

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

Université de Batna
Faculté des Sciences de l'Ingénieur
Département d'Electronique

Thèse

Préparée au
Laboratoire d'Electronique Avancée (LEA) Batna

Par :

Zidani Ghania

Ingénieur d'état en électronique

En vue de l'obtention du diplôme de :
Magister en Robotique

Thème

Exécution de Trajectoire Pour Robot Mobile d'Intérieur
-Réseaux de Neurones-

Soutenu le : 14 /01 /2009

Membres du Jury

Boulemden Mohamed	Professeur	Université de Batna	Président
Louchene Ahmed	Maître de Conférences	Université de Batna	Rapporteur
Baarir Zineddine	Maître de Conférences	Université de Biskra	Examineur
Khireddine M. Salah	Maître de Conférences	Université de Batna	Examineur

**Exécution de Trajectoire Pour Robot Mobile
d'Intérieur
-Réseaux de Neurones-**

إلى أمي و أبي

إلى جميع أفراد عائلتي و كل من أحبه في الله

REMERCIEMENTS

Je tiens à remercier tout particulièrement monsieur Dr A. LOUCHENE, mon directeur de mémoire, pour m'avoir confié et dirigé ce projet, pour ses remarques et ses critiques et qui n'a jamais manqué de me conseiller et de m'orienter tout au long de la réalisation de ce travail. Qu'il trouve ici l'expression de mon respect et de ma profonde reconnaissance.

Je remercie sincèrement mes collègues du laboratoire LEA -Laboratoire de l'Electronique Avancée- l'université de Batna, Mr D. DJRAH et Mr A. BENMAKHOUF.

Ma profonde gratitude s'adresse à monsieur Pr. N. NAIT SAID et Mme Pr. F. ZIDANI, leurs contributions ont beaucoup apporté au développement de ce mémoire

Je remercie les membres du jury qui ont bien voulu accepter d'examiner ce travail.

Merci à tous ceux qui de près ou de loin m'ont aidé.

Sommaire

Introduction générale	01
-----------------------------	----

Chapitre 1 : La robotique mobile

1.1. Qu'est ce qu'un robot mobile	04
1.2. Composantes d'un robot mobile	05
1.2.1. Composantes matérielles	05
1.2.2. Modules logiciels	05
1.3. Principales architectures de commande pour robots mobiles	07
1.3.1. Architecture délibérative	07
1.3.2. Architecture comportementale	08
1.3.3. Architecture hybride	10
1.4. Les robots mobiles à roues	10
1.5. Les robots à déplacement articulé	11
1.6. Modélisation de la cinématique de quelques robots mobile à roues	11
1.6.1. Modèle cinématique du robot unicycle	12
1.6.2. Modèle cinématique du robot tricycle	13
1.6.3. Modèle cinématique du robot de type voiture	14
1.6.4. Modèle cinématique du robot de type omnidirectionnel	15
Conclusion	16

Chapitre 2 : Réseaux de neurones -Aspects fondamentaux-

2.1. Historique	17
2.2. Application	19
2.3. Qu'est ce qu'un réseau de neurones	20
2.4. Modèle mathématique	20
2.5. Techniques d'apprentissages	21
2.5.1. L'apprentissage supervisé	21
2.5.2. L'apprentissage non supervisé	22
2.6. Fonctions de transfert	22
2.7. Quelques architectures neuronales	23
2.7.1. Perceptron Multi Couches «PMC»	23
2.7.2. Le réseau de Hopfield	25
2.7.3. Le réseau de Kohonen	26
2.7.4. Le réseau RBF	27
2.8. Propriétés des réseaux de neurones artificiels	27
2.8.1. Apprentissage et mémoire	27
2.8.2. Sous-apprentissage, généralisation et sur-apprentissage	28
Conclusion	30

Chapitre 3 : Contrôleurs neuronaux utilisés

3.1. Motivation pour l'emploi des réseaux de neurones	31
3.2. Identification de processus	32
3.3. Classification des méthodes de commande	33
3.4. Méthodes directes	33
3.4.1. Reproduction d'un contrôleur existant	33
3.4.2. Approche modèle inverse	36
3.4.3. Amélioration des régulateurs conventionnels	38
a. Amélioration d'un système linéaire	38
b. Commande avec un contrôleur PD neuronal	39
Auto ajustement des paramètres d'un PD	39
3.4.4. Utilisation directe de l'erreur en sortie du procédé	41
3.5. Méthodes indirectes	42
3.5.1 Apprentissage indirect du système de commande	42
3.5.2. Réalisation d'un système de commande par modèle prédictif	42
Conclusion	43

Chapitre 4 : Tests et validation

4.1. Simulation de l'approche « Reproduction d'un contrôleur existant »	45
4.1.1. Modèle du robot	45
4.1.2. Technique de poursuite	46
4.1.3. Choix du réseau	47
Conclusion	51
4.2. Simulation de l'approche « Modèle inverse »	51
4.2.1. Modèle du robot	51
4.2.2. Technique de poursuite	52
4.2.3. Choix du réseau	53
Conclusion	57
4.3. Simulation de l'approche « PD neuronal à coefficients adaptatifs »	57
4.3.1. Choix du réseau	57
4.3.2. Résultats de simulation	57
Conclusion	61

Conclusion générale	62
---------------------------	----

Bibliographie

INTRODUCTION GENERALE

INTRODUCTION GENERALE

L'un des défis de l'homme aujourd'hui est de copier la nature et de reproduire des modes de raisonnement et de comportement qui lui sont propre. Cette envie a fait naître le concept d'*intelligence artificielle*. Celui-ci se traduit par l'émergence des techniques avancées telles que les réseaux de neurones artificiels. Ces approches dites "intelligentes" se sont déjà imposées dans un grand nombre de domaines tels que la reconnaissance de forme, le traitement du signal, la gestion financière, la robotique, le contrôle des processus, etc.

Depuis 1970 le monde scientifique et industriel porte des efforts sans cesse croissants sur l'étude et la réalisation d'engins mobiles. Ces derniers sont capables d'effectuer des déplacements sans aucune liaison mécanique avec un point fixe de l'espace dans lequel ils évoluent, d'où le nom de *robots mobiles*. Aujourd'hui, l'industrie mondiale subit une véritable mutation qui se modifie profondément les données. La robotisation constitue une nouvelle avancée de l'automatisation grâce aux progrès de la micro-électronique permettant la mémorisation des informations de réglage et de contrôle pour un coût abordable et un volume réduit. Dans ce contexte, nous pouvons considérer que le robot mobile est synonyme de véhicule puisque ce dernier est composé de la structure matérielle et mécanique qui permet au robot d'*effectuer un déplacement*.

D'un autre côté, les techniques basées sur l'emploi des *réseaux de neurones artificiels* suscitent au aujourd'hui un intérêt croissant dans les domaines de la commande et de la robotique. La rapidité de traitement, les

capacités d'apprentissage et d'adaptation, mais aussi la robustesse de ces approches motivent en grande partie de nombreux travaux dans ce domaine - commande des robots mobiles.

Le travail présent a pour but d'étudier les possibilités d'utilisation de réseaux des neurones artificiels dans le cadre de la commande des robots mobiles à roues. C'est une étude qui permettra à ces derniers, utilisant des contrôleurs neuronaux artificiels, de suivre une trajectoire de référence préalablement définie.

Le mémoire s'articulera autour de quatre chapitres

- Dans le premier chapitre nous exposerons les principales architectures de commande pour les robots mobiles, celles-ci seront séparées en trois grandes familles : l'approche délibérative, l'approche réactive et l'approche hybride. Nous présenterons à la fin du chapitre la modélisation de la cinématique de quelques robots mobiles à roues.
- Le deuxième chapitre abordera les réseaux de neurones artificiels. Après une présentation de certains modèles, nous rappellerons les principales caractéristiques et propriétés des réseaux de neurones artificiels ainsi que leur principe de fonctionnement.
- Le troisième chapitre exposera le problème du suivi d'une trajectoire désirée. Nous présenterons une classification des principales méthodes de commande neuronales. Dans ce travail nous nous intéresserons uniquement aux trois approches à savoir : reproduction d'un contrôleur existant, approche modèle inverse et commande avec un contrôleur PD neuronal.

- Les différents résultats de simulation seront présentés au dernier chapitre, suivi d'une conclusion générale ainsi que quelques perspectives envisagées pour la poursuite de ce travail.

Chapitre 1

LA ROBOTIQUE MOBILE

La robotique mobile est un domaine hautement multidisciplinaire qui fait l'objet de beaucoup de recherches dans des disciplines très diversifiées. Pour cette raison, dans ce présent chapitre, nous ferons abstraction de capacités des robots mobiles car il doit posséder de nombreuses capacités, il doit être capable de percevoir son environnement et de se localiser dans celui-ci et capable de se déplacer. Ensuite les principales architectures de commande pour robots mobiles rencontrées dans la littérature, mais il est impossible d'établir une liste exhaustive de ces travaux, c'est pourquoi elles sont généralement séparées historiquement en trois groupes, ainsi que l'objectif de ce chapitre est de présenter les grandes classes de robots mobiles et leurs modèles.

1.1. Qu'est ce qu'un robot mobile

La robotique est la branche de l'intelligence artificielle concernée par l'étude de systèmes automatiques capables d'une interaction directe avec son environnement.

On distingue deux types de robots, les robots fixes qui sont utilisés généralement dans l'industrie pour réaliser les tâches dangereuses et les robots mobiles qui sont utilisés par exemple pour transporter des charges. La tâche essentielle des robots mobiles est le déplacement dans un environnement.

1.2. Composantes d'un robot mobile

A la base, un robot mobile est constitué de composantes matérielles et logicielles

1.2.1. Composantes matérielles

Parmi les composantes matérielles on retrouve une plateforme mobile à laquelle sont rattachées toutes les autres composantes comme les capteurs, les actionneurs et une source d'énergie.

Les capteurs ont pour fonction d'acquérir des données provenant de l'environnement. Ces capteurs peuvent être de type sonars à ultrasons, des capteurs laser de proximité, des encodeurs de roues, des caméras optiques et des microphones.

Un robot mobile est équipé d'actionneurs afin qu'il puisse bouger à l'intérieur de son environnement et interagir avec celui-ci. Par exemple un robot mobile est muni d'un ou plusieurs moteurs pouvant faire tourner ses roues afin d'effectuer des déplacements.

1.2.2. Modules logiciels

Plusieurs modules logiciels sont mis à contribution, afin de faire fonctionner un robot mobile. Ces modules peuvent servir à interpréter les données perçues par les capteurs afin d'en extraire des informations. Les modules les plus fréquemment utilisés sont les modules de localisation, de navigation, de vision.

a. Localisation

L'une des fonctions les plus importantes pour un robot est celle d'être capable de se localiser dans son environnement. À partir des données fournies par les capteurs, le module de localisation estime la position courante du robot. Typiquement, cette position est exprimée par un triplet « x, y, θ » représentant une position et une orientation sur un plan à deux dimensions [1].

b. Perception

Cette partie consiste à faire la lecture des capteurs puis le traitement de l'information ce qui permet de délivrer un message clair au robot. On peut imaginer la situation d'un robot mobile devant son environnement en comparant celui-ci avec un aveugle qui découvrirait un nouveau lieu. Le robot doit être capable d'analyser un objet localement, puis de positionner globalement tous les objets les uns par rapport aux autres. De cette manière il construit un modèle de son environnement et de son exploration.

Donc en analysant les images captées par les caméras, on peut extraire une multitude d'informations. Par exemple, à l'aide d'un algorithme de segmentation on peut reconnaître des objets de couleur en plus d'estimer leur position relative « angle » par rapport à la vue de la caméra. À l'aide de techniques de vision tridimensionnelle, il est aussi possible d'estimer certaines distances dans l'environnement. On peut aussi reconnaître des symboles, des caractères et lire des messages, comme des signaux de direction, ou des badges de conférence.

c. Navigation

Un module de navigation est responsable de déplacer un robot de sa position courante vers une destination désirée de façon sécuritaire et efficace. En

plus d'inclure des fonctions de perception de l'environnement et de localisation, le module de navigation a aussi la responsabilité de trouver un chemin reliant la position d'origine et de destination, formé d'une liste de points intermédiaires à atteindre, et de guider le robot à travers ce chemin [1].

Les moyens de déplacement sont nombreux et il convient d'appliquer un traitement différent selon que le mobile est censé se déplacer sur un sol ou bien au sein d'un même milieu par exemple avion. Nous traitons que les robots mobiles à roues.

1.3. Principales architectures de commande pour robots mobiles

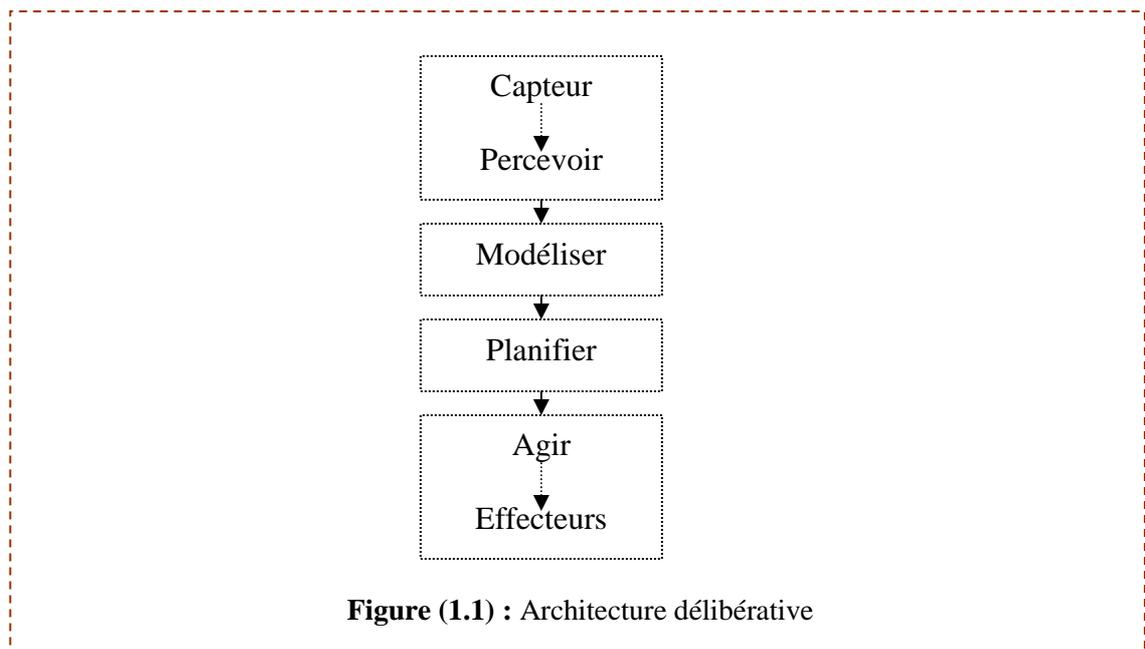
Un robot est composé d'un ensemble de modules, chacun étant responsable d'une ou de plusieurs capacités. L'un des premiers défis à résoudre est de déterminer comment relier efficacement les différents modules. Pour ce faire, il faut élaborer une architecture décisionnelle qui dicte les responsabilités de chacun des modules et comment l'information circule entre ces derniers. Depuis les débuts de la robotique, beaucoup d'architectures ont été proposées. Elles peuvent être généralement classées en trois grandes catégories : délibérative, comportementale et hybride.

1.3.1. Architecture délibérative

Les architectures délibératives sont les premières à avoir été proposées. Comme son nom l'indique, les architectures de ce type sont basées sur des processus complètement planifiés. Par exemple, afin d'exécuter un déplacement, un robot basé sur ce type d'architecture calcule un plan complet, lui disant d'avancer de x mètres, ensuite de tourner de y degré, et ainsi de suite. Lorsqu'un changement dans l'environnement est perçu, l'exécution est suspendue et un nouveau plan est généré [2].

Ces approches s'appuient sur les techniques de l'intelligence classique. Elles visent à reproduire le mode de raisonnement humain ou tout au moins une certaine vision du mode de raisonnement humain. Le traitement est décomposé en une série d'opérations successives décrites par la figure ci-dessous :

Ce type d'architecture souffre de plusieurs lacunes importantes. Premièrement, puisque les capteurs sont imprécis et que l'environnement est dynamique et partiellement observable, il est très difficile de tout prévoir à l'avance. Pour ces raisons, il n'est pas d'une très grande utilité de tout planifier à l'avance, puisque les plans seront constamment à refaire. Un autre problème avec ce type d'architecture est que la génération de plans précis demande beaucoup de ressources « temps de calcul et mémoire ».



1.3.2. Architecture comportementale

L'architecture comportementale, proposée par Brooks, est inspirée par le comportement des insectes. L'idée générale est de développer plusieurs petits modules simples et indépendants les uns des autres et, une fois regroupés, un comportement plus intelligent émerge sans qu'il ait été spécifiquement

programmé. Cette approche, radicalement différente de la précédente se distingue par l'abondance des phases de modélisation et de planification.

Celle-ci est composée de plusieurs modules simples appelés comportements et d'un module d'arbitrage. Dans sa conception, chaque comportement se limite à une seule fonctionnalité pour le contrôle du robot. Les comportements sont tous indépendants les uns des autres. Par exemple, on peut avoir des comportements pour l'évitement d'obstacles, le suivi de chemin, le suivi d'objets de couleur ou la manipulation d'objets. Les comportements sont exécutés parallèlement à une certaine fréquence. Lors d'une itération, chaque comportement calcule une ou plusieurs commandes motrices qui sont envoyées à un module d'arbitrage. Ce dernier fusionne l'ensemble des commandes reçues et calcule les commandes finales qui doivent être envoyées à chacun des actionneurs du robot.

Un exemple d'architecture comportementale est illustré à la figure (1.2.)

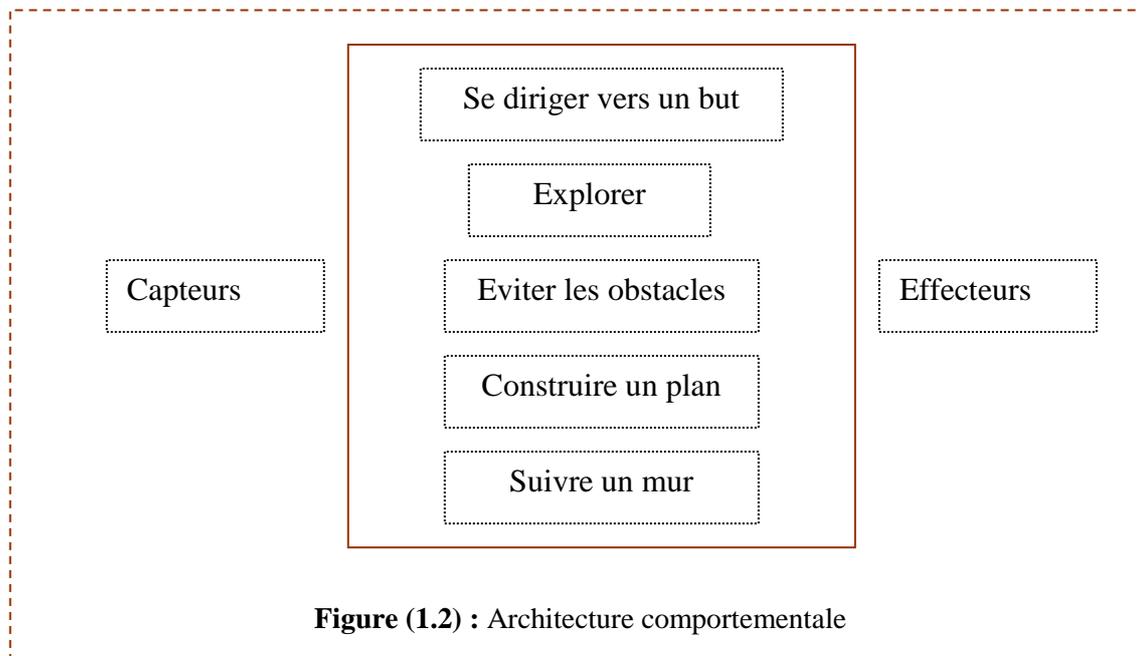


Figure (1.2) : Architecture comportementale

En contrepartie, les architectures comportementales ont de la difficulté à réaliser des tâches structurées, puisqu'elles ne contiennent aucun processus

délibératif. En effet, les tâches complexes requièrent la capacité du robot à prédire les conséquences futures de ses actions afin de sélectionner celles qui conviennent le mieux pour la réalisation de ses activités. Ces tâches complexes ont besoin d'être planifiées.

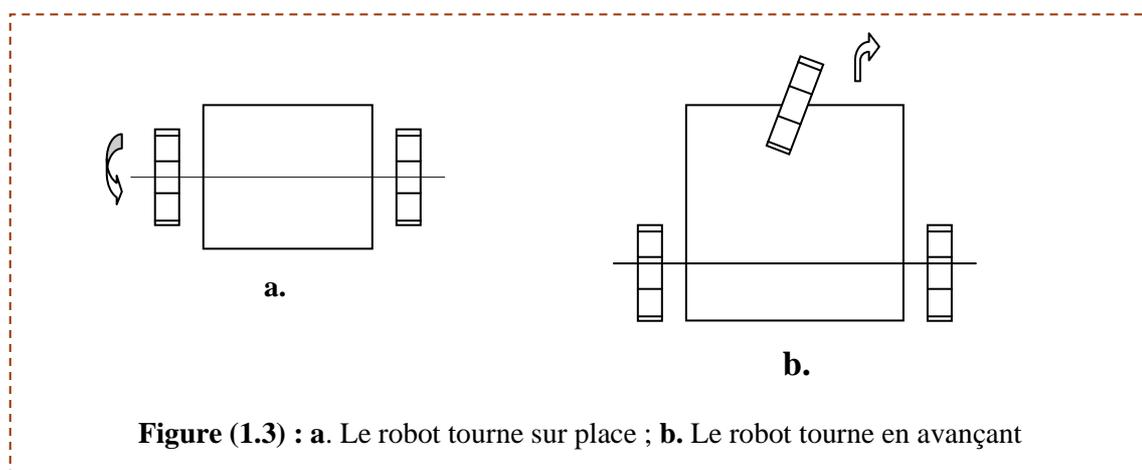
1.3.3. Architecture hybride

Les limitations des deux types d'architecture précédentes justifient l'émergence récente des architectures hybride, tentant de combiner les avantages des architectures délibératives et comportementales. Elles sont généralement décomposées en plusieurs niveaux.

