

Chapitre II

Identification et Commande

par

les réseaux de neurones

II.1 Identification avec réseaux de neurones

Comme approximateurs universels, les réseaux de neurones ont des capacités de modélisation très intéressantes. L'application des techniques neuronales pour l'identification des systèmes non linéaires peut fournir de nouvelles solutions pour ce problème.

II.1.1 Introduction

L'identification d'un système non linéaire dont la structure peut être inconnue consiste à trouver un ensemble d'équations non linéaires qui représentent d'une manière adéquate son comportement dynamique.

Plusieurs approches ont été proposées pour la modélisation des systèmes non linéaires à structure inconnue telles que :

- la modélisation par les séries de Volterra et Wiener ;
- la modélisation par les modèles poly nominaux d'Ivakhninko (GMDH : Group Method of Data Handling) ;
- la modélisation par des règles floues ;

Si bien qu'il existe plusieurs méthodes d'identification des systèmes linéaires et même pour les systèmes non linéaires à structure connue comme les méthodes de gradient et les filtres non linéaires, des méthodes générales et efficaces qui peuvent être appliquées pour l'identification des systèmes non linéaires à structure inconnue ne sont pas encore disponibles.

Avec leurs propriétés remarquables d'approximation, d'adaptativité et de généralisation, les réseaux de neurones peuvent fournir des solutions prometteuses pour ce problème. Cette approche a été adoptée par plusieurs chercheurs, son efficacité et ses performances ont été démontrées à travers plusieurs applications.

II.1.2 Propriété d'approximation des réseaux de neurones

L'entraînement d'un réseau de neurones en utilisant les données entrées – sorties d'un système non linéaire peut être considéré comme un problème d'approximation de fonctions non linéaires. A partir du théorème de Weierstrass, il est connu que les polynômes et d'autres structures d'approximation peuvent approcher une fonction continue quelconque avec une précision arbitraire. Récemment, l'application des techniques mathématiques pour l'étude des capacités d'approximation des réseaux de neurones était l'objet de plusieurs travaux.

Les résultats obtenus montrant les perceptrons multi-couches peuvent approcher n'importe quelle fonction continue, et précisent qu'un réseau à une seule couche cachée dont la fonction d'activation est la fonction sigmoïde peut approcher une fonction continue définie sur un sous ensemble compact de $\mathbb{R}^n$. L'utilisation d'autres fonctions autres que les sigmoïdes est possible et non constantes.

Si bien que ces résultats semblent être intéressants, ils ne sont pas suffisants pour caractériser les propriétés d'approximation des réseaux. En plus ces résultats supposent que ces réseaux contiennent un nombre suffisant de neurones cachés et que les poids des connexions sont correctement choisis. La possibilité de détermination des poids appropriés en utilisant les algorithmes d'apprentissage existants est un problème ouvert. Une étude mathématique plus approfondie est nécessaire pour permettre la définition du type et de la structure convenable pour un problème donné.

II.1.3 Identification

L'idée de base de l'approche connexionniste est de substituer aux modèles paramétriques classiques des modèles neuronaux dont les paramètres sont adaptés par une procédure d'apprentissage appropriée en utilisant les données entrées-sorties du système. En l'absence de résultats théoriques concrets concernant la stabilité et l'identifiabilité du système c'est-à-dire la capacité de la structure du modèle choisie à représenter le système d'une manière adéquate, on suppose que tous les systèmes qu'on est entrain d'étudier appartiennent à la classe de systèmes que le réseau choisi est capable de les représenter.

II.1.3.1 Modélisation directe

La modélisation directe consiste à entraîner un réseau de neurones pour représenter la dynamique directe du système. Dans ce cas le modèle est placé en parallèle avec le système. L'erreur entre les sorties est utilisée pour l'adaptation des paramètres du modèle. Cette structure correspond à un problème d'apprentissage supervisé classique, dans lequel le « professeur » (dans ce cas le système) génère les sorties désirées. Dans le cas des réseaux multi-couche, on utilise généralement la méthode de rétropropagation pour l'adaptation des poids.

La modélisation des caractéristiques dynamiques du système peut être faite par l'introduction de ces caractéristiques dans le réseau lui-même, par l'utilisation de réseaux récurrents, ou par l'introduction d'un comportement dynamique dans les neurones.

L'approche la plus évidente consiste à augmenter la dimension du vecteur d'entrée du réseau en ajoutant les signaux correspondant aux mesures précédentes des entrées et des sorties. supposons que le système à identifier est gouverné par l'équation aux différences non linéaire suivante :

$$Y(t+1) = f[Y(t), \dots, Y(t-n_y+1); U(t), \dots, U(t-n_u+1)] + e(t+1) \quad (\text{II.1})$$

Avec

$$Y(t) = [Y(t), \dots, Y_m(t)]^T \quad (\text{II.2})$$

$$U(t) = [U_1(t), \dots, U_r(t)]^T \quad (\text{II.3})$$

$$e(t) = [e_1(t), \dots, e_m(t)]^T \quad (\text{II.4})$$

Où $f(\cdot)$ est une fonction non linéaire, $Y(t)$, $U(t)$ et $e(t)$ sont les vecteurs de sortie, d'entrée et du bruit de mesure. A l'instant $t+1$, la sortie du système dépend (dans le sens définit par la fonction non linéaire f) des n_y valeurs précédentes de la sortie et des n_u valeurs précédentes de l'entrée. Le but de l'approche connexionniste est d'approcher la fonction f avec un réseau de neurones.

En s'inspirant des techniques d'identification classiques, il est possible d'utiliser deux structures d'identification.

