

LES RÉSEAUX DE NEURONES FORMELS POUR LE PILOTAGE DE ROBOTS MOBILES

Isabelle RIVALS, Laboratoire d'Électronique de l'ESPCI
(École Supérieure de Physique et de Chimie Industrielles)

Introduction

Les réseaux de neurones formels sont des systèmes de traitement de l'information dont la structure s'inspire de celle du système nerveux. Leurs deux grands domaines d'application sont d'une part la modélisation biologique, dont il ne sera pas question ici, et d'autre part, la réalisation de machines destinées à effectuer des tâches auxquelles les ordinateurs et les outils traditionnels semblent moins bien adaptés que les êtres vivants, telles que des tâches perceptives et motrices. Ainsi, les applications des réseaux de neurones formels à la reconnaissance de formes et à la classification ont-elles pris une place importante au sein des réalisations industrielles. Dans le domaine du traitement du signal et de l'automatique, les années 90 ont été marquées par des avancées significatives tant du point de vue fondamental que du point de vue des applications industrielles, signe incontestable de la maturation de ces outils. Nous allons ainsi présenter la réalisation du système de pilotage automatique d'un robot mobile à l'aide de réseaux de neurones effectuée au cours de la préparation d'un Doctorat, dans le cadre d'une convention CIFRE dont le partenaire industriel était la société SAGEM.

Les réseaux de neurones

Un neurone formel est un processeur très simple (simulé sur ordinateur ou réalisé sur circuit intégré) imitant grossièrement la structure et le fonctionnement d'un neurone biologique. La première version du neurone formel est celle de W. S. McCulloch et W. Pitts, et date de 1943. C'est un automate binaire qui réalise une somme pondérée de ses entrées, le *potentiel*, et compare ce potentiel à un seuil (nul) : s'il est supérieur, la sortie vaut +1 et le neurone est actif, s'il est inférieur, la sortie vaut -1 et le neurone est inactif. Il existe aujourd'hui d'autres types de neurones, mieux adaptés aux tâches de traitement du signal ou de classification, dont la sortie n'est pas le signe de leur potentiel, mais une fonction f non linéaire dérivable de ce potentiel, telle une tangente hyperbolique ; cette fonction est dite *fonction d'activation* du neurone (voir Figure 1). On peut également mettre en œuvre des réseaux de gaussiennes, d'ondelettes, etc.

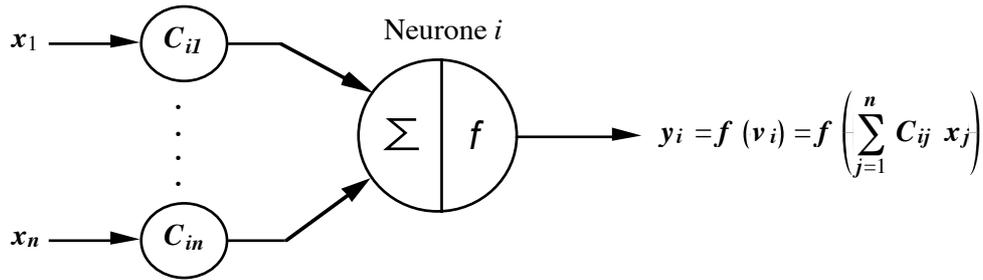


Figure 1 : Un neurone formel.

Un réseau de neurones est un système constitué de neurones interconnectés, qui reçoit des informations de l'environnement. On distingue deux familles de réseaux de neurones :

- les réseaux non bouclés, dans lesquels l'information circule des entrées vers les sorties, sans bouclage (voir Figure 2) ; ce sont des systèmes statiques, utilisés principalement pour effectuer des tâches de classification, ou de modélisation statique de processus ; l'opérateur réalisé par un réseau de neurones non bouclé (relation entrées-sorties) est une fonction algébrique ;

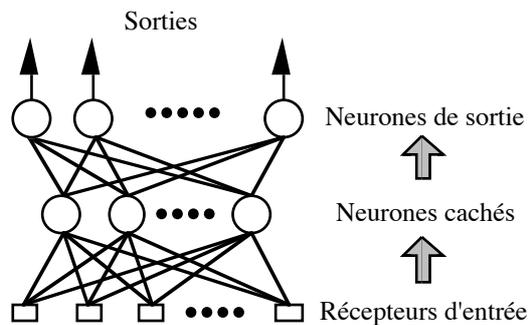


Figure 2 : Exemple de réseau de neurones non bouclé (dit à une couche de neurones cachés).