Dans la partie supérieure, on place les modules de type délibératif. Dans la partie inférieure, on retrouve les modules de type comportemental.

1.4. Les robots mobiles à roues

Dans le cas des robots utilisant des roues pour se déplacer le changement de direction est obtenu soit en faisant varier la vitesse des moteurs associés à chacune des roues latérales soit en faisant varier l'orientation des roues « véhicule automobile ».



1.5. Les robots à déplacement articulé

La moitié de la surface de la terre est inaccessible aux meilleurs véhicules à roues, Les recherches actuelles s'orientent vers l'étude de la marche à une ou plusieurs pattes. Le nombre de pattes détermine la complexité du système qui les commande. Les systèmes à une ou deux pattes comme l'homme ne fonctionnent qu'en mode dynamique, alors que quatre pattes ou plus autorisent une équilibre statique [3].

La reptation est le moyen utilisé pour progresser dans une galerie. La progression se fait au moyen de ventouses ou de griffes associées à un allongement des articulations [3]. On trouve différents types de déplacement : le type scolopendre, le type lombric et le type péristaltique ou à soufflet.

1.6. Modélisation de la cinématique de quelques robots mobiles à roues

La problématique de la commande des robots mobiles étant trop vaste, pour cela, nous introduisons un certain nombre d'hypothèses simplificatrices :

- nous nous restreignons aux robots mobiles de type unicycle et de type tricycle;
- les véhicules sont considérés comme rigides et évoluant sur un sol plan ;
- les véhicules sont dotés de roues conventionnelles : le point de contact entre la roue et le sol est réduit à un point I et la roue est soumise à la contrainte de roulement sans glissement, qui se traduit par une vitesse nulle entre le point I et le sol ;
- d'un point de vue schématiquement, le robot de type unicycle est composé de deux roues motrices coaxiales et indépendantes, d'une « ou sans » roue folle, destinée uniquement à assurer la stabilité statique du système.
- le système est dit non-holonyme lorsque ces contraintes sont sous la forme d'équations différentielles non complètement intégrables. Dans le cas des

véhicules sur roues, ces contraintes cinématiques résultent de l'hypothèse de roulement sans glissement.

1.6.1. Modèle cinématique du robot unicycle

On désigne par *unicycle*, un robot actionné par deux roues motorisées et indépendantes.

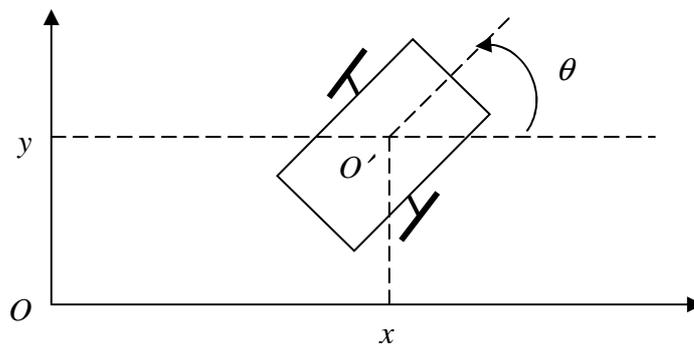


Figure (1.4) : Robot mobile de type unicycle

Centre Instantané de Rotation « CIR » : les roues ayant même axe de rotation, le CIR est un point sur cet axe, figure (1.5)

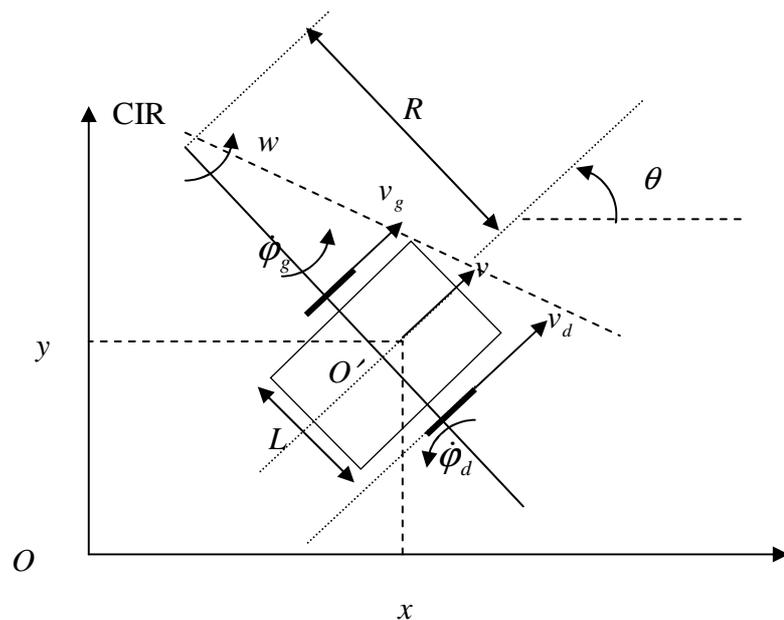


Figure (1.5) : Centre instantané de rotation d'un robot de type unicycle

Où v est la vitesse du centre de surface de robot, v_g est la vitesse de la roue gauche, v_d est la vitesse de la roue droite, r est le rayon de chaque roue, L est la distance entre les deux roues, x et y sont la position du robot mobile, et le θ est l'orientation du robot.

$$\dot{x} = v \cos \theta$$

Le modèle cinématique est : $\dot{y} = v \sin \theta$

$$\dot{\theta} = w$$

1.6.2. Modèle cinématique du robot tricycle

Ce robot est constitué de deux roues fixes de même axe et d'une roue centrée orientable placée sur l'axe longitudinal du robot. Le mouvement est conféré au robot par deux actions : la vitesse longitudinale et l'orientation de la roue orientable. De ce point de vue, il est donc très proche d'une voiture [4], figure (1.6).

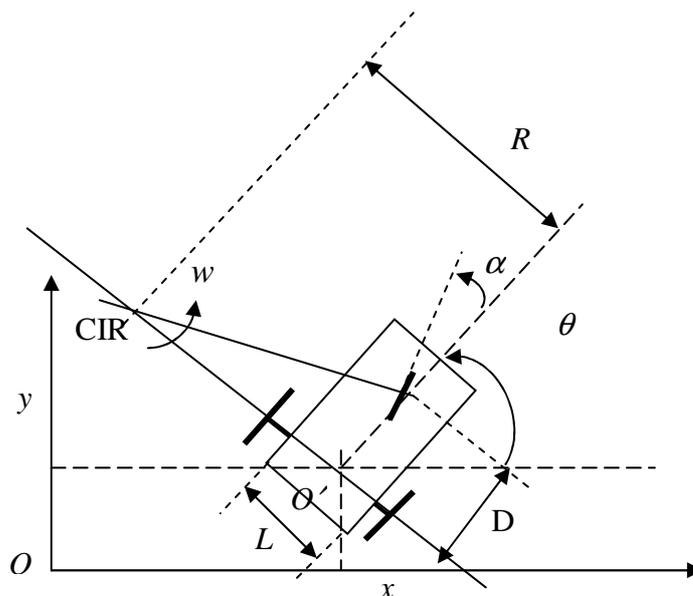


Figure (1.6) : Robot mobile de type tricycle et son CIR

L'angle de braquage α est déterminé comme suit :

$$\begin{aligned}\alpha &= \operatorname{arctg}\left(\frac{D}{R}\right) \\ R &= \frac{L(v_d + v_g)}{2(v_d - v_g)} \\ \alpha &= \operatorname{arctg}\left(\frac{2D(v_d - v_g)}{L(v_d + v_g)}\right)\end{aligned}\tag{1.1}$$

Ce type de robot peut se diriger en ligne droite pour $\alpha = 0$ et théoriquement tourner autour du point O' « on pourrait dire sur place » pour $\alpha = \pi/2$. Néanmoins, le rayon de braquage de la roue orientable, généralement limité, impose le plus souvent des valeurs de telles que $-\pi/2 < \alpha < \pi/2$, interdisant cette rotation du robot sur lui-même. L'écriture des contraintes sur chacune des roues et un raisonnement similaire à celui suivi dans le cas de l'unicycle permettent de déterminer les modèles cinématiques des robots de type tricycle, on établit les équations [4]:

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \frac{v}{D} \operatorname{tg} \alpha\end{aligned}$$

1.6.3. Modèle cinématique du robot de type voiture

Pour le robot de type voiture, schématisé sur la figure (1.7), on introduit la notion de roue directrice centrale. Cette roue, virtuelle dans le cas d'une voiture, correspondrait à la roue directrice d'un tricycle équivalent. Son introduction permet de simplifier les équations en faisant abstraction du mécanisme de couplage des roues directrices servant à respecter les contraintes de roulement sans glissement, et en ne considérant alors qu'un seul angle de direction [5].

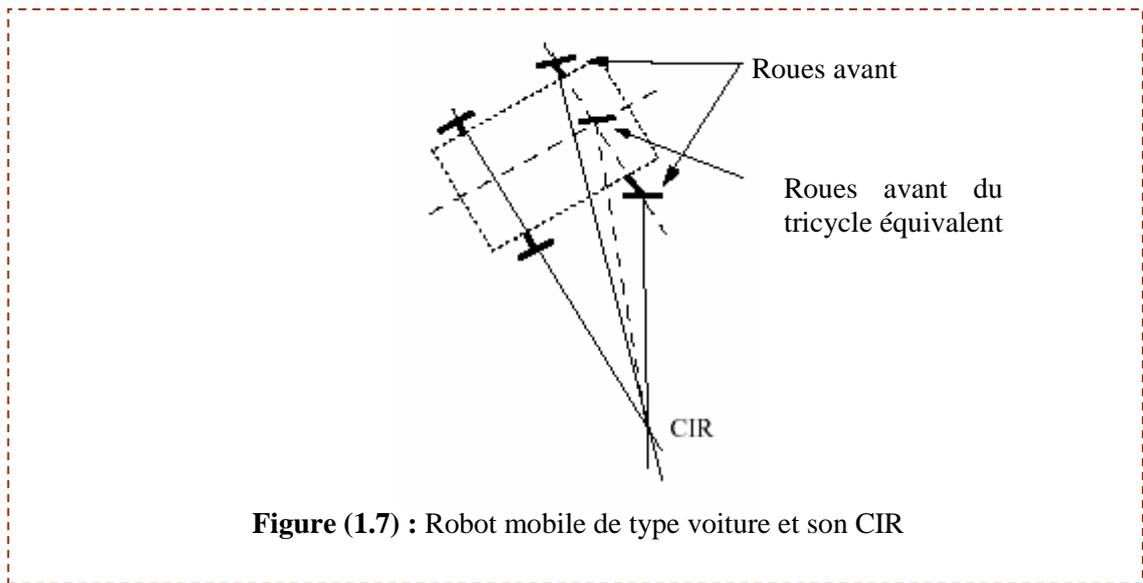


Figure (1.7) : Robot mobile de type voiture et son CIR

1.6.4. Modèle cinématique du robot de type omnidirectionnel

Un robot mobile est dit *omnidirectionnel* si l'on peut agir indépendamment sur les vitesses : vitesse de translation selon les axes \bar{x} et \bar{y} et vitesse de rotation autour de \bar{z} . D'un point de vue cinématique cela n'est pas possible avec des roues fixes ou des roues centrées orientables [Campion 96]. On peut en revanche réaliser un robot omnidirectionnel en ayant recours à un ensemble de trois roues décentrées orientables ou de trois roues suédoises disposées aux sommets d'un triangle équilatéral [4].

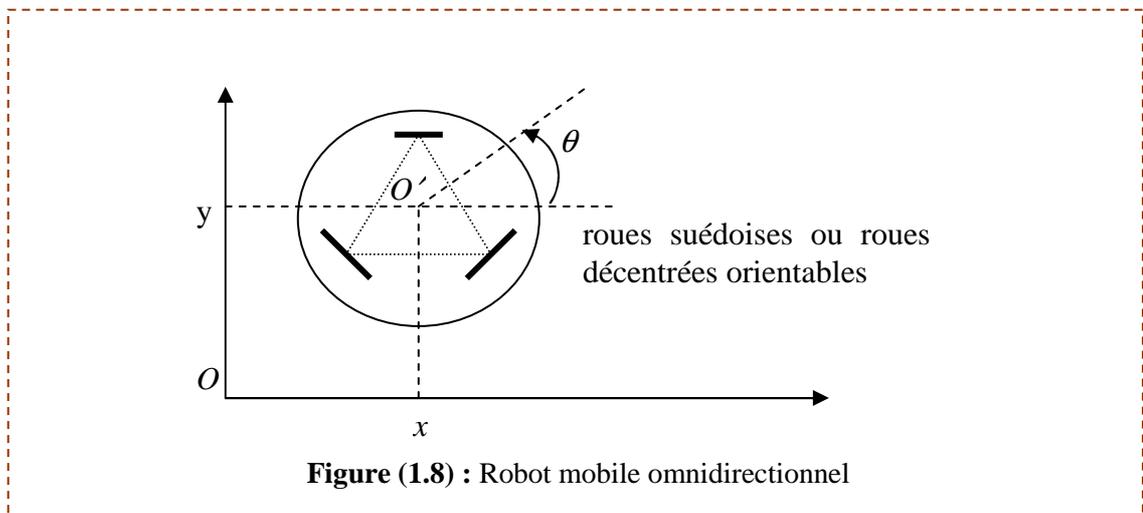


Figure (1.8) : Robot mobile omnidirectionnel

$$\dot{x} = v_1$$

Son modèle cinématique est : $\dot{y} = v_2$

$$\dot{\theta} = v_3$$

Conclusion

Les approches utilisées pour permettre à un robot de combiner des raisonnements avec une vitesse de réaction importante sont différentes, mais les principales sont : *l'approche délibérative*, d'une part, utilise une modélisation de l'environnement, connue a priori ou obtenue à partir des données capteurs, pour planifier à l'avance les commandes que le robot doit exécuter. Cette vision de la robotique autonome conduit à une décomposition séquentielle du traitement réalisée et à des systèmes fortement hiérarchiques; *l'approche réactive*, d'autre part, s'appuie sur un couplage étroit entre les capteurs et les actionneurs, pour générer en continu les commandes. Cette méthode va généralement de pair avec une décomposition sous forme de comportements élémentaires réalisant, chacun en parallèle, une fonction simple; *l'approche hybride* qui tente de combiner les deux approches précédentes afin de tirer partie des avantages respectifs de ces deux techniques.

La robotique mobile consiste d'application très intéressante et pour rendre ces robots intelligents, ils doivent posséder quelques capacités humaines, perception de l'environnement, acquisition de connaissances, prise de décision et capable de se déplacer d'un point à un autre.

Chapitre 2

RESEAUX DE NEURONES - ASPECTS FONDAMENTAUX -

RESEAUX DE NEURONES

ASPECTS FONDAMENTAUX

Dans ce chapitre de présentation des réseaux de neurones artificiels, nous commençons par donner quelques définitions relatives aux réseaux de neurones, ce qu'est un réseau de neurones, ce qu'est un apprentissage de réseaux de neurones et les principales architectures de réseaux de neurones que l'on retrouve dans la littérature, il ne s'agit pas de les étudier toutes car elles sont nombreuses, mais plutôt de comprendre les mécanismes internes fondamentaux et de savoir comment les utiliser.

2.1. Historique [2]

L'origine de l'inspiration des réseaux de neurones artificiels « R.N.A. » remonte à 1890 où W. James, célèbre psychologue américain, introduit le concept de mémoire associative. Il propose ce qui deviendra une loi de fonctionnement pour l'apprentissage des réseaux de neurones, connue plus tard sous le nom de loi de Hebb. Quelques années plus tard, en 1949, J. Mc Culloch et W. Pitts donnent leurs noms à une modélisation du neurone biologique « un neurone au comportement binaire ». Ce sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

C'est ensuite que D. Hebb, physiologiste américain, présente en *1949* les propriétés des neurones par le conditionnement chez l'animal. Ainsi, un conditionnement de type pavlovien tel que, nourrir tous les jours à la même heure un chien, entraîne chez cet animal la sécrétion de salive à cette heure précise même en l'absence de nourriture. La loi de modification des propriétés des connexions entre neurones qu'il propose, explique en partie ce type de résultats expérimentaux. Les premiers succès de cette discipline remontent à *1957*, lorsque F. Rosenblatt développe le modèle du perceptron. Il construit le premier neuro-ordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance des formes. Notons qu'à cette époque les moyens à sa disposition étaient limités et c'était une prouesse technologique que de réussir à faire fonctionner correctement cette machine plus de quelques minutes. C'est alors qu'en *1960*, l'automaticien Widrow développe le modèle Adaline (Adaptative Linear Element). Dans sa structure, le modèle ressemble au perceptron, cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétropropagation de gradient très utilisé aujourd'hui avec les Perceptrons Multi Couches.

Quelques années d'ombre se sont ensuite succédées de *1967* à *1982*. Le renouveau de cette discipline reprend en *1982* grâce à J. J. Hopfield, un physicien reconnu. Au travers d'un article court, clair et bien écrit, il présente une théorie du fonctionnement et des possibilités des réseaux de neurones. Alors que les auteurs s'acharnent jusqu'alors à proposer une structure et une loi d'apprentissage, puis à étudier les propriétés émergentes, J. J. Hopfield fixe préalablement le comportement à atteindre par son modèle et construit, à partir de là la structure et la loi d'apprentissage correspondant au résultat escompté. Ce modèle est aujourd'hui encore très utilisé pour des problèmes d'optimisation.

On peut citer encore la Machine de Boltzmann en *1983* qui était le premier modèle connu, apte à traiter de manière satisfaisante les limitations

recensées dans le cas du perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue « les temps de calcul sont considérables ».

C'est ensuite qu'en 1985 la rétro-propagation de gradient apparaît. C'est un algorithme d'apprentissage adapté au perceptron multi couches. Sa découverte est réalisée par trois groupes de chercheurs indépendants. Dès cette découverte, nous avons la possibilité de réaliser une fonction non linéaire d'entrée/sortie sur un réseau, en décomposant cette fonction en une suite d'étapes linéairement séparables. Enfin, en 1989 Moody et Darken exploitent quelques résultats de l'interpolation multi variables pour proposer le Réseau à Fonctions de base Radiales « *RFR* », connu sous l'appellation anglophone *Radial Basis Function network* « *RBF* ». Ce type de réseau se distingue des autres types de réseaux de neurones par sa représentation locale.

2.2. Application

Les réseaux de neurones servent aujourd'hui à toutes sortes d'applications dans divers domaines. Par exemple, ils étaient utilisés pour développer : un auto-pilote pour avion ou encore un système de guidage pour automobile, des systèmes de lecture automatique de chèques bancaires et d'adresses postales, on produit des systèmes de traitement du signal pour différentes applications militaires, un système pour la synthèse de la parole, des réseaux sont aussi utilisés pour bâtir des systèmes de vision par ordinateur, pour faire des prévisions sur les marchés monétaires, pour évaluer le risque financier ou en assurance, pour différents processus manufacturiers, pour le diagnostic médical, pour l'exploration pétrolière ou gazière, en robotique, en télécommunication. Les réseaux de neurones ont aujourd'hui un impact considérable.

2.3. Qu'est ce qu'un réseau de neurones

L'idée générale des réseaux de neurones artificiels est de trouver une nouvelle procédure de calcul pour produire une nouvelle génération d'ordinateurs. Cette idée inspirée de la biologie du cerveau humain, est différente de la méthode conventionnelle utilisée, actuellement, sur l'ordinateur. Cette nouvelle technique semble plus humaine que mécanique.

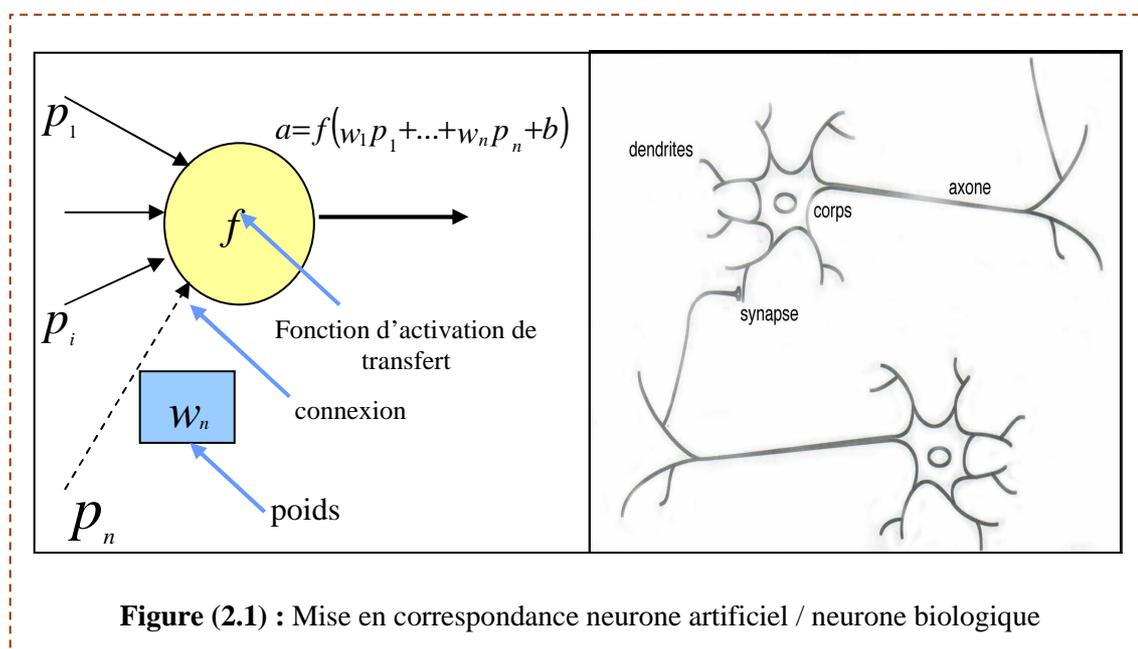
L'homme a toujours rêvé de la machine qui parle, pense, oublie et se rappelle. Le but n'est pas de produire une machine humaine, ce qui est impossible, mais de copier et d'imiter l'approche naturelle du cerveau humain au profit de la science. Un R.N.A est un système de traitement d'information par une nouvelle génération d'ordinateurs, inspirée par les systèmes biologique. Un R.N.A. peut être considéré comme une boîte noire, qui reçoit des signaux d'entrée et produit des signaux de sortie. C'est un modèle mathématique composé d'un grand nombre d'éléments de calculs organisés sous formes de couches interconnectées [6].