II.1.3.1.1 Identification série parallèle

Dans la première structure, représentée par la figure (II.1), on utilise des réseaux non récurrents (statiques) dont la structure entrée-sortie est donnée par :

$$Y^m(t) = \hat{f}[X^m(t)] \quad (\text{II.5})$$

$X^m(t)$ et $Y^m(t)$ sont les vecteurs d'entrée et de sortie du réseau. L'entrée à un instant t est :

$$X^m(t) = [Y^T(t-1), \dots, Y^T(t-n_y); U^T(t-1), \dots, U^T(t-n_u)]^T \quad (\text{II.6})$$

La dimension du vecteur d'entrée :

$$n_I = m.n_y + r.n_u \quad (II.7)$$

Le but de la procédure d'apprentissage est d'entraîner le réseau pour que la transformation non linéaire $\hat{f}$ soit une approximation de la fonction f .

Dans cette structure, l'entrée du modèle comporte les valeurs décalées de la réponse du système réel, elle correspond au modèle d'identification série-parallèle de l'identification classique.

Comme dans le cas des méthodes classiques, il apparaît une contrainte imposée a priori sur la connaissance de l'ordre du système (ou d'une borne supérieure sur cet ordre).

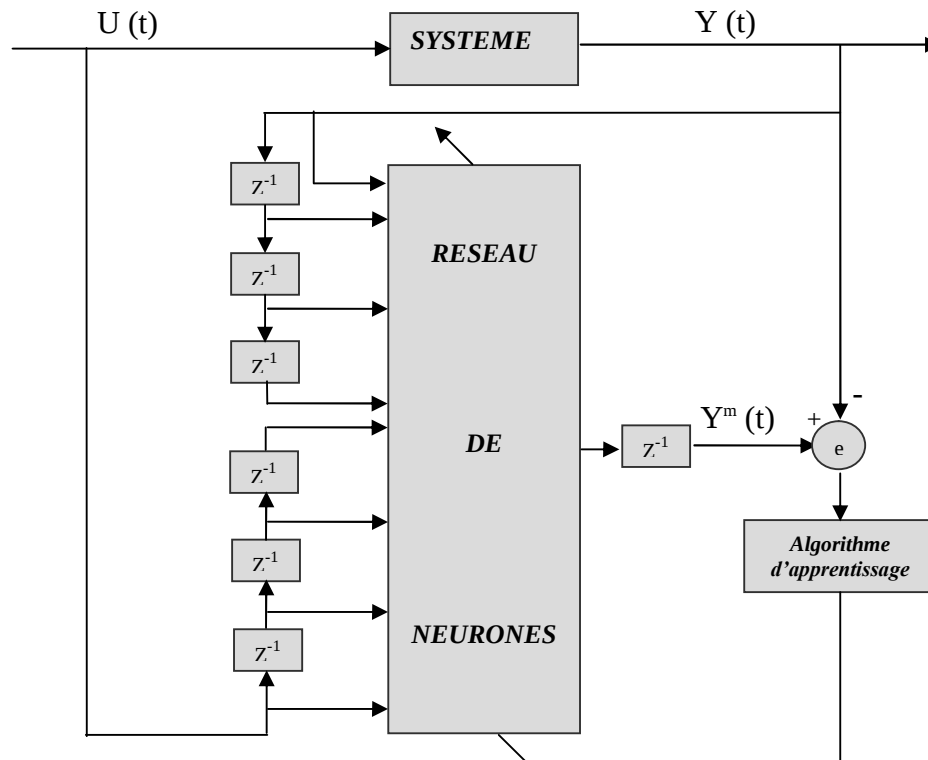


Figure (II.1) Structure d'identification série-parallèle

II.1.3.1.2 Identification parallèle

Dans la deuxième structure, représentée par la figure (II.2), on utilise des réseaux récurrents, l'entrée comporte les valeurs décalées de la réponse du modèle. Elle correspond au modèle d'identification parallèle dans ce cas, la sortie du modèle neuronal est donnée par :

$$Y^m(t) = \hat{f}[X^m(t)] \quad (\text{II.8})$$

Avec

$$X(t) = [Y^{mT}(t-1), \dots, Y^{mT}(t-n_y); U(t-1), \dots, U(t-n_u)]^T \quad (\text{II.9})$$

Cependant il n'y a aucune garantie de convergence si la structure parallèle est utilisée. Malgré plusieurs années de recherche, les conditions de convergence du modèle parallèle ne sont pas encore établies même dans le cas des systèmes linéaires.

Grâce à ses propriétés de stabilité et de convergence l'utilisation de la structure série-parallèle est préférable.

