS). Un des avantages de cette méthode est qu'il devient relativement simple et systématique d'assurer la stabilité de la boucle fermée complète. Ces stratégies de commande, aussi permettant d'assurer un suivi de trajectoire.

Chapitre 4

Stratégie de la commande intelligente d'un robot mobile

Introduction

Dans ce chapitre, On présente des stratégies de contrôle en combinant le contrôle conventionnel et le contrôle intelligent pour le suivi de trajectoire d'un robot mobile non holonome de type voiture. L'étude est basée sur la combinaison de la commande classique PIDs, les PIDs numériques, des modèles flous et des réseaux neurones artificiels.

Des tests d'une série de simulation ont été effectués pour évaluer les contrôleurs proposés, contrôleur PID- flou, contrôleur PID basé sur un réseau neuronal à fonction de base radiale et contrôleur PID basé sur un réseau neuronal à propagation arrière. Pour la mise en œuvre des commandes développées pour réaliser la poursuite de trajectoire pour le robot mobile, Les simulations ont été réalisées sur la plateforme de Matlab-Simulink.

4.1 Contrôleur PID-floue

Divers types de contrôleurs PID flous (y compris PI et PD) ont été proposés. Dans les littératures récentes, les contrôleurs PID flous sont divisés en deux grandes catégories, selon leurs structures [83,84]:

La première catégorie de contrôleurs PID flous, implique les contrôleurs PID conventionnels conjointement avec un ensemble de règles floues et un mécanisme de raisonnement flou pour régler les gains PID en ligne [85]. Les contrôleurs PID flous sont utilisés au lieu du contrôleur PID linéaire dans tous les contrôles classiques ou modernes. L'erreur entre la variable mesurée et la variable de référence a été convertie en commandes qui sont appliquées à l'actionneur d'un processus. Les informations sur leurs caractéristiques de transfert d'entrée-sortie équivalentes sont importantes dans une conception pratique.

La deuxième catégorie de contrôleurs PID flous est composée de contrôleurs à logique floue (FLC) typiques réalisés comme un ensemble de règles de contrôle heuristique. Afin d'être cohérent avec la nomenclature [86] et de se distinguer de la première catégorie de contrôleurs PID flous, les FLC de cette catégorie sont appelés FLC de type PID (PI-like ou PD-like). La plupart des recherches sur la conception de commandes à logique floue se réfèrent à cette catégorie [87].

4.1.1 Structure du contrôleur flou-PID

Le contrôleur flou-PID à réglage automatique contient deux parties: un contrôleur PID et un contrôleur à logique floue (FLC), comme illustré à la figure 4.1. Le contrôleur flou-PID proposé est un contrôleur auto-adaptatif conçu par un contrôleur à logique floue incrémentielle. Le contrôleur à logique floue est utilisé pour réguler les paramètres du contrôleur PID, K_p, K_I et K_d en ligne par des règles de contrôle à logique floue pour de meilleures performances de contrôle PID dans différentes situations.

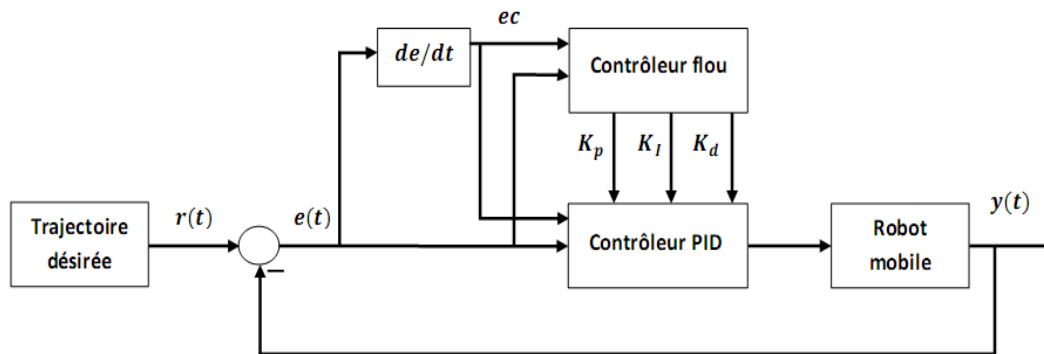


Figure 4.1 Structure du contrôleur flou-PID

Étant donné que le contrôleur PID auto-ajusté flou proposé vise à améliorer les performances de contrôle fournies par un contrôleur PID, il conserve la structure simple du contrôleur PID et il n'est pas nécessaire de modifier les parties matérielles du système de contrôle d'origine pour la mise en œuvre. La limitation de l'algorithme de contrôle traditionnel est largement rendue complète.

4.1.2 Modèle cinématique du robot mobile

Le robot que nous étudions est un robot mobile à quatre roues se déplaçant dans un plan idéal, de forme similaire à un rectangle. Il est supposé que pendant le mouvement, le plan de chaque roue reste continuellement perpendiculaire au sol. Dans chaque cas, le contact entre la roue et le sol est réduit à un seul point. La figure 4.2 illustre la position du robot sur un plan cartésien. La position et l'orientation du robot peuvent être définies par $[x_p \ x_p \ \theta]^T$, où $(x_p \ x_p)$ sont les coordonnées de la position du centre de robot, θ désigne l'orientation du robot dans le plan (o, \vec{x}, \vec{y}) [88]. Cependant, le robot non-Holonome exige que la vitesse du robot soit dans le plan des roues, ce qui donne le modèle du robot par la formule (4.1).

$$r(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

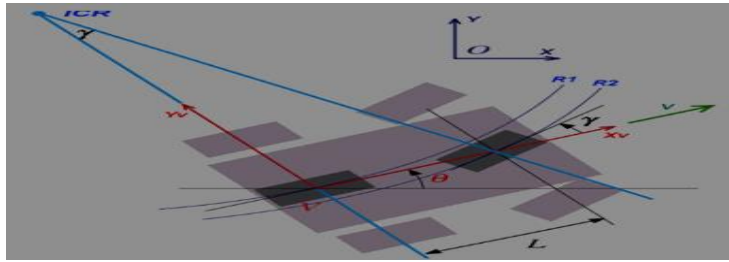


Figure4.2 modélisation cinématique du robot mobile

Comme le montre la figure 4.2, L est la distance entre les lignes avant et arrière, γ est l'angle d'attaque, R est le rayon de braquage, v est la vitesse linéaire [11]. Ainsi, le modèle cinématique du robot mobile peut être présenté par les équations suivantes (4.2, 4.3 et 4.4).

$$\dot{x} = v \cos \theta \quad (4.2)$$

$$\dot{y} = v \sin \theta \quad (4.3)$$

$$\dot{\theta} = \frac{v}{l} \tan \gamma \quad (4.4)$$

4.1.2.1 Modélisation du robot mobile

Dans cette partie, nous donnons une vue du modèle du robot sous Simulink, le robot sera modélisé par cinq variables: les entrées sont la vitesse de déplacement v (m/s) et γ la vitesse angulaire, les sorties sont : la position du robot mobile (x, y) et l'angle entre l'horizontal et la normale de la trajectoire. La figure 4.3 montre le modèle du robot sous Simulink.

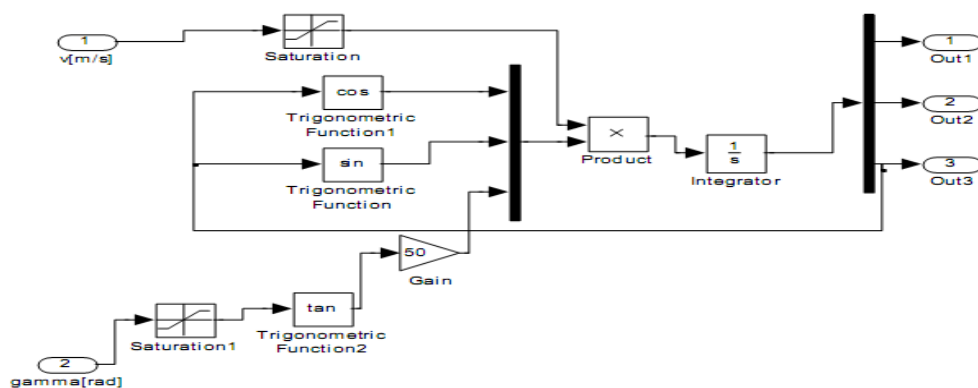


Figure4.3 Modèle Simulink du robot mobile

4.1.3 Algorithme de contrôle PID

Le robot est contrôlé par un contrôleur PID, qui utilise l'erreur du système, (e), et le taux de changement d'erreur, (ec); comme des entrées, qui sont présentées par les équations

(4-5 et (4-6). $e(k)$ est l'erreur entre la sortie réelle et la sortie désirée et peut être exprimée comme suit:

$$e(k) = r(k) - y(k) \quad (4.5)$$

$ec(k)$ est le taux de variation de $e(k)$ et qui est donné par :

$$ec(k) = e(k) - e(k - 1) \quad (4.6)$$

L'algorithme de contrôle PID peut être exprimé comme suit:

$$u(k) = k_p(e(k) - e(k - 1)) + k_i e(k) + k_d(e(k) - 2e(k - 1) + e(k - 2)) \quad (4.7)$$

Les performances du contrôle PID dépendent principalement de la sélection des paramètres K_p , K_I et K_d de PID. Le bloc flou de la stratégie de contrôle flou-PID est conçu pour ajuster automatiquement les valeurs des paramètres.

4.1.4 Conception du contrôleur flou

Le contrôleur de logique floue a pour fonction de régler les paramètres K_p , K_I et K_d du contrôleur PID, en ligne par des règles de contrôle à logique floue basées sur les variables (e) et (ec) dans le temps comme le montre la figure 4.4 [89].

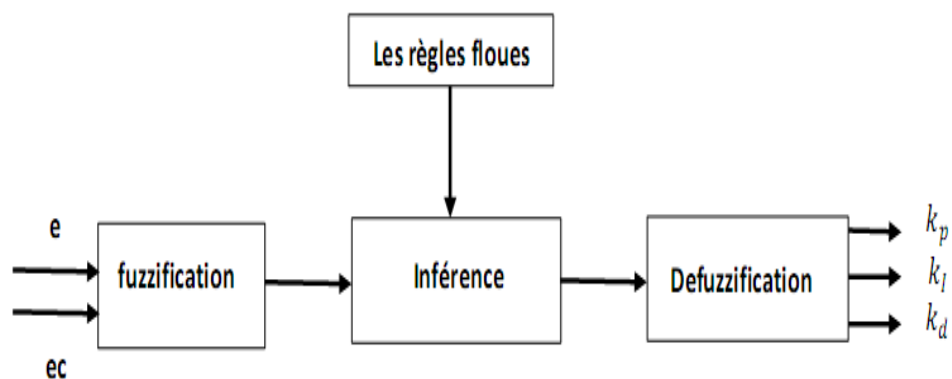


Figure 4.4 Schéma bloc du contrôleur flou

L'auto-réglage flou des paramètres PID consiste à trouver la relation floue entre les trois paramètres PID, (e) et (ec). Il mesure la sortie (y) du système et calcule ensuite (e) et (ec) basés sur la sortie (y) et l'entrée (r). Le contrôleur flou ajuste les paramètres par des règles de contrôle flou en ligne afin que le robot obtienne de meilleures performances. Par conséquent, il est nécessaire de comprendre la fonction de chaque paramètre PID. Il est possible de déterminer la relation entre la sortie floue K_p , K_I et K_d et l'entrée floue (e) et (ec)

pour construire les règles floues. Les fonctions de chaque paramètre PID sont décrétées de manière dont elles affectent les performances de contrôle et se rapportent à l'erreur du système et est répertoriée dans le tableau 4-1[90].

Tableau 4-1 Effets des réglages K_p , K_I et K_d

réponse en Boucle fermée	Temps de montée	temps de Réglage	Dépassement	Erreur de régime permanent	stabilité
Croissant de K_p	Diminution	augmentation	Petite augmentation	Diminution	Dégradation
Croissant de K_i	Petite diminution	augmentation	augmentation	grande augmentation	Dégradation
Croissant de K_d	Petite diminution	diminution	diminution	Minimal changement	Amélioration

Tableau 4-2 Base des règles floues de K_p

Δk_p e \ ec	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZO	ZO
NM	PB	PB	PM	PS	PS	ZO	NS
NS	PM	PM	PM	PS	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	PS	ZO	NX	NM	NM	NM	NB
PB	ZO	ZO	NM	NM	NM	NB	NB

La base des règles floues est établie selon la discussion des effets des paramètres PID, qui sont présentés dans le tableau 4.1. La base des règles floues contient trois matrices qui présentent la manière de changement de K_p , K_I et K_d , (ΔK_p , ΔK_I et K_D) lorsque les variables (e) et (ec) changent, comme indiqué dans le tableau 4.3. La base des règles floues est construite en utilisant plusieurs déclarations if-then, hypothèse et résultat pour chaque phrase qui sont des propositions floues. Le tableau 4-2 - le tableau 4-4 indique que 25 règles définissent la base des règles pour le contrôleur de type PID-flou. Les variables floues sont

définies pour la base de règles comme: e , ec , ΔK_p , ΔK_I et K_D (= {NB (Negative Big), NM (Negative Medium), NS (Negative Small), ZO (Zero), PS (Positive Small), PM (Positive Medium), PB (Positive Big)})

Tableau 4-3 Base des règles floues de K_I

Δk_i \ ec	NB	NM	NS	ZO	PS	PM	PB
e							
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NB	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	NS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	ND	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB

Table 4.4 Base des règles floues de K_D

Δk_a \ ec	NB	NM	NS	ZO	PS	PM	PB
e							
NB	PS	NS	NB	NB	NB	NM	PS
NM	PS	NS	NB	NM	NM	NS	ZO
NS	ZO	NS	NM	NM	NS	NS	ZO
ZO	ZO	NS	NS	NS	NS	NS	ZO
PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
PM	PB	NS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB

Une fois la base de règles définie, les fonctions d'appartenance pour e , ec , Δk_p , Δk_i et Δk_a doivent être déterminées comme indiqué à la figure 4.5 et à la figure 4.6. Une fonction d'appartenance est une courbe qui définit comment chaque point de l'entrée l'espace est mappé à une valeur d'appartenance (sur le degré d'appartenance) entre 0 et 1[91]. Dans ce cas, les fonctions d'appartenance combinées triangulaire et gauss sont utilisées pour toutes les variables. Le domaine physique [92] de (e) et (ec) est $\{-3, -2, -1, 0, 1, 2, 3, \}$; le domaine physique de ΔK_p est $\{-0,3, -0,2, -0,1, 0, 0,1, 0,2, 0,3\}$; le domaine physique de ΔK_I est $\{,06, -0,04, -0,02, 0, 0,02, 0,04, 0,06\}$ et celui de K_D est $\{-4, -3, -2, -1,0, 1, 2, 3, 4\}$.

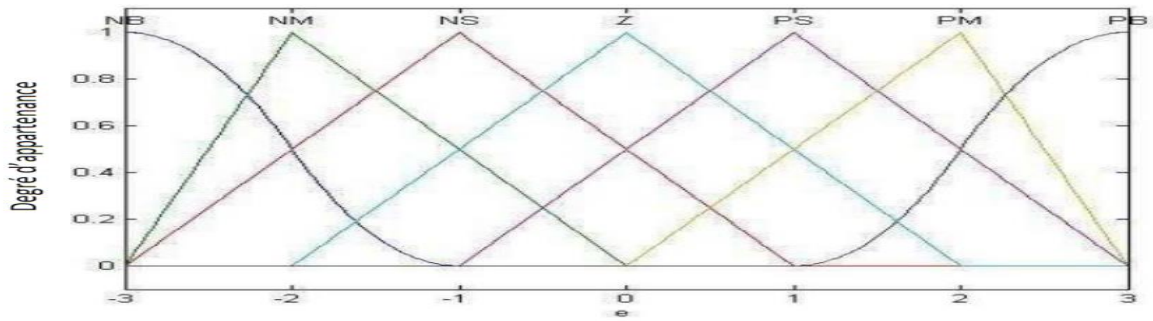


Figure 4.5 Fonction d'appartenance de l'erreur (e)

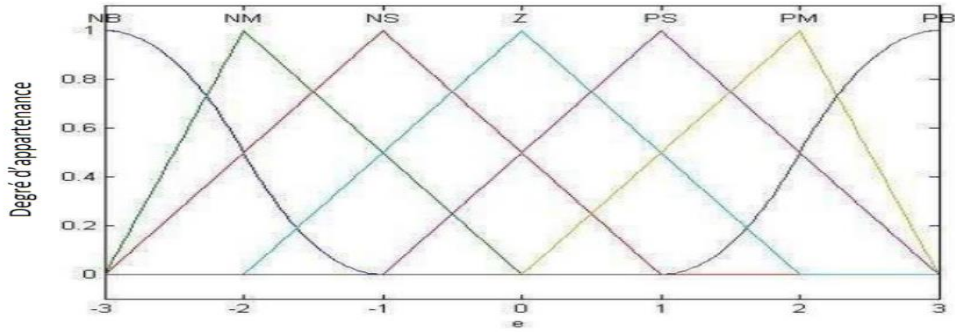


Figure 4.6 Fonction d'appartenance du taux de changement d'erreur (ec)