- les réseaux bouclés, dont le graphe des connexions contient des cycles (voir Figure 3) ; ce sont des systèmes dynamiques, utilisés comme filtres non linéaires, ainsi que pour la modélisation et la commande de processus ; l'opérateur réalisé par un réseau bouclé est un ensemble d'équations aux différences couplées.

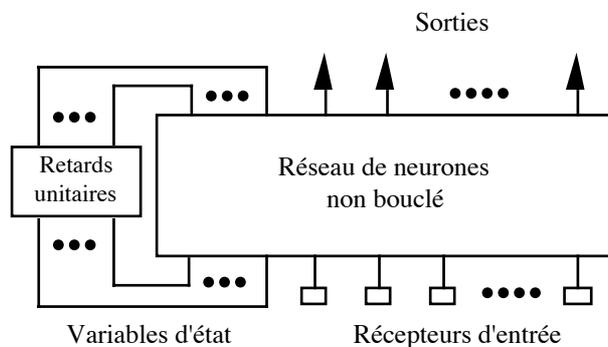


Figure 3 : Exemple de réseau de neurones bouclé.

L'opérateur non linéaire réalisé par un réseau, bouclé ou non, dépend des valeurs des coefficients de pondération C_{ij} (du neurone j vers le neurone i) du réseau. Pour qu'un réseau effectue une tâche

donnée, il faut donc ajuster la valeur de ses coefficients. Une tâche est définie par un ensemble d'exemples, ou couples {valeurs des entrées – valeurs des sorties désirées correspondantes}, tels les couples {forme – classe} en classification, ou les couples {commande – sortie mesurée du processus} en modélisation : ces couples constituent l'*ensemble d'apprentissage*. La procédure d'ajustement des coefficients de manière que les sorties du réseau soient aussi proches que possibles des sorties désirées est appelée *apprentissage*. Le principe général des *algorithmes d'apprentissage* repose sur la minimisation d'une fonction de coût quadratique des différences entre les sorties du réseau et les sorties désirées.

Les réseaux de neurones possèdent la *propriété d'approximation universelle*, qui stipule que toute fonction continue et bornée peut être approchée uniformément sur un domaine borné de ses entrées par un réseau de neurones de taille finie. En vertu de cette propriété, le comportement de tout système statique peut être approché par un réseau de neurones non bouclé approprié, et celui de tout système dynamique par un réseau bouclé. C'est cette propriété, conjuguée à l'existence d'algorithmes d'apprentissage performants, qui fait l'intérêt des réseaux de neurones formels pour l'automatique, beaucoup plus que leur parenté historique avec la circuiterie biologique.

Le pilotage de robots mobiles

La conception des véhicules automatisés, ou robots mobiles, à roues est un domaine de recherche en pleine expansion, et qui présente des enjeux industriels considérables. Ces véhicules sont en effet utilisés dans l'industrie comme moyen de transport, d'inspection, ou d'opération, et sont particulièrement adaptés à des interventions en environnement hostile.

Le robot REMI de la SAGEM

Ainsi, l'unité de recherche en Automatique de la Division Navigation et Défense de SAGEM a-t-elle équipé REMI (Robot Evaluator for Mobile Investigations), un 4x4 Mercedes de série, des actionneurs nécessaires au pilotage, et de capteurs pour la localisation du robot et celle d'obstacles éventuels (voir Figure 4). Le robot REMI n'est pas destiné à être commercialisé, mais les résultats de l'étude ont permis de poser les jalons d'applications industrielles futures, telles que l'automatisation de véhicules se déplaçant dans des forêts, des exploitations agricoles ou des chantiers de construction, de robots anti-incendie ou de déminage, afin d'éviter au maximum l'intervention humaine dans des tâches fastidieuses ou dangereuses. On peut également envisager dans un futur proche la réalisation en série de certaines fonctions de pilotage sur les automobiles, comme les régulateurs de vitesse, déjà largement répandus aux États-Unis.