Un neurone est une fonction algébrique non linéaire, paramétrée, à valeurs bornées, de variables réelles appelées entrées.

2.4. Modèle mathématique

Les réseaux de neurones biologiques réalisent facilement un certain nombre d'applications telles que la reconnaissance des formes, le traitement du signal, l'apprentissage par l'exemple, la mémorisation et la généralisation. 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.

La figure (2.1) 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 amont. A chacune de ces entrées est associée un poids w , abréviation de *weight* « poids », 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 aval. A chaque connexion est associée un poids.



2.5. Techniques d'apprentissages

Les procédures d'apprentissage peuvent se subdiviser, en deux grandes catégories : apprentissage supervisé ou apprentissage non supervisé.

2.5.1. L'apprentissage supervisé

Implique l'existence d'un « professeur » qui a pour rôle d'évaluer le succès « ou l'échec » du réseau quand il lui est présenté un stimulus connu « on dit que ce stimulus est un exemple appartenant à la base d'apprentissage ». Cette supervision consiste à renvoyer au réseau une information lui permettant de faire

évoluer ses connexions « parfois aussi sa propre architecture » afin de faire diminuer son taux d'échec [7].

L'information peut être explicite, sous la forme d'une mesure de l'erreur commise, exemple par exemple, ou globalement sur l'ensemble des exemples de la base. C'est ce qui se passe par exemple dans le cas du perceptron ou de la machine de Boltzmann. Mais elle peut être plus implicite « apprentissage par renforcement », sous forme d'une simple appréciation « bon » ou « mauvais », « punition » ou « récompense », sans mesure d'erreur, et même être globale, sur l'ensemble des tâches que le réseau doit exécuter.

Dans ce dernier cas, la difficulté majeure consistant pour le réseau à identifier les étapes du processus qui sont responsables de l'échec ou du succès « en anglais, *credit assignment problem* ».

2.5.2. L'apprentissage non supervisé

Implique la fourniture à un réseau autonome d'une quantité suffisante d'exemples contenant des corrélations « autrement dit de la redondance ». Ces réseaux sont souvent appelés « auto-organiseurs » ou encore « à apprentissage compétitif ». Bien entendu, l'architecture du réseau, préalablement définie par son utilisateur, est une forme de supervision. La tâche du réseau peut être par exemple dans ce cas, sans réponse désirée, de créer regroupement de données selon des propriétés communes « catégorisation ».

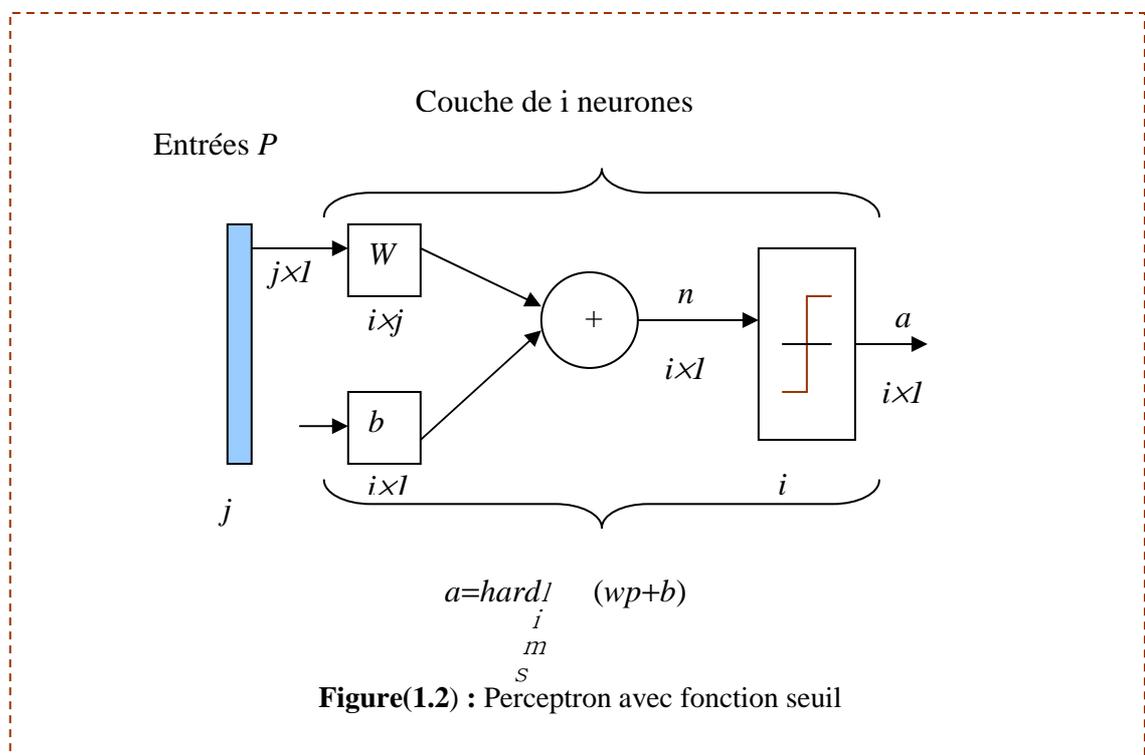
2.6. Fonctions de transfert

Différentes fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone, les trois les plus utilisées sont les fonctions *seuil* « en anglais -hard limit- », *linéaire* et *sigmoïde*.

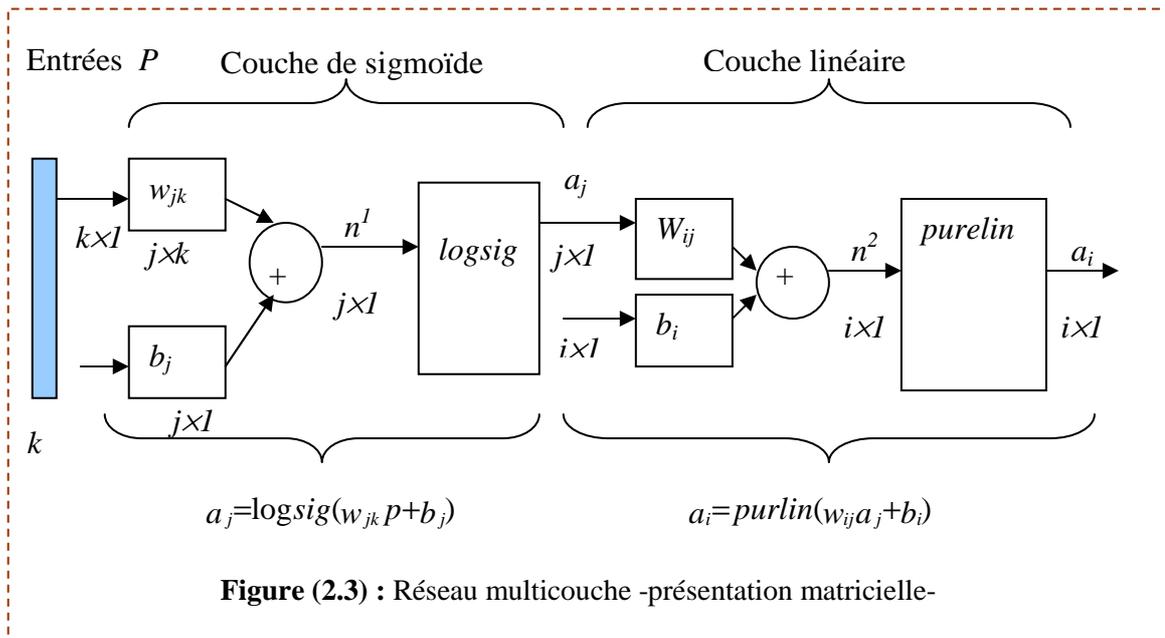
2.7. Quelques architectures neuronales

2.7.1. Perceptron Multi Couches «PMC»

Le perceptron est un réseau à deux couches, une couche d'entrée et une couche de sortie de type *feedforward* « propagation avant ». Les neurones de la couche d'entrée ont pour rôle de fournir au réseau les données externes. Chaque neurone de la couche de sortie effectue une somme pondérée de ses entrées.



La capacité du perceptron a été jugée limitée à la décision linéaire et aux fonctions logiques simples. Par contre en mettant plus d'un perceptron en cascade «PMC», le réseau de neurones peut résoudre beaucoup de problèmes tel que la capacité d'implémenter les fonctions non linéaires et la capacité de classifier les objets, à condition d'employer une fonction de transfert non linéaire et de disposer de suffisamment de neurones sur les couches cachées.



Règle Delta

Contrairement au modèle de Rosenblatt où les neurones ont des fonctions d'activation à seuils, Widrow et Hoff en 1960 ont proposé un modèle de perceptron avec neurones linéaires « f : fonction d'activation linéaire » :

$$f(a_i) = a_i \tag{2.1}$$

L'avantage d'utiliser des unités linéaires est qu'elles permettent de calculer une fonction de coût qui évalue l'erreur que commet le réseau. Cette erreur peut être définie en fonction des erreurs entre réponses désirées et réponses obtenues par le réseau. Cette erreur est donc fonction des poids du réseau :

$$E(w) = \frac{1}{2} \sum_i (p_i - a_i)^2 = \frac{1}{2} \sum_i (p_i - \sum_k w_{ik} p_k)^2 \tag{2.2}$$

$$\Delta w_{ik} = -\eta \frac{\partial E}{\partial w_{ik}} = \eta \sum (p_i - a_i) p_k \tag{2.3}$$

avec ;

$$w_{ik}^{\text{nouveau}} = w_{ik}^{\text{ancien}} + \Delta w_{ik} \tag{2.4}$$

Où $\Delta_{w_{ik}}$ est la quantité ajoutée au poids w_{ik} pour chaque exemple de l'ensemble des exemples d'apprentissage et le paramètre η est appelé taux d'apprentissage.

Rétropropagation

La rétropropagation est le paradigme des RNA le plus utilisé. La rétropropagation est basée sur des principes mathématiques très solides, cette méthode a donné des bons résultats dans beaucoup d'apprentissage; Pour l'appliquer il suffit d'avoir des données d'entrées et de données de sortie «cible». La condition d'interconnexion complète des couches n'est pas une obligation, car certaines applications sont réalisées avec des formes d'interconnexions incomplètes. L'algorithme de rétropropagation utilise une étape de propagation en avant «chaîne directe», suivie d'une étape de rétropropagation «chaîne inverse».

- *Propagation avant:*
 - a. Choix de la paire de l'ensemble d'entraînement et l'application du vecteur d'entrée au RNA
 - b. Calcul de la sortie du RNA; $a=f(p \times w)$
- *Propagation dans le sens inverse:*
 - c. Calcul de l'erreur entre la sortie et la cible
 - d. Ajustement des poids pour permettre de minimiser l'erreur totale
 - e. Répétition des étapes a, b, c et d pour chaque vecteur dans l'ensemble d'apprentissage jusqu'à ce que l'erreur totale soit acceptable.

2.7.2. Le réseau de Hopfield

Le modèle de Hopfield est basé sur le concept de mémoire adressée par le contenu : *Mémoire Associative*. A partir d'un réseau entièrement connecté, utilisant le plus souvent la règle d'apprentissage Hebb, avec un apprentissage supervisé. Chaque information mémorisée représente un point stable de l'espace d'état vers lequel l'évolution du système aboutit à partir d'un point initial voisin correspondant à une version déformée de l'information mémorisée. Autrement

dit, l'espace d'état du système comporte des attracteurs qui correspondent aux informations mémorisées.

L'architecture du réseau est telle que chaque neurone est connecté à tous les autres sauf à lui-même. L'architecture du réseau de Hopfield est symétrique.

Le problème consiste à déterminer, s'il existe, l'ensemble des poids w_{ij} qui permettent au réseau de se comporter comme une mémoire adressée par le contenu. La condition irrévocable de convergence du réseau de Hopfield vers un état stable est que la matrice des poids des connexions w soit symétrique et de diagonale égale à zéro. La particularité du réseau de Hopfield est que son évolution vers un état stable est caractérisée par une fonction appelée *fonction d'énergie*. Le réseau de Hopfield converge vers un des états stables « états attracteurs » correspondant à un minimum local de la fonction d'énergie. Pour définir les états stables à mémoriser par le réseau, il faut donc trouver la matrice des poids w qui minimise la fonction d'énergie [8].

2.7.3. Le réseau de Kohonen [6]

Treuvo Kohonen s'est inspiré de l'apprentissage biologique pour établir sa règle, utilisé uniquement dans l'apprentissage non supervisé. Dans cette règle les éléments de traitement entrent en compétition pour effectuer l'apprentissage. L'élément de traitement de plus grande valeur de sortie est déclaré vainqueur et a la possibilité d'inhiber, éliminer, ses opposants compétiteurs et en même exciter ses voisins ; Seul, le vainqueur est toléré à donner une sortie et le vainqueur ainsi ses voisins sont les seuls autorisés à ajuster leurs poids.

La taille du voisinage peut changer durant l'apprentissage. Il est préférable de démarrer avec une large voisinage et diminuer sa taille eu fur et à mesure que l'apprentissage se déroule. Le RNA s'auto-organise et s'établi et apprend. Ses neurones sont complètement connectés. Ce RNA converge rapidement.

2.7.4. Le réseau RBF

Le réseau RBF est un réseau de neurones supervisé. Il s'agit d'une *spécialisation* d'un PMC. Le modèle RBF "**Radial Basis Function**" est constitué uniquement de trois couches :

- La couche d'entrée : elle retransmet les inputs sans distorsion.
- La couche RBF : couche cachée qui contient les neurones RBF.
- La couche de sortie : simple couche qui contient une fonction linéaire.

Chaque couche est « *fully connected* » à la suivante. Chaque neurone RBF contient une *gaussienne* qui est centrée sur un point de l'espace d'entrée. Pour une entrée donnée, la sortie du neurone RBF est la *hauteur* de la gaussienne en ce point. La fonction gaussienne permet aux neurones de ne répondre qu'à une petite *région* de l'espace d'entrée, région sur laquelle la gaussienne est centrée. La sortie du réseau est simplement une *combinaison linéaire* des sorties des neurones RBF multipliés par le poids de leur connexion respective. Il y a quatre paramètres principaux à régler dans un réseau RBF :

- Le *nombre* de neurones RBF, nombre de neurones dans l'unique couche cachée.
- La *position* des centres des gaussiennes de chacun des neurones.
- La *largeur* de ces gaussiennes.
- Le *poids* des connexions entre les neurones RBF et le(s) neurone(s) de sortie.

Tout modification d'un de ces paramètres entraîne directement un changement du comportement du réseau.

2.8. Propriétés des réseaux de neurones artificiels

2.8.1. Apprentissage et mémoire

L'une des caractéristiques les plus complexes du fonctionnement de notre cerveau est bien la phase d'apprentissage. C'est une phase au bout de laquelle certaines modifications s'opèrent entre les connexions des neurones : certaines

sont renforcées et d'autres affaiblies ou carrément inhibitrices. Le cerveau converge alors vers un comportement souhaité.

Ceci nous emmène à la notion de mémoire qui donne au cerveau la capacité de retrouver des expériences passées. Le cerveau possède plusieurs types de mémoires et ce que nous pouvons retenir c'est que le cerveau humain procède par association. Cela permet par exemple de retrouver une information à partir d'éléments incomplets ou imprécis « bruités ». Par exemple, le fait de voir un bout d'une photographie qu'on connaît déjà est suffisant pour que notre cerveau soit capable de la reconnaître. Cette caractéristique des réseaux de neurones artificiels est connue comme *capacité de généralisation*.

Le mécanisme de l'association permet aussi au cerveau de converger vers un état à partir d'un autre état. Par exemple, le fait de passer devant une boulangerie nous fait rappeler qu'on devait acheter du pain. Cette deuxième importante caractéristique est aussi connue sous le nom de mémoire adressée par le contenu, dont le modèle de Hopfield s'en inspire. Par analogie avec les réseaux de neurones biologiques, les réseaux de neurones artificiels tentent de reproduire les caractéristiques les plus importantes du comportement biologique, à savoir *l'apprentissage, la généralisation et l'association*.

2.8.2. Sous-apprentissage, généralisation et sur-apprentissage

La capacité de généralisation est l'une des raisons qui motivent l'étude et le développement des réseaux de neurones artificiels. Elle peut être définie par la capacité d'élargir les connaissances acquises après apprentissage à des données nouvellement rencontrées par le réseau de neurones. On peut distinguer deux types de généralisation : *locale* et *globale*.

Dans l'approche *locale*, chaque neurone est associé à une région d'activation « région d'influence » localisée dans l'espace des données. Seule

une partie des neurones participe donc à la réponse du réseau. Deux types d'architectures neuronales possèdent cette particularité : les réseaux à base de fonctions radiales et la carte de Kohonen. Par contre, dans l'approche *globale*, l'ensemble des neurones du réseau participe à l'élaboration de la sortie du réseau.

L'information est donc distribuée dans le réseau tout entier. C'est le cas des réseaux de neurones de type perceptron multi couches ou le modèle de Hopfield.

L'approche globale est supposée plus robuste aux *pannes* éventuelles de quelques neurones isolés. En d'autres termes, tandis que le réseau apprend dans une région de l'espace des données, le modèle peut *oublier* ce qu'il a appris dans d'autres régions.

Cette capacité de généralisation est très liée à la notion de sur-apprentissage. Ces deux caractéristiques sont complètement antagonistes. On parle de sur-apprentissage quand le réseau a trop parfaitement appris les exemples proposés. Il sera donc incapable de généraliser. Ceci est appelé calcul de la *complexité* du réseau de neurones. En pratique, on effectue un apprentissage sur un sous ensemble S de l'espace de données D . Le réseau est alors testé sur un ensemble de test T ne faisant pas partie de l'apprentissage. On calcule alors la moyenne des erreurs quadratiques sur l'ensemble S appelée '*erreur base apprentissage*' et sur l'ensemble de test T appelée '*erreur base test*'. Plus on agrandit l'ensemble S , plus l'*erreur base apprentissage* diminue, plus l'*erreur base test* augmente. Le réseau perd dans ce cas là ses capacités de généralisation.

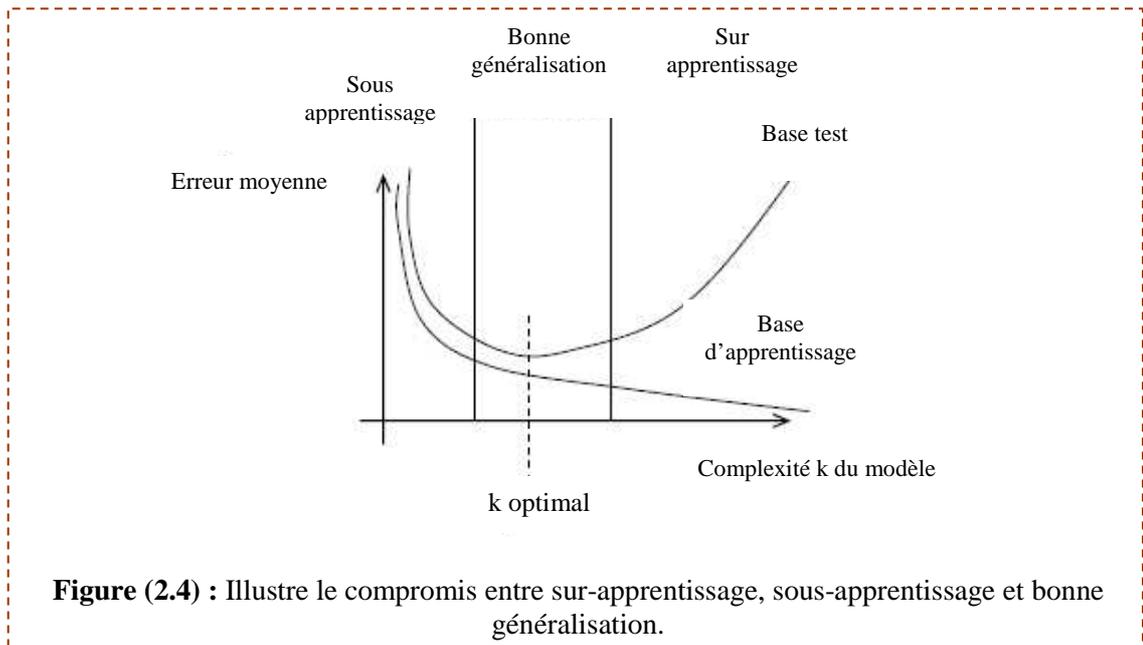


Figure (2.4) : Illustre le compromis entre sur-apprentissage, sous-apprentissage et bonne généralisation.

Conclusion

En raison des propriétés fondamentales que nous venons de mentionner, les réseaux de neurones artificiels sont capables d'intervenir dans la résolution de nombreux problèmes de classification, de contrôle ...

Chapitre 3

CONTROLEURS NEURONAUX UTILISES

CONTROLEURS NEURONAUX UTILISES

La commande automatique d'un processus consiste à développer les techniques de contrôle à lui appliquer, de manière à lui assurer un comportement dynamique spécifié par un cahier de charges. Ces techniques sont délivrées par un organe de commande comprenant un correcteur ou un régulateur. Le processus et son organe de commande constituent le système de commande. Dans ce contexte l'utilisation de réseaux de neurones artificiels « RNA » pour la réalisation de système de commande a connu un essor important au cours des dernières années. Dans ce chapitre nous allons présenter un état d'art non exhaustif des techniques de contrôle basées sur les réseaux de neurones ainsi que les modalités de leur utilisation, dans ce domaine.