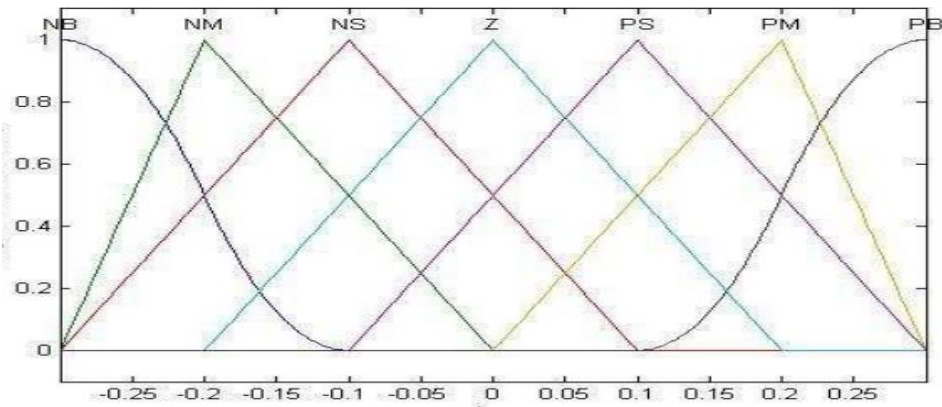


Figure 4.7 Fonction d'appartenance de K_p

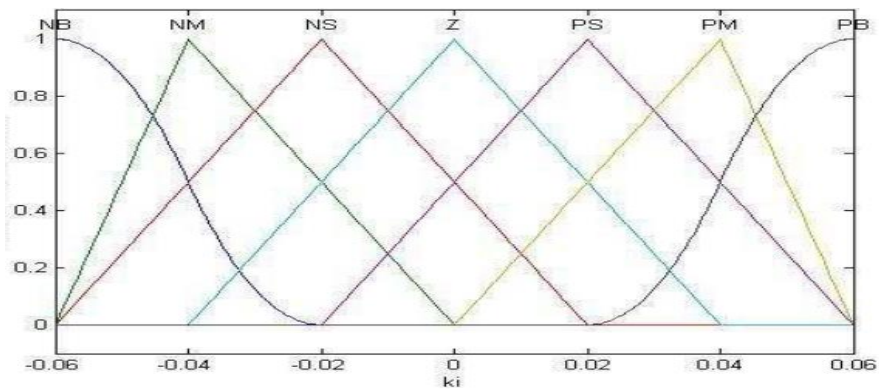


Figure 4.8 Fonction 'appartenance de K_i

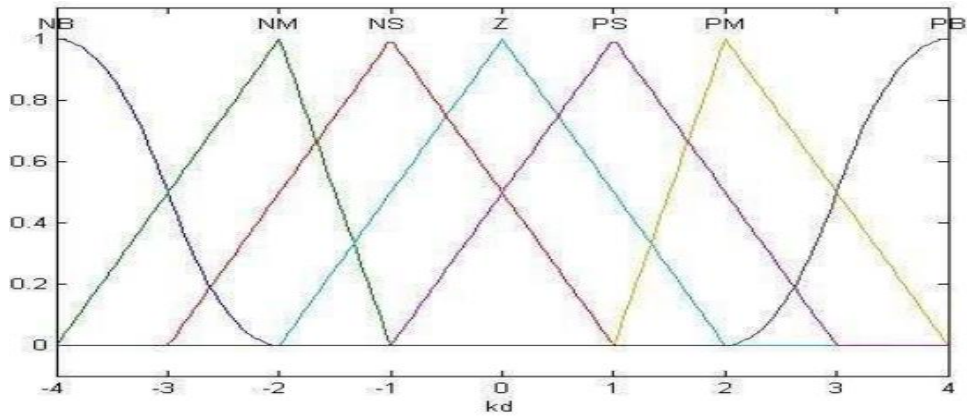


Figure 4.9 Fonction d'appartenance de K_D

Puisque nous pouvons obtenir les valeurs de $(\Delta K_p, \Delta K_I$ et $K_D)$ basée sur des règles floues et la fonction d'appartenance. Ensuite, les valeurs de K_p, K_I et K_d peuvent être calculés en utilisant les équations suivantes:

$$k_p(k+1) = f_{kp}(e, ec) = \hat{k}_p(k) + \Delta k_p(k) \quad (4.8)$$

$$k_i(k+1) = f_{ki}(e, ec) = \hat{k}_i(k) + \Delta k_i(k) \quad (4.9)$$

$$k_d(k+1) = f_{kd}(e, ec) = \hat{k}_d(k) + \Delta k_d(k) \quad (4.10)$$

Les valeurs de K_p, K_I et K_d désirées peuvent être obtenues à l'aide du FLC et transférées au Contrôleur PID afin de guider le robot de suivre la trajectoire souhaitée et faire la correction de la trajectoire en temps réel.

Le processus de contrôle du contrôleur PID-flou est résumé comme suit et est illustré à la figure 4-15 :

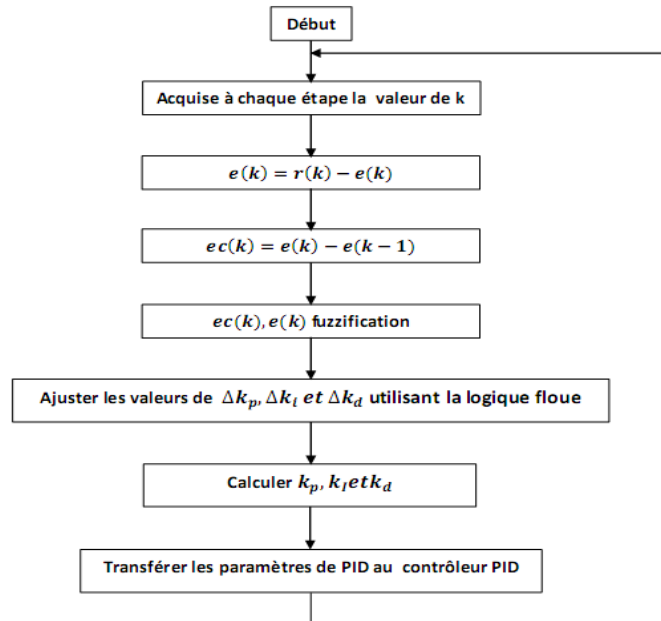
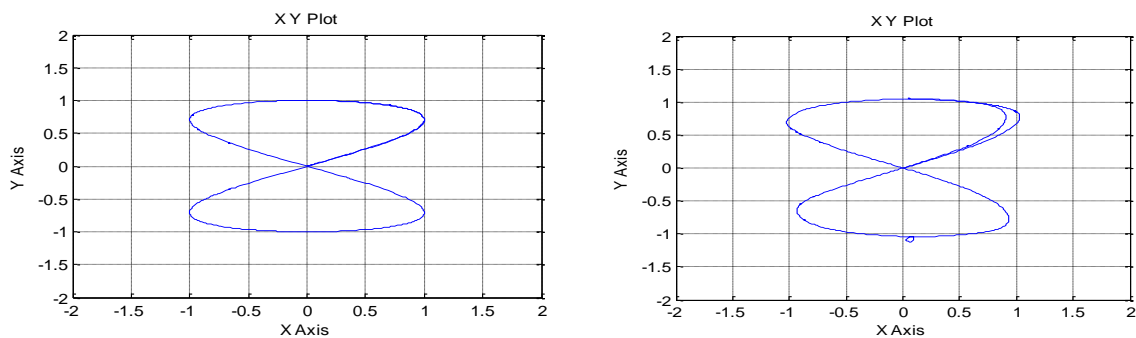


Figure 4.10 Organigramme du contrôleur de PID-flou

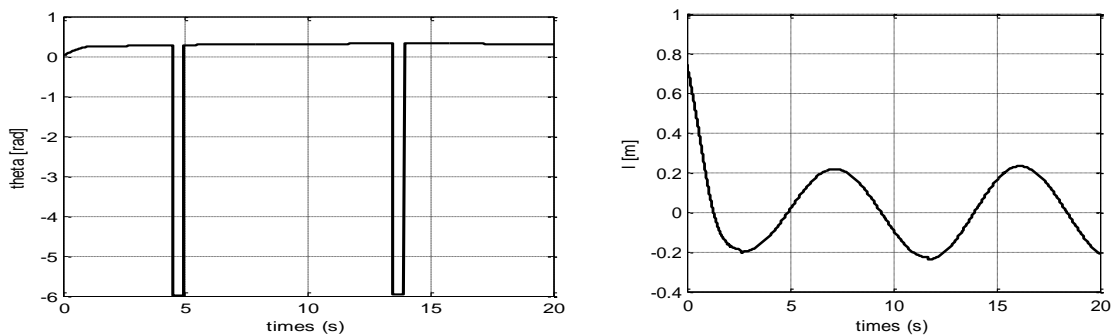
Après plusieurs tests avec l'application d'une trajectoire sous forme de huit (8), Les résultats de simulation obtenus sous matlab-simulink sont illustrés ci-dessous :



a) trajectoire désirée

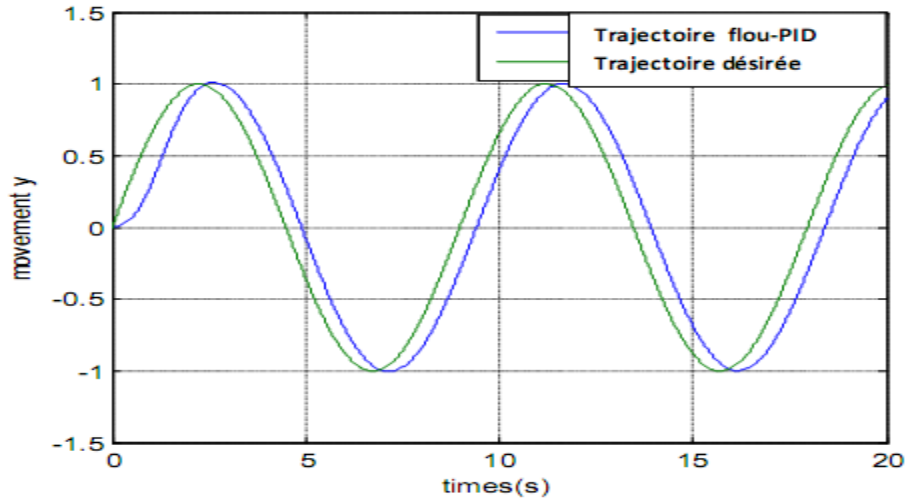
b) trajectoire suivie

Figure 4.11 Suivi de trajectoire de forme8



a) Erreur de l'orientation theta

b) déplacement suivant l'axe x



b) Déplacement suivant l'axe y

Figure 4.12 Le chemin suivi par le robot, l'erreur de distance l et l'erreur d'orientation θ

4.2 Contrôleur flou-PI+D

Le contrôleur analogique PI+D conventionnel illustré à la figure 4.19. Nous le discrétisons d'abord en appliquant la transformation bilinéaire, puis on conçoit séparément le PI flou et le D flou, et enfin les combiner ensemble dans un système en boucle fermée.

En commençant par le système de contrôle PI + D analogique conventionnel illustré à la figure 4.19, la sortie U_{PI} du contrôleur PI analogique conventionnel dans le domaine de fréquence, est donnée par

$$u_{PI}(s) = \left[K_p^c + \frac{K_I^c}{s} \right] E(s), \quad (4.11)$$

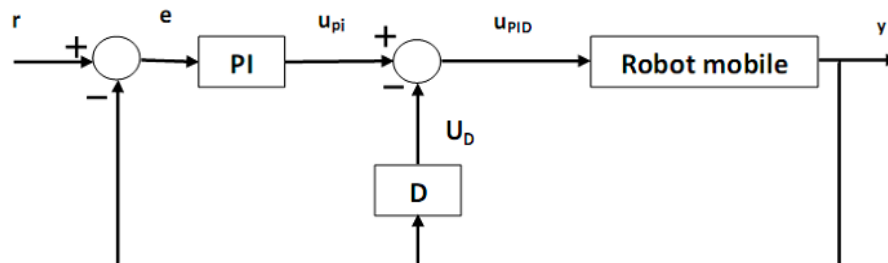


Figure 4.13 Système de contrôle classique PI + D à temps continu.

Où K_p^c et K_I^c sont les gains proportionnel et intégral, respectivement, et $E(s)$ est le signal d'erreur de suivi, après la transformation de Laplace. Cette équation peut être

transformée en version discrète en appliquant la transformation bilinéaire (4.12), ce qui conduit à la forme:

$$u_{PI}(z) = \left[K_p^c - \frac{K_i^c T}{2} + \frac{K_i^c T}{1-z^{-1}} \right] E(z) \quad (4.12)$$

$$K_p = K_p^c - \frac{K_i^c T}{2}$$

$$K_i = K_i^c T, \quad (4.13)$$

puis en prenant la transformation Z inverse, on obtient :

$$u_{PI}(nT) - u_{PI}(nT - T) = K_p [e(nT) - e(nT - T)] + K_i T e(nT)$$

En divisant cette équation par T, on obtient :

$$\Delta u_{PI}(nT) = K_p v(nT) + K_i e(nT), \quad (4.14)$$

Où

$$\Delta u_{PI}(nT) = \frac{u_{PI}(nT) - u_{PI}(nT - T)}{T}$$

$$v(nT) = \frac{e(nT) - e(nT - T)}{T}$$

Il s'ensuit que :

$$u_{PI}(nT) = u_{PI}(nT - T) + T \Delta u_{PI}(nT)$$

Dans la conception du contrôleur PI flou, nous remplacerons le coefficient T par un gain de contrôle flou $k_{u,PI}$, de sorte que :

$$u_{PI}(nT) = u_{PI}(nT - T) + K_{u,PI} \Delta u_{PI}(nT) \quad (4.20)$$

Le contrôleur D du système de contrôle PI+D, comme illustré à la figure 4.19, avec u_D comme entrée et y comme sortie. Il est clair que :

$$U_D(s) = s K_d^c Y(s)$$

où k_d^c est le gain de contrôle et Y (s) le signal de sortie. Sous la transformation bilinéaire, l'équation ci-dessus devient :

$$U_D(Z) = \frac{2}{T} \left(\frac{z-1}{z+1} \right) K_d^c Y(z)$$

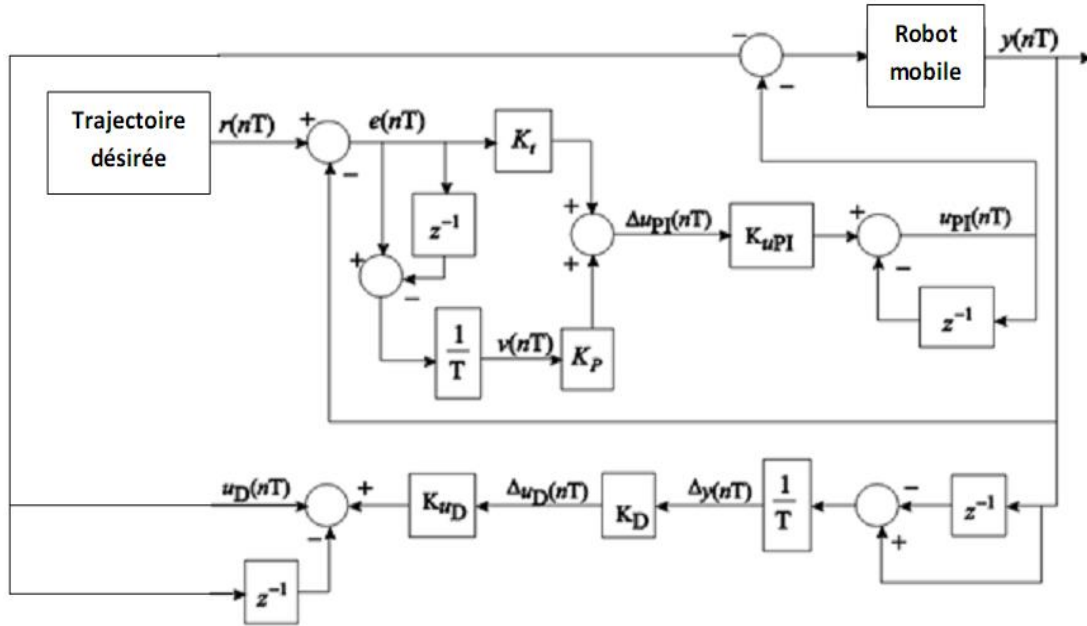


Figure 4.14 Le système de contrôle numérique PI+D conventionnel.

Où

$$U_D(z) = K_d^c \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) Y(z)$$

Par conséquent,

$$u_D(nT) + u_D(nT - T) = \frac{2K_d^c}{T} [y(nT) - y(nT - T)]$$

Ensuite, en divisant cette équation par T, on obtient :

$$\Delta u_D(nT) = K_d \Delta y(nT) \quad (4.21)$$

Où

$$\Delta u_D(nT) = \frac{u_D(nT) + u_D(nT - T)}{T}$$

est la sortie incrémentale du contrôleur flou D, et

$$\Delta y(nT) = \frac{y(nT) - y(nT - T)}{T}$$

Enfin, la loi de commande globale floue-PI+D peut être obtenue en additionnant algébriquement la loi de contrôle des PI flous et la loi D floue ensemble. Le résultat est

$$u_{PID}(nT) = u_{PI}(nT) - u_D(nT),$$

ou plus précisément,

$$u_{PID}(nT) = u_{PI}(nT - T) + K_{u,PI}\Delta u_{PI}(nT) + u_D(nT - T) - K_{u,D}\Delta u_D(nT). \quad (4.23)$$

Cette équation sera appelée loi de commande floue-PI+D et est résumée ci-dessous:

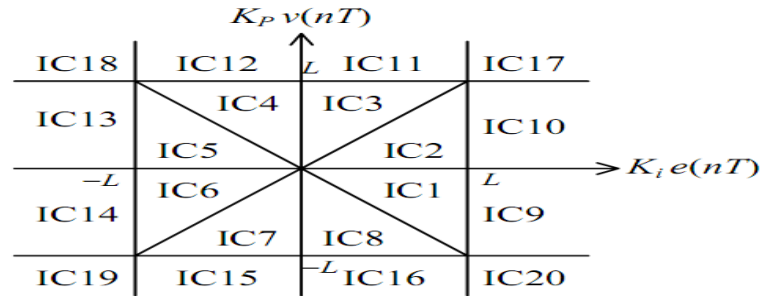


Figure 4.16 Régions des valeurs de combinaison d'entrées du contrôleur flou (PI).

Le système de contrôle flou-PI+D est illustré par la figure 4.15, dans lequel cinq gains de contrôle constants peuvent être ajustés: K_I , K_P , $K_{U,PI}$, K_{DI} et $K_{U,D}$ où on définit $K_{U,PI} = K_{U,D} = T$ pour simplifier la conception. Dans cette implémentation, les deux actions de contrôle incrémentielles $\Delta u_{PI}(nT)$ et $\Delta u_D(nT)$ ont les formules analytiques indiquées ci-dessous:

$$\Delta u_{PI}(nT) = \frac{L[K_i e(nT) + K_P v(nT)]}{2(2L - K_i |e(nT)|)}, \quad \text{en IC1, IC2, IC5, IC6} \quad (4.24)$$

$$= \frac{L[K_i e(nT) + K_P v(nT)]}{2(2L - K_P |e(nT)|)}, \quad \text{en IC3, IC4, IC7, IC8}, \quad (4.25)$$

$$= \frac{1}{2} [L + K_P v(nT)] , \quad \text{en IC9, IC10}, \quad (4.26)$$

$$= \frac{1}{2} [L + K_i e(nT)], \quad \text{en IC11, IC12}, \quad (4.27)$$

$$= \frac{1}{2} [-L + K_i e(nT)] , \quad \text{en IC15, IC16}, \quad (4.28)$$

$$= 0, \quad \text{en IC18, IC20}, \quad (4.29)$$

$$= -L, \quad \text{en IC17}, \quad (4.30)$$

$$= L, \quad \text{en IC19}, \quad (4.31)$$

Pour le contrôleur PI, nous divisons aussi le plan en vingt régions du contrôleur flou (PI) en combinant les gains $(k_i e(nT), k_p e(nT))$ comme illustré à la figure 4.16. Le contrôle incrémentiel du contrôleur PI flou est calculé ensuite par les formules (4.24) - (4.31).