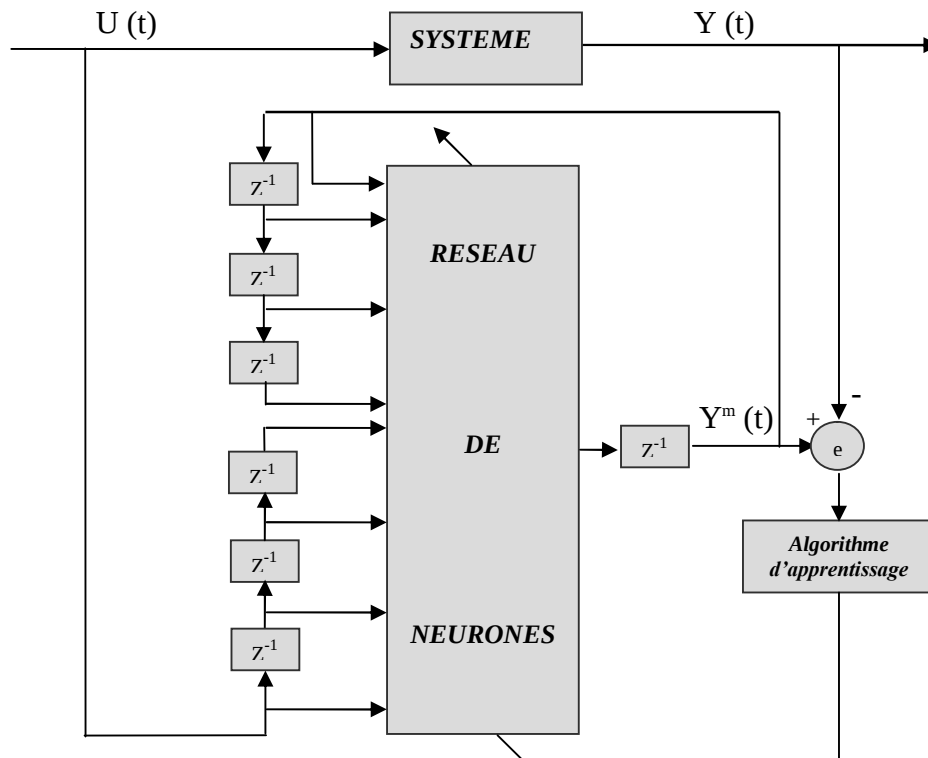


Figure (II.2) Structure d'identification parallèle

L'avantage du modèle parallèle est qu'il présente des performances meilleures dans le cas de perturbations puisqu'il permet d'éviter les problèmes du biais causés par les bruits sur la sortie du système réel.

Généralement, lors d'une procédure d'identification, on commence par une identification série-parallèle. En supposant qu'après une période d'entraînement convenable le modèle neuronal représente d'une manière adéquate le comportement du système (les sorties du modèle sont suffisamment proches de celles du système), le modèle série parallèle peut être remplacé par un modèle parallèle. Ceci a un avantage pratique si le modèle d'identification est utilisé par exemple pour des procédures d'entraînement hors ligne d'un contrôleur neuronale.

II.1.3.2 Modèle inverse

Les modèles inverses des systèmes dynamiques jouent un rôle très important dans les structures de contrôle. La modélisation inverse par réseaux de neurones consiste à entraîner un modèle neuronal pour qu'il génère les commandes nécessaires permettant de déplacer le système de son état actuel $Y(t)$ à un état désiré $Y^d(t+1)$.

A partir de l'équations (II.1) décrivant le comportement dynamique du système, son comportement dynamique inverse peut être décrit par :

$$U(t) = f^{-1}[Y(t+1), \dots, Y(t-n_y+1); U(t-1), \dots, U(t-n_u+1)] \quad (\text{II.10})$$

La fonction inverse f^{-1} nécessite la connaissance de la valeur future $Y(t+1)$, cette valeur est remplacée par la valeur désirée $Y^d(t+1)$ qui est supposée disponible à l'instant t . La relation entrée-sortie du modèle inverse neuronal est donnée par :

$$U^m(t) = \hat{f}^{-1}[X^m(t)] \quad (\text{II.11})$$

Avec

$$X^m(t) = [Y^T(t), \dots, Y^T(t-n_y+1); U^T(t-1), \dots, U^T(t-n_u+1); Y_d^T(t+1)]^T \quad (\text{II.12})$$

La dimension du vecteur d'entrée dans ce cas est :

$$n_I = m.(n_y + 1) + r.n_u \quad (\text{II.13})$$

Le but de la procédure d'entraînement est l'adaptation des paramètres du réseau pour qu'il approche la fonction inverse f^{-1} .

Cependant la commande désirée n'est pas connue a priori. Il faut donc arriver à évaluer l'erreur sur la commande qui sera ensuite utilisée pour l'entraînement du réseau. On peut utiliser pour cela plusieurs structures d'apprentissage.

II.1.3.2.1 Apprentissage généralisé

L'approche la plus simple est la modélisation inverse directe illustrée par la figure (II.3), cette structure est appelée structure d'apprentissage généralisé.

Dans ce cas, en utilisant la connaissance qualitative que l'on a sur le comportement du système, des signaux d'entraînement compatibles avec son fonctionnement lui sont appliqués. La sortie du système sert alors d'entrée au modèle neuronal dont la sortie est comparée avec le signal d'entraînement. En utilisant cette erreur pour l'entraînement du réseau, cette structure force le réseau à apprendre le comportement inverse du système.