3.1. Motivation pour l'emploi des réseaux de neurones

Dans le monde réel un grand nombre de processus sont caractérisés par un comportement dynamique non linéaire complexe, et rendent impossible l'utilisation des outils classiques de l'automatique. Il n'existe pas aujourd'hui de théorie systématique et applicable de manière générale à la commande de tels processus.

Pour résoudre ce problème une des solutions proposées consiste à avoir recours à une phase d'apprentissage pour identifier le modèle du processus ou le contrôleur. Le terme « apprentissage » désigne le fait de modifier la structure

et/ou les paramètres du système, de manière à améliorer ses performances futures.

Sachant que les réseaux de neurones artificiels sont principalement des procédures permettant d'approcher n'importe quelle fonction linéaire ou non [8]. C'est cette propriété qui motive leur utilisation pour la réalisation de systèmes de commande non linéaires par apprentissage. La conception de l'organe de commande est précédée d'une phase de modélisation du processus.

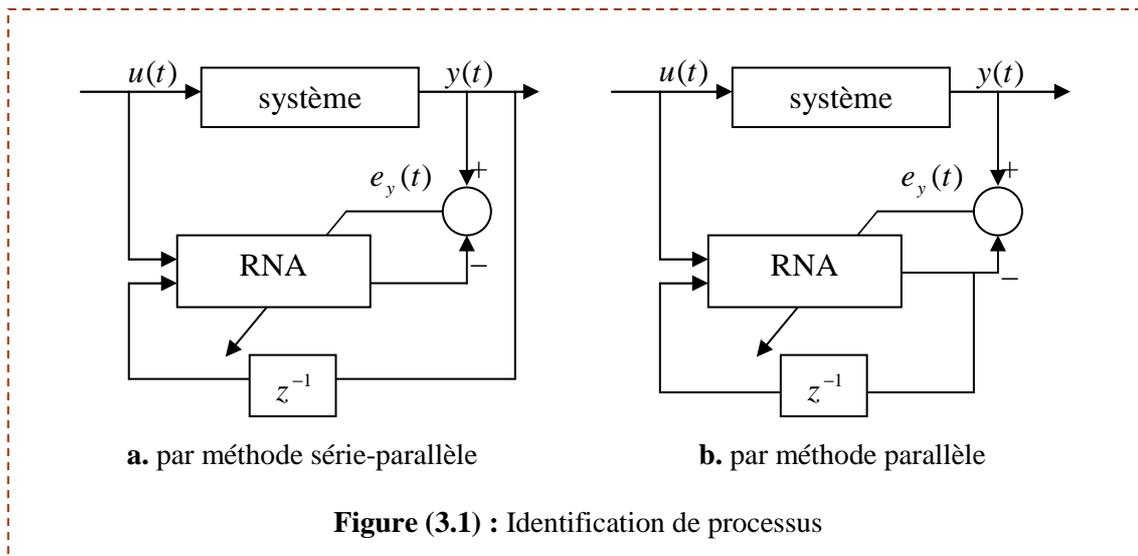
Nous entamons cette section par une présentation des méthodes d'identification de processus utilisant des RNA ensuite quelques méthodes de commande.

3.2. Identification de processus

Il s'agit d'une première phase nécessaire dans certaines approches, par identification d'un processus nous entendons l'entraînement d'un réseau, à reproduire une fonction donnant les sorties ou l'état du processus à partir des entrées qui lui sont appliquées.

Le principe général de l'identification est simple et consiste à placer en parallèle le réseau et le processus à identifier, Le réseau reçoit en entrée la commande $u(t)$ appliquée et éventuellement la sortie $y(t-1)$ précédente du processus. Il est entraîné à produire la nouvelle sortie « ou le nouvel état » $y(t)$ du processus.

Cette méthode d'identification est souvent appelée méthode série-parallèle, par opposition à la méthode parallèle, comme indiqué sur la figure (3.1). L'approche série-parallèle est souvent considérée comme plus stable car le réseau est régulièrement recalé en utilisant l'état réel du processus.



3.3. Classification des méthodes de commande [6][8][9]

Dans cette partie nous présentons une classification des principales méthodes de commande neuronales. La plupart d'entre elles sont indépendantes du type de réseau utilisé et du type de processus commande.

La grande majorité des architectures rencontrées utilise des réseaux multicouches entraînés par rétropropagation du gradient, mais d'autres modèles sont possibles. Généralement ces approches sont séparées en deux classes principales, méthodes de commande indirectes et méthodes de commande directes. Notre travail sera basé sur la deuxième.

3.4. Méthodes directes [6][8][9]

3.4.1. Reproduction d'un contrôleur existant

La première méthode utilisée pour la réalisation d'un système de commande neuronal consiste à reproduire le fonctionnement d'un contrôleur existant. Même si cette approche semble au premier lieu peu intéressante puisqu'elle nécessite l'existence d'un autre contrôleur. Elle peut s'avérer utile si ce

dernier est trop complexe ou trop lent pour être utilisé en temps réel, ou encore s'il utilise des données qui ne sont pas disponibles en permanence [6].

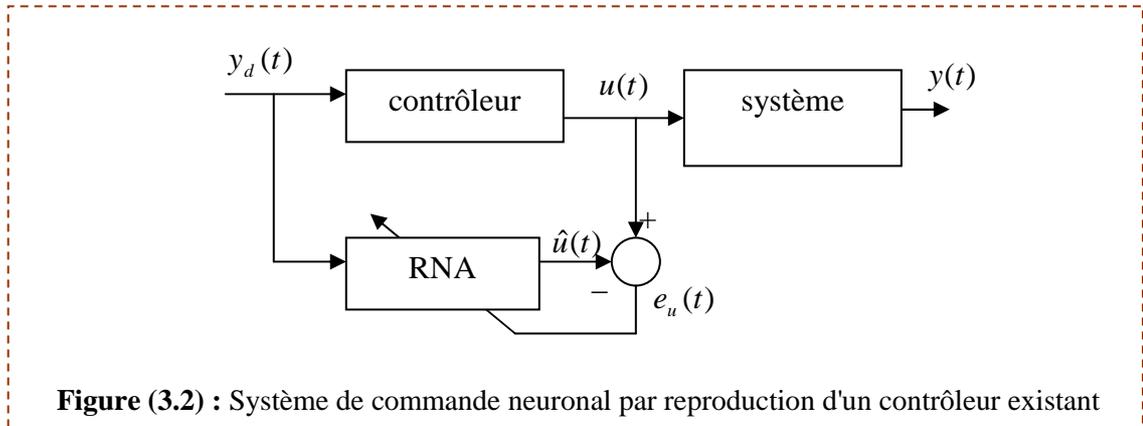


Figure (3.2) : Système de commande neuronal par reproduction d'un contrôleur existant

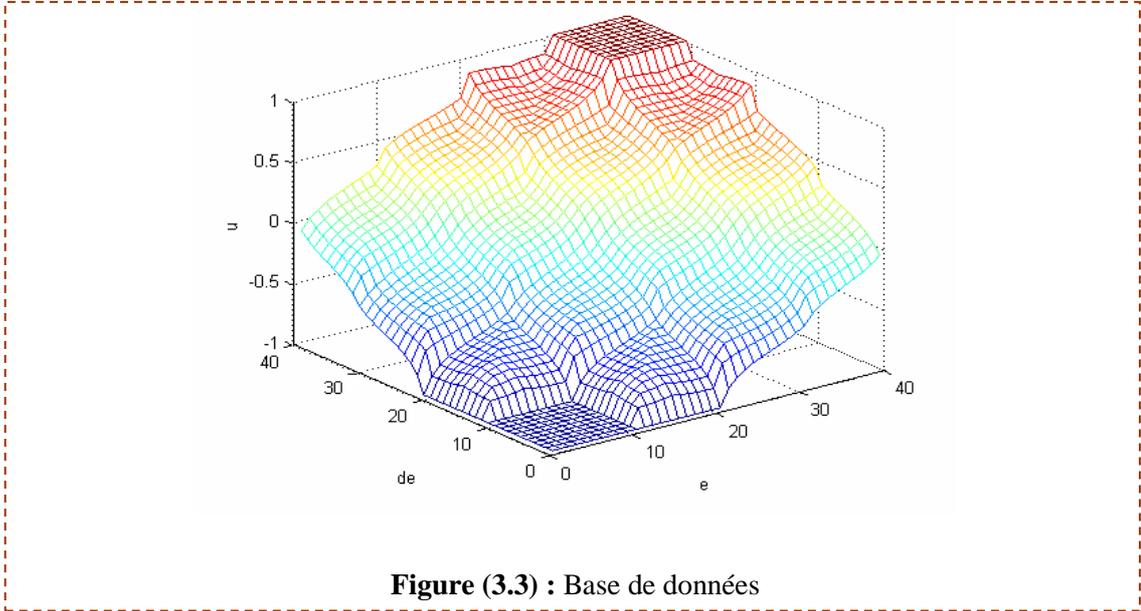
La figure ci-dessous représente l'architecture générale de cette méthode qui consiste à apprendre au réseau à reproduire la commande $\hat{u}(t)$ préconisée par le premier contrôleur à partir de la sortie désirée $y_d(t)$ et éventuellement de la sortie précédente.

Un des premiers exemples de système de commande neuronal, proposée par B. Widrow et F. W. Smith en 1964, utilise cette technique pour résoudre le problème de la commande d'un pendule inversée. Nous pouvons noter que cette approche demande de parcourir, lors de la phase d'apprentissage, tous les modes de fonctionnement du système commandé et il est nécessaire de posséder une bonne connaissance a priori des conditions d'utilisation du contrôleur.

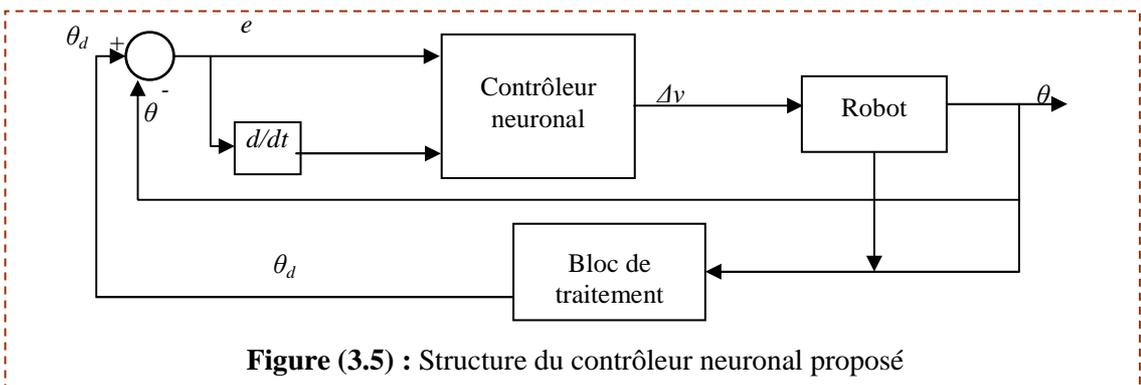
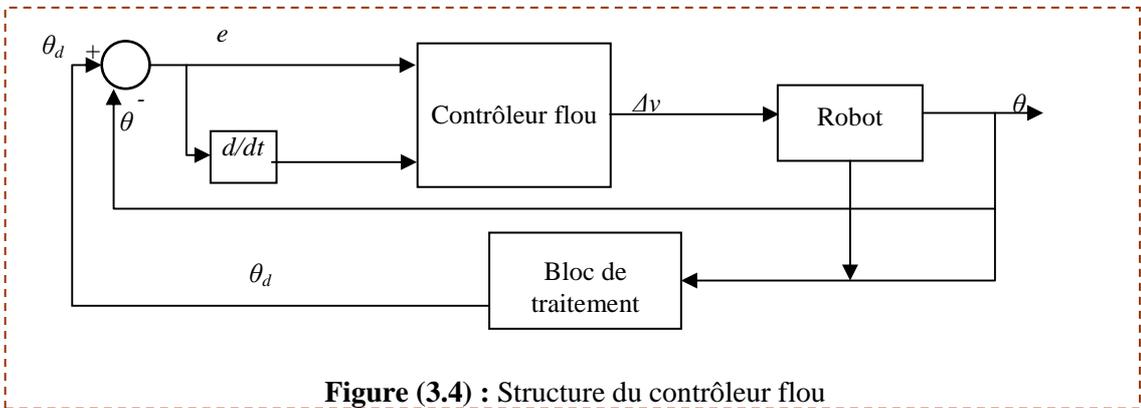
Cette technique sera utilisée dans notre travail comme première partie. La base de données utilisée est extraite d'un travail déjà effectué a sein du laboratoire « L.E.A. », Laboratoire de l'Electronique Avancée. Il s'agit d'une commande d'un robot mobile utilisant un contrôleur logique flou.

La figure suivante montre cette base de données, tel que les deux variables « e » et « de » représentent les vecteurs d'entrées et « u » le vecteur de sortie,

sachant que « e » c'est l'erreur de l'angle d'orientation, « de » est sa dérivée et « $u = \Delta v$ » est la commande du robot mobile.



Nous proposons dans cette partie de remplacer le contrôleur flou par un contrôleur neuronal, comme le montre les figures au dessous :

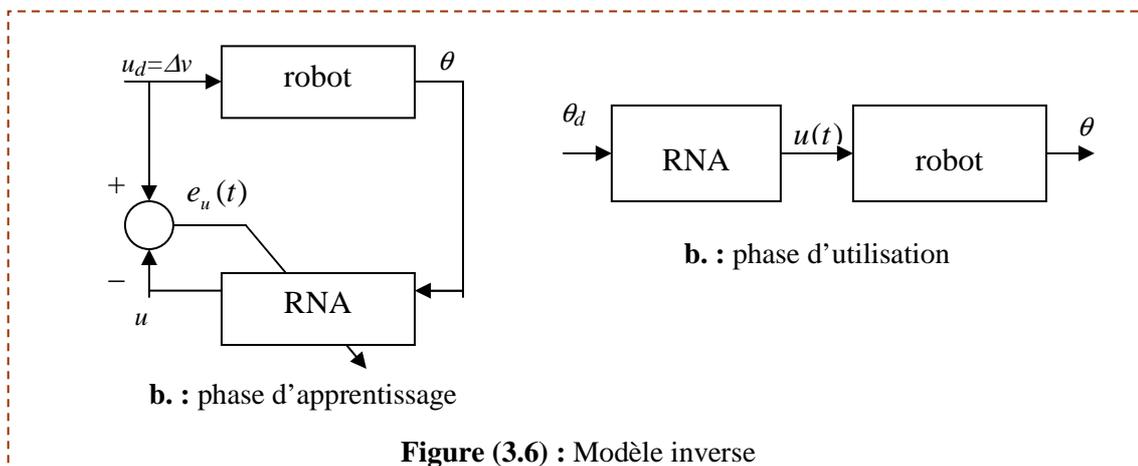


Cette base de données sera séparée en deux ensembles, le premier sert à l'apprentissage et le deuxième pour le test. Les résultats de cette partie seront exposés dans le dernier chapitre.

3.4.2. Approche modèle inverse

Nous utiliserons cette approche comme deuxième technique dans notre travail, elle nécessite deux phases séparées, une pour l'apprentissage puis l'autre pour l'utilisation du réseau. Durant la phase d'apprentissage le réseau et le processus sont placés en parallèle, une commande u est fournie au processus. De façon les sorties θ du processus seront considérées comme entrées du réseau qui est entraîné de façon à retrouver en sortie les commandes u . Le réseau apprend ainsi un modèle inverse du processus, c'est à dire une fonction donnant la commande appliquée $u(t)$ à partir de la sortie actuelle $\theta(t)$ et éventuellement de sa sortie passée $\theta(t-1)$.

Après la phase d'apprentissage le réseau est donc théoriquement capable de fournir la commande $u(t)$ nécessaire pour obtenir une sortie $\theta_d(t)$ fournie en entrée. Il est donc placé directement en série avec le système commandé, comme le montre la figure (3.6.b). Lors de la phase d'apprentissage, il est nécessaire de faire parcourir au processus l'ensemble de ses états possibles, ou tout au moins l'ensemble des états qui seront utilisés durant la phase de commande.



La structure générale de notre système est illustrée par figure suivante :

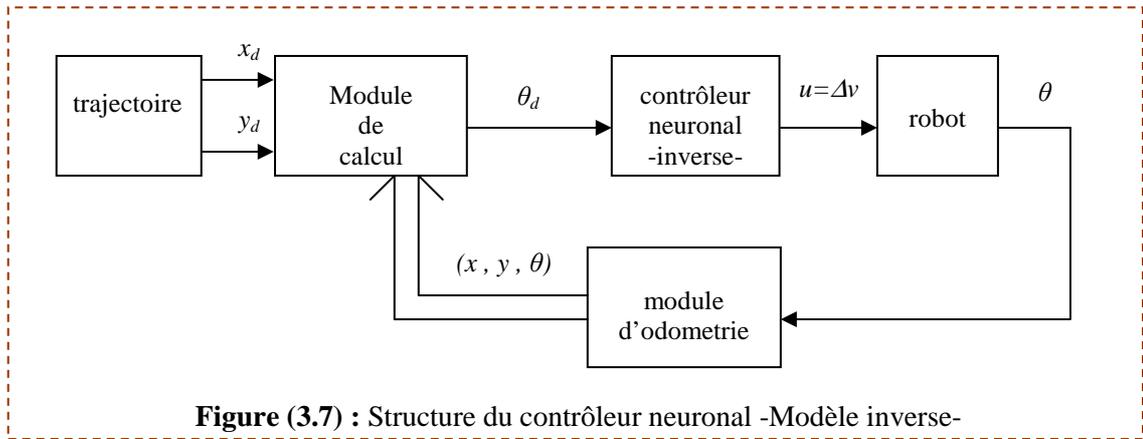


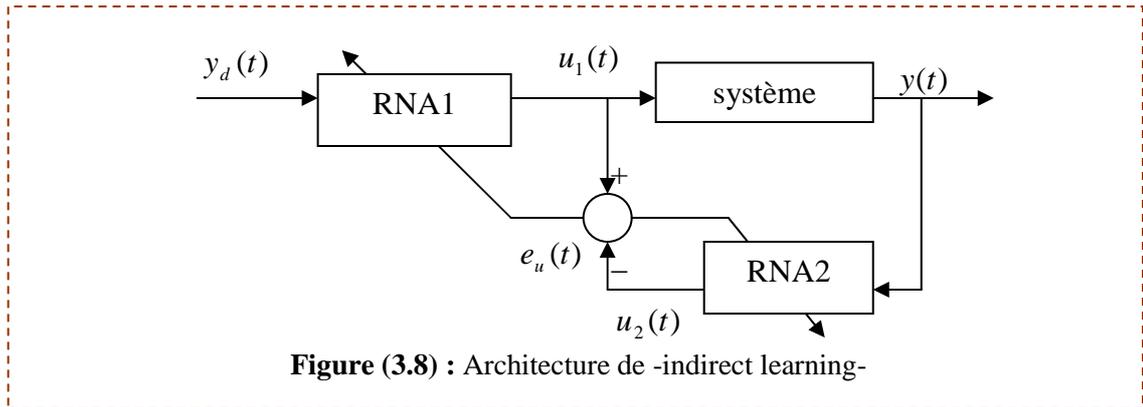
Figure (3.7) : Structure du contrôleur neuronal -Modèle inverse-

L'application ainsi que la validation de cette méthode seront exposées dans le chapitre qui suit.

Les travaux de Y. Kuroe et al. utilisent cette approche pour la commande d'un bras robotique. D. Psaltis et al. proposent une variante de cette architecture, appelée *Indirect Learning Architecture*, illustrée par la figure (3.8).

Dans cette architecture les deux réseaux partagent les mêmes poids c'est à dire le deuxième réseau est une copie du premier. La valeur de sortie désirée $y_d(t)$ est propagée à travers le premier réseau pour produire une commande $\hat{u}(t)$ qui est appliquée au système. La valeur de sortie y mesurée est alors utilisée comme entrée du deuxième réseau pour obtenir une deuxième consigne, $e_u(t)$ est utilisée pour corriger les réseaux.

L'idée exploitée par cette architecture est que si l'erreur tend vers zéro, la différence $y - y_d$ tend également vers zéro. On constate cependant que le système converge souvent vers une solution triviale donnant toujours la même commande, ce qui rend l'architecture peu utilisable.



3.4.3. Amélioration des régulateurs conventionnels

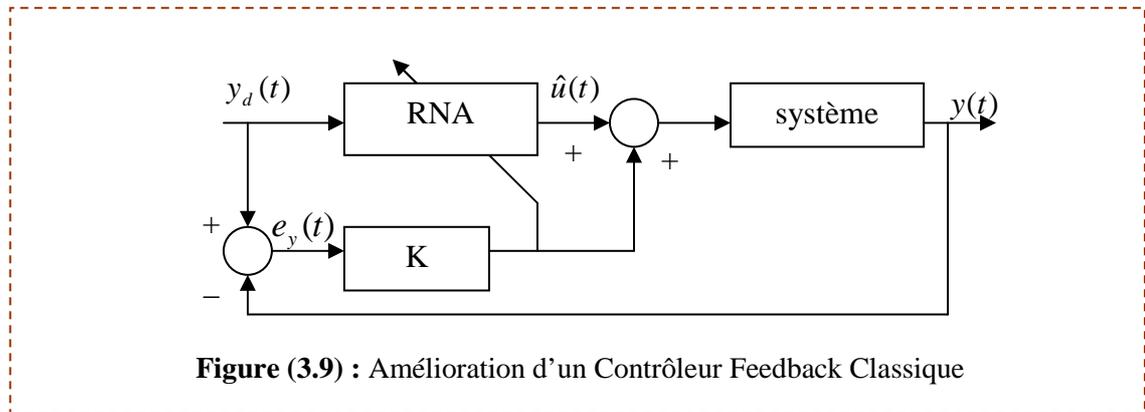
a. Amélioration d'un système linéaire

L'idée principale de cette approche est d'utiliser conjointement un contrôleur linéaire classique et un contrôleur neuronal et de réaliser une somme des commandes issues des deux contrôleurs, en augmentant progressivement l'importance donnée à la commande $\hat{u}(t)$ préconisée par le réseau, au fur et à mesure de l'apprentissage de ce dernier.