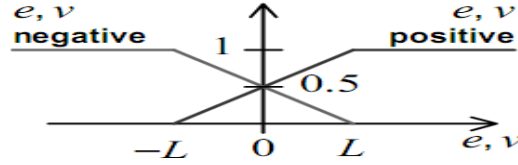


Figure 4.17 fonctions d'appartenance de l'entrée pour le contrôleur PI.

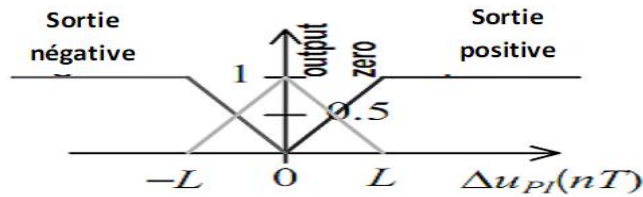


Figure 4.18 fonctions d'appartenance de la sortie pour le contrôleur PI.

Pour le contrôleur flou D, les vingt régions de la combinaison d'entrées sont illustrées à la figure 4.19. Le contrôle incrémentiel du contrôleur flou D est calculé par les formules suivantes:

$$\Delta u_D(nT) = \frac{L[Ky_d(nT) + K_d \Delta y(nT)]}{2(2L - K|y_d(nT)|)}, \quad \text{en IC1, IC2, IC5, IC6,} \quad (4.32)$$

$$= \frac{L[Ky_d(nT) + K_d \Delta y(nT)]}{2(2L - K_d|y_d(nT)|)}, \quad \text{en IC3, IC4, IC7, IC8,} \quad (4.32)$$

$$= \frac{1}{2}[L - K_d \Delta y(nT)], \quad \text{en IC9, IC10,} \quad (4.33)$$

$$= \frac{1}{2}[-L + Ky_d(nT)], \quad \text{en IC11, IC12,} \quad (4.34)$$

$$= \frac{1}{2}[-L - K_d \Delta y(nT)], \quad \text{en IC13, IC14,} \quad (4.35)$$

$$= \frac{1}{2}[L + Ky_d(nT)], \quad \text{en IC15, IC16,} \quad (4.36)$$

$$= 0, \quad \text{en IC18, IC20,} \quad (4.37)$$

$$= -L, \quad \text{en IC17,} \quad (4.38)$$

$$= L, \quad \text{en IC19,} \quad (4.39)$$

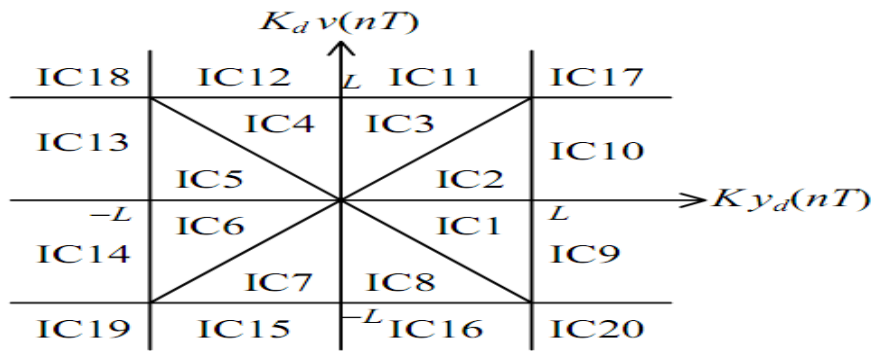


Figure 4.19 Régions des valeurs de combinaison d'entrées du contrôleur flou D

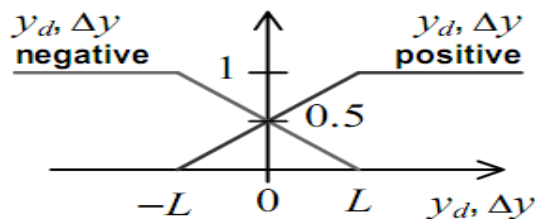


Figure 4.20 fonctions d'appartenance de l'entrée pour le contrôleur D.

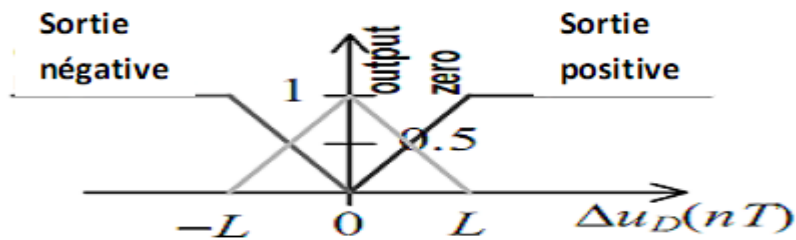


Figure 4.21 fonctions d'appartenance de sortie pour le contrôleur D.

4.2.1 Fuzzification du Contrôleur flou-PI+D

La fuzzification des contrôleurs PI et D du système de contrôle Flou-PI+D se fait individuellement et procéder ensuite à l'établissement des règles de contrôle floues souhaitées pour chacun d'eux en tenant compte de la loi de commande floue-PI+D interconnectée donnée dans équation (4.23). Les fonctions d'appartenance en entrée et en sortie du PI sont illustrées aux figures 4.17 et 4.18, respectivement.

En utilisant ces fonctions d'appartenance, les règles de contrôle suivantes sont établies pour le contrôleur PI flou:

R^1 : IF $e = e_n$ AND $v = v_n$ THEN PI-output = o_n .

R^2 : IF $e = e_n$ AND $v = v_p$ THEN PI-output = o_z .

R^3 : IF $e = ep$ AND $v = vn$ THEN PI-output = oz .

R^4 : IF $e = ep$ AND $v = vp$ THEN PI-output = op

Dans ces règles, $e = r - y$ est l'erreur, $v = \dot{e} = 0 - \dot{y} = -\dot{y}$ est le taux de changement de l'erreur, la sortie de contrôle PI flou Δu_{PI} (nT), «ep» signifie «Erreur positive», «op» signifie «sortie positive», etc. En outre, AND est la logique ET opérateur définie comme précédemment.

Les fonctions d'appartenance en entrée et en sortie pour le contrôleur flou D sont montrées aux figures 4.20 et 4.21, respectivement.

De même, à partir des fonctions d'appartenance du contrôleur flou D, les règles de contrôle suivantes sont utilisées pour le contrôleur D:

R^5 : IF $y_d = y_{dp}$ AND $\Delta y = \Delta y_p$ THEN D-output = oz .

R^6 : IF $y_d = y_{dp}$ AND $\Delta y = \Delta y_n$ THEN D-output = op .

R^7 : IF $y_d = y_{dn}$ AND $\Delta y = \Delta y_p$ THEN D-output = on .

R^8 : IF $y_d = y_{dn}$ AND $\Delta y = \Delta y_n$ THEN D-output = oz

Ces huit règles donnent les actions de contrôle pour la loi de contrôle flou-PI+D.

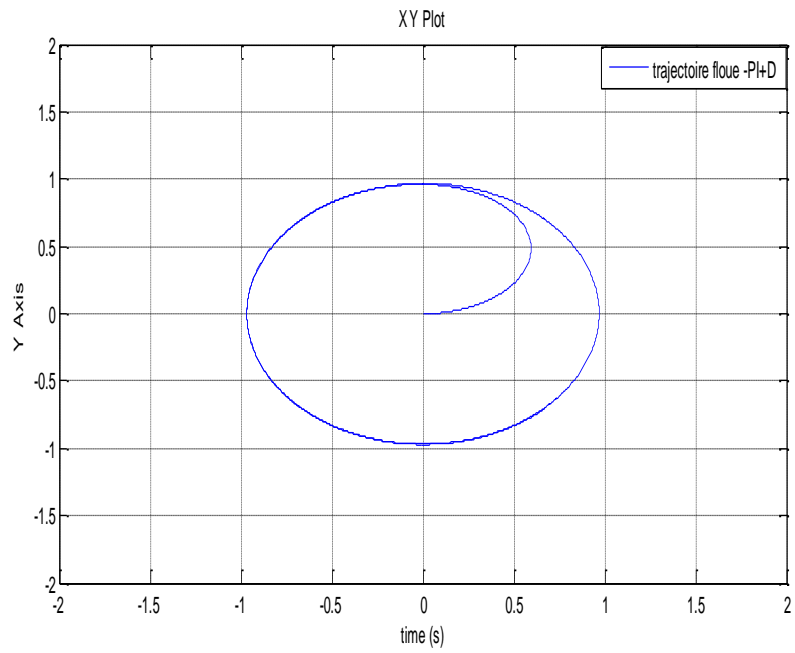
4.2.2 Defuzzification du Contrôleur flou-PI+D

Dans l'étape de défuzzification, pour les deux contrôleurs flous PI et D, la même formule moyenne pondérée est utilisée pour défuzzifier le contrôle incrémental de la loi de commande floue:

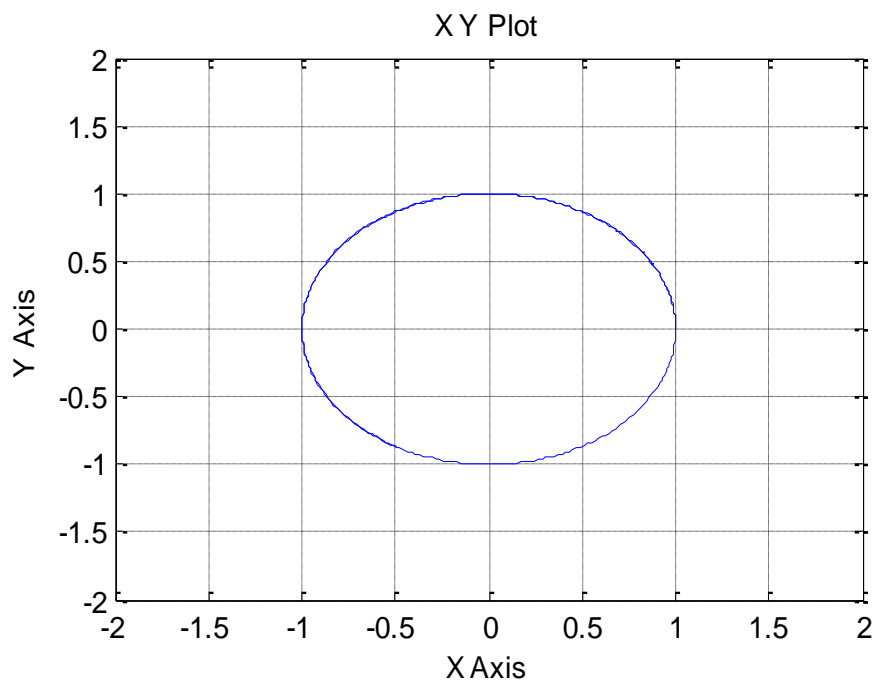
$$\Delta u (nT) = \frac{\sum(\text{valeur d'appartenance de l'entrée}) \times \text{valeur correspondante de la sortie}}{\sum(\text{valeur d'appartenance de l'entrée})} \quad (4.40)$$

Pour le contrôleur PI flou, les plages de valeurs des deux entrées, l'erreur et le taux de variation de l'erreur sont décomposées en vingt régions adjacentes combinées des différentes entrées (IC), comme illustré à la figure 4.19.

Les résultats de simulation sous Matlab-Simulink nous ont donné :

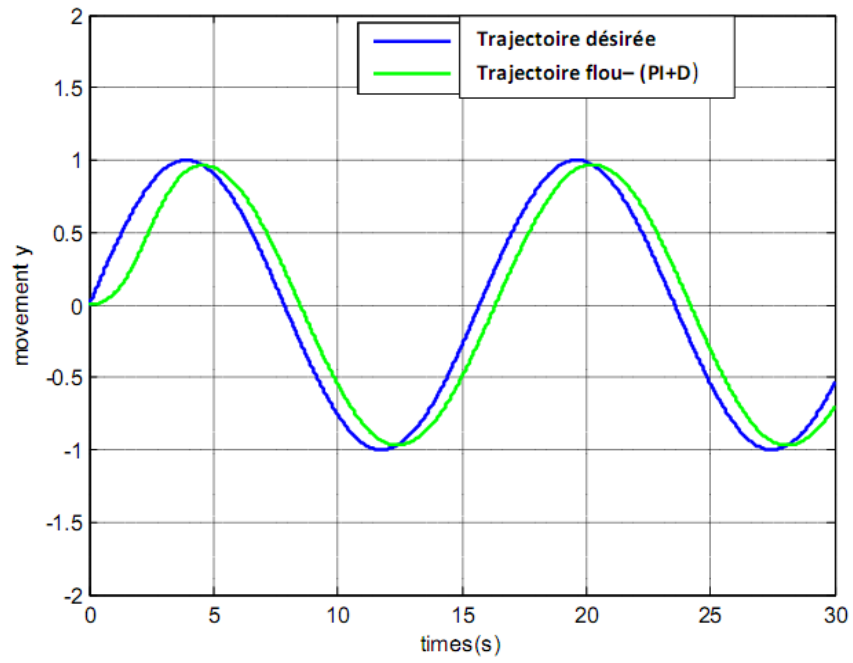


a. trajectoire désirée

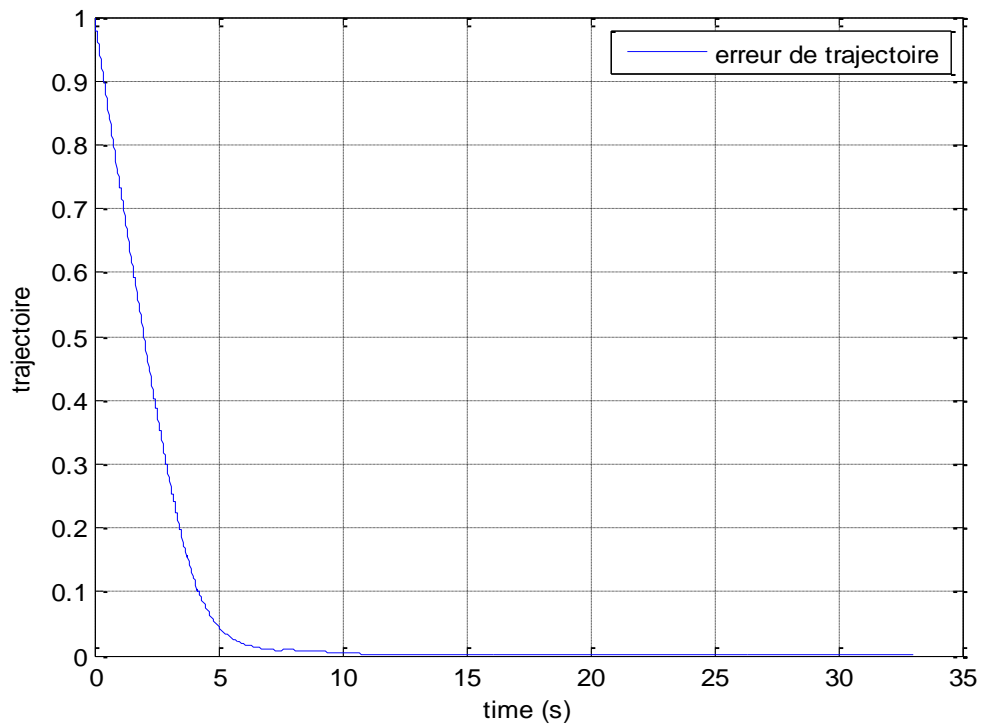


b. trajectoire suivie

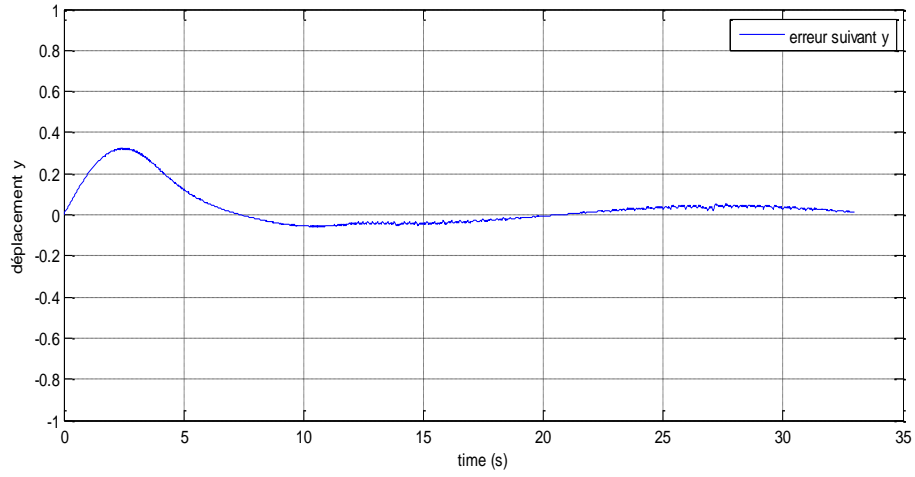
Figure 4.22 Suivi des trajectoires sous forme circulaires



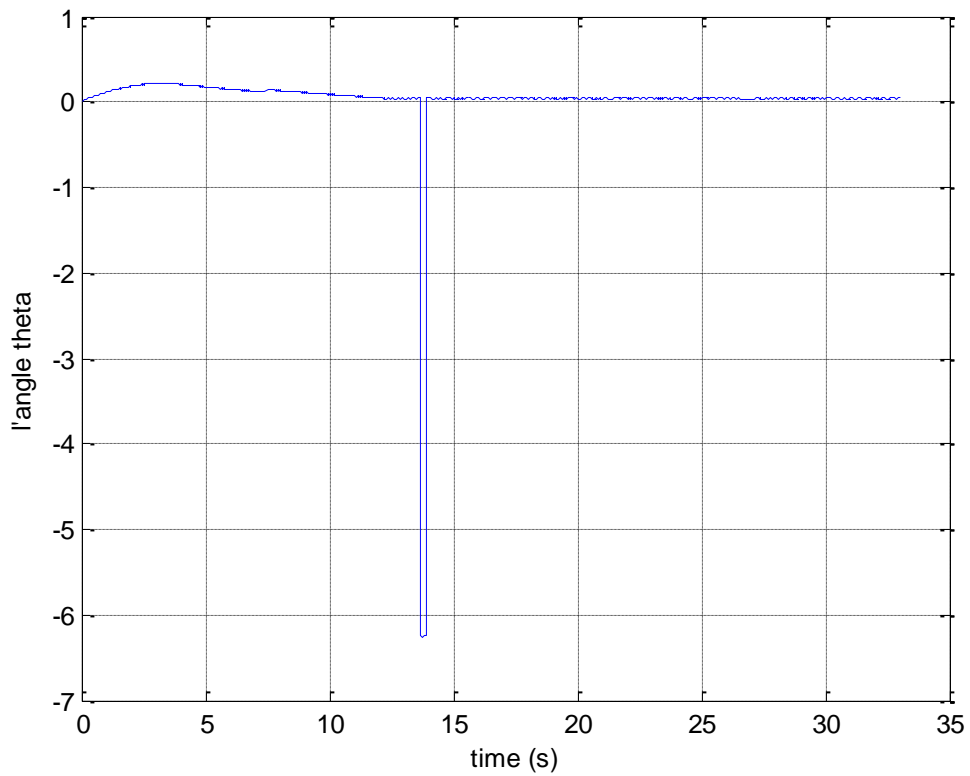
a. déplacement suivant l'axe y



b. Erreur de la trajectoire du robot



c. Erreur de déplacement suivant l'axe y



d. Erreur de l'orientation θ

Figure 4.23 (a) déplacement suivant l'axe y, (b) Erreur de la trajectoire du robot, (c) Erreur de déplacement suivant l'axe y, (d) l'erreur d'orientation θ

Nous avons détaillé le contrôleur numérique de type flou PI+D, car il est différent des PID classiques. Dans la suite de notre travail, on va introduire d'autres techniques intelligentes pour le réglage du contrôleur PI+D.