Problèmes posés par le pilotage de robots mobiles en tout-terrain

Les robots mobiles à roues ont fait l'objet de nombreux travaux : conception mécanique, systèmes de vision et de localisation, planification du déplacement, et commande proprement dite. L'élaboration de cette dernière pose en effet de grandes difficultés : un véhicule à roues est un système intrinsèquement non linéaire de par sa cinématique et ses caractéristiques dynamiques

(actionneurs, moteur thermique) ; de plus, et particulièrement en tout-terrain, il peut être soumis à de nombreuses perturbations, tels que des changements d'adhérence du terrain ou de pente, des coups de vent, etc. La commande d'un tel système est donc un problème qui, pour être résolu de façon satisfaisante, doit prendre ces *non-linéarités* et ces *perturbations* en considération.

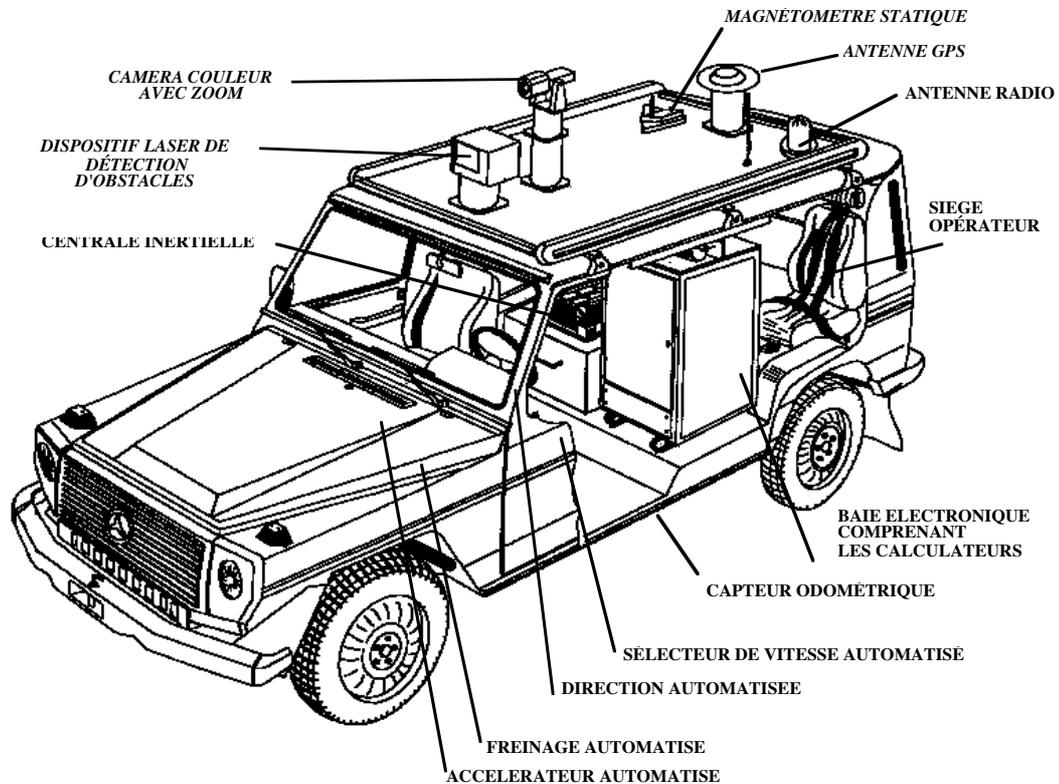


Figure 4 : Le robot REMI.

Définition de la tâche de pilotage

La mission du robot, définie au départ par un opérateur humain, consiste en une destination finale et des contraintes temporelles. À bord du robot se trouve un module de *planification* qui, en fonction de la position du robot, de son environnement et de sa mission, élabore une liste de points de passage assortis d'une vitesse moyenne. Cette liste est transmise à un module de *guidage* qui la transforme en une séquence de positions, d'attitudes, et de vitesses de consigne. La fonction de *pilotage* consiste en l'élaboration des commandes des actionneurs de la direction (le volant) et de la vitesse (l'accélérateur et les freins) de manière à respecter ces consignes.