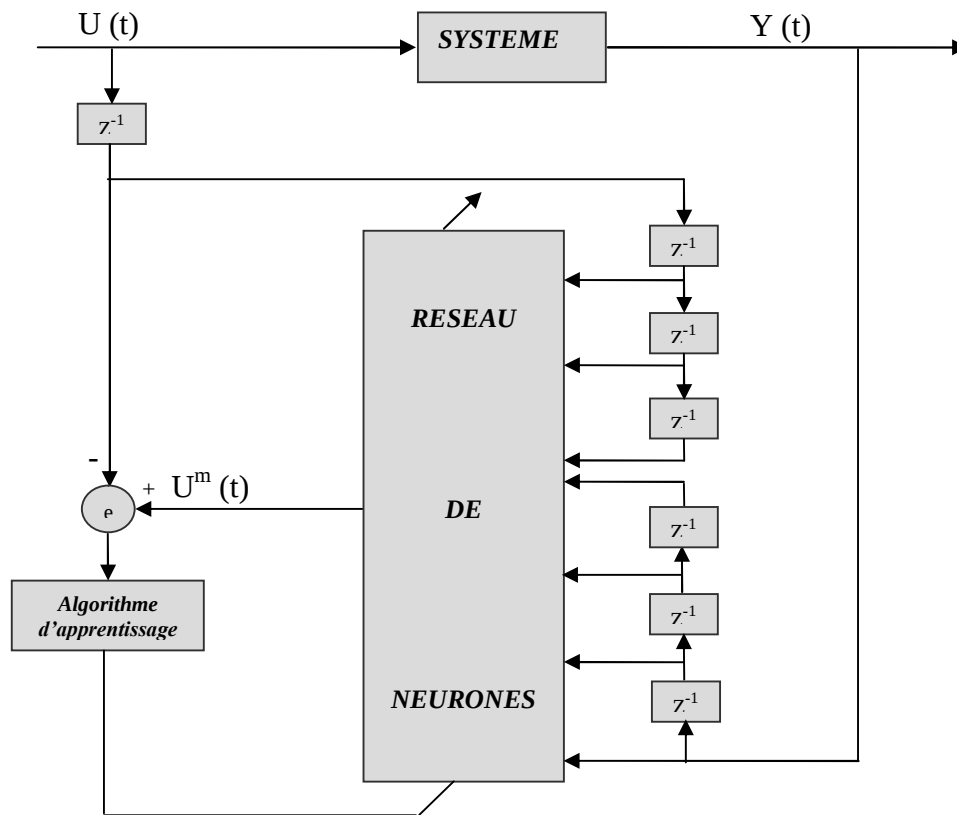


Figure (II.3) Structure d'apprentissage généralisé

Une fois entraîné, le réseau doit être capable de produire la commande nécessaire assurant de déplacer l'état du système à un état suffisamment proche de l'état désiré.

Pendant cette structure souffre de deux inconvénients majeurs :

- La procédure d'apprentissage n'utilise pas des commandes désirées implicites. Puisque les commandes opérationnelles sont difficiles à définir a priori le signal d'entraînement doit être choisi sur un intervalle de l'espace d'entrée du système plus large que celui nécessaire.
- Si le système n'est pas une correspondance un à un on peut obtenir un modèle inverse incorrect.

Le premier point est étroitement lié au concept général d'excitations persistantes qui concerne le choix des entrées utilisées pour l'entraînement. Il faut trouver les conditions assurant les excitations persistantes c'est -à- dire les excitations qui assurent la convergence des paramètres. Pour les réseaux de neurones, les méthodes de caractérisation des excitations persistantes seront très importantes. Une étude préliminaire a été fait par Narendre et Parthasarathy.

Le succès de cette méthode est lié a la capacité du réseau à généraliser et à apprendre à répondre correctement aux entrées pour les quelles il n'a pas été entraîné. Pour surmonter ses difficultés, cette méthode peut être combinée avec une méthode d'apprentissage spécialisée.

II.1.3.2.2 Apprentissage spécialisé

Une seconde approche de modélisation inverse dont le but est de dominer les problèmes cités précédemment est connue sous le nom de l'apprentissage spécialisé, sa structure est représentée par la figure (II.4).

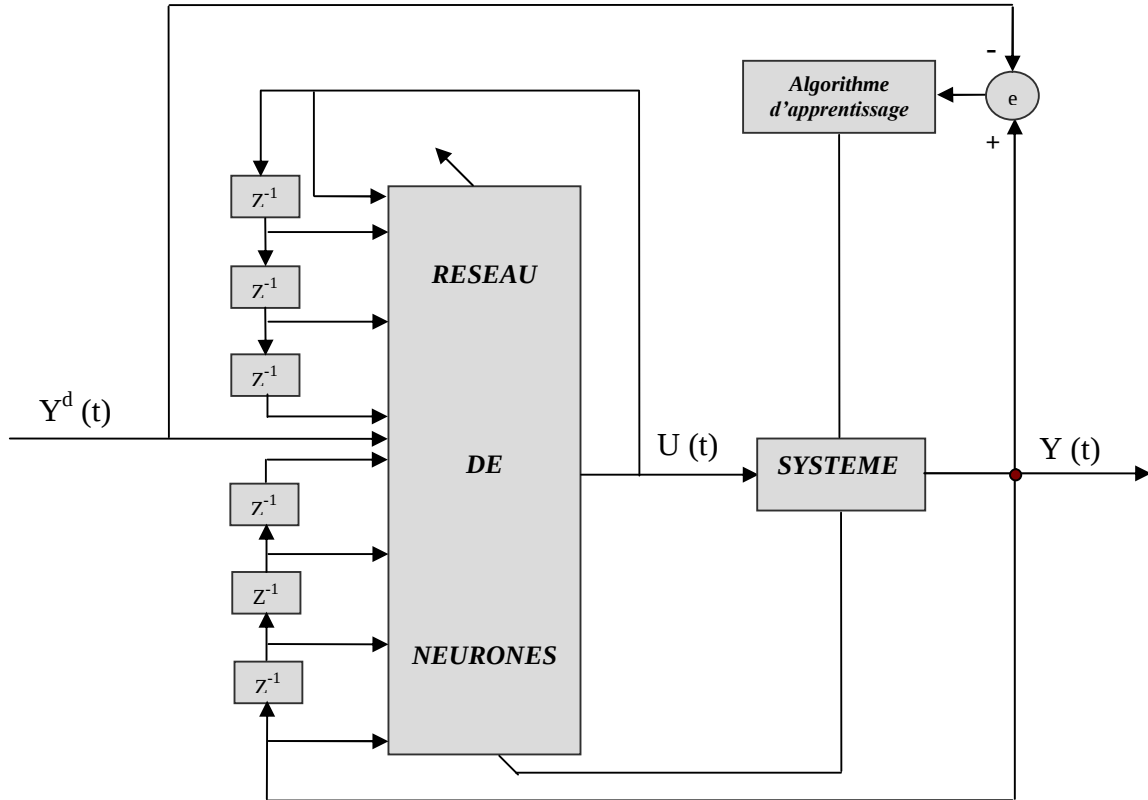


Figure (II.4) Structure d'apprentissage spécialisé

Dans cette approche le modèle inverse neuronal est placé en série avec le système, et reçoit à son entrée un signal d'entraînement appartenant à l'espace des sorties opérationnelle désirée du système (le système de référence ou le signal de commande).