4.3 Contrôleur neuro-PID

Bien que le contrôle PID-flou soit conçu pour le contrôle de suivi de trajectoire de robot mobile, il est toujours utile d'analyser davantage l'utilisation d'autres méthodes avancées permettant d'améliorer la qualité de réglage des PID. La technique proposée et réalisée dans notre thèse n'est autre que l'exploitation des réseaux de neurones qui sera illustrée dans cette partie.

Le contrôleur PID intelligent basé sur un réseau neuronal de fonction (RBF) est introduit pour le contrôle de robot. Les performances du contrôleur RBFNN-PID proposé sont vérifiées par la simulation à l'aide de l'environnement Matlab.

4.3.1 Structure du contrôleur neuro-PID

La stratégie intelligente de régulation PID basée sur l'algorithme RBFNN est conçue pour le contrôle du robot, comme illustré à la figure 4.24. Elle présente la structure du contrôleur RBFNN-PID comprenant deux parties principales: un contrôleur PID et un réseau de neurones RBF utilisé pour la régulation des paramètres : K_p , K_I et K_d

Le réseau neuronal (RBF) est utilisé pour réguler automatiquement les paramètres DU contrôleur PID, cela réduit les coûts de temps liés à la conception du contrôleur et augmentera l'adaptabilité. Par conséquent, le principe de cette stratégie consiste à combiner les avantages de la méthode de contrôle conventionnelle PID et la technique de contrôle avancée (réseau de neurones à fonction de base radiale).

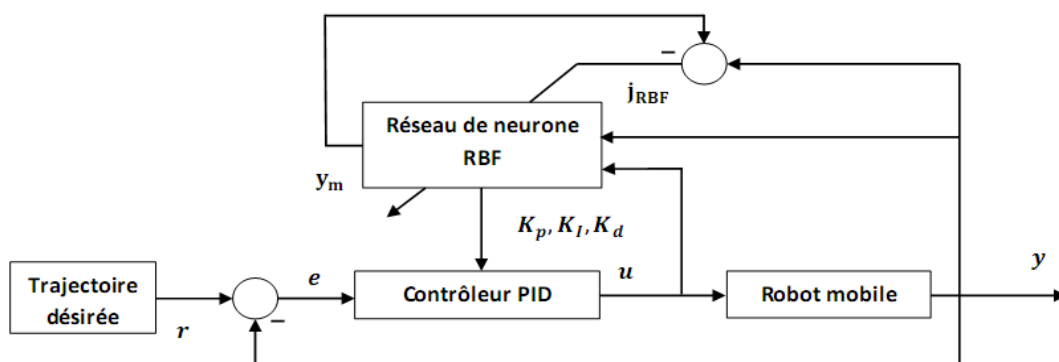


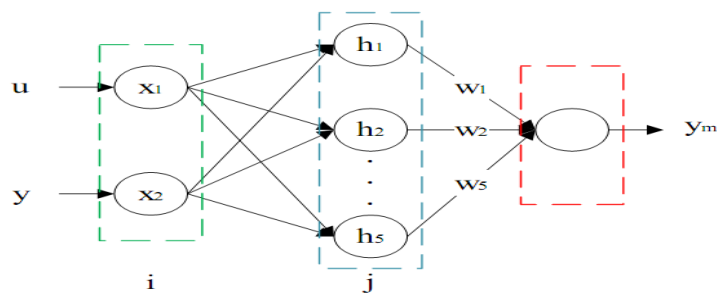
Figure 4.24 Contrôleur PID basé sur un RBFNN

4.3.2 Algorithme RBFNN pour la régulation des K_p , K_i et K_d

Le réseau à fonction de base radiale (RBF) est un système neuronal artificiel à réseau de trois couches qui utilise des fonctions de base radiales en tant que fonctions d'activation. La sortie du réseau est une combinaison linéaire des fonctions de base radiales des entrées du réseau de neurone. Les réseaux à fonctionnement de base radiale ont de nombreux usages, y compris la fonction approximation, prévision de série chronologique, classification et contrôle du système. Ils ont l'avantage de la rapidité d'apprentissage et sont capables d'éviter le problème de minimum local dans le champ de contrôle du système. Par conséquent, le RBFNN est utilisé pour ajuster les paramètres PID dans la stratégie de contrôle.

Le réseau RBF conçu comporte trois couches: une couche d'entrée, une seule couche cachée et une couche de sortie, comme le montre la figure 4.25. Il y a deux entrées et le vecteur d'entrée du réseau RBF est donné comme:

$$X = [X_1, \dots, X_i]^T = [u, y]^T \quad \{I = 1, 2\} \quad (4.41)$$



La couche d'entrée La couche cachée La couche de sortie

Figure 4.25 réseau neuronal RBF

Dans un réseau neuronal RBF, la fonction gaussienne est utilisée comme fonction d'activation. Les neurones cachés implémentent la fonction gaussienne comme fonction de base et les éléments du vecteur base radiale $H = [h_1, h_2, \dots, h_j, \dots, h_5]^T$ peuvent être exprimés par la fonction gaussienne comme suit:

$$\Delta c_{ji} = j_{RDF} \cdot w_j \frac{x_j - c_{ji}}{b_j^2} \quad (4.42)$$

Et la règle d'ajustement de la couche cachée du vecteur d'entrée peut être exprimée comme suit:

$$c_{ji}(k) = c_{ji}(k-1) + \eta \Delta c_{ji} + \alpha (\Delta c_{ji}(k-1) c_{ji}(k-2)) \quad (4.42)$$

Ainsi, la matrice jacobienne peut être obtenue sur la base du calcul précédent:

$$\frac{\partial y(k)}{\partial u(k)} = \sum w_j h_j \frac{c_{ji}(k) - u(k)}{b_i(k)^2} \quad (4.43)$$

On sait que les performances de contrôle PID sont basées sur la valeur des paramètres PID K_p, K_I et K_d . Le contrôleur PID peut avoir d'excellentes performances avec des paramètres appropriés, sinon le contrôleur ne peut pas obtenir le contrôle souhaité. Ainsi, la régulation des paramètres PID est une tâche importante et le réseau RBF conçu est capable d'accorder K_p, K_I et K_d avec précision dans différentes situations en utilisant la matrice jacobienne.

Premièrement, la fonction d'erreur du réseau est définie comme suit:

$$E(k) = \frac{1}{2} (r(k) - (k))^2 \quad (4.44)$$

Ensuite, la règle de d'autoréglage de K_p, K_I et K_d est conçue, basée sur la méthode de la descente d'itération de gradient comme suit:

$$\Delta k_p = -\eta \frac{\partial E}{\partial k_p} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial p} = \eta \cdot e(k) \frac{\partial y}{\partial u} \cdot xc_1(k) \quad (4.45)$$

$$\Delta k_i = -\eta \frac{\partial E}{\partial i} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial i} = \eta \cdot e(k) \frac{\partial y}{\partial u} \cdot xc_2(k) \quad (4.46)$$

$$\Delta k_d = -\eta \frac{\partial E}{\partial d} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial k_d} = \eta \cdot e(k) \frac{\partial y}{\partial u} \cdot xc_3(k) \quad (4.47)$$

Où $\frac{\partial y}{\partial u}$ est la matrice jacobienne donnée par l'équation (4.41). $xc_1(k), xc_2(k)$ et $xc_3(k)$ sont des sorties de contrôleur PID.

Par conséquent, les paramètres PID optimaux peuvent être obtenus puis communiqués au contrôleur PID afin d'obtenir les performances de contrôle souhaitées. L'algorithme de contrôle PID est présentée dans la section suivante.

4.3.3 Algorithme du contrôle PID

L'erreur de système entre la sortie souhaitée du système et la sortie réelle du système, comme illustré à la figure 4.24, est donnée par :

$$e(k) = r(k) - y(k) \quad (4.48)$$

les entrées de contrôle PID peuvent être exprimées comme suit :

$$xc_1(k) = e(k) - e(k - 1) \quad (4.49)$$

$$xc_2(k) = e(k) \quad (4.50)$$

$$xc_3(k) = e(k) - 2e(k - 1) + e(k - 2) \quad (4.51)$$

L'algorithme de contrôle PID est donné par :

$$u(k) = u(k - 1) + k_p(e(k) - e(k - 1)) + k_i e(k) + k_d(e(k) - 2e(k - 1) + e(k - 2)) \quad (4.52)$$

4.3.4 Résumé du contrôle PID basé sur les RBFNN

Le processus de contrôle de RBFNN-PID présenté à la figure 4.24 peut être résumé comme suivant l'organigramme :

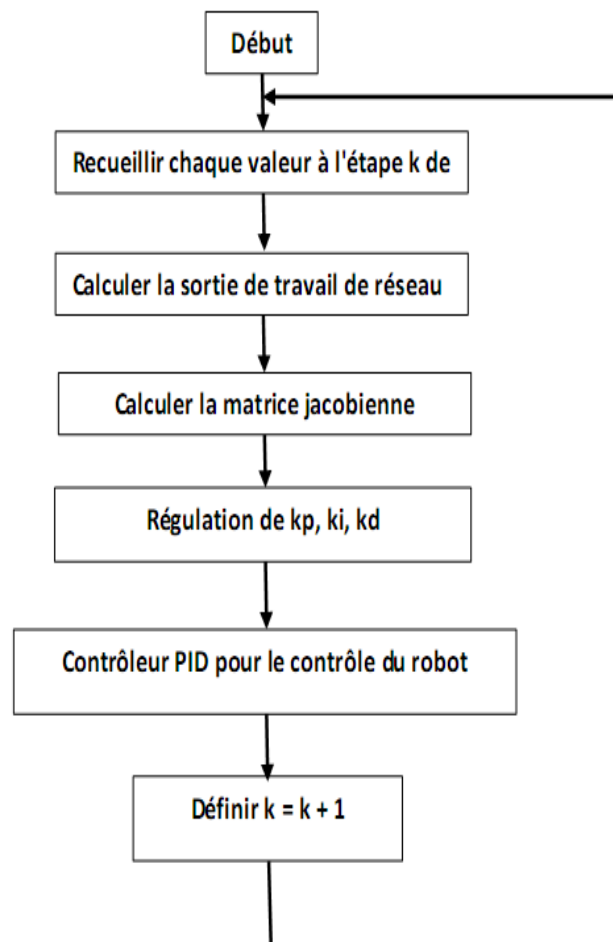
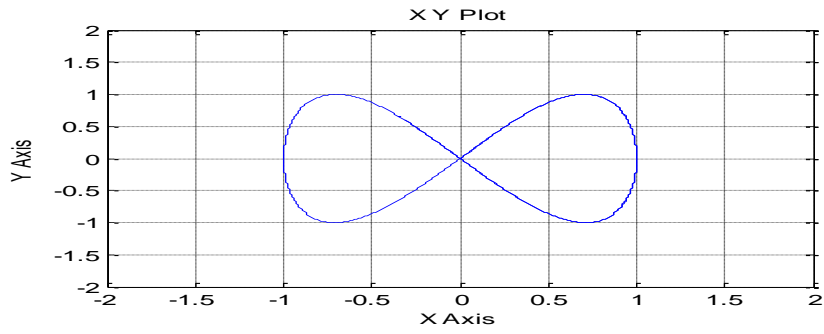
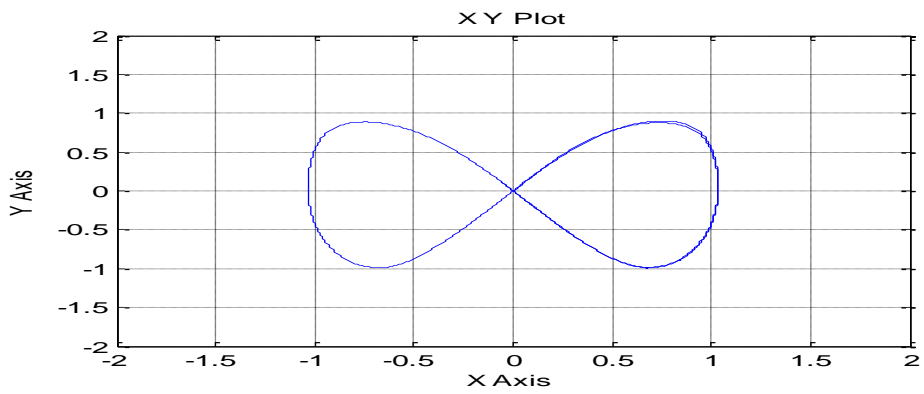


Figure 4.26 Organigramme du contrôleur RBFNN-PID

Les performances de contrôle de la stratégie de contrôle RBFNN-PID proposée est validée par une simulation sous l'environnement Matlab.

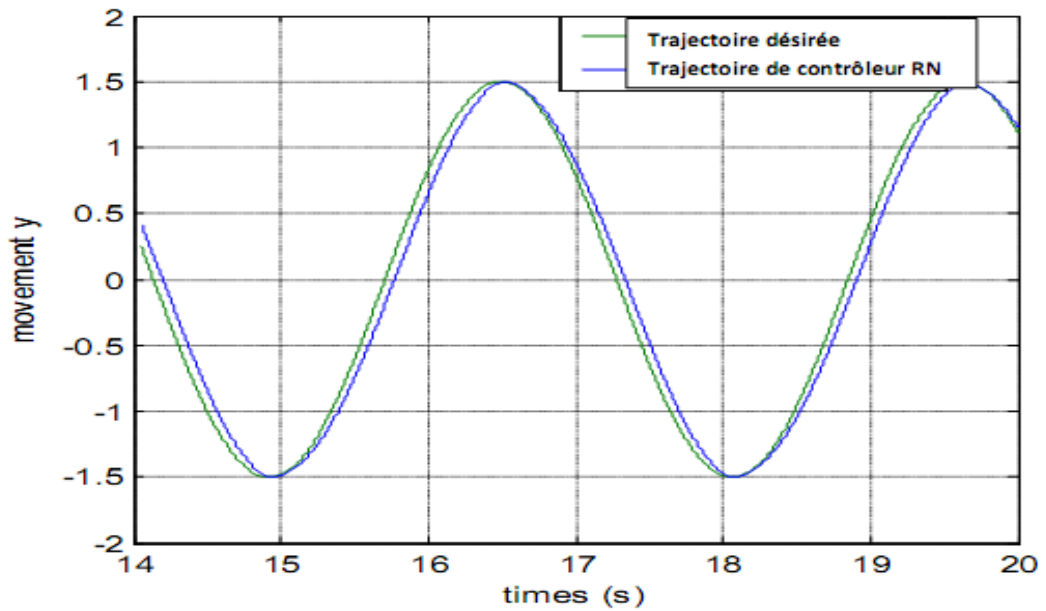


a. trajectoire désirée

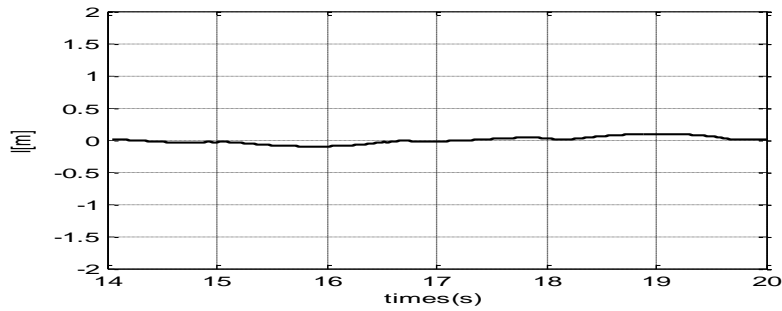


b. trajectoire suivie

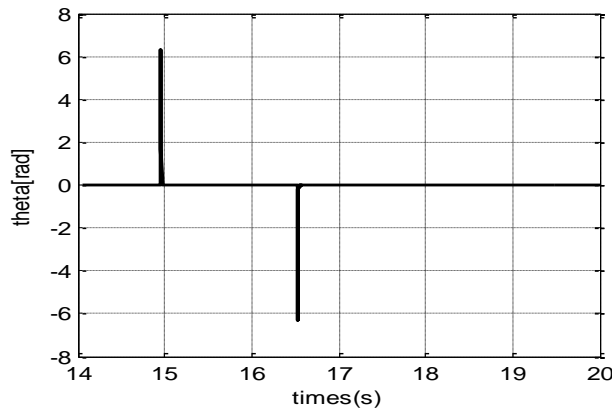
Figure 4.27. Suivi des trajectoires sous forme huit



a. déplacement suivant l'axe y



b. Erreur de déplacement suivant l'axe y



c) Erreur de l'orientation thêta

Figure 4.28 (a) déplacement suivant l'axe y, (b) l'erreur de déplacement

(c) l'erreur d'orientation thêta

4.4 Contrôleur neuro-flou-PI+D

Nous avons proposé aussi une méthode de contrôle hybride associant logique floue aux réseaux neuronaux pour le suivi de trajectoires d'un robot mobile. Ces techniques ont prouvé leur efficacité dans une grande variété de systèmes de contrôle [93]. Avec l'utilisation de contrôleur PI+D numérique que nous avons détaillé dans ce chapitre, nous appliquons une autre technique de réglage permettant d'améliorer beaucoup plus les performances de commande.