Or, dans toutes les études précédemment effectuées par la SAGEM, le pilotage du véhicule était réalisé de façon classique, c'est-à-dire principalement à l'aide de techniques *linéaires* : le but de la collaboration avec le Laboratoire d'Électronique de l'ESPCI était l'évaluation des méthodes de modélisation et de commande *non linéaires neuronales* pour réaliser les lois de commande du pilotage (voir les correcteurs neuronaux représentés Figure 5).

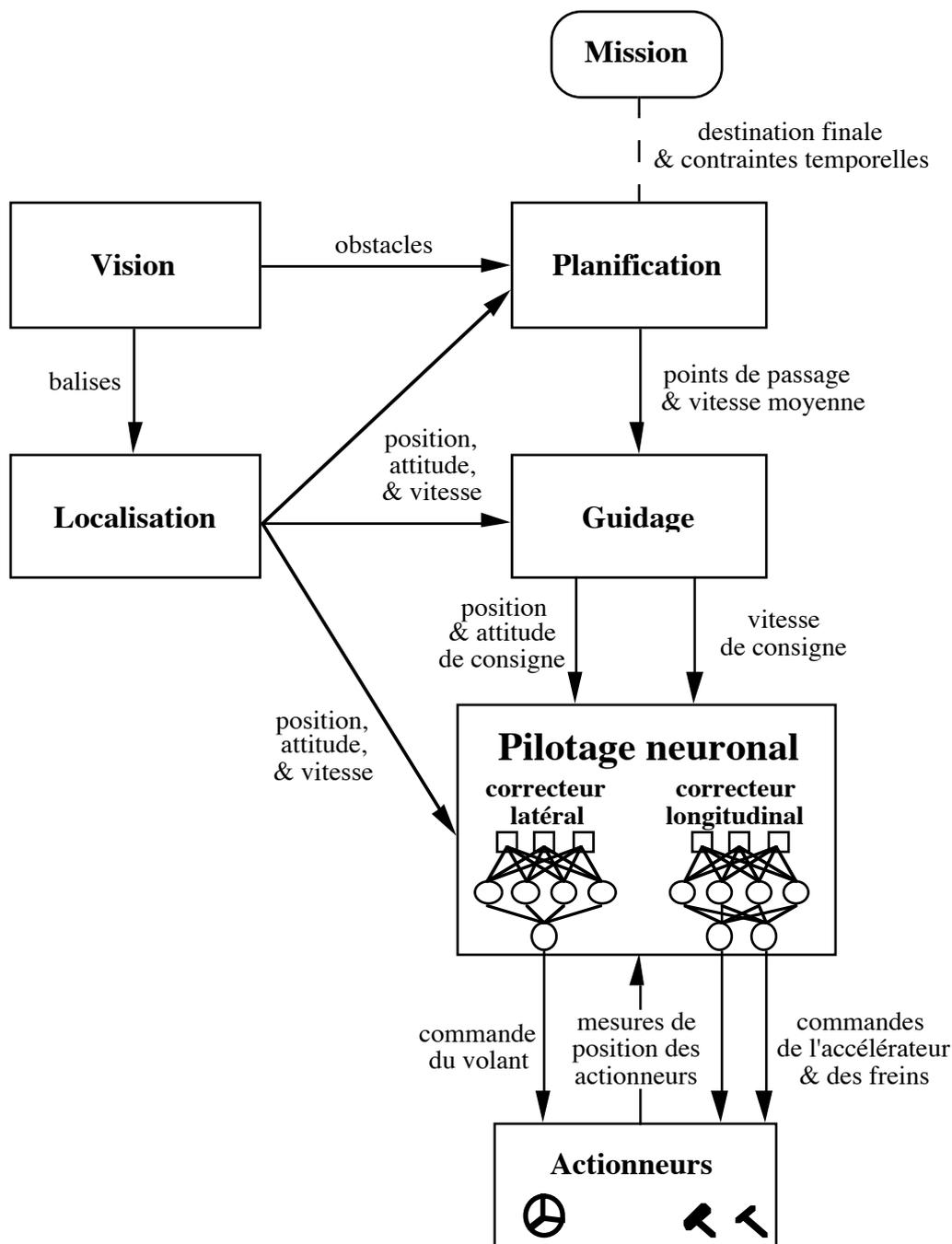


Figure 5 : Architecture d'un robot mobile autonome piloté par réseaux de neurones.