M. Kawato propose d'utiliser un *Contrôleur de type Feedback Classique* « CFC ». Dans sa version la plus simple un tel contrôleur se contente d'appliquer un gain sur l'erreur y mesurée en sortie du processus. La commande est donc proportionnelle à cette erreur. Le réseau reçoit en entrée la sortie désirée $y_d(t)$ pour le processus commandé, et il utilise comme signal d'erreur la sortie du CFC.

Lorsque l'apprentissage avance, le réseau apprend donc à minimiser par ses commandes la sortie du CFC et par conséquent l'erreur en sortie du processus. La sortie du réseau prend par la même occasion une part de plus en plus importante dans la commande. On trouve parfois pour cette architecture le nom de *feedback error learning*.

H. Miyamoto utilise cette approche pour la commande d'un bras robotique.
H. Ohno l'utilise pour la commande du système de freinage sur une automobile.
La figure suivante décrit cette architecture.



b. Commande avec un contrôleur PD neuronal

Le régulateur Proportionnel Dérivateur PD est la combinaison de deux modules, le module P « proportionnel » qui assure la fonction de réglage de base et le module D « dérivateur » qui améliore la stabilité et accélère le réglage. Ce régulateur fournit un signal de commande proportionnel à l'écart de réglage et à sa dérivée.

En exploitant la faculté d'apprentissage des réseaux de neurones, nous élaborerons un système permettant d'estimer ces deux paramètres.

Auto ajustement des paramètres d'un PD

Dans cette approche le réseau de neurone sera utilisé pour ajuster les paramètres du contrôleur conventionnel PD de la même manière que lorsqu'ils sont réglés par un opérateur humain. La structure de principe permettant d'estimer ces paramètres est détaillée sur la figure suivante.

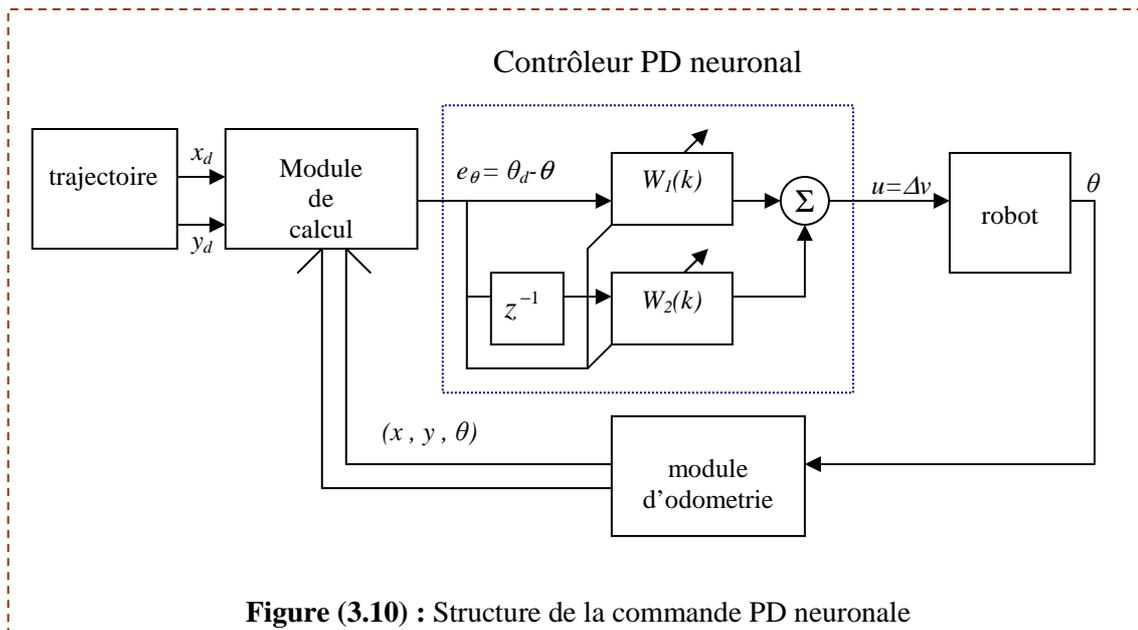


Figure (3.10) : Structure de la commande PD neuronale

Les gains K_p et K_d , gains proportionnel et dérivé sont déterminés en temps réel par le réseau de neurones. Cette approche est l'application directe des techniques traditionnelles de commande incluant une méthode de commande adaptative.

Le vecteur d'entrée du réseau possède deux composantes : l'erreur et sa dérivée. Le poids $w_1(k)$ pondérant l'entrée erreur sera associé au facteur P et le poids $w_2(k)$ pondérant l'entrée de la dérivée de l'erreur sera quand à lui associé au facteur D .

L'erreur d'apprentissage sera $e = \theta_d - \theta$ et l'algorithme de mise à jour des poids est celui de Widrow-Hoff. L'apprentissage sera arrêté lorsque le système arrive à suivre le chemin prévu selon les critères fixés par l'utilisateur. Le réseau se comporte comme un PD non adaptatif. Si des variations importantes surviennent au niveau du système à commander, l'apprentissage peut reprendre.

Le dernier chapitre contient l'application et la validation de cette approche.

3.4.4. Utilisation directe de l'erreur en sortie du procédé

Cette approche consiste à d'utiliser directement l'erreur $e_y(t)$ pour adapter le contrôleur, cette méthode consiste à considérer le processus comme une couche supplémentaire du réseau à travers laquelle on rétropropage l'erreur, la figure (3.11) illustre cette architecture.

Pour pouvoir utiliser cette technique, parfois appelée *apprentissage spécialisé*, il est nécessaire de connaître le *Jacobien* du processus, c'est à dire la quantité $\partial y / \partial u$, il est alors possible d'utiliser l'algorithme de rétropropagation du gradient en considérant que le processus est une extension fixe du réseau.

La règle de calcul du gradient de l'erreur par rapport aux poids w est :

$$\frac{\partial e_y}{\partial w} = \frac{\partial e_y}{\partial y} \frac{\partial y}{\partial u} \frac{\partial u}{\partial w} \quad (3.1)$$

$\frac{\partial y}{\partial u}$: ce terme est le Jacobien

$\frac{\partial u}{\partial w}$: ce terme est obtenu par l'algorithme de rétropropagation du gradient.

Y. Zhang utilise cette technique pour la commande en direction d'un navire.

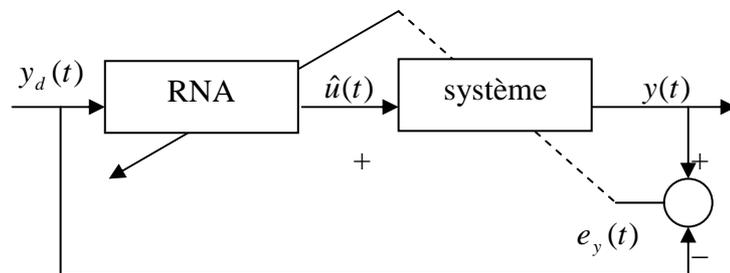


Figure (3.11) : Apprentissage spécialisé

3.5. Méthodes indirectes [6][8][9]

3.5.1 Apprentissage indirect du système de commande

Cette architecture de commande utilise un modèle du processus commandé réalisé par un premier RNA, comme indiquée dans la section précédente « modèle différentiable du processus ».

Le principe de l'architecture est d'utiliser l'algorithme de rétropropagation du gradient en considérant que le système de commande et le modèle ne forment qu'un seul réseau. L'erreur est propagée à travers le modèle sans modifier ses poids, et seul le réseau donnant la commande subit un apprentissage, la figure (3.12) décrit cette technique.

Cette architecture est utilisée par K. Ishii pour la commande d'un robot sous-marin et elle est appliquée par Nguyen et Widrow au problème connu du parking parallèle d'un camion tractant une remorque.

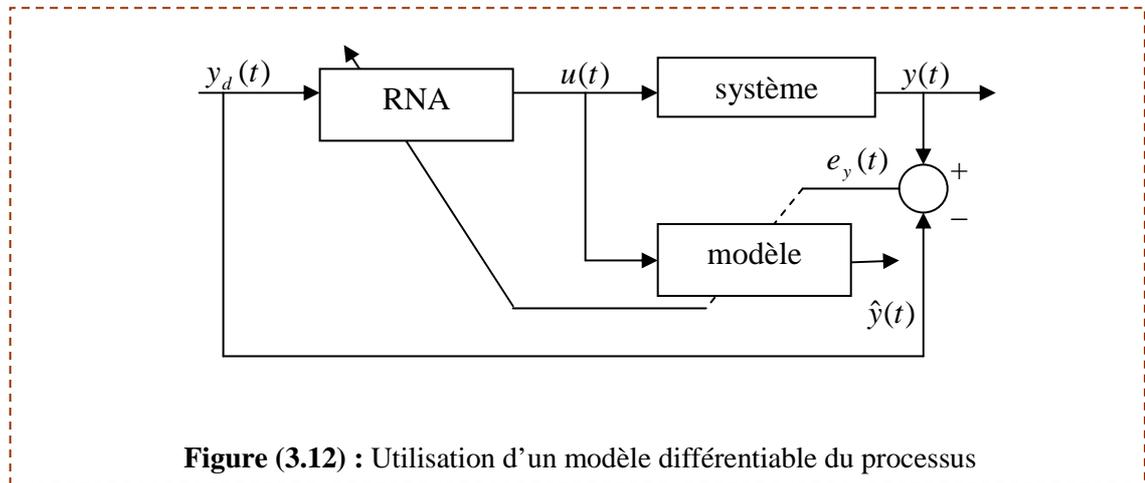


Figure (3.12) : Utilisation d'un modèle différentiable du processus

3.5.2. Réalisation d'un système de commande par modèle prédictif

Les architectures de commande présentées ci-dessus ont toutes pour principe de tenter d'obtenir par apprentissage un modèle neuronal inverse du

fonctionnement du processus. Que ce soit par une méthode directe ou par une méthode indirecte, le but est en effet toujours de construire un système donnant la commande à partir de l'état ou de la sortie actuel et de l'état ou de la sortie désiré.

Le principe mis en œuvre par la commande par modèle prédictif est totalement différent. Les réseaux sont ici utilisés pour construire un modèle direct du processus, comme décrit dans la section (3.2), afin de prédire les états futurs de ce dernier.

Le système de commande consiste alors à utiliser un algorithme d'optimisation numérique non linéaire afin de minimiser une fonction de coût dépendant des états futurs prédits, la figure (3.13) illustre cette architecture.

H. Fritz utilise une méthode similaire, également pour la commande de véhicules autonomes.

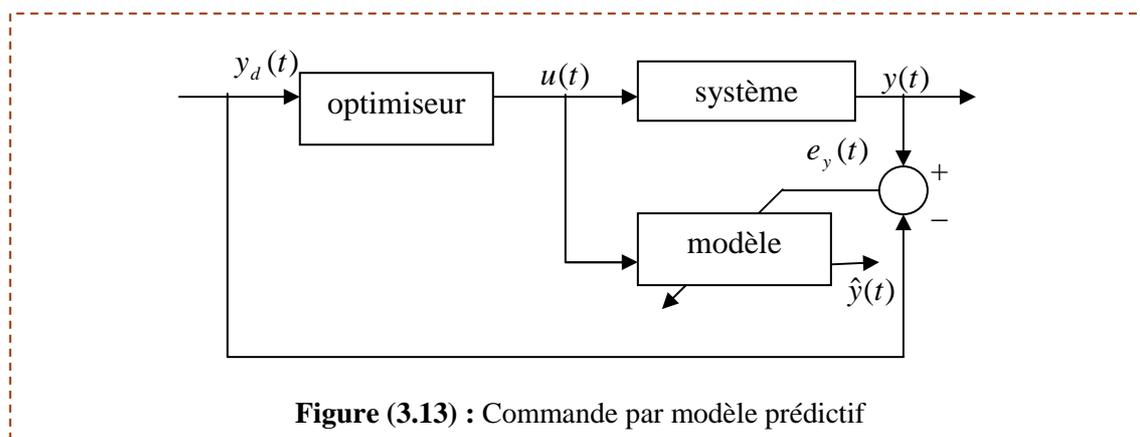


Figure (3.13) : Commande par modèle prédictif

H. Fritz utilise une méthode similaire, également pour la commande de véhicules autonomes.

Conclusion

Dans ce chapitre nous avons essayé de présenter les différentes techniques de commande des robots mobiles utilisant les réseaux de neurones artificiels,

sachant que certaines architectures sont absentes dans cette classification. Les trois premières approches citées précédemment seront validées dans le dernier chapitre.

Nous concluons ce chapitre par l'avis de C. Torras : *la planification et le contrôle des mouvements impliquent deux types différents de traitement. D'une part la planification globale implique un traitement symbolique séquentiel, et d'autre part les méthodes visant à éviter localement des obstacles impliquent un traitement distribué de manière fortement parallèle, par exemple R.N.A.*

Chapitre 4
TESTS ET VALIDATION

Dans ce chapitre nous exposerons les résultats des approches utilisées basées sur l'utilisation des réseaux de neurones artificiels pour la commande d'un robot mobile

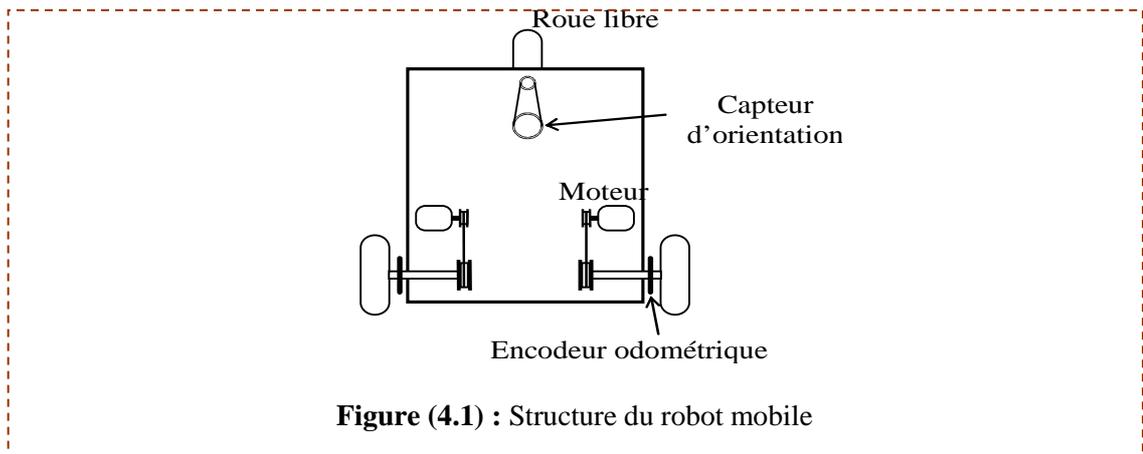
Nous nous intéresserons au suivi d'une trajectoire de référence désirée.

4.1. Simulation de l'approche « Reproduction d'un contrôleur existant »

Nous représentons dans un premier temps les résultats obtenus de la première technique utilisée, reproduction d'un contrôleur existant, dont laquelle nous avons remplacé le contrôleur flou par un contrôleur neuronal, voir section (3.4.1).

4.1.1. Modèle du robot

Le robot possède deux roues motrices arrières, dotées chacune d'un moteur indépendant muni d'un encodeur odométrique pour la mesure de la position et la vitesse angulaires. La roue avant est une roue libre, son rôle est d'assurer la stabilité du robot pendant son mouvement, elle est dotée d'un capteur d'orientation qui donne la mesure de l'angle de braquage du robot.



Le modèle cinématique est le suivant, voir section (1.6.2) :

$$\begin{aligned}
 \dot{x} &= V \cos \theta \\
 \dot{y} &= V \sin \theta \\
 \dot{\theta} &= \frac{V}{D} \operatorname{tg} \alpha
 \end{aligned}
 \tag{4.1}$$

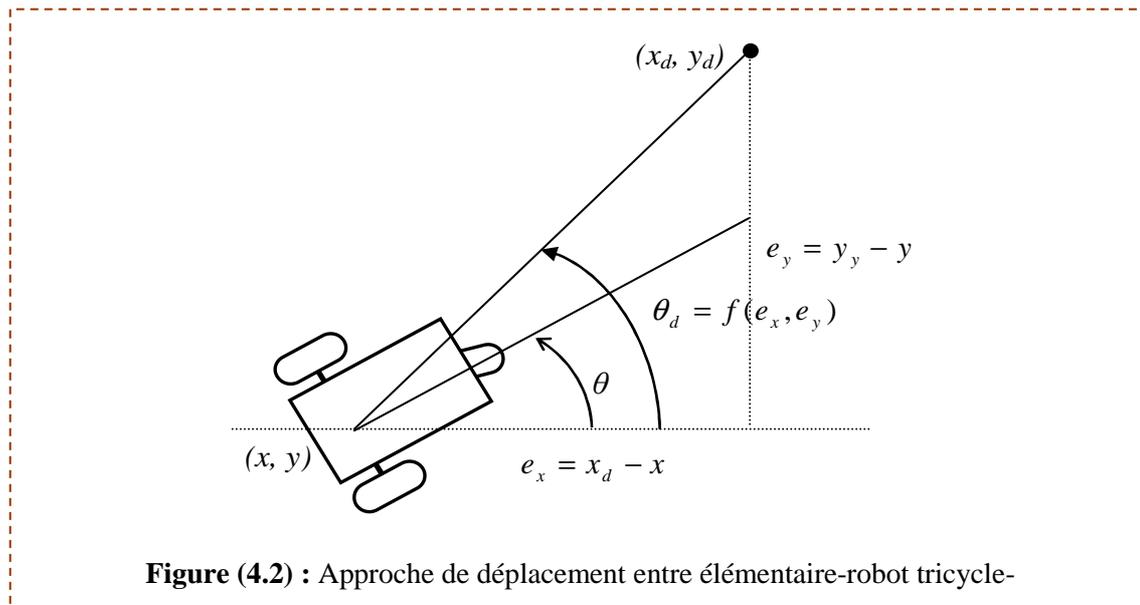
L'intégration des équations cinématiques est faite pour des déplacements élémentaires entre deux instants t et $(t + \Delta t)$

4.1.2. Technique de poursuite

Concernant l'approche présentée au chapitre précédent « reproduction d'un contrôleur existant », la trajectoire est représentée par un ensemble de points de passage définissant le chemin de poursuite désiré. Le chemin à suivre peut être une trajectoire d'évitement d'obstacle issue d'un planificateur, ou un nombre de points de passage qui peuvent être des postes de travail. La trajectoire choisie pour la validation de cette approche est désignée par un ensemble de points défini par le triplet $\{(x, y), \theta\}$ tel que : x est l'abscisse, y est l'ordonnée et θ est l'angle d'orientation.

L'objectif visé est l'aboutissement du robot au point d'arrivée tout en traversant des points précis avec une certaine erreur de précision.

L'angle d'orientation θ_d est généré par l'ensemble contrôleur neuronal–robot dont la structure générale est représentée dans le chapitre précédent. La figure au dessous illustre le principe de déplacement du robot du point actuel au point cible.



L'angle d'orientation θ_d est une fonction $\theta_d = f(e_x, e_y)$, tel que les erreurs de position e_x et e_y validées à partir de sa position courante de coordonnées (x,y) et sa position désiré (x_d, y_d) .