La logique floue et les réseaux de neurones sont des techniques complémentaires dans la conception des systèmes intelligents. Chaque méthode a ses avantages et ses inconvénients. Les réseaux de neurones sont utilisés pour reconnaître des modèles. Les techniques de logique floue traitent souvent des problèmes tels que le raisonnement à un niveau supérieur à celui des

réseaux de neurones. Cependant, il n'existe pas de consensus absolu, mais en général, le terme flou désigne un type de système caractérisé pour une structure de contrôleur flou, où les règles floues sont ajustées à l'aide de tunnels de réseaux de neurones [94].

La technique hybride proposée est appliquée à notre conception, où le contrôle PI + D est ajusté efficacement pour produire de meilleurs résultats pour le suivi de trajectoire.

4.4.1 Méthodologie

La figure 4.29 montre la représentation sous forme de diagramme d'éléments de contrôle. Ces éléments de contrôle sont conçus pour accorder le contrôleur PI + D. Il est composé de deux blocs, un contrôleur flou et un contrôleur neuronal, où le bloc flou fournit un vecteur d'entrée à un réseau neuronal multicouche. Dans ce travail, nous avons utilisé un schéma fonctionnel à deux éléments, le premier pour réguler le contrôleur PI et le deuxième pour le contrôleur dérivée D. Le bloc flou comporte deux entrées et une sortie, $k_I e(nT)$ et $k_P v(nT)$ sont les entrées du contrôleur PI et $\Delta u_{PI}(nT)$ représente sa sortie. $y_D(nT)$ et $\Delta y(nT)$ constituent les entrées du contrôleur D et $\Delta y(nT)$ sa sortie. Le bloc doit générer les sorties et la décision de commande souhaitée.

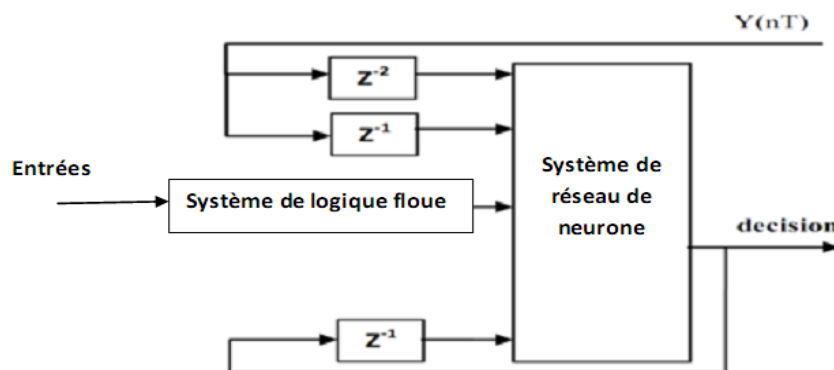


Figure 4.29 Schéma fonctionnel de l'élément de contrôle

4.4.2 Le système de réseau de neurone

Le but de ce système est d'entraîner un réseau neuronal à agir comme l'inverse du processus de robot mobile. Le modèle inverse est ensuite utilisé pour le contrôle inverse direct. L'ensemble de données considérées sont celui présenté sur la figure 4.30. Le modèle de robot a été déterminé à l'avance. Ensuite, le réseau de neurones a été formé pour agir comme l'inverse de ce modèle. La représentation finale de l'inverse sur le réseau est donnée par

$$\hat{u}(k) = \hat{g}^{-1}[r(k+1), y(k), \dots, y(k-n_a+1), u(k), \dots, u(k-n_b)] \quad (4.53)$$

Le réseau est ensuite formé pour minimiser le critère

$$j(\theta, Z^n) = \frac{1}{2N} \sum_{k=1}^N [u(k) - \hat{u}(k|\theta)]^2 \quad (4.54)$$

Les sorties du système sont remplacées par la composante de qui correspondante de l'entrée de commande donnée par

$$u_{ff}(k) = \hat{g}[r(k-1) \dots r(k-n_a+1), u_{ff}(k-1), \dots, u_{ff}(k-n_b)] \quad (4.55)$$

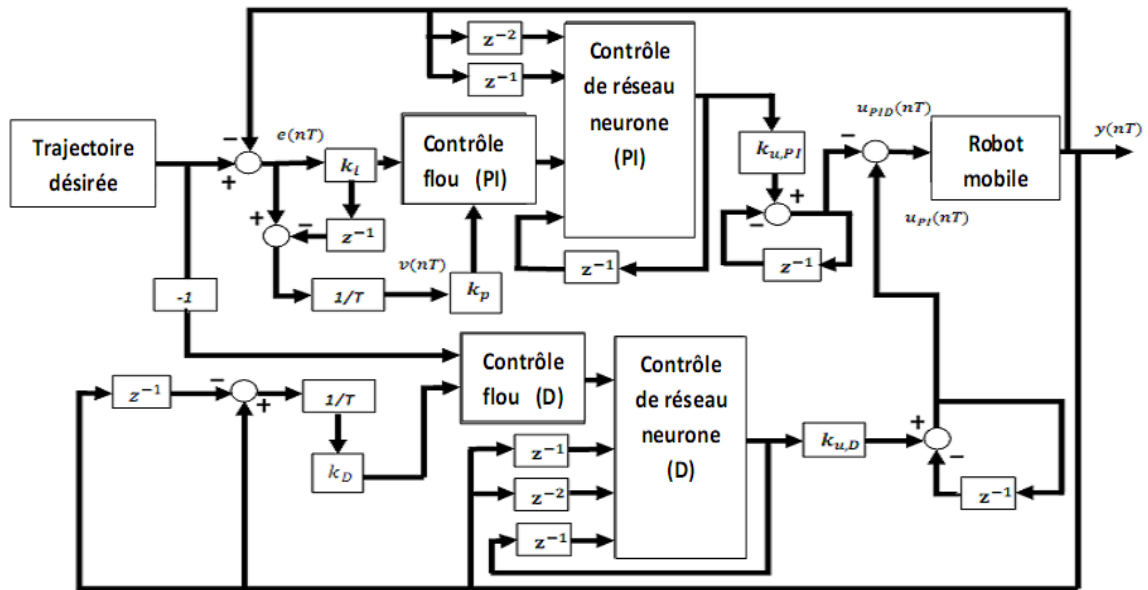


Figure 4.30 Schéma de principe du contrôleur neuro-flou-PI + D

4.4.3 Structure du contrôleur flu-PI+D

Il existe différents types de contrôle par la logique floue. L'expert doit développer un savoir-faire technologique en termes d'analyse et d'expérience pour raisonner logiquement sur des règles floues [95]. La tâche associée à cette partie consiste à afficher les performances de conception d'un contrôleur PI + D ajustées automatiquement par deux contrôleurs flous PI et D. Les entrées du contrôleur PI sont $K_I e(nT)$ et $K_P v(nT)$, la sortie est $\Delta u_{PI}(nT)$. Les entrées du contrôleur D sont $y_D(nT)$ et $\Delta y(nT)$. La sortie est $\Delta u_D(nT)$.

Les entrées des deux contrôleurs présentent l'erreur (e) et le taux d'erreur (ec). Le principe de ce système de contrôle est illustré aux figures 4.31 et figure 4.32. La loi de commande est présentée par :

$$u_{PID}(nT) = u_{PI}(nT - T) + K_{u,PI} \Delta u_{PI}(nT) + u_D(nT - T) - K_{u,D} \Delta u_{DPI}(nT) \quad (4.56)$$

Dans la mise en œuvre du robot sous contrôle, la comparaison de la boucle fermée détecte une erreur (e) et un changement d'erreur (ec). Pour améliorer les performances du système de contrôle, les paramètres de PI + D sont modifiés en temps réel en fonction du contrôle flou. Ce contrôle permet au robot de corriger le mouvement le long de la trajectoire souhaitée [96]. Pour remédier aux inconvénients du contrôleur PI + D, nous proposons une combinaison entre la logique floue et PI + D. La conception des paramètres fuzzy PI + D à autoréglage est illustrée aux figure 4.31 et la figure 4.32. La sortie à ajustement automatique permet de contrôler la vitesse v et la vitesse angulaire γ du robot.

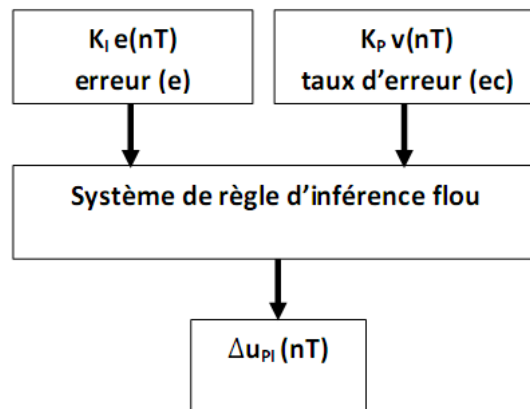


Figure 4.31 les entrées floues du contrôleur PI

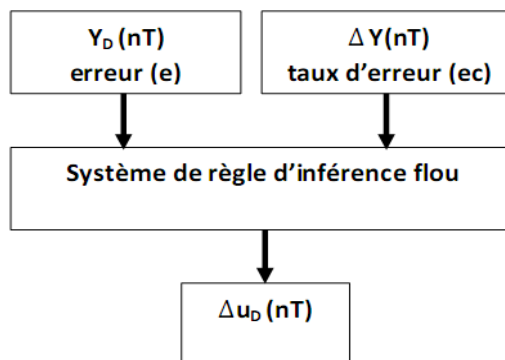


Figure 4.32 sorties floues du contrôleur PI

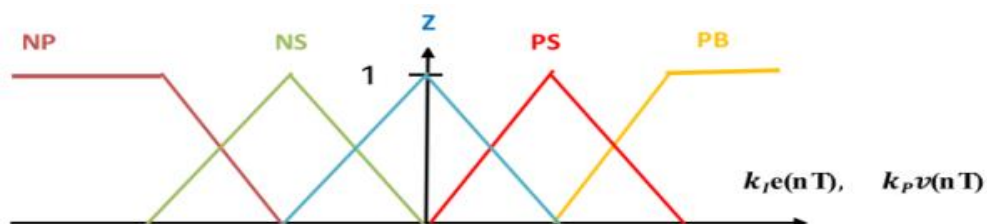


Figure 4.33 fonctions d'appartenance des entrées du contrôleur flou PI

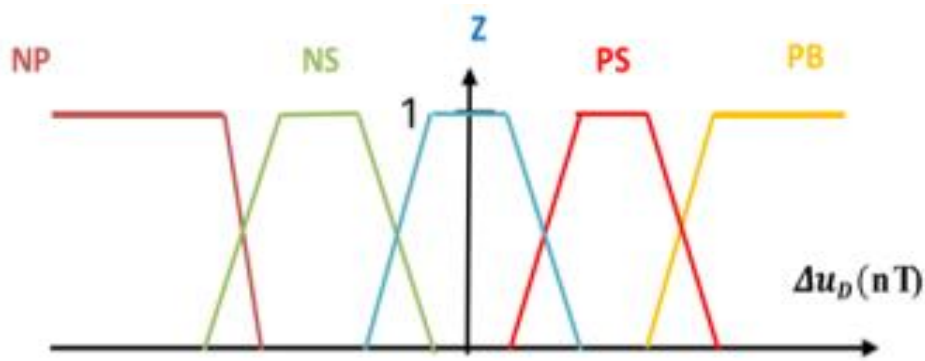


Figure 4.34 fonctions d'appartenance de la sortie du contrôleur flou PI

Chaque entrée sera représentée par trois variables linguistiques. Les règles d'inférence permettant de déterminer le comportement du contrôleur flou. Les fonctions d'appartenance de ces entrées sous forme de règles floues qui sont illustrées aux figure 4.35 et la figure 4.36 ci-dessous. Les niveaux de variables linguistiques sont attribués comme suit: grand négatif (NB), petit négatif (NS), zéro (Z), petit positif (PS) et grand positif (PB).

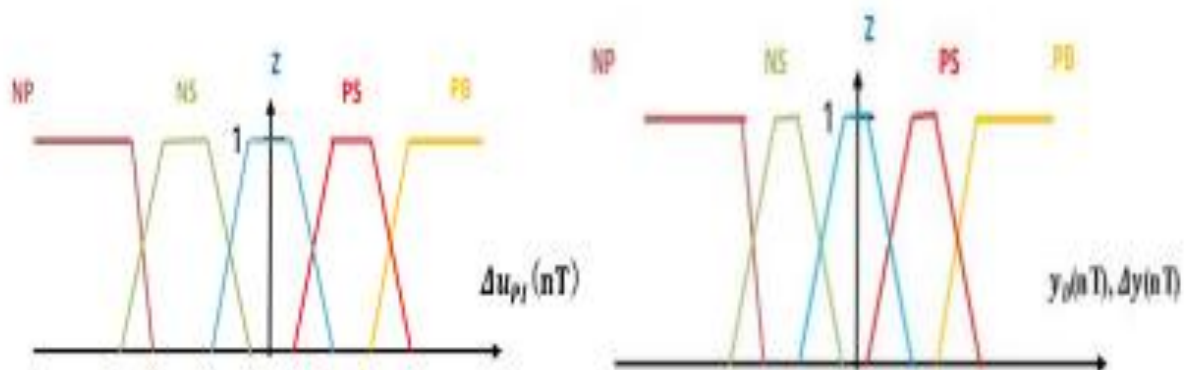


Figure 4.35 fonctions d'appartenance des entrées et de la sortie du contrôleur flou D

Tableau 4.4. Table des règles floues du contrôleur D

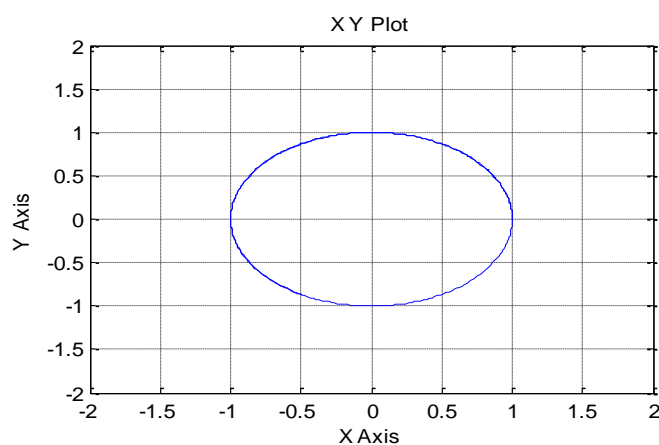
		error rate (ec) $\Delta y(nT)$				
		NB	NS	Z	PS	PB
error (e) $y_D(nT)$	NB	NS	NS	Z	PS	PS
	NS	NB	NS	Z	PS	PB
	Z	NB	NS	Z	PS	PS
	PS	NB	NS	Z	PS	PB
	PB	NB	PS	NB	NS	PS

Tableau 4.5. Table des règles floues du contrôleur P

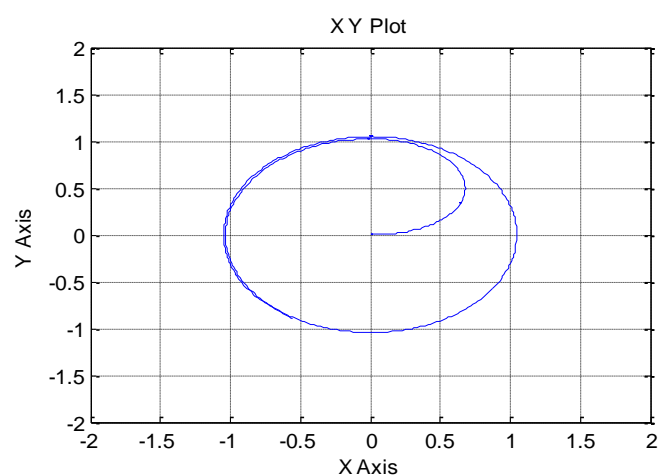
		error rate (ec) $k_f e(nT)$				
		NB	NS	Z	PS	PB
Error(e) $k_p e(nT)$	NB	NB	NS	Z	Z	Z
	NS	NB	NS	PS	PB	PB
	Z	NB	NS	Z	PS	PB
	PS	NB	NB	NS	Z	PS
	PB	NB	NB	NB	NS	Z

La défuzzification consiste à transformer la décision de sortie linguistique en décision numérique connue pour ajuster les paramètres PI + D. Ainsi. Pour notre étude de cas, nous avons utilisé la méthode du centre de gravité [97]. On a aussi exploité Simulink pour implémenter le contrôleur neuro-flou PI+D pour commander le robot mobile figure 4.30.