La modélisation et la commande par réseaux de neurones

La démarche neuronale consiste tout d'abord à modéliser le comportement du processus à l'aide d'un réseau de neurones bouclé. Le but est d'obtenir un modèle aussi précis que possible, c'est-à-dire éventuellement meilleur que les modèles généraux répertoriés dans la littérature, et prenant en compte les non-linéarités et caractéristiques du robot particulier étudié (saturations, dynamique des actionneurs...).

À l'aide de ce modèle, on peut alors effectuer l'apprentissage d'un réseau de neurones correcteur de manière à ce que le comportement dynamique du robot respecte le cahier des charges. Comme le pilotage concerne à la fois la direction et la vitesse du robot (ainsi qu'il apparaît sur la Figure 5),

nous avons en fait élaboré deux modèles, un modèle du comportement latéral et un modèle du comportement longitudinal, et mis au point les deux correcteurs correspondants.

1) Modélisation de processus

Il s'agit de réaliser l'apprentissage d'un modèle dynamique neuronal du processus, un réseau bouclé de coefficients C_m . Les séquences d'entrées et de sorties désirées définissant la tâche de modélisation sont la séquence des entrées de commande $\{u(k)\}$ appliquées au robot et les sorties du processus $\{y_p(k)\}$ mesurées pendant le fonctionnement (voir Figure 6).

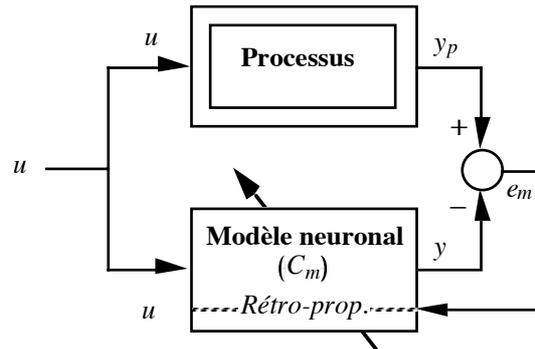


Figure 6 : Système d'apprentissage d'un modèle neuronal.

Le but de l'apprentissage est d'ajuster les coefficients C_m du réseau modèle de manière telle que ses sorties $\{y(k)\}$ soient aussi proches que possible des sorties désirées $\{y_p(k)\}$. Pour cela, on définit une fonction de coût J_m , qui est la somme des carrés des différences entre les sorties du réseau et les sorties désirées :

$$J_m = \sum_{k=1}^T e_m(k)^2 = \sum_{k=1}^T (y_p(k) - y(k))^2$$

On modifie itérativement les coefficients C_m de manière à minimiser la fonction de coût J_m selon $\Delta C_m = \mu D$, où D est une direction de descente, et μ un scalaire (le pas dans la direction de descente) ; en général, on calcule le gradient de la fonction de coût à l'aide de l'algorithme dit de *rétropropagation*, et on l'utilise soit directement comme direction de descente, soit plus efficacement comme ingrédient d'une méthode du second ordre (méthode de quasi-Newton par exemple).

Application au véhicule REMI

Dans le cas du modèle latéral, l'entrée est la commande du volant et la sortie le cap du véhicule. Le modèle neuronal obtenu a l'avantage sur son homologue classique, un simple modèle "bicyclette", de modéliser précisément le comportement dynamique de l'actionneur du volant, un moteur à courant continu sur la colonne de direction (qui présente des saturations en vitesse angulaire et en position), ainsi que la non-linéarité géométrique du mécanisme de direction.

Dans le cas du modèle longitudinal, les entrées sont les commandes du frein et de l'accélérateur, et la sortie, la vitesse. Nous avons élaboré un modèle longitudinal qui présente des performances analogues à celles du modèle utilisé habituellement, mais offre l'avantage d'une plus grande

simplicité et d'une compacité remarquable (les modèles latéraux et longitudinaux ne comportent que cinq neurones chacun).