4.1.3. Choix du réseau

Nous avons utilisé un réseau de type PMC :

Nombre de neurones cachés : 5 ;

Les résultats sont :

Erreur Quadratique Moyenne Apprentissage EQMA = 0.0100 ;

Erreur Quadratique Moyenne Test EQMT = 0.0300 ;

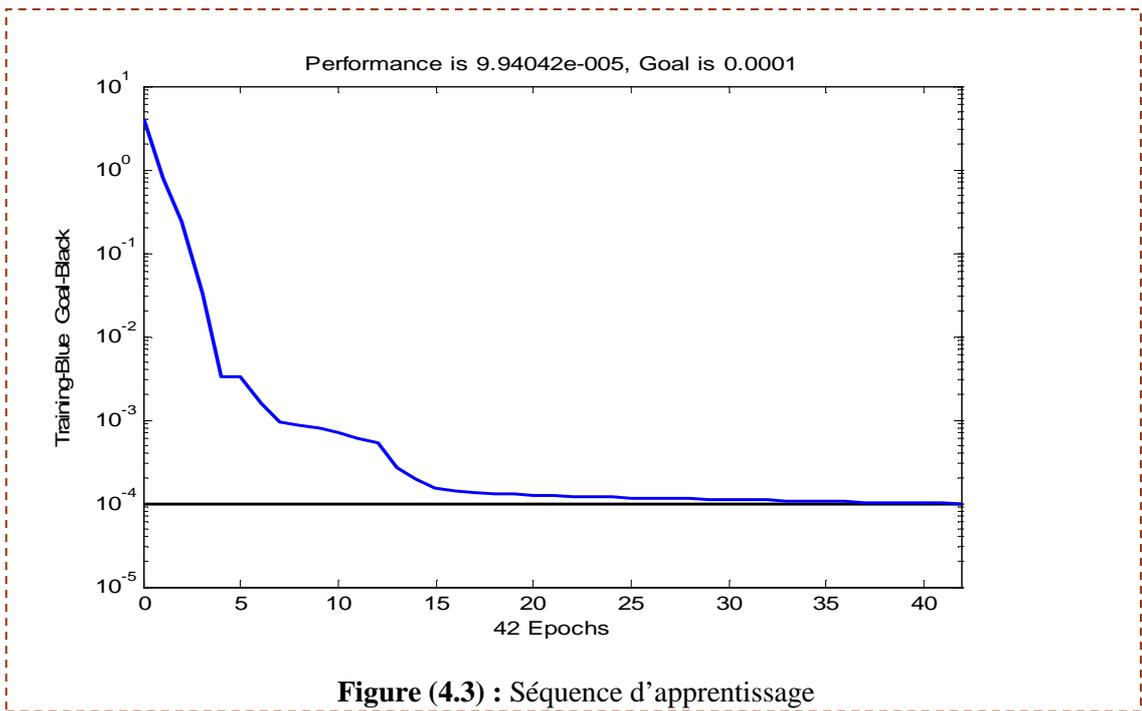


Figure (4.3) : Séquence d'apprentissage

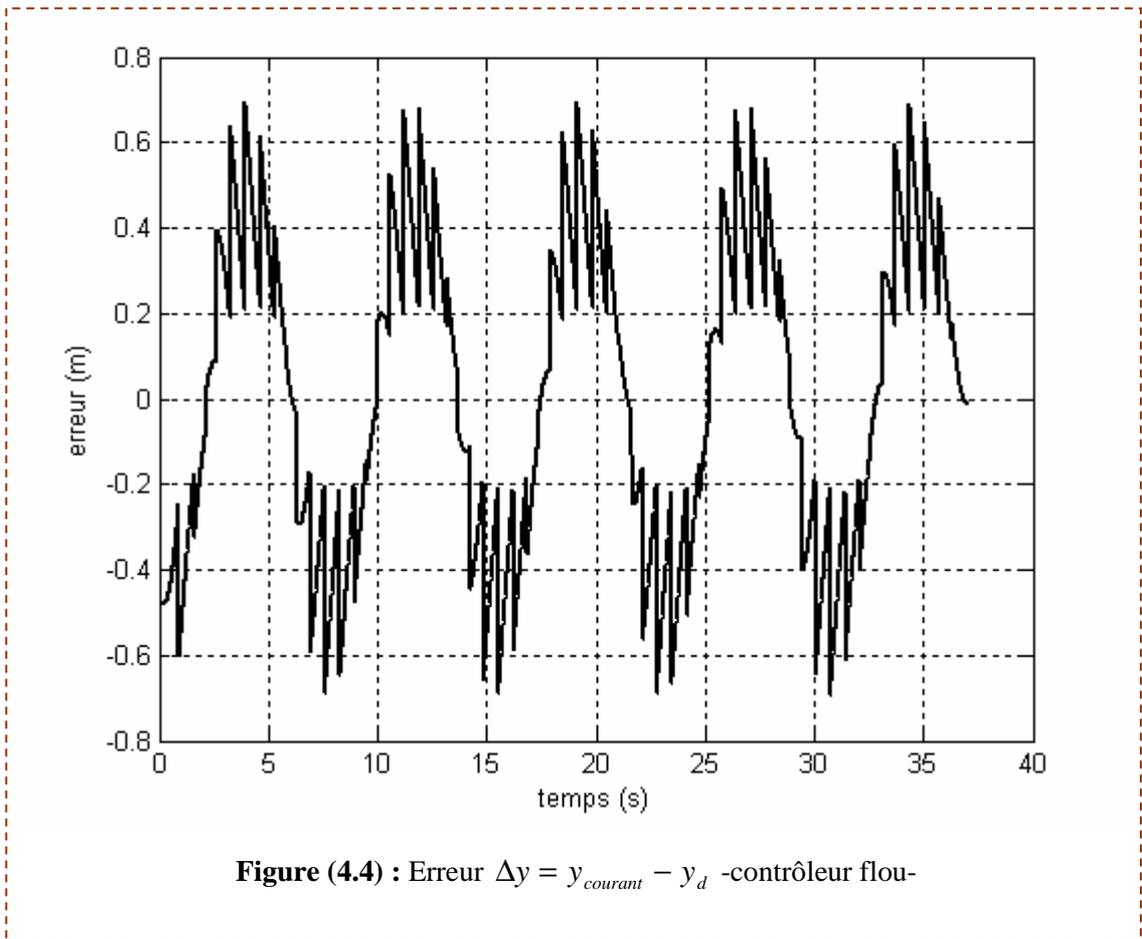


Figure (4.4) : Erreur $\Delta y = y_{courant} - y_d$ -contrôleur flou-

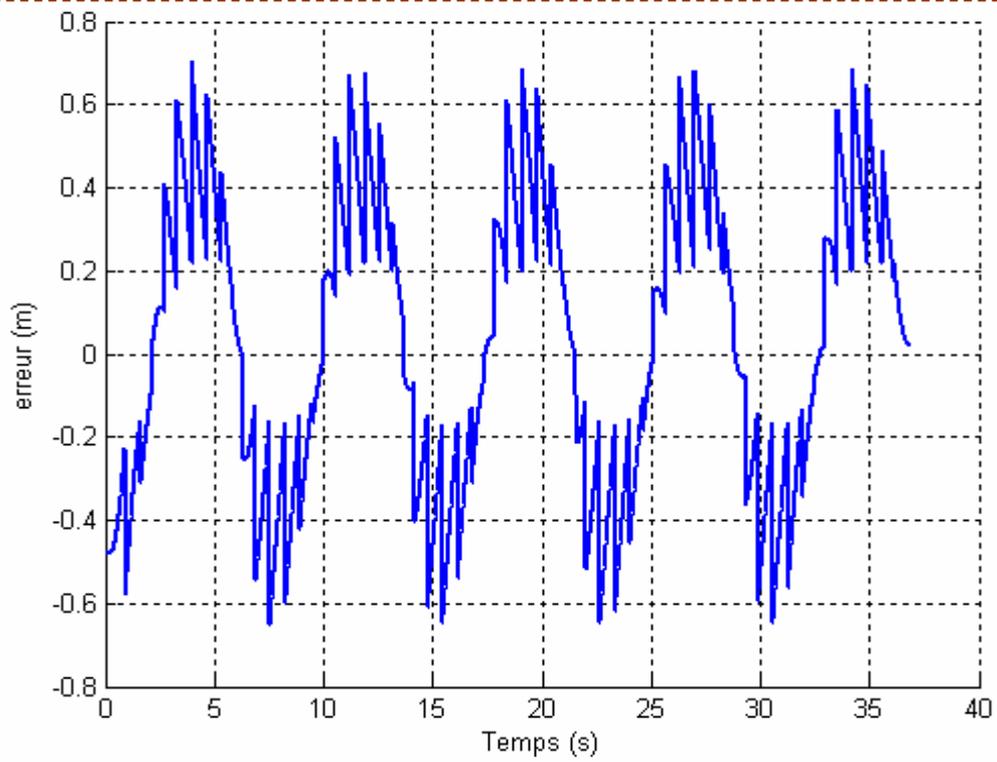


Figure (4.5) : Erreur $\Delta y = y_{courant} - y_d$ -contrôleur neuronal-

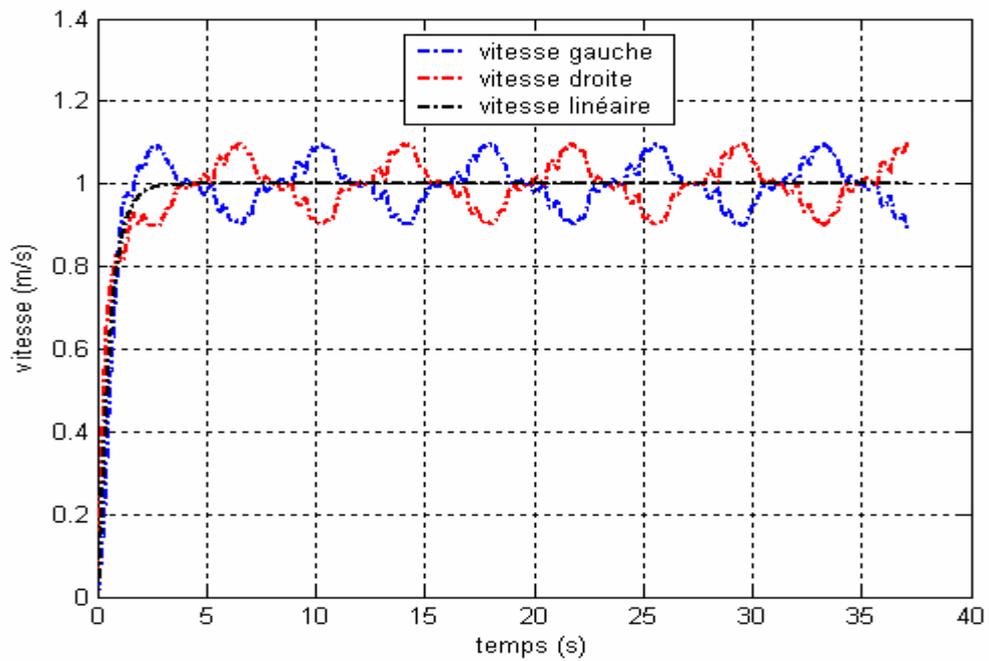


Figure (4.6) : Vitesses -contrôleur flou-

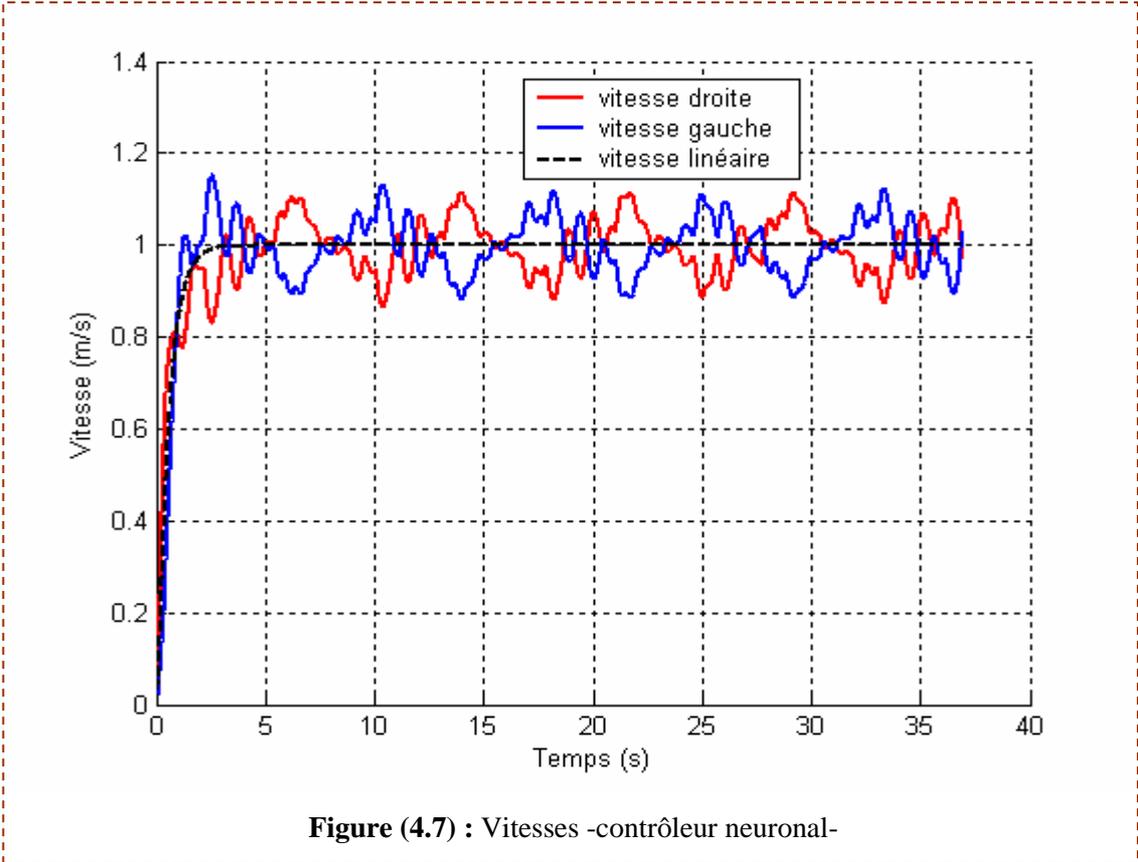


Figure (4.7) : Vitesses -contrôleur neuronal-

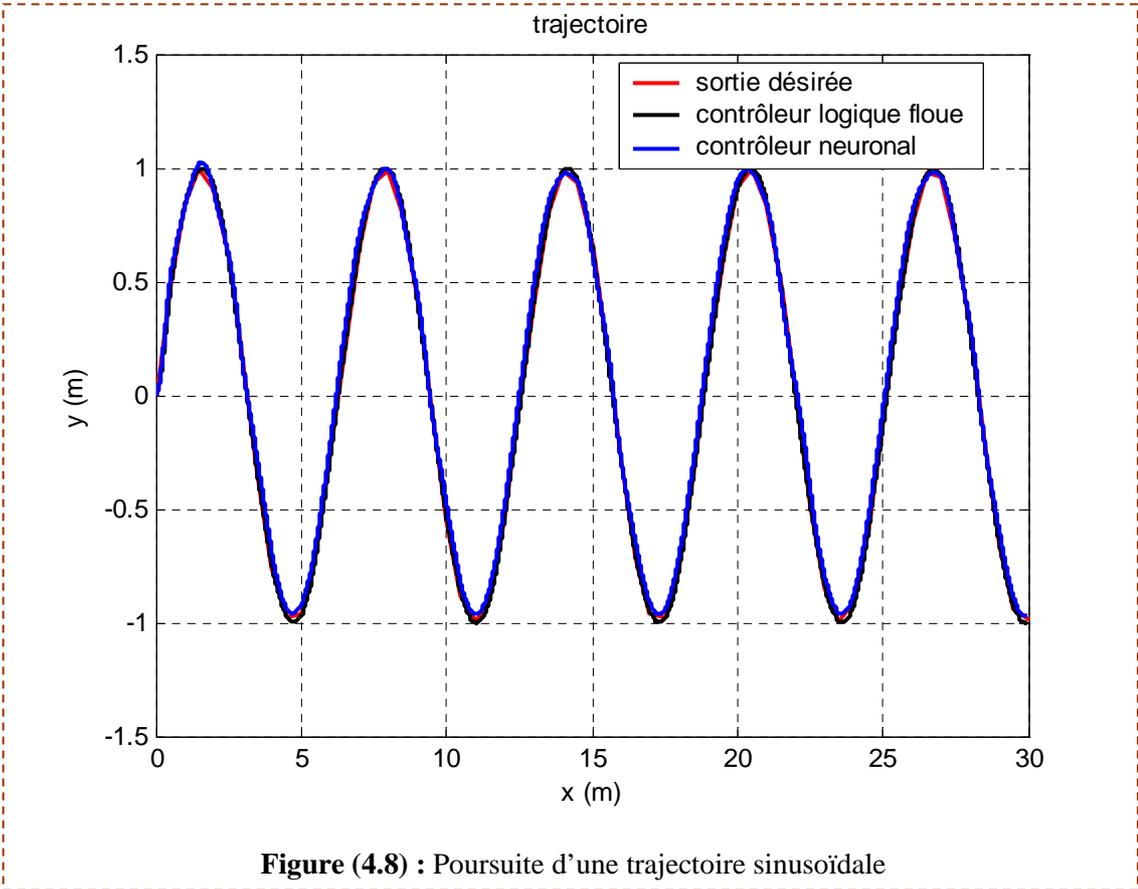


Figure (4.8) : Poursuite d'une trajectoire sinusoïdale

Conclusion

Notons que les résultats de simulation du système à base de contrôleur neuronal sont identiques à ceux obtenus par le système à base du contrôleur flou. En réalité ceci est logique vu que le contrôleur neuronal conçu dans cette partie est une copie du contrôleur flou.

L'avantage de cette tentative est la maîtrise de cette technique neuronale ainsi que la justification du recours à un certain nombre de couches de neurones.

4.2. Simulation de l'approche « Modèle inverse »

La deuxième approche employée est le modèle inverse, voir section (3.4.2) Le but attendu par le robot est toujours la poursuite d'une trajectoire bien définie.

La tâche du contrôleur est de diriger le robot vers des points précis sur une trajectoire d'une manière continue avec la meilleure précision possible. Le robot doit suivre une trajectoire dans l'espace cartésien à partir d'une configuration initiale indiquée. Il n'est pas nécessaire que le robot passe exactement par les points sur la trajectoire, mais passe au moins dans leur proximité et arrive à la destination finale. Le robot mobile exécute sa trajectoire indépendamment de temps avec une vitesse linéaire constante et estime sa position avec ses propres sondes odométriques.

4.2.1. Modèle du robot

Il s'agit d'un robot unicycle, voir section (1.6.1) Dans cette conception des encodeurs sont montés sur les deux moteurs d'entraînement. En employant des équations géométriques simples afin de calculer la position momentanée du

robot relativement à une position de départ connue. Par conséquent, la navigation est commandée par le changement de vitesse de chaque côté du robot. La figure (1.5) montre les paramètres cinématiques du robot mobile

Selon le principe de mouvement de la cinématique rigide de corps, le mouvement d'un robot mobile peut être décrit comme suit :

$$\begin{aligned} v_d &= r\omega_d \\ v_g &= r\omega_g \end{aligned} \tag{4.2}$$

$$\begin{aligned} v &= \frac{v_d + v_g}{2} \\ \omega &= \frac{v_d - v_g}{L} \end{aligned} \tag{4.3}$$

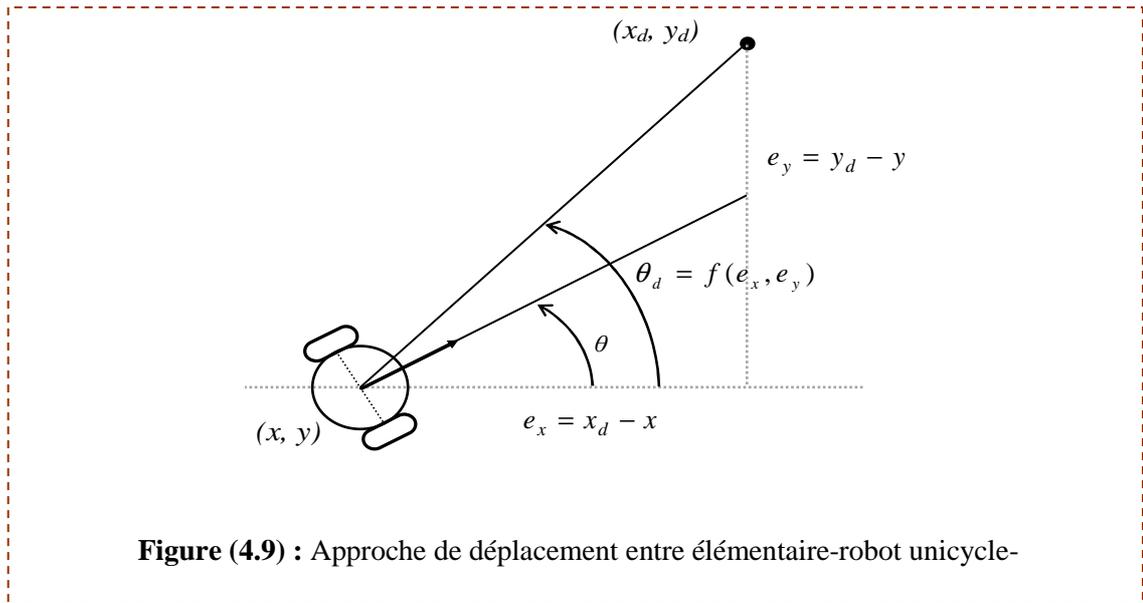
Où :

ω_d et ω_g sont les vitesses angulaires des roues droite et gauche respectivement et ω est la vitesse angulaire du centre de surface. A partir des équations (4.2) et (4.3) on obtient le modèle cinématique :

$$\begin{aligned} \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega \end{aligned} \tag{4.4}$$

4.2.2. Technique de poursuite

La trajectoire à suivre est stockée dans la mémoire sous forme de vecteur de trois éléments (x_d, y_d, θ_d) . Dans cette partie nous avons pris en compte uniquement un vecteur avec deux éléments (x_d, y_d) le troisième paramètre sera évalué calculé par le robot pendant le déplacement.