4.4.4 Résultats de simulation



(a)



(b)

Figure 4.36 Trajectoires de suivi, a) la trajectoire souhaitée, b) trajectoire neuro -floue- PI + d obtenue

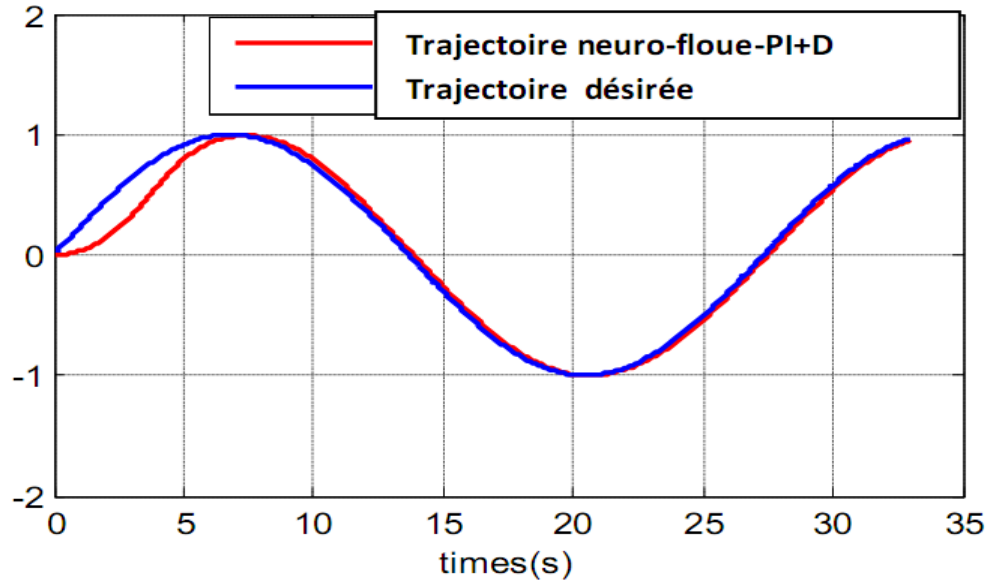


Figure 4.37 déplacement Y et trajectoire souhaitée du contrôleur neuro-floue (PI + D)

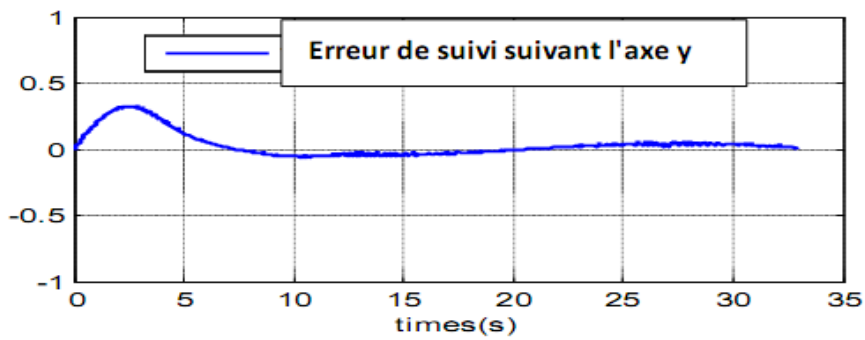


Figure 4.37 erreur de suivi de trajectoire en position Y

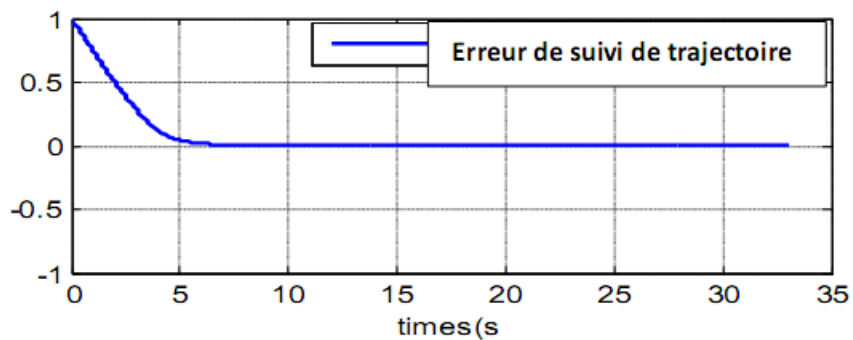


Figure 4.38 erreur de suivi de trajectoire

Les simulations montrant les performances du neuro-floue (PI +D), la bonne stabilité du robot

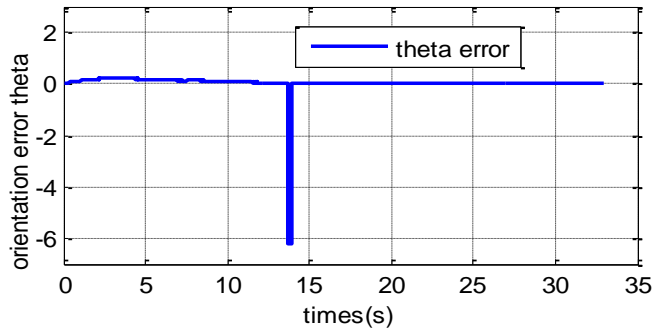


Figure 4.39 L'erreur d'orientation du contrôleur neuro-floue PI + D

4.5 Contrôleur neuro- PI+D

Dans cette section, nous présentons d'autre technique qui est le contrôleur du réseau de neurones. Les ANNs sont organisées autour d'un ensemble de cellules, ou neurones, interconnectés par des connexions qui leur permettent de disposer d'un canal pour envoyer et recevoir des signaux d'autres cellules du réseau. Le modèle se comporte comme le calcul formel par un modèle de neurone d'une somme pondérée de ses entrées, suivie d'une non-linéarité appelée fonction d'activation, responsable du comportement du neurone [20].

Les entrées représentées par les nœuds sources fournissent un poids d'accès physique pour l'application des signaux d'entrée. Les neurones de la couche cachée se comportent comme des détecteurs de caractéristiques, tandis que les neurones de la couche de sortie fournissent à l'utilisateur les conclusions atteintes par le réseau, figure4.40.

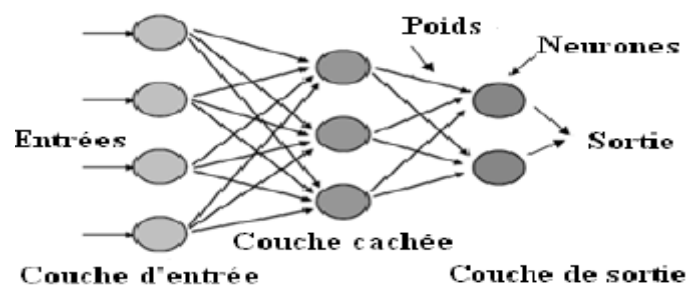
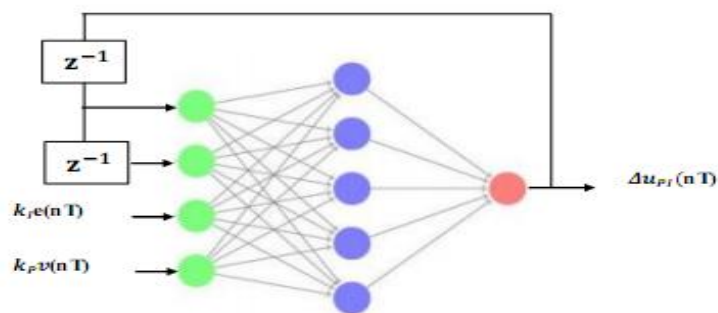


Figure 4.40 La structure d'un RN typique

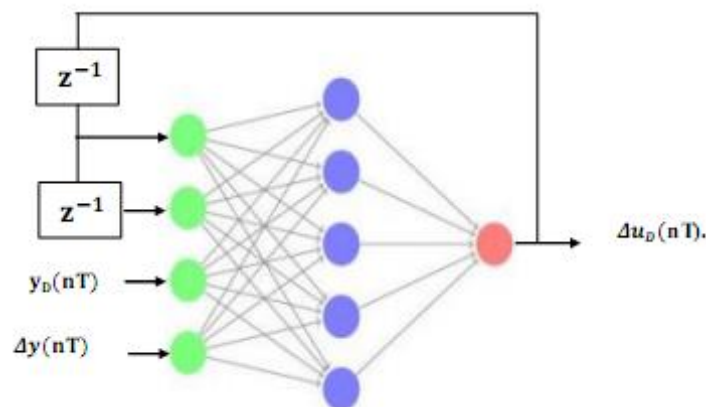
Dans cette partie, le contrôleur RN est utilisé pour régler les paramètres du contrôleur PI + D. Figure4.41 montrent la structure de ce contrôleur.

L'idée de base du contrôle direct inverse est de contrôler un système non linéaire, notre cas, qui est un robot mobile. Le contrôleur de réseau de neurones (CRN) agira comme

modèle inverse du système pour annuler sa dynamique. Il affiche de bonnes performances et est donc utilisé pour contrôler des systèmes dynamiques. L'architecture généralisée est illustrée à la figure 4.42. Le modèle inverse du système sous contrôle est représenté par des réseaux de neurones et doit être formé jusqu'à l'obtention d'une erreur minimale prédéterminée avant sa mise en œuvre en tant que contrôleur. Un perceptron à trois couches: une couche d'entrée connectée aux variables d'entrée $K_{pe}(nT)$ et $K_{pv}(nT)$ pour le contrôleur PI, $y_D(nT)$ et $\Delta y(nT)$ pour le contrôleur D est une couche cachée, ses fonctions d'activation de le sigmoïde est une couche de sortie connectée aux variables de sortie $\Delta u_{PI}(nT)$ et $\Delta u_D(nT)$. Une fois la structure fixée, il faut passer par le processus d'apprentissage, en fonction des informations d'entrée grâce auxquelles les poids seront ajustés en même temps. Les sorties $\Delta u_{PI}(nT)$ et $\Delta u_D(nT)$ sont ajustées à satisfaire un critère d'optimisation. Le mécanisme d'apprentissage utilisé pour ce modèle inverse direct supervisé est la méthode d'apprentissage par propagation arrière.



a) Réseau de neurones de contrôle PI



b) réseau de neurones de contrôle D

Figure 4.41 Réseau de neurones de contrôle PI et du réseau de neurones de contrôle D

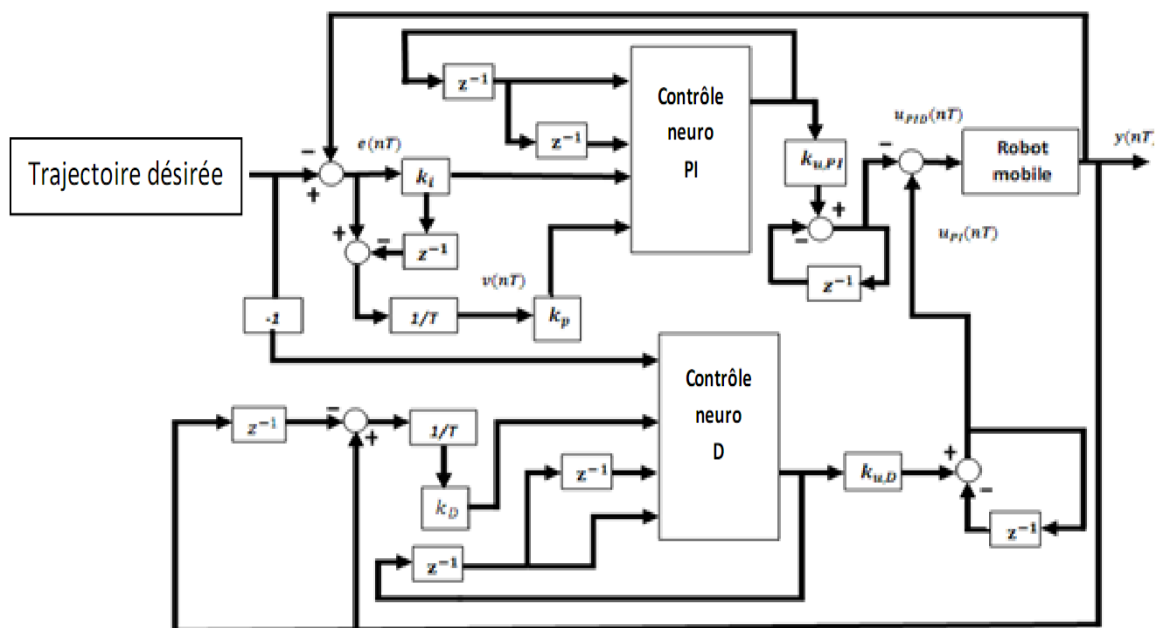


Figure 4.42 contrôleur neuro-PI+D

L'entrée et la sortie des nœuds des couches masquées et de sortie peuvent être représentées sous la forme [99] – [100].

$$\text{entrées} \left\{ \begin{array}{l} in_i^{(2)}(k) = \sum_{j=1}^3 w_{ij}^{(2)} out_j^{(1)}(k) \\ i = 1,2,3,4,5 \\ out_i^{(2)}(k) = f[in_i^{(2)}(k)] \end{array} \right. \quad (4.57)$$

$$\text{sorties} \left\{ \begin{array}{l} in_i^{(3)}(k) = \sum_{j=1}^5 w_{ij}^{(3)} out_j^{(2)}(k) \\ i = 1,2,3 \\ out_i^{(3)}(k) = g[in_i^{(3)}(k)] \end{array} \right. \quad (4.58)$$

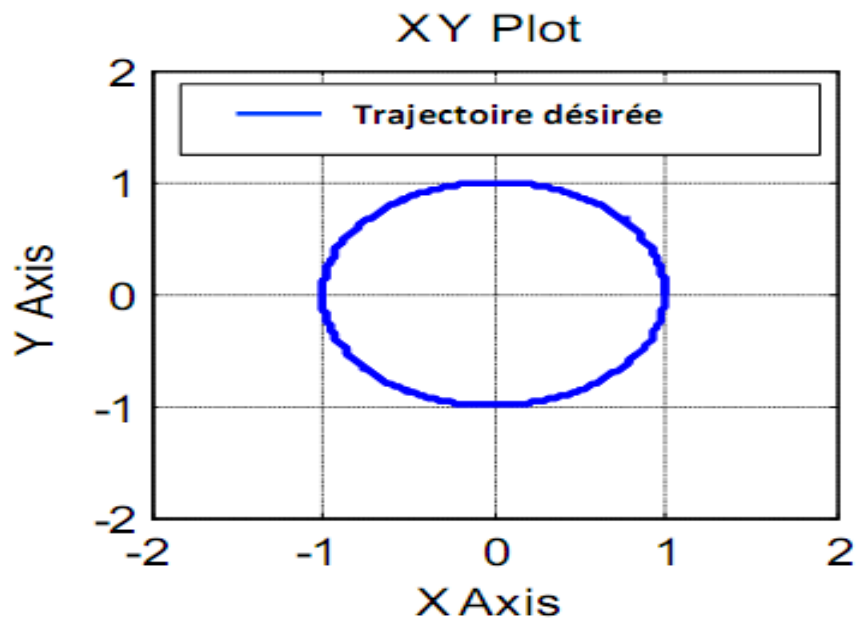
$$\text{sorties du contrôleur NN} \left\{ \begin{array}{l} out_1^{(3)}(k) = \Delta u_D(nT) \\ out_2^{(3)}(k) = \Delta u_{PI}(nT) \end{array} \right. \quad (4.59)$$

$w_{ij}^{(2)}$: est le poids de la connexion entre la couche d'entrée et la couche cachée

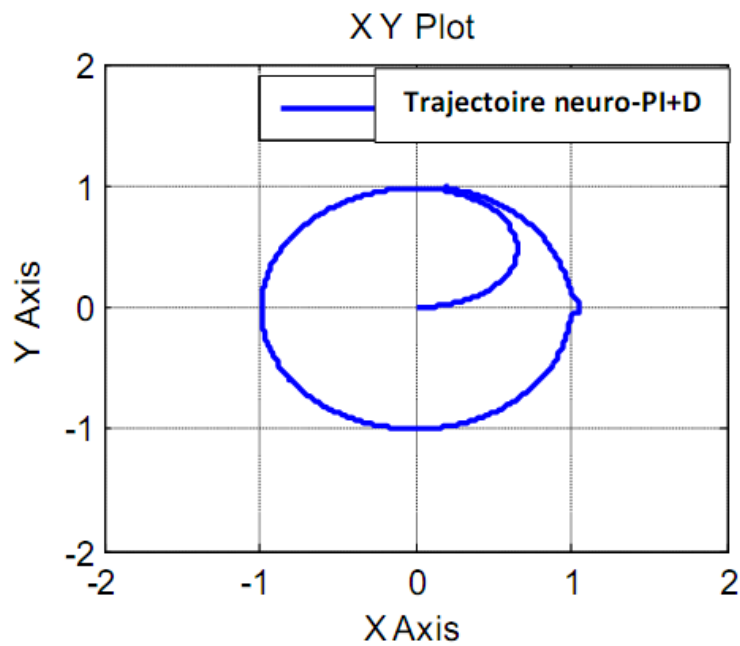
$w_{ii}^{(3)}$: est le poids de connexion entre la couche cachée et la couche de sortie

g, f : sont des fonctions d'activation.

Après avoir détaillé la conception du contrôleur basée sur les réseaux de neurones, nous présentons la mise en œuvre de la simulation avec Simulink.