Notons que, contrairement à une idée largement répandue selon laquelle la modélisation neuronale ne permet de mettre au point que des modèles "boîte noire", c'est-à-dire dont les paramètres n'ont aucune signification physique, la démarche neuronale peut, comme les approches classiques, tirer profit de l'étude des phénomènes physiques mis en jeu, et intégrer ces connaissances dans les modèles (en mêlant aux neurones standards - à tangente hyperbolique - des "neurones" réalisant des fonctions déterminées, en fixant les valeurs de certains coefficients...); nous avons de cette manière utilisé avec profit les modèles mathématiques disponibles.

2) Commande de processus

La mise au point d'un système de commande non adaptatif suppose deux phases, une phase préalable d'apprentissage au cours de laquelle sont estimés les coefficients du réseau correcteur, et une phase d'utilisation, où le correcteur, dont les coefficients sont fixés, est mis en œuvre avec le processus.

a) Phase d'apprentissage

Il s'agit de réaliser l'apprentissage d'un correcteur neuronal à l'aide du modèle précédemment établi. Les séquences d'entrées et de sorties désirées définissant la tâche de commande sont la séquence des entrées de consigne $\{r(k)\}$ et les sorties $\{y_r(k)\}$ d'un modèle de référence définissant le comportement dynamique désiré pour le système de commande, conformément au cahier des charges (voir Figure 7).

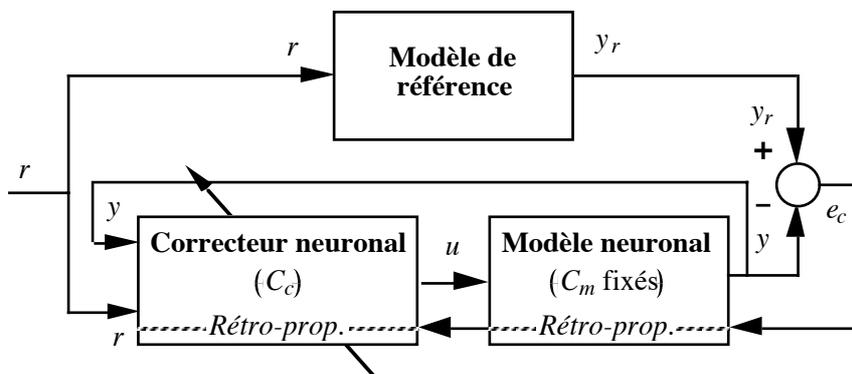


Figure 7 : Système d'apprentissage d'un correcteur neuronal.

De manière similaire à la modélisation, le but de l'apprentissage est d'ajuster les coefficients C_c du réseau correcteur de manière telle que les sorties du modèle neuronal $\{y(k)\}$ soient aussi proches que possible des sorties désirées $\{y_r(k)\}$. Pour cela, on définit une fonction de coût J_c , qui est la somme des carrés des différences entre les sorties du réseau et les sorties désirées :

$$J_c = \sum_{k=1}^T e_c(k)^2 = \sum_{k=1}^T (y_r(k) - y(k))^2$$

De même, on modifie itérativement les coefficients C_c de manière à minimiser la fonction de coût J_c . Ici, le modèle neuronal est nécessaire pour évaluer la valeur du gradient par rapport aux coefficients C_c du correcteur.

b) Phase d'utilisation

Après la phase d'apprentissage, le correcteur est utilisé avec le processus. La solution la plus simple consiste en un simple bouclage avec ce dernier (voir Figure 8). Dans ces systèmes de commande, le signal de rétroaction est la sortie mesurée y_p du processus.

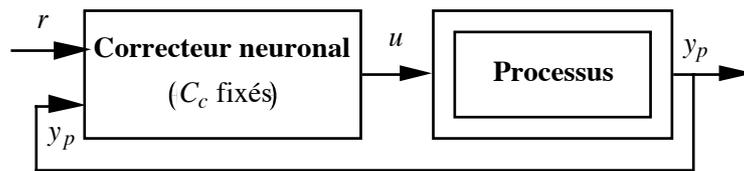


Figure 8 : Système de commande en simple bouclage.