Dans le cas d'un entraînement par rétropropagation, le système est considéré comme une couche supplémentaire dont les poids sont fixes. Le but de la procédure d'apprentissage est de minimiser l'erreur E sur les sorties du modèle inverse, qui est donnée par :

$$E = \frac{1}{2} \sum_{i=1}^r (U_i - U_i^d)^2 \quad (\text{II.14})$$

Or les commandes désirées ne sont pas connues, il faut donc évaluer les erreurs sur la commande à partir des erreurs sur la sortie.

L'erreur sur la sortie E' est donnée par :

$$E' = \frac{1}{2} \sum_{i=1}^m (Y_i(t) - Y_i^d(t))^2 \quad (\text{II.15})$$

La dérivée de l'erreur par rapport à un poids W_{ij} est :

$$\frac{\partial E'}{\partial W_{ij}(t)} = \sum_{i=1}^r \frac{\partial E'}{\partial U_i(t)} \frac{\partial U_i(t)}{\partial W_{ij}(t)} \quad (\text{II.16})$$

La dérivée de l'erreur E' par rapport à la commande :

$$\frac{\partial E'}{\partial W_{ij}} = \sum_{i=1}^r \frac{\partial E'}{\partial U_i(t)} \frac{\partial U_i(t)}{\partial W_{ij}(t)} \quad (\text{II.17})$$

L'évaluation de l'erreur nécessite le calcul du Jacobien du système.

Le Jacobien du système peut être obtenu par l'approximation des dérivées partielles des sorties du système par rapport à ses entrées, ou par l'utilisation d'un modèle direct entraîné du système par rapport à ses entrées, ou par l'utilisation d'un modèle direct entraîné du système (un modèle neuronal par exemple) qui sera placé en parallèle avec le système.

Dans ce cas, le signal d'erreur pour l'algorithme d'entraînement peut être la différence entre le signal d'entraînement et la sortie du système, ou la différence entre le signal d'entraînement et la sortie de modèle direct. Jordan et Rumelhart ont montré que l'utilisation de la sortie du système réel peut produire un modèle inverse exact même si le modèle direct est inexact, ce qui n'est pas évidemment vrai lorsqu'on utilise la sortie du modèle direct. Mais l'avantage de l'utilisation de la sortie du modèle est que le système réel n'est pas nécessaire pendant la procédure d'entraînement, ce qui est très important lorsque l'utilisation du système réel n'est pas possible.

L'adaptation des poids est faite par la rétro-propagation de l'erreur à travers le modèle direct et le modèle inverse, seuls les poids du modèle inverse sont ajustés pendant d'entraînement [4].

II.2 Commande

II.2.1 Introduction

La théorie de contrôle fournit des outils d'analyse et de synthèse adaptés aux systèmes linéaires. En pratique, l'application des techniques analytiques classiques basées sur la théorie des systèmes linéaires souffre de problèmes de robustesse à cause des incertitudes, perturbations et non linéarités.

Le traitement massivement parallèle, les capacités de modélisation et les possibilités d'apprentissage par expérience des réseaux de neurones sont des facteurs motivant pour le développement de système de contrôle à base des modèles connexionnistes.

Depuis leur apparition, les réseaux de neurones et en particulier les réseaux multicouches ont été appliqués dans plusieurs domaines vu leur capacité à commander les systèmes non linéaires et les systèmes dont les modèles sont peu connus ou ayant des paramètres variables dans le temps. De plus ils présentent une bonne résistance au bruit en entrée. Donc les applications des techniques neuronales peuvent fournir des solutions plus intelligentes avec des modérés.

Il existe plusieurs manières d'intégrer les réseaux de neurones dans les structures de commande des systèmes. Cela dépend essentiellement du schéma de réglage et de l'apprentissage choisi [4][18].

II.2.2 Les étapes de la conception d'un organe de commande

Commander un processus, c'est déterminer les commandes à lui appliquer, de manière assurer aux variables d'état ou aux sorties qui nous intéressent un comportement précisé par un cahier des charges. Ces commandes sont délivrées par un organe de commande ; le processus et son organe de commande constituent le système de commande. L'élaboration de l'organe de commande s'articule en plusieurs étapes.

II.2.2.1 Choix d'un modèle du processus

Un modèle du processus est nécessaire à la synthèse de l'organe de commande. La modélisation consiste à rassembler les connaissances que l'on a du comportement dynamique du processus, par une analyse physique des phénomènes mis en jeu, et une analyse des

données expérimentales. Ces analyses conduisent à la définition des grandeurs caractérisant le processus, c'est-à-dire ses entrées, ses variables d'état et ses sorties.

II.2.2.2 Estimation des paramètres du modèle (identification)

Le but de cette étape est de configurer et de sélectionner le meilleur modèle parmi différents modèles de la structure du modèle hypothèse (modèle choisi), sur la base d'un critère de performances. Le véritable but de cette démarche est la conception d'un organe de commande à partir d'un modèle, le meilleur d'entre eux est celui qui conduit aux meilleures performances du système de commande, au sens du cahier des charges. Il est évidemment plus économique, et donc préférable, de se fonder sur un critère qui ne nécessite pas la réalisation complète du système de commande pour sélectionner ce modèle : le meilleur modèle est défini comme celui dont l'erreur de prédiction est la plus faible. Pour obtenir ce modèle, il faut le chercher au sein d'une famille de modèles paramétrés. L'estimation des paramètres d'un modèle est donc effectuée de manière à minimiser l'erreur de prédiction, à partir de mesures effectuées sur le processus (ensemble d'apprentissage). La famille de modèles paramétrés que nous considérons dans ce mémoire est celle des réseaux de neurones, qui a l'intérêt de posséder la propriété d'approximation universelle. Leurs paramètres sont les coefficients des réseaux, et leur estimation correspond à la phase d'apprentissage. Dans le cadre de ce mémoire, les réseaux obtenus en fin d'identification sont essentiellement utilisés comme modèles de simulation pour la synthèse, des organes de commande.