(a)



(b)

Figure 4.43 Trajectoires de suivi de forme cercle, a) la trajectoire souhaitée, b) trajectoire neuro-PI+D

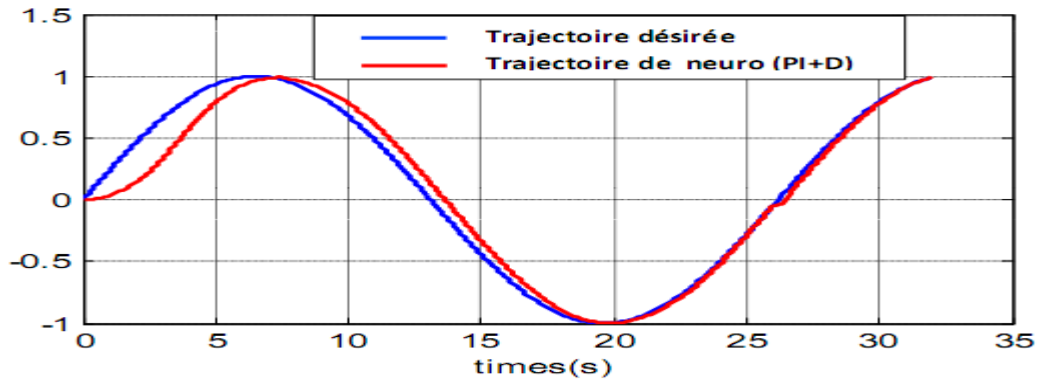


Figure 4.44 déplacement Y et trajectoire souhaitée du contrôleur neuro (PI + d)

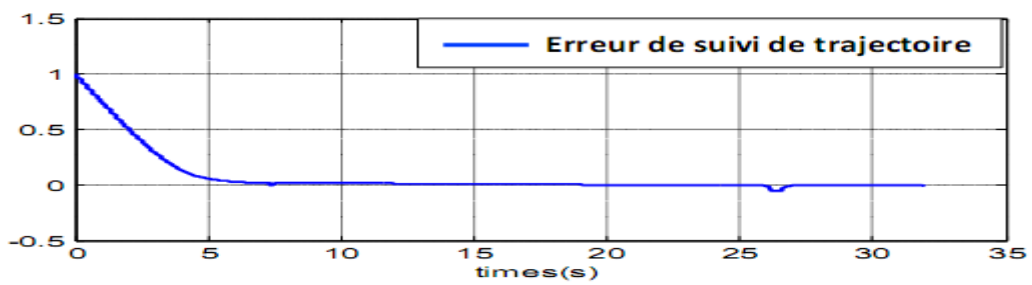


Figure 4.45 erreur de suivi de trajectoire du contrôleur neuro- (PI + D)

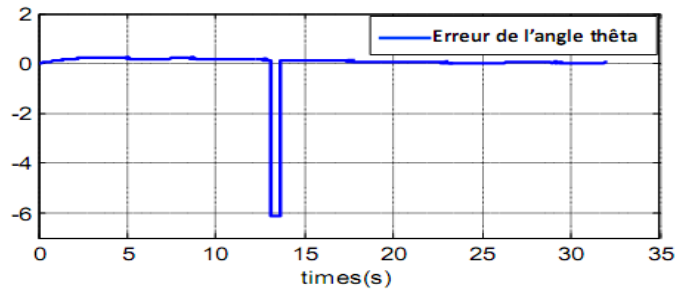


Figure 4.46 L'erreur d'orientation du contrôleur neuro -PI + D

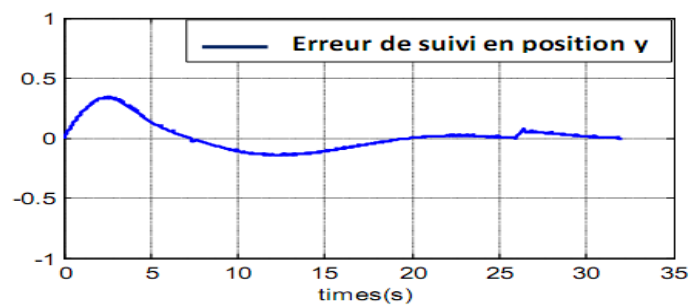


Figure 4.47 erreur de suivi de trajectoire en position Y du contrôleur neuro-PI+D

Conclusion 4.6

Le contrôleur flou-PID a été conçu pour atteindre l'objectif de minimiser l'erreur entre la trajectoire souhaitée et la trajectoire suivie. La structure de ce contrôleur contient deux parties principales: une commande PID classique typique pour le contrôle du robot mobile et un contrôleur de logique floue destiné à la régulation des paramètres du contrôleur PID. Le contrôle de la logique floue a été utilisé donc pour un réglage optimal des paramètres PID afin de garantir une adaptabilité à différentes situations. Une meilleure sélection de ces paramètres de contrôle PID peut assurer la sortie système désirée.

Conclusion générale

Le contrôle de systèmes complexes, point de départ de cette thèse, est un domaine riche, trouvant ses racines dans de nombreux champs d'application et dont les notions de base sont présentées dans le premier chapitre. Notre étude nous a rapidement menés vers les contrôleurs PID, qui sont les plus répandus dans l'industrie. Des nouveaux contrôleurs basés sur les PID conventionnels et les PID numériques pour suivi de trajectoire d'un robot mobile non holonome ont été conçus et mis en œuvre.

Dans cette thèse, une stratégie de conception de contrôleurs visant à améliorer le suivi de trajectoire d'un robot mobile non holonome est donc introduite en combinant le contrôleur conventionnel et la technique de contrôle intelligente pour développer une nouvelle stratégie de contrôle. Parmi les contrôleurs abordés, nous avons développé un contrôleur flou PID, un contrôleur RBFNN-PID, un Contrôleur numérique FLOU (PI+D), un contrôleur neuro-flou (PI+D) et un contrôleur neuro (PI+D). Les contrôleurs proposés sont tous conçus sur la base des contrôleurs PID conventionnel, PIDs numériques et les contrôleurs intelligents. La structure principale des contrôleurs conçus sont présentés dans le chapitre quatre. Le contrôleur PID est utilisé pour contrôler le robot mobile de type voiture qui reste toujours pratique et peut être utilisé pour une variété de contrôles même si de nombreux chercheurs affirment qu'elle a beaucoup d'inconvénients. Le contrôle intelligent est utilisé pour régler les paramètres PID automatiquement pour la meilleure performance de contrôle. Donc le but de ce concept est d'exploiter les avantages de chaque méthode de contrôle et d'éviter les inconvénients.

Le contrôleur PID est très pratique et largement utilisé pour le contrôle des robots. La performance du contrôleur PID est principalement basée sur la sélection des trois paramètres

du contrôleur PID, Le réglage des paramètres nécessite beaucoup de temps et est basé sur l'expérience des concepteurs.

Par conséquent, le contrôleur intelligent est conçu pour assurer une sortie de contrôle optimisée. Ainsi, l'adaptabilité et la stabilité du contrôleur PID peuvent être améliorées en appliquant les techniques FLC et RN conçus de manière appropriée à la stratégie de contrôle. De cette façon, les performances de contrôle des PID conventionnels peuvent être considérablement améliorées en travaillant en association avec les technologies de contrôle intelligent.

Les avantages des contrôleurs de type (PI+D) intelligents par rapport au contrôle des méthodes conventionnel ayant une meilleure adaptabilité et un état stable dans toutes les situations. Les contrôleurs intelligents tels que le contrôle basé sur la logique floue et sur le réseau de neurones a permis au robot de suivre la trajectoire désirée avec une erreur minimale, une légère oscillation transitoire et un temps court pour corriger l'erreur inter-distance par rapport à la référence. Néanmoins le contrôleur basé sur le réseau de neurones PI + D présente une oscillation transitoire considérée importante.

La combinaison du contrôleur PID conventionnel et des contrôleurs intelligents proposés a amélioré les performances de contrôle. Les résultats présentés dans le chapitre 4 prouvent que les performances des contrôleurs (PI+D) numériques sont acceptables et peuvent être utilisés pour une conception d'un contrôleur intelligent.

Comme perspective, il est nécessaire d'améliorer le comportement des robots et de simuler cette stratégie de contrôle pour étudier la navigation du robot mobile dans un environnement inconnu comportant par exemple divers types d'obstacles.

Références bibliographiques

Chapitre 1

- [1] Minorsky, Nicolas (1922). « Directional stability of automatically steered bodies ». In : Journal of ASNE 34.2, p. 280–309 (cf. p. 3).
- [2] Sontag, Eduardo D. (1998). *Mathematical control theory : deterministic finite dimensional systems*. Springer (cf. p. 4).
- [3] Astrom, Karl Johan et Hagglund, Tore (1995). *PID Controllers : Theory, Design, and Tuning* Second. Research Triangle Park, NC : Instrument Society of America (cf. p. 12).
- [4] Ziegler, John G. et Nichols, Nathaniel B. (1942). « Optimum settings for automatic control-ers ». In : trans. ASME 64.11 (cf. p. 13).
- [5] Valério, Duarte et Sá da Costa, José (2006). « Tuning of fractional PID controlle Ziegler–Nichols-type rules ». In : *Signal Processing* 86.10, p. 2771–2784 (cf. p. 13).
- [6] Visioli, Antonio (2001). « Tuning of PID Controllers With Fuzzy Logic ». In : *IEE Proceedings -Control Theory and Applications* 148.1 (jan. 2001), p. 1–8 (cf. p. 13)
- [7] Chang, Wei-Der, Hwang, Rey-Chue et Hsieh, Jer-Guang (2002). « A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach ». In : *Journal of Process Control* 12.2, p. 233–242 (cf. p. 14).
- [8] Carvajal, James, Chen, Guanrong et Ogmen, Haluk (2000). « Fuzzy PID controller : Design, performance evaluation, and stability analysis ». In : *Information Sciences* 123.3–4, p. 249–270 (cf. p. 14).
- [9] Lee, Yongho, Park, Sunwon et Lee, Moonyong (1998). « PID Controller Tuning To Obtain Desired Closed Loop Responses for Cascade Control Systems ». In : *Industrial & Engineering Chemistry Research* 37.5, p. 1859–1865 (cf. p. 14).
- [10] Shu, Huailin et Pi, Youguo (2000). « PID neural networks for time-delay systems ». In : *Computers and Chemical Engineering* 24.2–7, p. 859–862 (cf. p. 14).
- [11] Mizumoto, Ikuro, Ikeda, Daisuke, Hirahata, Tadashi et Iwai, Zenta (2010). « Design of discrete time adaptive PID control systems with parallel feedforward compensator ». In : *Control Engineering Practice* 18.2, p. 168–176 (cf. p. 14).
- [12] Whitaker, H.Philip, Yamron, Joseph et Kezer, Allen (1958). *Design of Model Reference Adaptive Control Systems for Aircraft*. Issue 164 of Report Massachusetts Institute of Technology Instrumentation Laboratory R. M.I.T. Instrumentation Laboratory (cf. p. 15).

- [13] Mareels, Iven M.Y., Anderson, Brian D.O., Bitmead, Robert R., Bodson, Marc et Sastry Shankar S. (1986). « Revisiting the MIT rule for adaptive control ». In : Proceedings of the 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing, p. 161–166 (cf. p. 16).
- [14] Braziunas, Vytautas (1992). Design of the MRAC Scheme with PID Controllers for Robotic Manipulators. Northern Illinois University (cf. p. 16).
- [15] Sastry, Shankar et Bodson, Marc (1994). Adaptive control : stability, convergence and robustness. Courier Dover Publications (cf. p. 16).
- [16] Soderstrom, Torsten et Stoica, Petre (1988). System identification. Upper Saddle River, NJ, USA : Prentice-Hall, Inc. (cf. p. 16).
- [17] Chen, Sheng, Billings, Stephen A. et Luo, Wan (1989). « Orthogonal least squares methods and their application to non-linear system identification ». In : International Journal of control 50.5, p. 1873–1896 (cf. p. 16).
- [18] Van Der Merwe, Rudolph et Wan, Eric A. (2001). « The square-root unscented Kalman filter for state and parameter-estimation ». In : Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on. T. 6. IEEE, p. 3461–3464 (cf. p. 16).
- [19] Pintelon, Rik et Schoukens, Johan (2004). System identification : a frequency domain approach. Wiley-IEEE Press (cf. p. 16).
- [20] Xue, Xiaocen, Dong, Zhanbo, Xiang, Wenguo et Lu, Jianhong (2012). « A new neuro-fuzzy approach for nonlinear system identification based on differential evolution ». In : 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). IEEE, p. 09–413 (cf. p. 16).
- [21] Nikolaou, Michael (2001). « Model Predictive Controllers : A Critical Synthesis of Theory and Industrial Needs ». In : Advances in Chemical Engineering 26, p. 131–204 (cf. p. 17).
- [22] Isidori, Alberto (1999). Nonlinear control systems. T. 2. Springer (cf. p. 18).
- [23] Deng, Jiamei, Becerra, Victor M. et Stobart, Richard (2009). « Input constraints handling in an MPC/feedback linearization scheme ». In : International Journal of Applied Mathematics and Computer Science 19.2, p. 219–232 (cf. p. 18).
- [24] Müller, Matthias A., Reble, Marcus et Allgöwer, Frank (2011). « A general distributed MPC framework for cooperative control ». In : Proceedings of the 18th IFAC World Congress, p. 7987–7992 (cf. p. 18, 87).

- [25] Feldbaum, Alexander Aronovich (1961). « Dual Control Theory, I-IV ». In : Automation Remote Control 21-22 (cf. p. 18).
- [26] Arimoto, Suguru, Kawamura, Sadao et Miyazaki, Fumio (1984). « Bettering operation of dynamic systems by learning : A new control theory for servomechanism or mechatronics systems ». In : The 23rd IEEE Conference on Decision and Control. T. 23. IEEE, p. 1064–1069 (cf. p. 19).
- [27] Owens, David H. et Daley, Steve (2008). « Iterative Learning Control—Monotonicity and Optimization ». In : International Journal of Applied Mathematics and Computer Science 18.3, p. 279–293 (cf. p. 20).
- [28] Wang, Youqing, Gao, Furong et Doyle, Francis J. III (2009). « Survey on iterative learning control, repetitive control, and run-to-run control ». In : Journal of Process Control 19.10, p. 1589–1600 (cf. p. 20).
- [29] Pargfrieder, Joachim et Jörgl, H.P. (2002). « An integrated control system for optimizing the energy consumption and user comfort in buildings ». In : Computer Aided Control System Design, 2002. Proceedings. 2002 IEEE International Symposium on. IEEE, p. 127–132 (cf. p.20).
- [30] Casals, Alicia (2013). « Adaptive Control in Neurorehabilitation ». In : Converging Clinical and Engineering Research on Neurorehabilitation. Springer, p. 123–127 (cf. p. 20).
- [31] Mitchell, Tom Michael (2006). The discipline of machine learning. Carnegie Mellon University, School of Computer Science, Machine Learning Department (cf. p. 21).
- [32] Cornuéjols, Antoine et Miclet, Laurent (2010). Apprentissage artificiel : Concepts et algorithmes, 2ème édition. Eyrolles (cf. p. 21, 42).
- [33] Russell, Stuart Jonathan et Norvig, Peter (2010). Artificial intelligence : a modern approach, 3rd edition. Prentice Hall, Person Education Inc. (cf. p. 22, 70).

Chapitre2

- [34] ActivMedia 04 ActivMedia. P3-DX : World's Most Popular Intelligent WheeledRobot, 2004.
<http://www.activrobots.com>.
- [35] Airmar 04 Airmar. Transducers and Sensors, 2004.
<http://www.airmar.com>.

- [36] G. Campion, G. Bastin et B. D'Andr' ea-Novel. Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. IEEE Transactions on Robotics and Automation, vol. 12, no. 1, pages 47–62, 1996
- [37] Bar-Shalom 88 Y. Bar-Shalom et T. E. Fortmann. Tracking and data association, volume 179 of Mathematics in Science and Engineering. Academic Press, 1988.
- [38] Berthoz 93 A. Berthoz. Physiologie de la perception et de l'action, leçon inaugurale, 1993.
<http://www.college-de-france.fr/media/phyper/UPL52025BerthozR01-02.pdf/>.
- [39] Borenstein 94 J. Borenstein et L. Feng. UMBmark - A Method For Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots. Rapport technique UM-MEAM-94-22, 1994, University of Michigan, 1994.
<http://citeseer.ist.psu.edu/borenstein95umbmark.html>.
- [40] Borenstein 96 J. Borenstein, B. Everett et L. Feng. Navigating mobile robots : Systems and techniques. A. K. Peters, 1996.
www-personal.umich.edu/johannb/shared/pos96rep.pdf.
- [41] Campion 96 G. Campion, G. Bastin et B. D'Andr' ea-Novel. Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. IEEE Transactions on Robotics and Automation, vol. 12, no. 1, pages 47–62, 1996.
- [42] Christensen 02 H. Christensen. SLAM Summerschool, 2002.
<http://www.cas.kth.se/SLAM>.
- [43] Dana 01 Peter H. Dana. Global Positioning System Overview, 2001.
<http://www.colorado.edu/geography/gcraft/notes/gps/gpsf.html>.
- [44] de Wit 96 C. Canudas de Wit, B. Siciliano et G. Bastin, éditeurs. Theory of robot control. Springer, 1996.
- [45] Dudek 00 G. Dudek et M. Jenkin. Computational principles of mobile robotics. Cambridge University Press, 2000.
- [46] A. Elfes. Using Occupancy Grids for Mobile Robot Perception and Navigation. Computer, vol. 22, pages 46–57, juin 1989.
- [47] EMS. Industrial Sensors and Controls, 2004. <http://www.emssensors.com>.
- [48] J.-Y. Fourquet et M. Renaud. Coordinated Control of a Non-Holonomic Mobile Manipulator. In ISER'1999, pages 115–125, Sydney, Australie, mars 1999.
- [49] M. Ghallab. La robotique mobile, chapitre Planification et d'ecision. Hermes, 2001.
<http://www.laas.fr/malik/#lectures>.
- [50] K-team. Khepera II user manual, 2002. <http://www.k-team.com>.

- [51] S. M. LaValle. Planning algorithms. Published online, 1999-2004. <http://misl.cs.uiuc.edu/planning>.
- [52] V. Lee-Shue. Motion Planning. Université de Carnegie Mellon, Pittsburg, Etats-Unis, 2004. <http://generalrobotics.org/syllabus.html>.
- [53] G. Dudek et M. Jenkin. Computational principles of mobile robotics. Cambridge University Press, 2000.
- [54] J.-Y. Fourquet et M. Renaud. Coordinated Control of a Non-Holonomic Mobile Manipulator. In ISER'1999, pages 115–125, Sydney, Australie, mars 1999.
- [55] J. L. Jones, A. M. Flynn et B. A. Seiger. Mobile robots : Inspiration to implementation. AK Peters, Ltd., 1999.
- [56] K-team. Khepera II user manual, 2002. <http://www.k-team.com>.
- [57] J.-C. Latombe. Motion Planning : A Journey of Robots, Molecules, Digital Actors, and Other Artifacts. The International Journal of Robotics Research, vol. 18, no. 11, pages 1119–1128, 1999.
- [58] J.-C. Latombe. Motion Planning. Université de Stanford, Etats-Unis, 2004. <http://robotics.stanford.edu/latombe/cs326/2004/index.htm>.
- [59] S. M. LaValle. Planning algorithms. Published online, 1999-2004. <http://misl.cs.uiuc.edu/planning>.
- [60] L. Extrinsic. Capteurs et modules électroniques pour robotique, 2004. <http://www.lextronic.fr>.
- [61] P. Morin et C. Samson. Introduction à la commande par retour d'état des systèmes non-holonomes. Ecole des Mines de Paris, 2003.
- [62] OmniBot. Projet Robea. De la vision omnidirectionnelle à la commande de robots mobiles, 2004. <http://www.laas.fr/robea/omnibot.pdf>.
- [63] T. Pajdla. Omnidirectionnel Vision Course. Rapport technique, ICCV 2003, 2003. <http://cmp.felk.cvut.cz/pajdla/Pajdla-Omni-Vision-ICCV-2003>.
- [64] C. Petres. Underwater obstacle avoidance path planning. Master's thesis, Ecole Nationale Supérieure de Physique de Strasbourg, 2004.
- [65] T. Siméon. Nic's homepage, 2004. <http://www.laas.fr/nic>.
- [66] U. R. Zimmer. Mobile Robotics : Path Planning and Motion Control. Université Nationale d'Australie, Camberra, Australie, 2001
- [67] E. Wijn, "Unstable behavior of a unicycle mobile robot with tracking control," Report No. DCT 2004.63, Eindhoven, Eindhoven University of technology, 2004.