Si le processus est susceptible de subir des perturbations importantes, ou si le modèle est peu fidèle, nous avons montré qu'il est intéressant de mettre en œuvre le correcteur neuronal dans un système de commande *avec modèle interne*, qui est le modèle neuronal utilisé pour l'apprentissage du correcteur (voir Figure 9). Le principal signal de rétroaction est ici la différence e_{mi} entre la sortie du modèle interne et celle du processus. Cette différence est représentative des défauts de modélisation et de l'effet de perturbations, et sa prise en considération rend le système de commande *robuste* vis-à-vis de ces défauts et des perturbations. Ce type de système de commande n'est pas particulier aux réseaux de neurones, mais l'utilisation de ces derniers permet d'améliorer considérablement les performances de la commande avec modèle interne dans le cas de processus non linéaires.

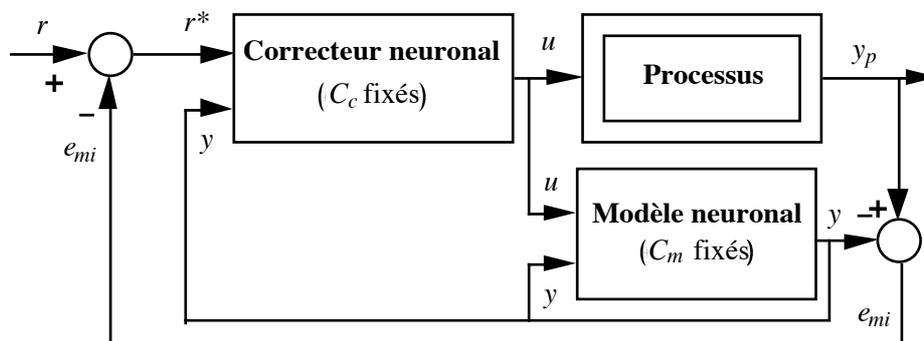


Figure 9 : Système de commande avec modèle interne.

Application au véhicule REMI

Pour la commande latérale, l'entrée est constituée de la position et de l'attitude de consigne. Nous avons exprimé le problème de l'asservissement latéral comme un problème de commande optimale au sens d'un coût quadratique, méthode qui se prête donc bien à l'élaboration d'un

correcteur neuronal selon les principes présentés plus haut. La recherche *ab initio* d'une loi de commande optimale, étendue à des modèles non linéaires et à l'utilisation d'un correcteur neuronal, est une généralisation de la commande linéaire quadratique. Le modèle latéral étant très précis, ce système a été mis en œuvre en simple bouclage.

Pour la commande longitudinale, l'entrée est la consigne de vitesse. Le problème de la poursuite de la consigne de vitesse était plus délicat en raison de la moindre précision du modèle longitudinal du véhicule. Nous avons donc développé un système de commande neuronal avec modèle interne, afin de pallier les imperfections du modèle longitudinal, ainsi que l'effet des perturbations de pente ou de changements de terrain.

À l'instar des modèles, les correcteurs ne mettent en œuvre qu'une dizaine de neurones, et ne posent aucun problème de portabilité (ils ont été implantés en langage C sur un processeur 68030).

3) Performances obtenues et comparaison avec le système classique

Le module de pilotage neuronal a été testé à grande vitesse sur route (jusqu'à 72 km/h), ainsi qu'en tout-terrain sur des pistes d'essais de caractéristiques calibrées de l'ETAS (Établissement Technique d'Angers). Les trajets présentaient de fortes variations de courbure (courbure maximale de $0,1 \text{ m}^{-1}$), des pentes allant jusqu'à 40%, et des dévers maximaux de 30%, sur bitume et en tout-terrain. Les performances du système de commande du cap se sont révélées excellentes, avec une erreur latérale n'excédant jamais 40 cm. Le système de commande de la vitesse avec modèle interne garantit une erreur statique nulle, même en présence de perturbations importantes (pente de 40%), et un écart-type de l'erreur de l'ordre de 0.3 m/s en tout-terrain.

La supériorité du système neuronal sur son homologue classique est due principalement à la prise en considération des non-linéarités du véhicule *via* le modèle : celle de l'actionneur du volant (saturations en vitesse et en position) et du moteur thermique (aux grandes accélérations et ouvertures du papillon des gaz). Toutefois, si la performance du système de commande du volant est supérieure à celle du correcteur classique, celle du système de commande de la vitesse est équivalente. Cependant, le système neuronal avec modèle interne est potentiellement mieux adapté au pilotage en tout-terrain, car il est plus robuste vis-à-vis des perturbations de terrain de par sa conception même.