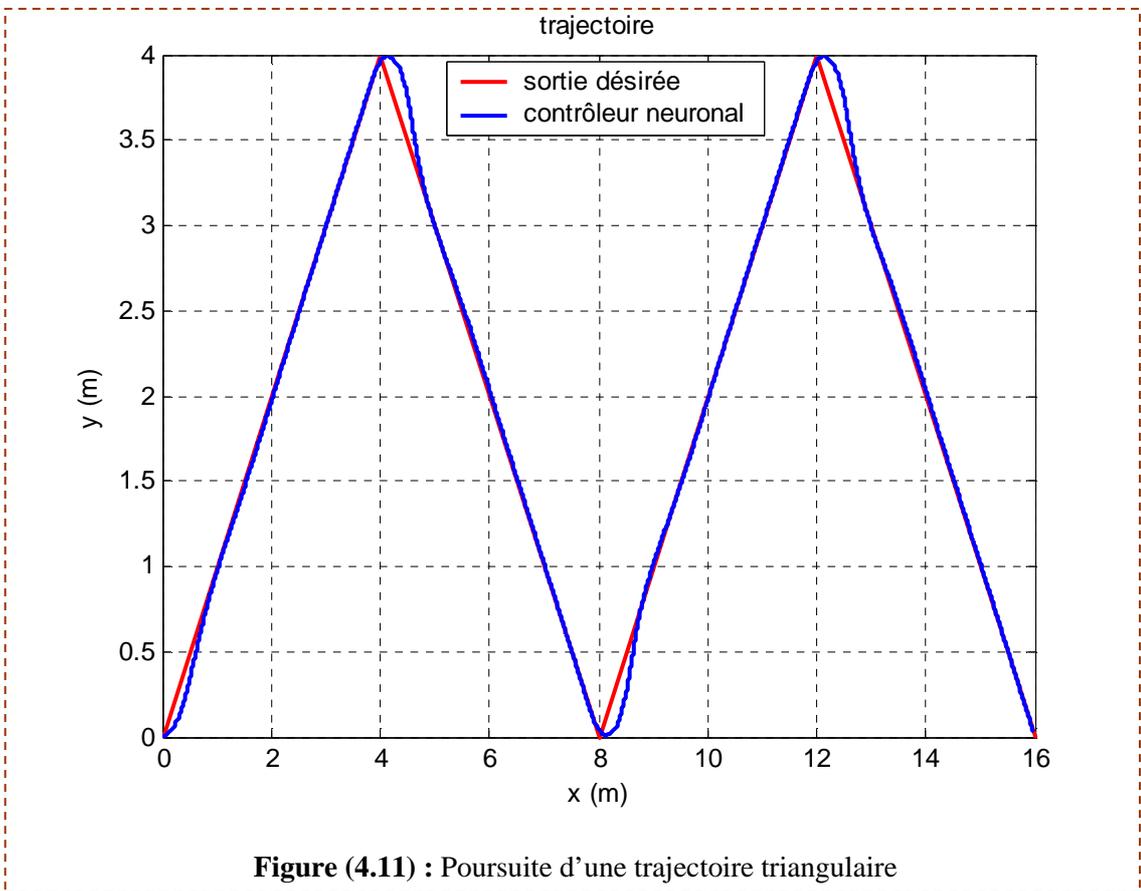
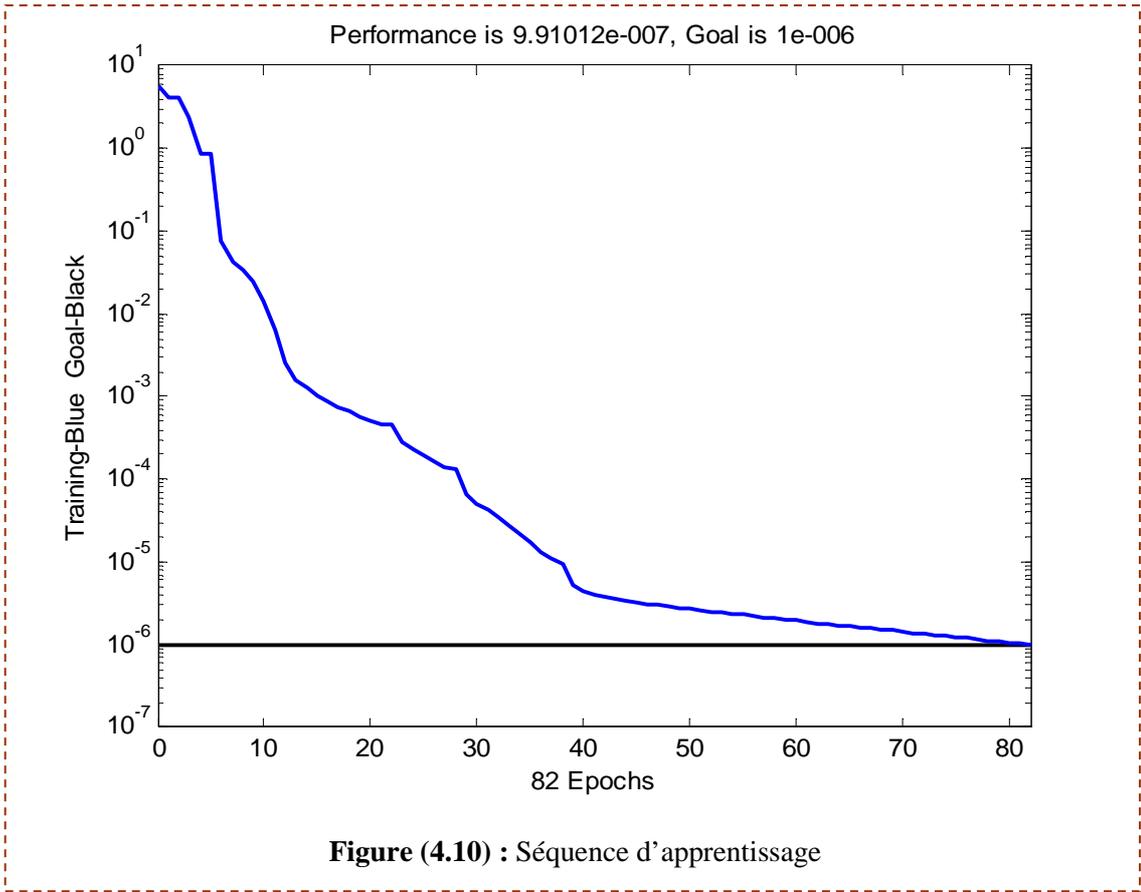
La figure (4.9) montre la technique de déplacement du robot entre deux points successifs. Le principe de fonctionnement est identique à celui expliqué dans le paragraphe 4.1.2.

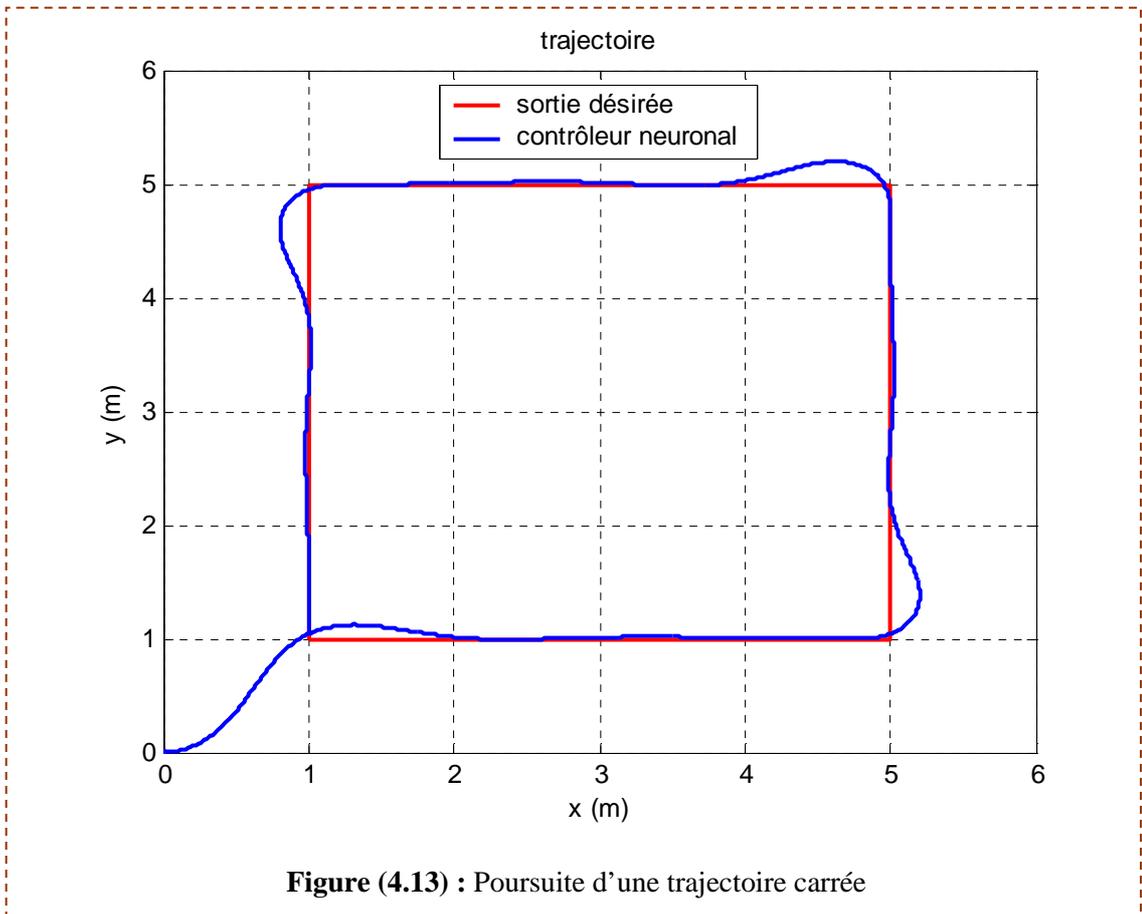
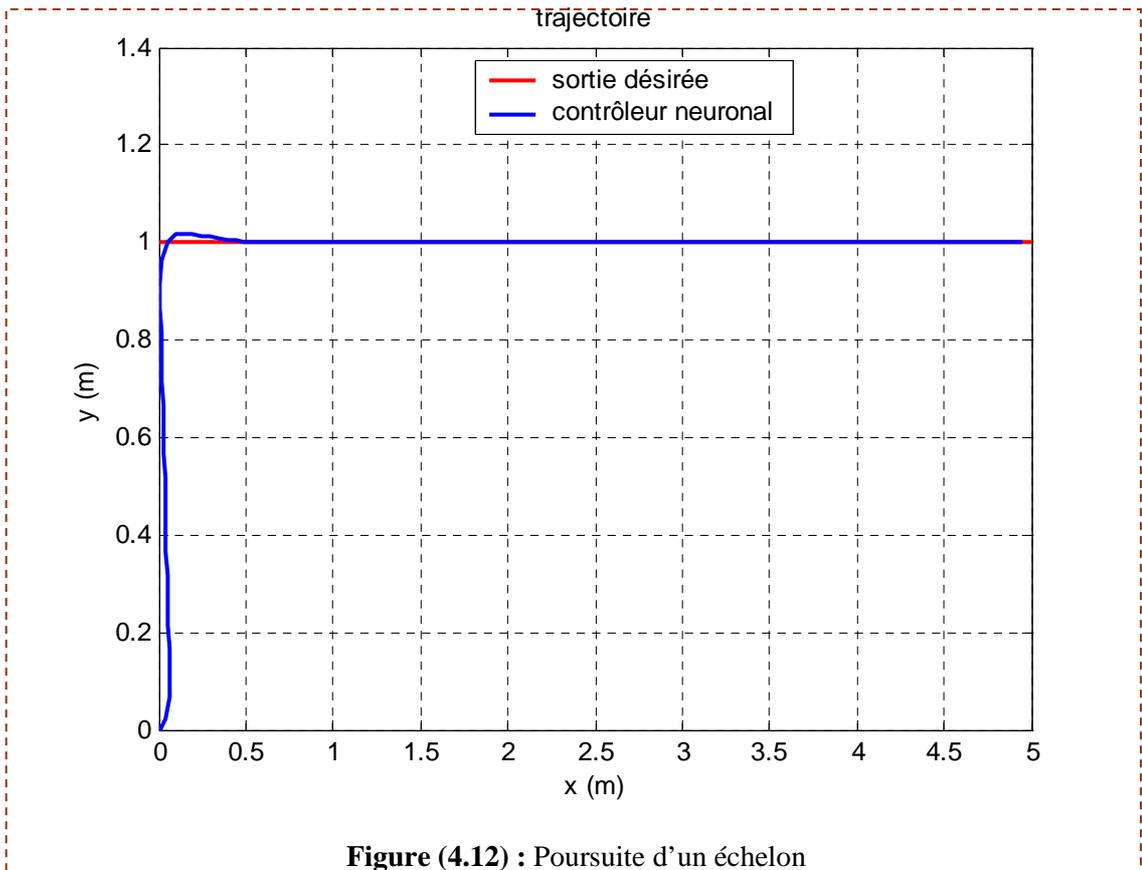
4.2.3. Choix du réseau

Un réseau multicouche utilisé comme neurocontrôleur possède une seule entrée, deux couches cachées de 5 neurones à fonctions d'activations sigmoïdes et une sortie à fonction d'activation linéaire. Son apprentissage est réalisé à l'aide de l'algorithme de rétropropagation du gradient basé sur l'erreur $e = \text{sortie désirée} - \text{sortie réelle}$, les résultats sont :

$$\text{EQMA} = 9.9550\text{e-}004 ;$$

$$\text{EQMT} = 9.2389\text{e-}004 ;$$





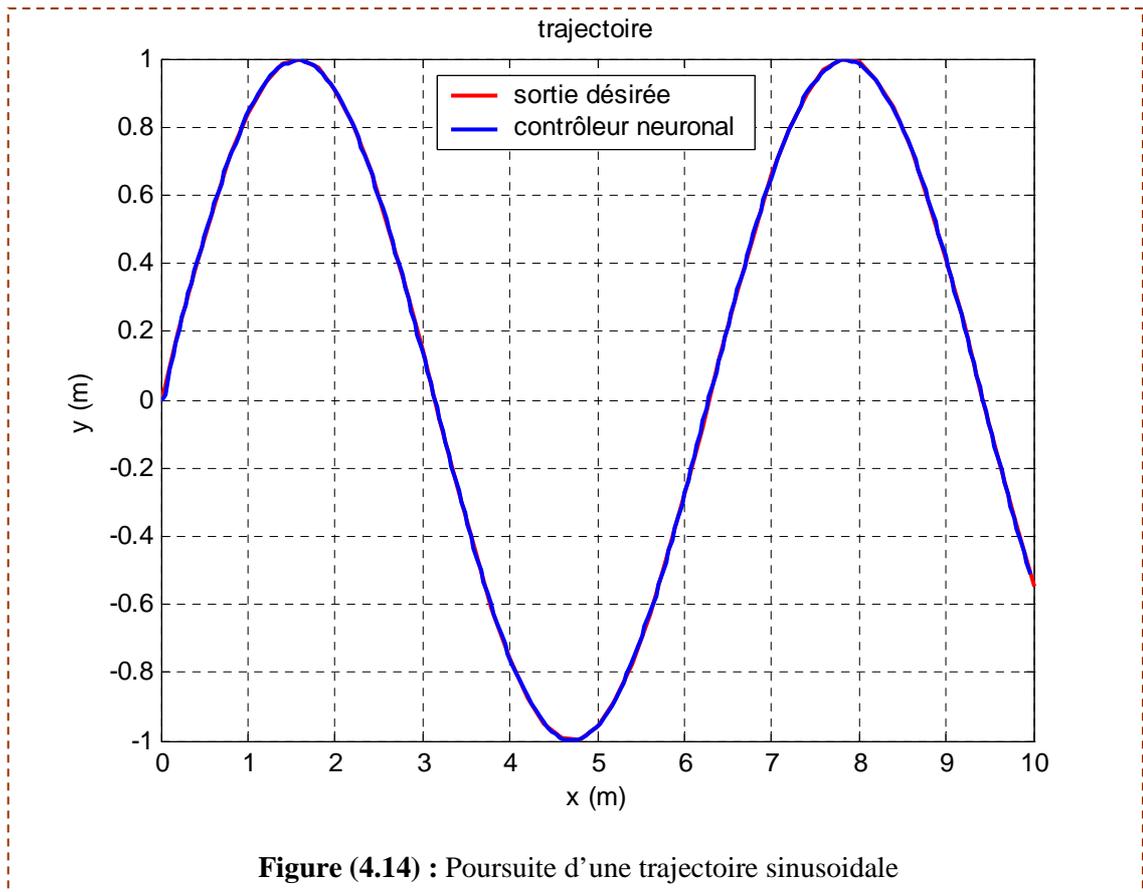


Figure (4.14) : Poursuite d'une trajectoire sinusoidale

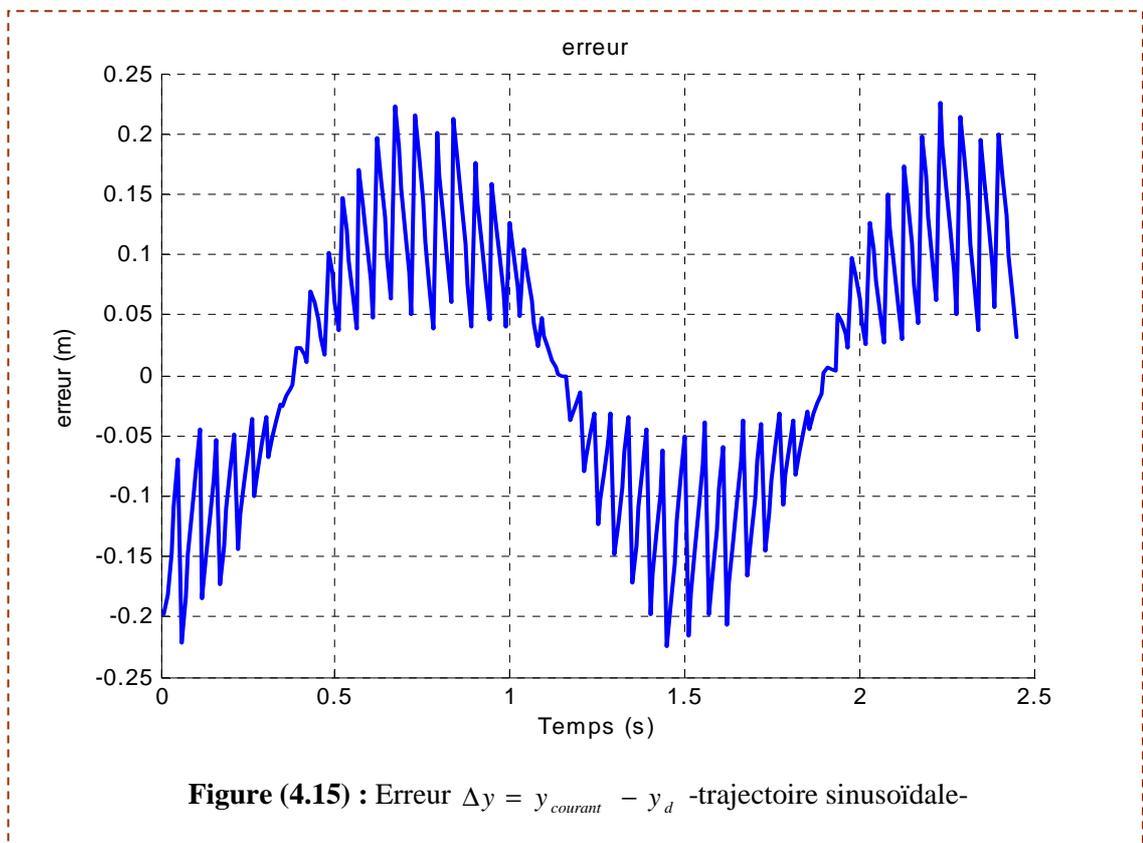


Figure (4.15) : Erreur $\Delta y = y_{courant} - y_d$ -trajectoire sinusoidale-

Conclusion

La tâche du robot est de suivre le chemin désiré avec un minimum d'erreur, Les figures précédentes montrent les performances du contrôleur neuronal inverse en poursuite de la trajectoire. On constate que cette approche assure la souplesse des mouvements.

4.3. Simulation de l'approche « PD neuronal à coefficients adaptatifs »

L'idée de base de cette dernière approche est d'employer un contrôleur neuronal « PD à coefficients adaptatifs », nous avons appliqué cette technique sur le même modèle de robot que la première approche, robot mobile tricycle, ainsi que nous avons suivi la même stratégie de poursuite de trajectoire désirée.

4.3.1. Choix du réseau

Cette dernière méthode s'inspire du principe de fonctionnement d'un PI classique. Le PI neuronal utilise un Adaline avec deux entrées pour déterminer les paramètres proportionnel et dérivateur, voir section (3.4.3-b).

4.3.2. Résultats de simulation

Pour tester ce contrôleur sur les performances de poursuite, nous avons appliqué cette approche pour différents chemins et pour différentes valeurs de l_r , coefficient d'apprentissage et de m , coefficient du terme moment.