- [68] R. Barzamini and A. Afshar, "Dynamic adaptive tracking control for wheeled mobile robots," AmirKabir University of Technology, Tehran,2006.
- [69] G. Klancar and I. Skrjanc, "Tracking-error model-based predictive control for mobile robots in real time," Robotics and autonomous systems , 460-469,2007.
- [70] J. Ye, "Tracking control for nonholonomic mobile robots: Integrating the analog neural network into backstepping technique,". Neurocomputing , 3373-3378. (2007).
- [71] J. Velagic, N. Osmic and B. Lacevic, " Neural Network controller for mobile robot motion control," International journal of intelligent systems and technologies,2008 .
- [72] X. Jiang and M. Yung, " Predictive fuzzy logic controller for trajectory tracking of a mobile robot," IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications, pp. 29 – 32,2005
- [73] O. Castillo and L. T. Aguilar , " Fuzzy logic tracking control for unicycle mobile robots," Advanced online publications,2006.
- [74] F. Pourboghraat and M. P. Karlsson, " Adaptive control of dynamic mobile robots with nonholonomic constraints," Carbondale, IL 62901-6603, USA: Department of Electrical and Computer Engineering, Southern Illinois University,2002
- [75] R. M. DeSantis, "Modeling and path-tracking control of a mobile wheeled robot with differential drive," Robotica, volume 13, pp 401-410, Cambridge university press, 1995.
- [76] G. Campion and G. Bastin, "On adaptive linearizing control of omni-directional mobile robots," MTNS, (pp. 531-538), Amsterdam,1989
- [77] Y. Kanayama and Y. Matsuo, " A Stable tracking control method for an autonomous mobile robot," IEEE Transactions on Robotics and Control , arizona,1990

Chapitre 3

- [78] Boukezzoula01 R. Boukezzoula, S. Galichet, L. Foulloy, "Robust fuzzy control for a class of continuous nonlinear systems using input-output linearization: Real-time implementation for a robot wrist", Proc. of the 2001 IEEE Int. Conf. on Control Applications (CCA2001), Mexico, Mexique, Sept.2001, pp 311-316.
- [79] Bouslaa92 F. Bouslama, A. Ichikawa, "Fuzzy control rules and their natural control laws", Fuzzy Sets and Systems, vol. 48, pp. 65-86, 1992.
- [80] Chao96 C.T. Chao, Y.J. Chen, C.C. Teng, "Simplification of fuzzy-neural systems using similarity analysis", IEEE Trans. on Systems, Man, and Cybernetics, Part B, vol. 26, no. 2, pp. 344-354, April 96.

- [81] Charbonnier00 S. Charbonnier, S. Galichet, G. Mauris, J.P. Siché, “Statistical and Fuzzy Models of Ambulatory Systolic Blood Pressure for Hypertension Dia-gnosis”, IEEE Trans. on Instrumentation and Measurement, Vol. 49, N° 5, October 2000, pp. 998-1003.
- [82] Cordon00 O. Cordon, F. Herrera, “A proposal for improving the accuracy of linguistic modeling”, IEEE Trans. on Fuzzy Systems, vol. 8, no. 3, pp. 335-344, June 2000.

Chapitre 4

- [83] B. Hu, G. Mann, R.G. Gosine, A systematic study of fuzzy PID controllers-Function-based evaluation approach, IEEE Transaction on Fuzzy Systems 9 (2001) 693–712
- [84] Xu, C. Hang, C. Liu, Parallel structure and tuning of a fuzzy PID controller, Automatica 36 (2000) 673–684.
- [85] Z.Y. Zhao, M. Tomizuka, S. Isaka, Fuzzy gain scheduling of PID controllers, IEEE Transactions on Systems Man and Cybernetics 23 (1993) 1392–1398.
- [86] Volosencu, C. Pseudo-Equivalence of Fuzzy PID Controllers, WSEAS Transactions on Systems and Control, Issue 4, Vol. 4, April 2009, p. 163-176.
- [87] Y. Zhao, E.G. Collins, Fuzzy PI control design for an industrial weigh belt feeder, IEEE Transactions on Fuzzy Systems 3 (2003) 311–319.
- [88] ISO, International Standard 7730. 1984, ISO Geneva; revised 1990.
- [89] Z.Y. Zhao, M. Tomizuka, S. Isaka, Fuzzy gain scheduling of PID controllers, IEEE Transactions on Systems Man and Cybernetics 23 (1993) 1392–1398.
- [90] Ang, K.H., Chong, G., (2005), PID control system analysis, design and technology, IEEE Transaction on control systems technology, 13(4), pp.559-576
- [91] Soyguder, S., Karakose, M., Alli, H. (2009) Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system. Expert Systems with Applications, 36, 4566-4573
- [92] Soyguder, S., Karakose, M., Alli, H. (2009) Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system. Expert Systems with Applications, 36, 4566-4573
- [93] Khodakarami J, Nasrollahi N. Thermal comfort in hospitals—a literature review. Renewable and Sustainable Energy Reviews 2012;16(6):4071–7.
- [94] Ormandy D, Ezratty V. Health and thermal comfort: from WHO guidance to housing strategies. Energy Policy 2012;49(0):116–21.

- [95] Humphreys MA, Nicol JF. Understanding the adaptive approach to thermal comfort, field studies of thermal comfort and adaptation. *ASHRAE Technical Data Bulletin* 1998;14(1):1–14.
- [96] Chen A, Chang VWC. Human health and thermal comfort of office workers in Singapore. *Building and Environment* 2012;58(0):172–8.
- [97] Leyten JL, Kurvers SR, Raue AK. Temperature, thermal sensation and workers’ performance in air-conditioned and free-running environments. *Architectural Science Review* 2013;56(1):14–21. <http://dx.doi.org/10.1080/00038628.2012.745391>.
- [98] Liu, Y., Yan, H., Lam, J.C., Thermal comfort and building energy consumption implications –A review, *Applied Energy*, 115 (2014) 164-173
- [99] Pablo Gil, Youcef Mezouar “Robotic Perception of the Sight and Touch to Interact with Environments,” Hindawi Publishing Corporation *Journal of Sensors*, Article ID 1751205, 2 pages Volume, 2016.
- [100] U. Libal, J. P plaskonka, “Noise sensitivity of selected kinematic path following controllers for a unicycle,” *Bull. Pol. Ac*, Vol. 62, No. 1, Mar.2014.
- [101] Bin Hua, Endri Rama “A Human-Like Robot Intelligent Navigation in Narrow Indoor Environments,” *International Journal of Information and Electronics Engineering*, Vol. 6, No. 5, Sep. 2016.
- [102] Kai Chen, Fangkai Yang, and Xiaoping Chen “Planning with Task-Oriented Knowledge Acquisition for a Service Robot,” *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence IJCAI-2016*.
- [103] Yogita Gigras, Kusum Gupta, “Artificial Intelligence in Robot Path Planning,” *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Volume-2, Issue-2, May. 2012.
- [104] Bruno Damas and Jose Santos-Victor, “Avoiding Moving Obstacles: the Forbidden Velocity Map,” *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* St. Louis, USA 11-15, Oct. 2009.
- [105] WentaoYu , Jun Peng, Xiaoyong Zhang, Kuo-chi Lin2, “A Cooperative Path Planning Algorithm for a Multiple Mobile Robot System in a Dynamic Environment,” *International Journal of Advanced Robotic Systems* ISSN: 1729-8806, Aug. 2014.
- [106] Matthew Turpin, Kartik Mohta, “Goal Assignment and Trajectory Planning for Large Teams of Aerial Robots,” *Robotics: Science and Systems* Berlin, Germany, 24-28, Jun. 2013.

- [107] JAN CVEJN, “PID control of FOPDT plants with dominant dead time based on the modulus optimum criterion,” Archives of Control Sciences Volume 26(LXII), No.1, pages 5–17, 2016.
- [108] Mahmood Ali Moqbel Obaid, Abdul Rashid Husain, Ali Abdo Mohammed Al-kubati, “Robust Back stepping Tracking Control of Mobile Robot Based on Nonlinear Disturbance Observer,” International Journal of Electrical and Computer Engineering (IJECE) Vol. 6, No. 2, pp. 901~908, Apr. 2016.
- [109] Dazhao Cheng, Yanfei Guo, Changjun Jiang, Xiaobo Zhou, “Self-Tuning Batching with DVFS for Performance Improvement and Energy Efficiency in Internet Servers,” ACM Transactions on Autonomous and Adaptive Systems (TAAS) TAAS Homepage archive Volume 10 Issue 1, Mar. 2015.
- [110] Selma Dilek, Hüseyin Çakır and Mustafa Aydın, “Applications of artificial intelligence techniques to cyber crimes,” a review International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 6, No. 1, Jan. 2015.

Sommaire

Introduction	1
Chapitre1	5
Le contrôle des systèmes complexes	5
1.1 Le contrôle	5
1.1.1 De la révolution industrielle à nos jours	5
1.1.2 Définition et concepts fondamentaux	6
1.1.2.1 La notion de système	6
1.1.2.2 La notion de contrôle	8
1.2 Contrôle de systèmes complexes et apprentissage artificiel	11
1.2.1 Les classiques du contrôle des systèmes	11
1.2.1.1 Contrôleurs PID	11
1.2.1.1.1 Méthode de Ziegler-Nichols	13
1.2.1.1.2 limite des PID	13
1.2.1.1.3 Bilan des PID	14
1.2.1.2 Contrôle adaptatif	14
1.2.1.2.1 Contrôle avec modèle de référence	15
1.2.1.2.2 Contrôle avec identification de système	16
1.2.1.2.3 Commande prédictive	16
1.2.1.2.4 Contrôle dual	18
1.2.1.2.5 Contrôle par apprentissage itératif	19
1.2.1.2.6 Bilan du contrôle adaptatif	20
1.2.1.3 Bilan des approches classiques de contrôle	21
1.2.2 L'apprentissage artificiel	21
1.2.2.1 Apprentissage supervisé	21
1.2.2.1.2 Réseaux de neurones artificiels	22
1.2.2.2 Apprentissage non-supervisé	23
1.2.2.3 Apprentissage actif	24
1.2.3 Le contrôle intelligent	24
1.2.3.1 le PID intelligent	25
1.3 Conclusion	26
Chapitre 2	27
2 Le robot mobile	27
2.1 Introduction	27
2.2 Modélisation et propriétés structurelles	28
2.2.1 Définition	28
2.2.2 Roulement sans glissement et contraintes non holonomes	29
2.2.2.1 Roulement sans glissement	29
2.2.2.2 Contraintes non holonomes	30

2.2.3 Les grandes classes de robots mobiles et leurs modèles	32
2.2.3.1 Disposition des roues et centre instantané de rotation	32
2.2.3.2 Robots mobiles de type tricycle et de type voiture	35
2.2.3.2.1 Présentation des robots de type voiture	35
2.2.3.2.2 Robots mobiles de type unicycle	38
2.3 Planification de trajectoire	39
2.3.1 Approches polynomiales pour la génération de trajectoires	40
2.3.2 Décomposition en cellules	41
2.3.3 Approche par champs de potentiel	42
2.3.4 Utilisation des roadmaps	43
2.3.4.1 Diagramme de Voronoï	43
2.3.4.2 Graphe de visibilité	43
2.4 Poursuite de trajectoire	44
2.4.1 Formulation du problème	44
2.4.2 Stabilisation par un retour d'état non stationnaire continu	45
2.4.2.1 Commande non linéaire non stationnaire	45
2.4.2.2 Commande de système du robot de type unicycle sous forme chaînée	47
2.4.2.3 Approche par fonction transverse	48
2.4.3 Stabilisation par un retour d'état discontinu	49
2.5 Conclusion	51
Chapitre 3	52
La logique floue et la régulation intelligente	52
3. 1 Logique floue	52
3.1.1 Relations floues et logique floue	52
3.1.2 Principes et définitions de la logique floue	52
3.1.3 Ensembles flous	53
3.1.4 Variable linguistique	54
3.1.5 Fonctions d'appartenance	55
3.1.5.1 Caractéristique de la fonction d'appartenance	55
3.1.5.1.1 Le noyau	55
3.1.5.1.2 Le support	55
3.1.5.1.3 Les Frontières	55
3.1.6 Opérations sur les ensembles flous	57
3.1.7 Propositions floues	57
3.1.7.1 Propositions floues élémentaires	57
3.1.7.2 Propositions floues générales	57
3.1.8 Raisonnement en logique floue	58
3.1.8.1 Implication floue	58

3.1.8.2	Inférence floue	58
3.1.8.3	Agrégation des règles	60
3.1.9	Fuzzification	60
3.1.9.1	Intuition	61
3.1.9.2	Inférence	61
3.1.9.3	Le classement	61
3.1.9.4	Les réseaux de neurones	62
3.1.9.5	Algorithmes génétiques	63
3.1.9.6	Assaisonnement inductif	64
3.1.10	Defuzzification	65
3.1.10.1	Principe d'appartenance maximum	66
3.1.10.2	Méthode centroïde	66
3.1.10.3	Méthode de moyenne pondérée	66
3.1.10.4	moyenne-maximale d'appartenance	67
3.1.10.5	Centre des sommes	67
3.1.10.6	Centre de la plus grande surface	68
3.1.10.7	Premier des maximums ou dernier des maximums	69
3.2	Conception d'un contrôleur flou	69
3.2.1	Fusion de la logique floue et les PID	74
3.2.2	Inversion des systèmes flous	77
3.2.3	Inversion numérique des systèmes flous	78
3.3	Conclusion	79
Chapitre 4		82
Stratégie de la commande intelligente d'un robot mobile		82
Introduction		82
4.1	Contrôleur PID-floue	82
4.1.1	Structure du contrôleur flou-PID	83
4.1.2	Modèle cinématique du robot mobile	83
4.1.2.1	Modélisation du robot mobile	84
4.1.3	Algorithme de contrôle PID	84
4.1.4	Conception du contrôleur flou	85
4.2	Contrôleur flou-PI+D	91
4.2.1	Fuzzification du Contrôleur flou-PI+D	97
4.2.2	Defuzzification du Contrôleur flou-PI+D	98
4.3	Contrôleur neuro-PID	102
4.3.1	Structure du contrôleur neuro-PID	102
4.3.2	Algorithme RBFNN pour la régulation des K_p , K_i et K_d	103

4.3.3 Algorithme du contrôle PID	104
4.3.4 Résumé du contrôle PID basé sur les RBFNN	105
4.4 Contrôleur neuro-flou-PI+D	107
4.4.1 Méthodologie	108
4.4.2 Le système de réseau de neurone	108
4.4.3 Structure du contrôleur flou- PI+D	109
4.4.4 Résultats de simulation	112
4.5 Contrôleur neuro- PI+D	114
Conclusion 4.6	119
Conclusion générale	119
Références bibliographiques	121

Résumé

De nos jours, le contrôle classique PID conventionnel est les PID numériques est le plus utilisés, mais pour obtenir un contrôle souhaité, il faut que les utilisateurs connaissent les caractéristiques du système à contrôler. Aujourd'hui Les PIDs se trouvent sous forme d'un module électronique ayant les entrées et les sorties. Ce circuit est flexible pour le réglage des différents gains. Et lui appliqués les algorithmes qui satisfais le contrôle adéquates.

Cette thèse s'intéresse et illustre la mise en œuvre du réglage du contrôleur PID à l'aide de technique évolutive qui est la logique floue et les réseaux de neurones artificiels. Cette nouvelle proposition est appliqué pour traiter le problème de suivie de trajectoire pour un robot mobile non holonome de type voiture. L'enjeu principal de la robotique mobile actuelle consiste à développer des systèmes de navigation intelligents, où la navigation autonome qui vise à donner à une machine la capacité de se mouvoir dans un environnement sans assistance, ni intervention humaine pour accomplir un but désiré. Les concepts proposés dans ce document sont testés et simulés sous le logiciel matlab.simulink. Les résultats obtenus prouvent l'efficacité des techniques de réglage intelligentes proposées.

summary

Nowadays, the conventional conventional PID control is the digital PID is the most used, but to obtain a desired control, users must know the characteristics of the system to control. Today PIDs are in the form of an electronic module with inputs and outputs. This circuit is flexible for adjusting the different gains. And applied to it the algorithms that satisfy adequate control.

This thesis is interested in and illustrates the implementation of PID controller tuning using an evolutionary technique which is fuzzy logic and artificial neural networks. This new proposal is applied to address the problem of trajectory following for a non-holonomic mobile robot of the car type. The main challenge of current mobile robotics is to develop intelligent navigation systems, where autonomous navigation which aims to give a machine the ability to move in an environment without assistance or human intervention to achieve a desired goal. The concepts proposed in this document are tested and simulated under the software matlab.simulink. The results obtained prove the effectiveness of the intelligent adjustment techniques proposed.

المخلص

في الوقت الحاضر ، عنصر التحكم التقليدي PID هو PID الرقمي وهو الأكثر استخدامًا ، ولكن للحصول على عنصر التحكم المطلوب ، يجب على المستخدمين معرفة خصائص النظام للتحكم. اليوم PIDs يوجد على شكل وحدة إلكترونية مع المدخلات والمخرجات. هذه الدائرة مرنة لضبط المكاسب المختلفة. وتسمح بتطبيق الخوارزميات التي تحقق التحكم الكافي.

تهتم هذه الرسالة وتوضح تنفيذ تعديل وحدة تحكم PID باستخدام تقنية تطويرية هي الشبكات العصبية والمنطقية. يتم تطبيق التحكم الأمثل PID. يتم تطبيق هذا الاقتراح الجديد لمعالجة مشكلة تتبع المسار لروبوت متنقل من نوع سيارة. يتمثل التحدي الرئيسي للروبوتات المتنقلة الحالية في تطوير أنظمة ملاحية ذكية ، حيث الملاحية المستقلة التي تهدف إلى منح الآلة القدرة على التحرك في بيئة دون مساعدة أو تدخل بشري لتحقيق الهدف المنشود. المفاهيم المقترحة في هذا المستند تم اختبارها ومحاكاتها تحت برنامج matlab.simulink. النتائج التي تم الحصول عليها تثبت فعالية تقنيات التكيف الذكية.