II.2.2.3 Conception de l'organe de commande

Cette étape est celle du choix de l'architecture de l'organe de commande et de la structure des éléments qui le composent, choix effectué en fonction du modèle du processus mis au point et du cahier des charges, qui spécifie les performances désirées pour le système de commande. L'organe de commande comprend nécessairement un élément, le correcteur, qui effectue le calcul de la commande à appliquer au processus. Il comprend en général aussi d'autres éléments : un modèle de référence, un observateur, ou encore un " modèle interne " (émulateur).

II.2.2.4 Estimation des paramètres du correcteur

La dernière étape consiste à configurer le correcteur pour que l'organe de commande assure la tâche définie à l'étape précédente. Cette configuration est effectuée à l'aide du modèle : ceci caractérise les méthodes indirectes de synthèse du correcteur. Comme pour la modélisation, nous considérons des correcteurs réalisés par des réseaux de neurones, dont la structure a été fixée lors de la définition de l'organe de commande. L'estimation des coefficients du correcteur correspond à la phase d'apprentissage du réseau de neurones. Nous nous intéressons à la synthèse d'organes de commande non adaptatifs, c'est-à-dire pour lesquels l'apprentissage du correcteur est entièrement réalisé préalablement à son utilisation avec le processus. En réalité, la conception d'un organe de commande est une procédure itérative, surtout lorsque celui-ci n'est pas adaptatif. Car, comme nous l'avons signalé plus haut, un réseau modèle ne peut être définitivement rejeté ou validé qu'en fonction des performances du système de commande élaboré à partir de ce modèle [17].

II.2.3 Principe de la commande

La commande par réseaux de neurones consiste essentiellement à remplacer le régulateur par un réseau de neurones. Ce dernier préalablement entraîné va synthétiser la commande appropriée au système à commander.

Dans certaines situations, lorsqu'il est impossible de construire un contrôleur automatique en utilisant les techniques standard de la théorie de contrôle, les actions de contrôle sont délivrées par un opérateur humain pour réaliser des tâches particulières.

Dans quelque applications, il est peut être désirable de réaliser un contrôleur automatique qui imite les actions de l'opérateur humain. Un réseau de neurones peut être utilisé pour réaliser de telle tâche .le problème se réduit ainsi à un problème d'apprentissage supervisé. Dans ce cas, les entrées du réseau sont les informations que reçoit l'opérateur humain sur l'état du système. Les sorties désirées sont les actions délivrées par l'opérateur [18].

II.2.4 Les structures de contrôle

Malgré la diversité des structures de contrôle neuronal, nous allons présenter deux types différentes, telle que l'utilisation du modèle inverse et l'utilisation d'un modèle de référence

II.2.4.1 Contrôle par modèle inverse

II.2.4.1.1 Contrôleur Feed-forward

Dans ce type de contrôleur, on utilise un modèle inverse pour le contrôle direct du système, l'obtention d'un modèle inverse se fait par les techniques étudiées dans la partie de l'identification. Le modèle inverse, comme montre la figure (II.5) est placé en série avec le système à contrôler.

Les deux types d'apprentissage (généralisé et spécialisé) peuvent être utilisé pour l'entraînement du réseau, En principe, le modèle inverse est capable de générer la commande nécessaire pour réaliser la tâche désirée.

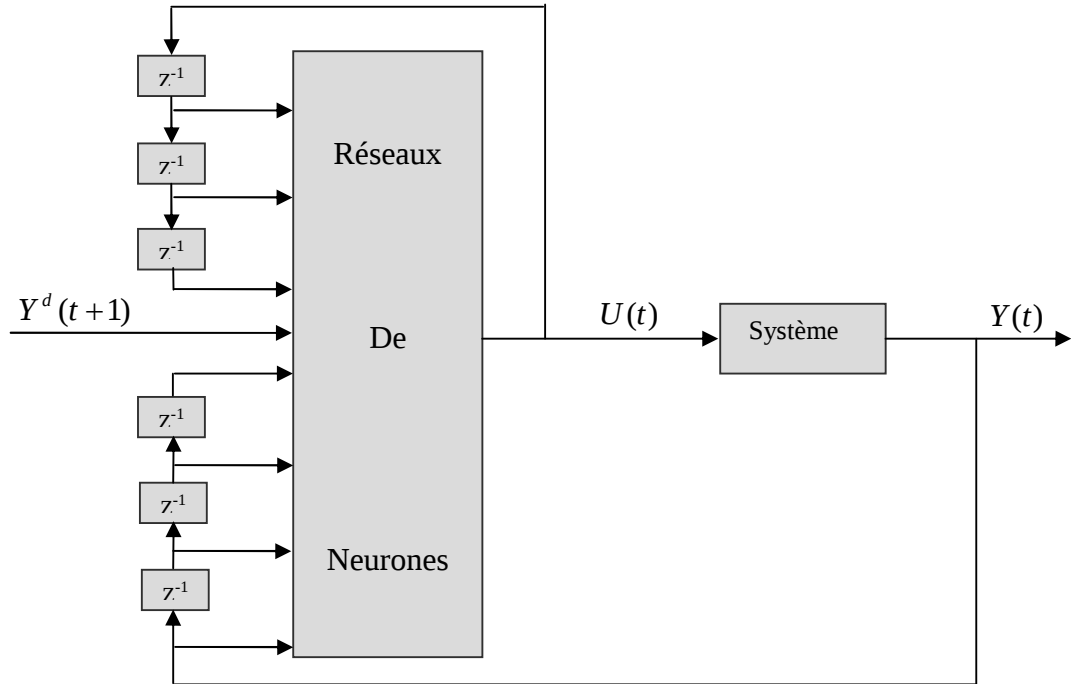


Figure (II.5) Structure de contrôle par modèle inverse (feed-forward)

Dans cette figure, on a adjoint au réseau de neurones qui constitue le modèle de processus un réseau de neurones qui calcule la loi de commande. Ce réseau est aussi un réseau non bouclé qui a pour entrées sortie du système $y(t)$ (état mesuré à l'instant t ou avec retard), la sortie désirée $y^d(t+1)$ (état au temps suivant) et la commande $u(t)$ (au temps précédent) dans le cas où l'on souhaite que $y^d(t+1)$ soit variable. La sortie du contrôleur neuronal est la

commande au temps t , qui lors de l'apprentissage, est appliquée à l'entrée de commande du modèle, et qui, lors de l'utilisation, est appliqué à l'entrée du système.