Conclusion

Nous avons présenté ici les schémas les plus simples d'apprentissage et de mise en œuvre des réseaux de neurones pour la modélisation et la commande de processus. Il peut être nécessaire en effet d'envisager des architectures adaptatives (les coefficients du correcteur sont alors ajustés en permanence pendant son utilisation avec le processus), comme c'est le cas si le processus est non stationnaire. Mais quel que soit le type de système, adaptatif ou non, la démarche neuronale présente l'avantage d'être indépendante de la complexité des modèles et, partant, des correcteurs à réaliser. La facilité d'apprentissage et de mise en œuvre des réseaux de neurones est l'un de leurs atouts principaux pour la modélisation et la commande de processus dont la complexité provient de

non-linéarités. De plus, nous avons vu que la conception des réseaux de neurones peut facilement intégrer les informations concernant le processus à commander prodiguées par les experts, démarche qui permet notamment d'augmenter encore leur compacité. Toutefois, l'application des réseaux de neurones à l'automatique peut toujours bénéficier des résultats et des méthodes de l'automatique non linéaire classique, en particulier pour la démonstration des propriétés de stabilité et de robustesse des systèmes neuronaux.

Bibliographie commentée

Pour une introduction générale aux réseaux de neurones, voir l'ouvrage :

Bishop C. M. (1995) Neural Networks for Pattern Recognition, Clarendon Press, Oxford.

Pour une première approche de la modélisation et de la commande, voir par exemple l'article :

Rivals I., Personnaz L., Dreyfus G., Ploix J.-L. (1995) "Modélisation, classification et commande par réseaux de neurones : principes fondamentaux, méthodologie de conception, et illustrations industrielles", in Récents progrès en génie des procédés 9, Lavoisier technique et documentation, Paris.

et, pour une approche plus approfondie, le mémoire de thèse :

Nerrand O. (1992) "Réseaux de neurones pour le filtrage adaptatif, l'identification et la commande de processus", Thèse de doctorat de l'Université Paris VI.

Les travaux concernant spécifiquement le pilotage du véhicule ont fait l'objet de deux articles, pour les pilotages respectivement latéral et longitudinal :

Rivals I., Personnaz L., Dreyfus G., Canas D. (1993) "Real-time control of an autonomous vehicle : a neural network approach to the path following problem", Proceedings of the 5th International Conference on Neural Networks and their Applications (NeuroNîmes'93).

Rivals I., Canas D., Personnaz L., Dreyfus G. (1994) "Modeling and control of mobile robots and intelligent vehicles by neural networks", Proceedings of the IEEE Conference on Intelligent Vehicles, 24-26 octobre 1994, Paris.

ainsi que du deuxième volume de ma thèse :

Rivals I. (1995) Modélisation et commande de processus par réseaux de neurones; application au pilotage d'un véhicule autonome, Thèse de Doctorat de l'Université Paris VI.

Les aspects théoriques de la modélisation neuronale sont développés dans le premier volume de cette même thèse, ainsi que dans le chapitre de livre :

Rivals I., Personnaz L. (1996) "Black-box modeling with state-space neural networks", in Neural Adaptive Control Technology I, R. Zbikowski and K. J. Hunt eds., World Scientific.

Isabelle Rivals est ingénieur ESPCI (École Supérieure de Physique et de Chimie Industrielles), et docteur en Sciences Physiques de l'Université Paris VI. Elle a soutenu sa thèse, qui faisait l'objet d'une collaboration avec la société SAGEM (contrat CIFRE), sur le sujet "Modélisation et commande de processus par réseaux de neurones ; application au pilotage d'un véhicule autonome". Elle est actuellement Maître de Conférences au Laboratoire d'Électronique de l'ESPCI, où elle poursuit des recherches sur la modélisation et la commande de processus par réseaux de neurones.

Le Laboratoire d'Électronique de l'ESPCI mène des recherches dans le domaine des réseaux de neurones depuis 1982, domaine où il a joué le rôle de pionnier. Depuis, ses activités sont tournées d'une part vers la modélisation et la commande non linéaires de processus, et la classification automatique, et, d'autre part, vers la modélisation de systèmes neuronaux vivants.