- $l_r = 0.00001$, $m = 0.8$

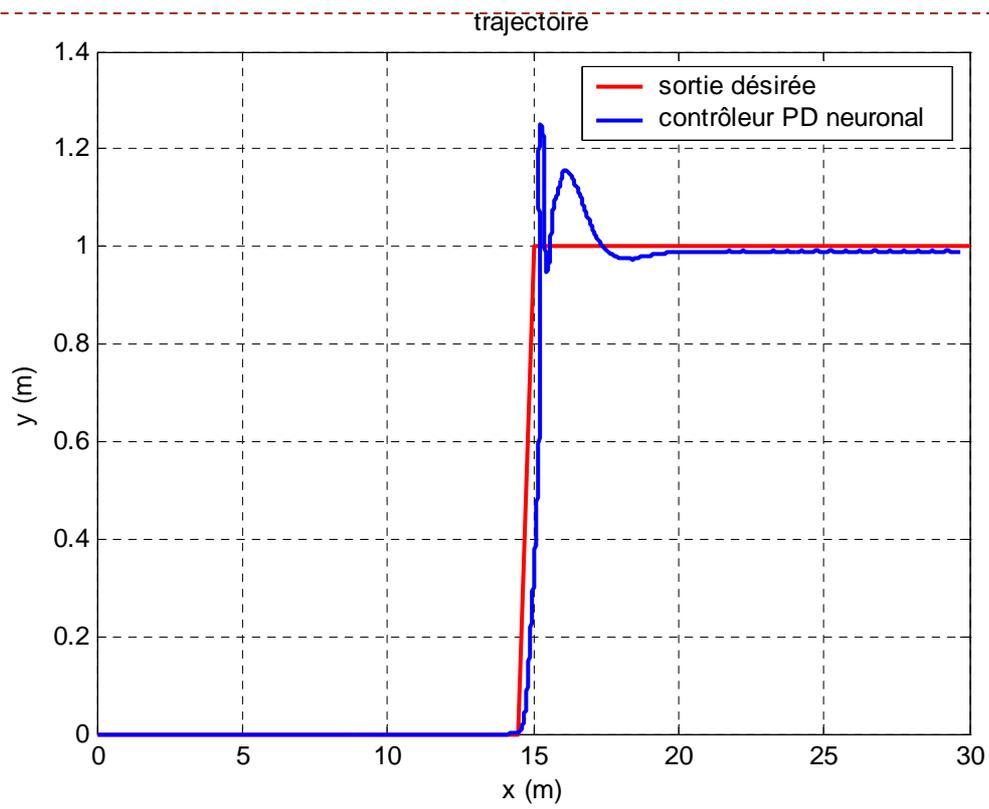


Figure (4.16) : Poursuite d'un échelon

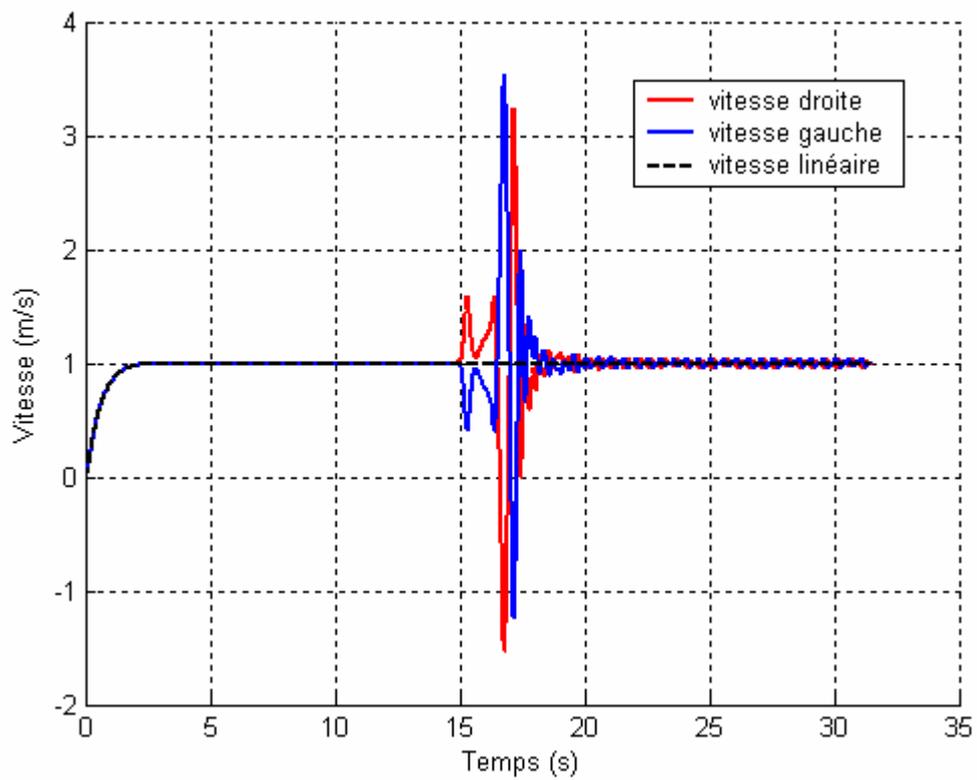


Figure (4.17) : Vitesses des moteurs droit et gauche

- $l_r = 0.000009$, $m = 0.9$;

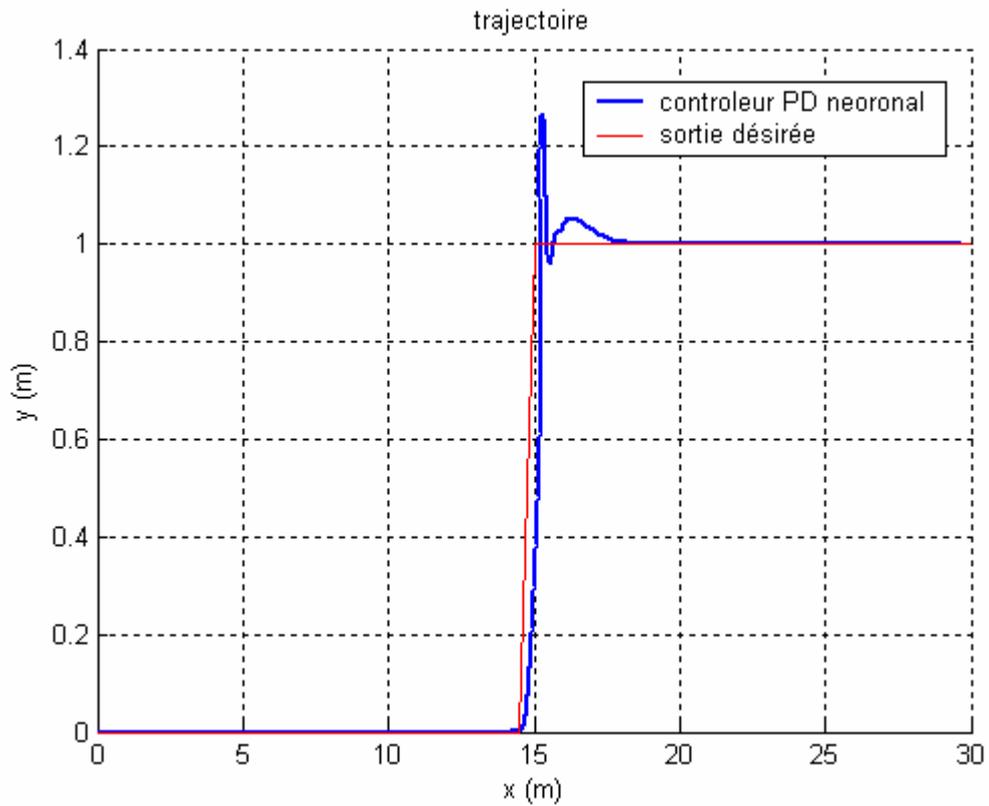


Figure (4.18) : Poursuite d'un échelon

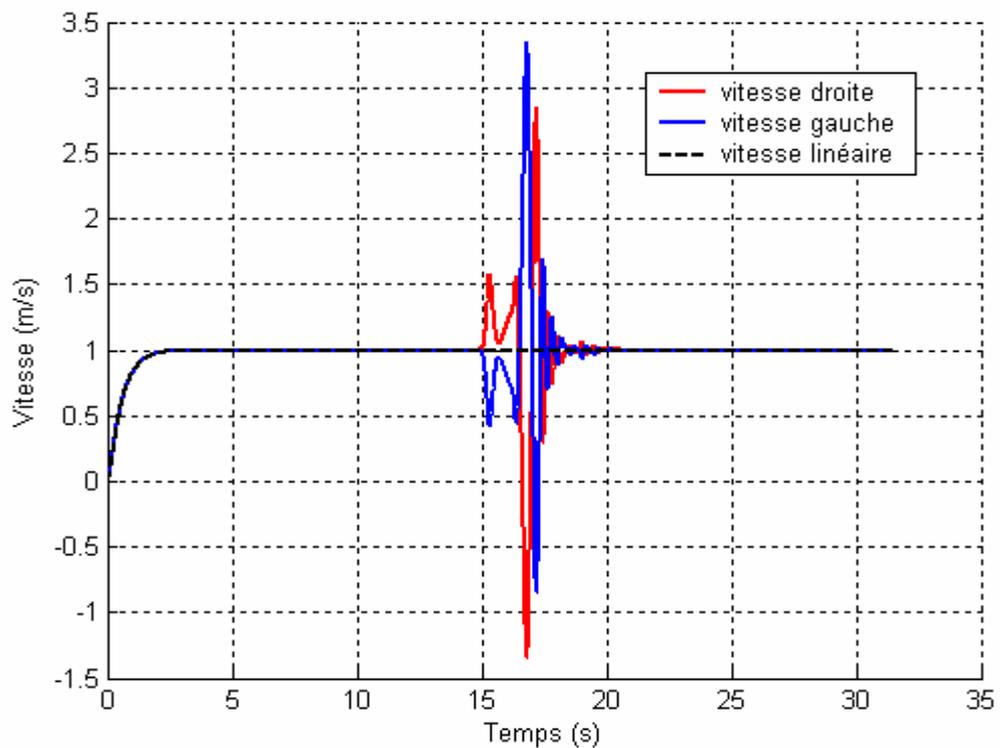


Figure (4.19) : Vitesses des moteurs droit et gauche

- $L_r = 0.00002$; $m = 0.9$

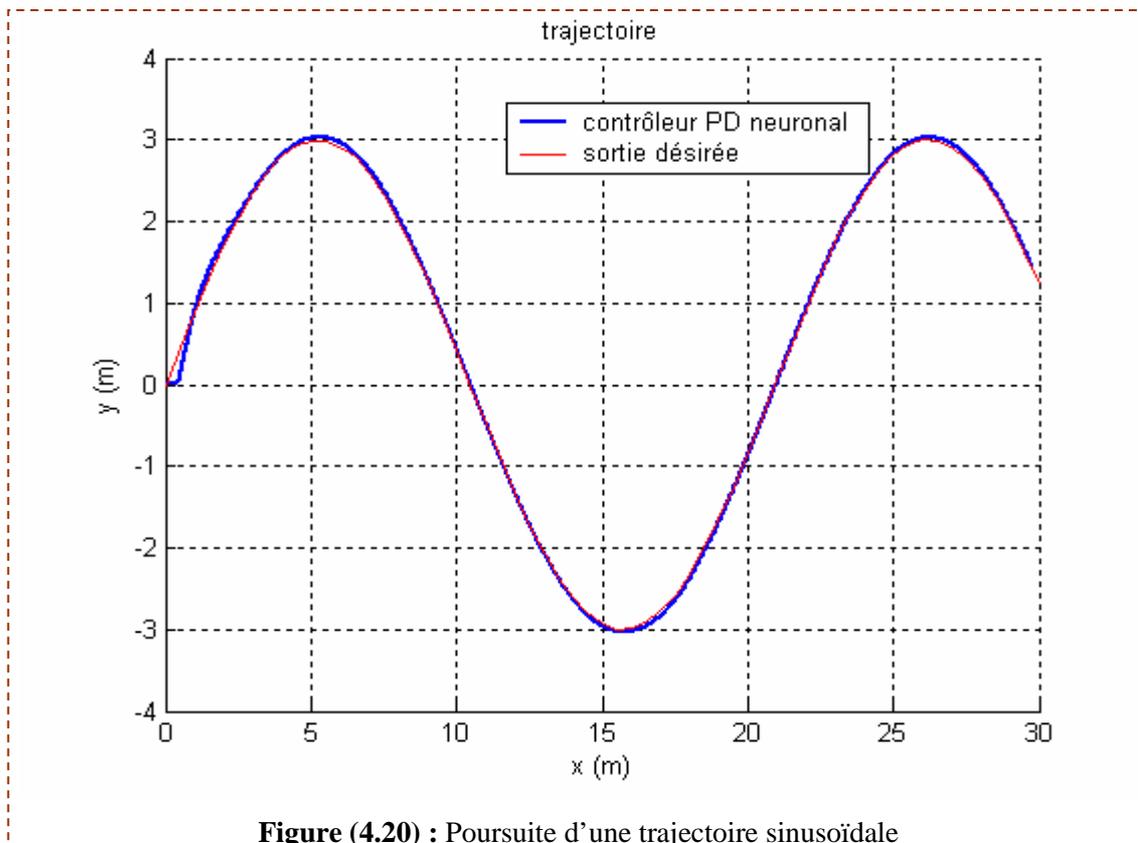


Figure (4.20) : Poursuite d'une trajectoire sinusoïdale

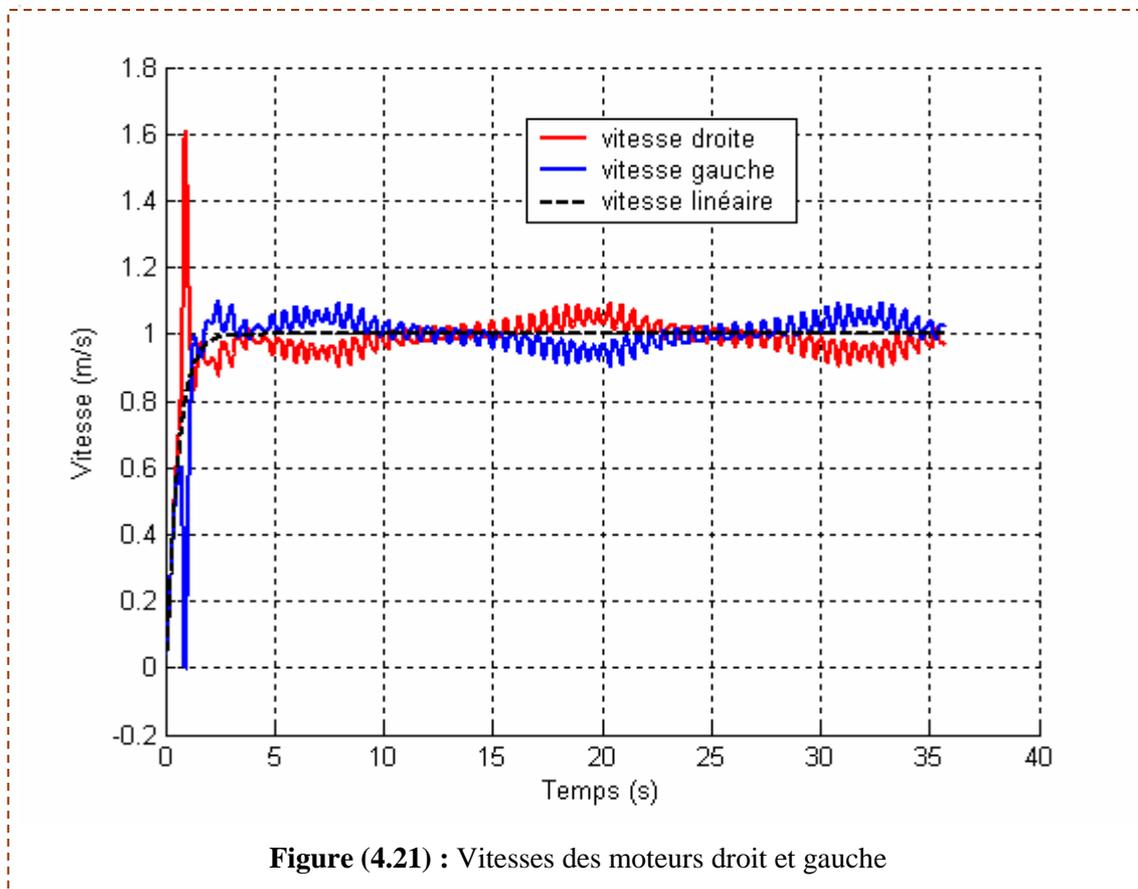


Figure (4.21) : Vitesses des moteurs droit et gauche

Conclusion

On note que avec ce contrôleur neuronal « PD à coefficient adaptatifs l'apprentissage se fait en ligne, nous l'avons appliqué d'abord pour la poursuite un échelon, changement brusque de trajectoire pour déterminer la stabilité et la précision de poursuite ensuite pour la poursuite d'une trajectoire sinusoïdale, on remarque que le changement des coefficients l_r et m , coefficient d'apprentissage et coefficient du terme moment respectivement, a une grande influence sur les résultats. On constate que l'utilisation de cette dernière approche donne des résultats satisfaisants. Le but de cette approche est de concevoir un contrôleur PI adaptatif et d'exploiter la simplicité de réglage des réseaux Adaline.

CONCLUSION GENERALE

CONCLUSION GENERALE

Ce travail qui se situe à l'intersection des domaines de la robotique mobile et les réseaux de neurones artificiels, a été pour nous un pas vers un mode très vaste et très intéressant. Une recherche bibliographique dans ce domaine, nous a permis d'enrichir nos connaissances et d'acquérir une base théorique assez exhaustive.

Dans ce travail nous avons essayé d'apporter une contribution aux possibilités offertes par les réseaux de neurones artificiels dans le cadre de la commande d'un robot mobile à roues.

Deux approches de commande utilisant les réseaux de neurones artificiels multicouches ont été réalisées. Une troisième approche reprend le principe d'un régulateur PD et emploie un réseau pour adapter les paramètres proportionnel et dérivateur.

Les différents modèles relatifs au robot mobile ont été étudiés et présentés au niveau du premier chapitre. Nous nous sommes intéressés aux modèles cinématiques vu la simplicité.

Les principales structures des réseaux de neurones artificiels ainsi que leur propriété ont été également discutés dans le deuxième chapitre. L'intérêt est de donner un aperçu sur cet outil très sollicité dans le domaine de contrôle, classification et autres...

Trois approches ont été utilisées dans ce travail, la première est une copie « imitation » d'un contrôleur flou déjà réalisé au sein de notre laboratoire de recherche. Dans cette approche une base de données déjà fournie par le contrôleur flou a été exploitée par le réseau de neurones. Nous avons pu constater que le robot suit la trajectoire désirée. Les performances obtenues avec les réseaux de neurones artificiels sont semblables à ceux du contrôleur flou avec erreur permanente relativement la même.

La deuxième approche « modèle inverse » a donné des bons résultats. Une amélioration est observée au niveau la valeur maximale de l'erreur de poursuite.

Concernant la dernière approche, un régulateur PD neuronal à coefficients adaptatifs a été étudié et simulé. L'apprentissage est effectué en ligne. Le recours aux réseaux de neurones a constitué un moyen efficace pour la détermination des paramètres du PD en l'occurrence paramètre proportionnel et paramètre dérivé. Plusieurs essais de validations de cette approche ont été testés, les résultats obtenus ont été jugés acceptables en particulier la réponse du système suite à l'application d'un échelon.

Nous pouvons affirmer que l'objectif, fixé au départ, a pu être atteint. Comme conclusion, les trois approches utilisées permettent la poursuite du robot aux différentes trajectoires. Les réseaux de neurones constituent un moyen crédible pour le contrôle des robots mobiles.

Comme perspectives, des améliorations peuvent être apportées à ce travail afin de le perfectionner et de l'enrichir. Il s'agit en particulier des contrôleurs hybrides « les réseaux neuro-flous ». Comme exemple, les systèmes utilisent à la fois une base de règles pour les connaissances à priori et un algorithme d'adaptation pour le réajustement des sous-ensembles flous. Cette combinaison permettra d'entirer profit des avantages outils et par conséquent l'amélioration des performances obtenues.

BIBLIOGRAPHIE

Bibliographie

- [1] Mohamed Ryad ZEMOURI, « Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance », Thèse doctorat, L'UFR des Sciences et Techniques de l'Université de Franche-Comté, Novembre 2003.
- [2] Marc Parizeau, « Réseaux de Neurones », Université de Laval, Automne 2004.
- [3] Eric Beaudry, « Planification en intelligence artificielle pour la robotique mobile », mémoire de maîtrise, Faculté des sciences, Université de Sherbrooke, Août 2006.
- [4] Frédéric Giamarchi, « Petits robots mobiles », ed. Dunod, 2006.
- [5] Bernard Bayle, « Robotique mobile », Ecole Nationale Supérieure de Physique de Strasbourg, 2004–2005.
- [6] Eric GAUTHIER, « Utilisation des Réseaux de Neurones Artificiels pour la Commande d'un Véhicule Autonome », thèse doctorat, Institut national polytechnique de Grenoble, Janvier 1999.
- [7] Michel WEINFELD, « Réseaux de Neurones », Directeur de Recherche au CNRS, Laboratoire d'Informatique de l'École polytechnique, est strictement interdite, Techniques de l'Ingénieur.
.
- [8] Léon Personnaz, Isabelle Rivals, « Réseaux de neurones formels pour la modélisation, la commande et la classification », ed. CNRS, Paris , 2003.
- [9] Isabelle Rivals, « Modélisation et commande de processus par réseaux de neurones ; application au pilotage d'un véhicule autonome -Le robot REMI de la SAGEM-», Thèse doctorat, Université Pierre et Marie Curie, janvier 1995.

- [10] Isabelle Rivals, « Les réseaux de neurones formels pour le pilotage de robots mobiles », revue de l'association amicale les ingénieurs SUPÉLEC : La robotique mobile, septembre-octobre 1996.
- [11] A. Louchene, « Système de navigation pour robot mobile », thèse doctorat, Université de Batna, 2004.
- [12] G ; Dreyfus, J. M. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, L. Hérault, «°Réseaux de neurones – méthodologie et application- », ed. Eyrolles, 2002.
- [13] Naceur-eddine Boukezzoula, «°Simulation et mise en œuvre des réseaux de neurones formels », thèse doctorat, Université de Sétif, 2006.
- [14] J. Wesley Hines, «°Fuzzy and neural approaches in engineering Matlab supplement », New York, 1997.
- [15] Michel Marie, Mohand Mokhtari, «°Applications de Matlab et Simulink », ed. Springer, 1998.
- [16] Hamada Bnnassar, «°Les réseaux de neurones artificiels dans les commandes de systèmes », Wichita State University USA, 1993.
- [17] Nicola Tomatis, «°Hybrid, metric-topologic, mobile robot navigation », thèse doctorat, Ecole polytechnique fédérale de Lausanne, 2001.
- [18] M. Norgaard, O. Ravn, N. K. Poulsen, L. K. Hasen, «°Neural networks for modelling and control of dynamic systems »,ed Springer.
- [19] Bernard BAYLE, «°Robotique mobile », Ecole nationale supérieure de physique de Strasbourg, 2004-2005.
- [20] David Filliat, «°Robotique mobile », ed. ENSTA, Octobre 2004.
- [21] Haowen Yang, Simon X. Yang and Gauri S . Mittal, «°Tracking control of a nonholonomic mobile robot by integrating feedback and neural dynamics techniques », Proceedings of the 2003 IEEURSJ, Intl. Conference on Intelligent Robots and Systems Las Vegas. Nevada, October 2003.

- [22] Ben Krose, Patrick van der Smagt, «°An introduction to neural networks »,ed Springer, ed. Eighth, November 1996.
- [23] Claude Touzet, «°Les réseaux de neurones artificiels, introduction au connexionnisme cours, cours, exercices et travaux pratiques », Juillet 1992.
- [24] R. Fierro, F. L. Lewis, «°Control of a nonholonomic mobile robot using neural networks », IEEE transasctions on neural networks, Juillet 1998.
- [25] Johann Borenstein and Yoram Koren , «°Motion control anatysis of a mobile robot », Transactions of ASME, Journal of dynamics, measurement and control, Vol. 109, No. 2, pp. 73-79, Decembre 1995.
- [26] Satoshi Vamaguchi, Hidekiyo Italurra, «°A modular neural network of mobile robots », Chiba institute of technology, Japan, 1999.
- [27] M. Zerikat, M. Bendjebbar, N. Benouzza, «° Dynamic Fuzzy-Neural Network Controller for Induction Motor Drive », Proceedings of world academy of science, engineering and technology volume 10, Decembre 2005.