L'ensemble (contrôleur + modèle placés en série) constitue un réseau de neurones non bouclé qui admet pour sortie $y(t+1)$. L'apprentissage s'effectue en minimisant la différence entre l'état désiré $y^d(t+1)$ et la sortie du système $y(t+1)$. Seuls les paramètres du contrôleur (poids et biais) sont variables et modifiés par le processus d'apprentissage.

En effet, si le système est supposé décrit par l'équation différentielle non linéaire suivante :

$$y(t+1) = f[y(t), \dots, y(t-n_y+1); u(t), \dots, u(t-n_u+1)] \quad (\text{II.18})$$

Pour une sortie désirée $y_d(k+1)$, la commande $u(k)$ délivrée par la commande est la suivante :

$$y(t+1) = \hat{f}^{-1}[y(t), \dots, y(t-n_y+1); u(t-1), \dots, u(t-n_u+1); y_d(t+1)] \quad (\text{II.19})$$

Où $\hat{f}^{-1}$ est une approximation de la fonction inverse de f .

Cette méthode ne donne de bons résultats qu'aux problèmes simples où l'objectif peut s'exprimer instantanément en fonction de l'état $y(t)$, sinon cette méthode ne peut être mise en œuvre.

Il est évident que l'efficacité de cette approche dépend de la qualité, de fidélité et de la capacité de génération du modèle inverse. L'inconvénient de ce type de contrôle est qu'il souffre de robustesse [4][18].

II.2.4.1.2 Contrôleur Feed-back

Ce contrôleur est utilisé surtout comme contrôleur dynamique des systèmes linéaires qui est caractérisé par sa robustesse. Son utilisation est parallèle à celle des régulateurs connus dans la commande des systèmes linéaires. C'est à dire que son entrée est le signal d'erreur (différence entre la consigne et la mesure), ou la sortie d'un régulateur classique PI, alors que son signal de sortie et la commande à appliquer au système, comme la montre la figure (II.6)

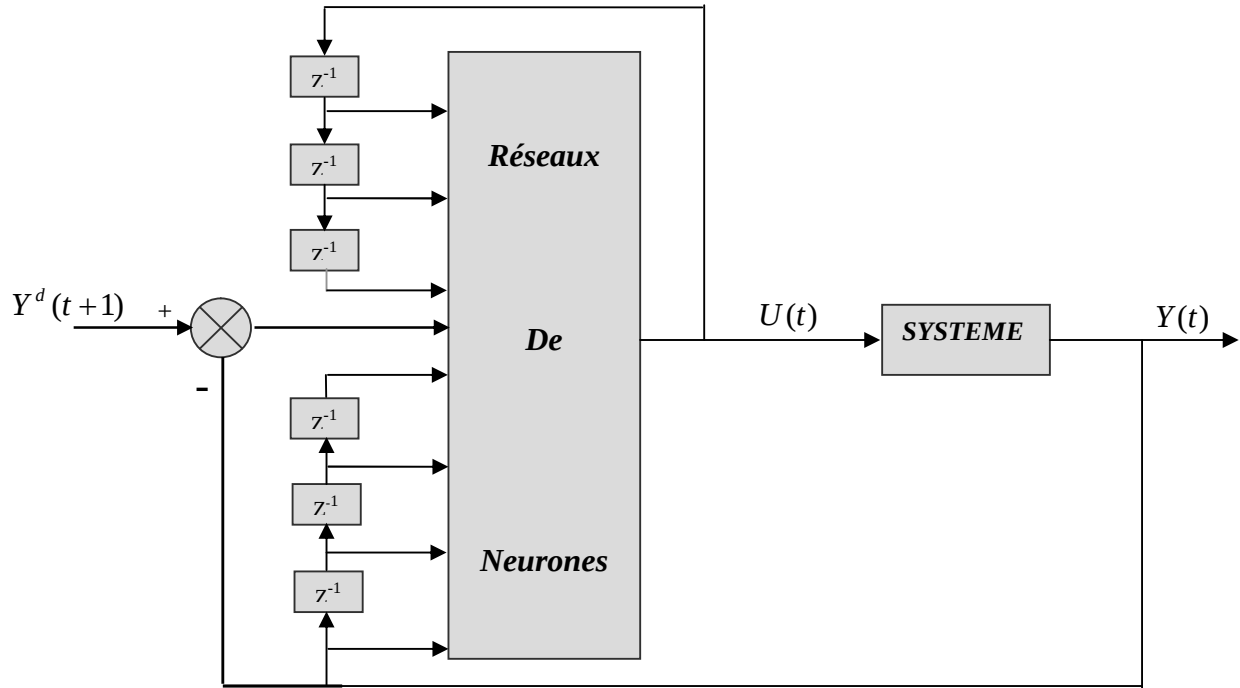


Figure (II.6) Structure de contrôle par modèle inverse (feed-back)

La différence entre les deux structures (Feed-back et Feed-Forward) réside dans l'entraînement des réseaux, le premier utilise l'apprentissage généralisé bien que le second utilise les deux types d'apprentissage (Généralisé et spécialisé) [4][18].

II.2.4.2 Utilisation d'un modèle de référence

L'utilisation d'un modèle de référence notamment mais non exclusivement en commande adaptative permet de bénéficier plus rationnellement de la connaissance à priori du système, pour synthétiser la commande. Dans ce type de contrôle, les performances désirées du système en boucle fermée sont spécifiées par un modèle de référence stable, défini par ses couples entrées/sorties $(r(k), y_r(k))$. Le principe général de l'utilisation d'un modèle de référence est donné par la Figure (II.7) [4][17].

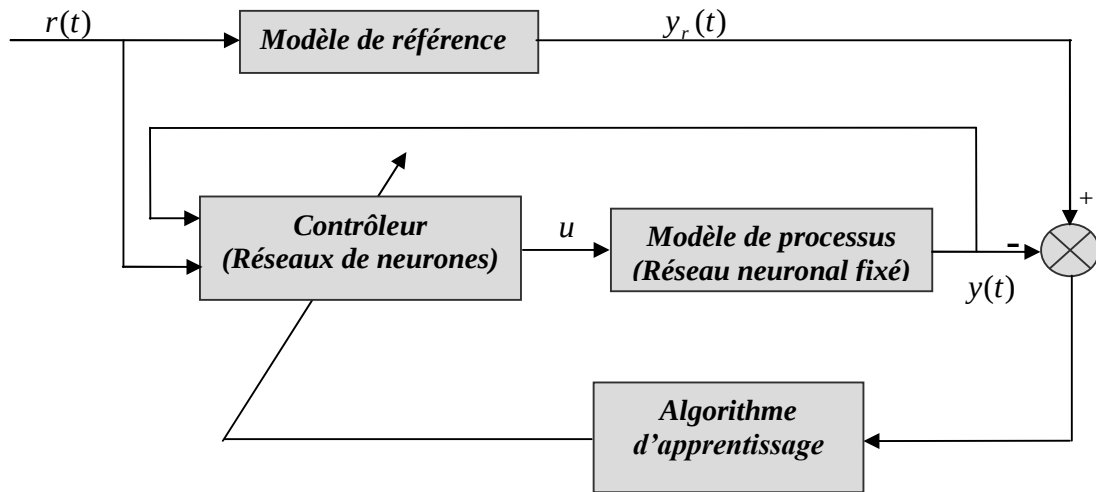


Figure (II.7) Principe de l'apprentissage d'une commande par modèle de référence

Le rôle du contrôleur est de commander la sortie du système de façon à minimiser l'écart entre la sortie du système et la sortie du modèle de référence.

L'erreur $y_r(k) - y(k)$ est utilisée pour ajuster les paramètres du contrôleur neuronal, la procédure d'entraînement force le contrôleur à se comporter comme un modèle inverse réglé dans le sens est défini par le modèle de référence (dans la commande par inversion du modèle, un modèle de référence est utilisé implicitement, il se réduit à un simple retard).

Deux structures de contrôle en utilisant un modèle de référence ont été proposées par Narendra et Parthasarathy ; la structure de contrôle direct et la structure de contrôle indirect.

II.2.4.2.1 Structure de contrôle direct

La structure de contrôle direct est illustrée dans la Figure (II.8). Dans ce cas la sortie du contrôleur est au même temps le signal de commande pour le système réel. Les entrées du contrôleur sont des sorties retardées du système réel et des sorties retardées du contrôleur lui-même [4][17].

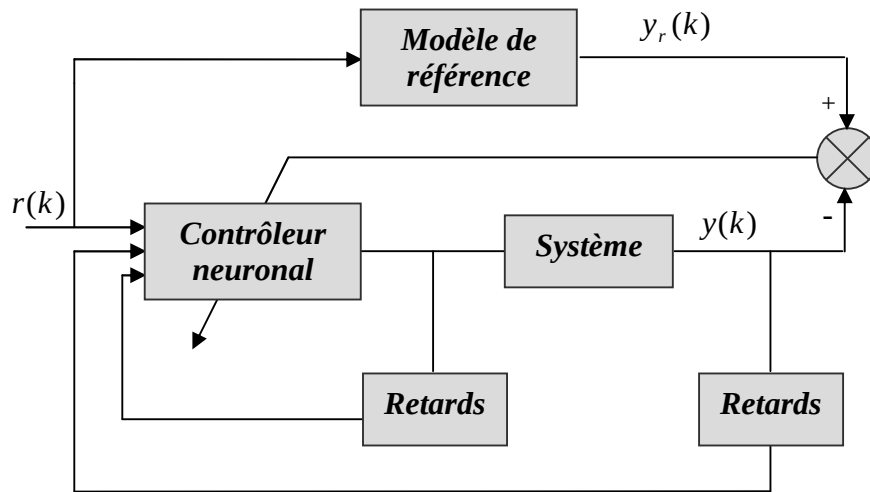


Figure (II.8) Structure de contrôle direct

II.2.4.2.2 Structure de contrôle indirect

Pour la structure de contrôle indirect, les paramètres du contrôleur sont ajustés de façon à minimiser l'écart entre la sortie du système et celle du modèle de référence. Le signal de commande généré par le contrôleur est utilisé comme une entrée pour le modèle neuronal et le système réel. L'erreur entre la sortie du système et celle du modèle sera utilisée pour ajuster les paramètres du modèle neuronal.

L'objectif du contrôleur est de générer les commandes nécessaires pour deux buts :

- Le premier est d'adapter les paramètres du modèle de façon à avoir une bonne identification, le second est que le système réel suit le comportement du modèle de référence.
- Notons enfin que les performances obtenues par le contrôle indirect sont plus intéressantes que celles obtenues par le contrôle direct. La Figure (II.9) représente la structure de contrôle indirect [4][21].

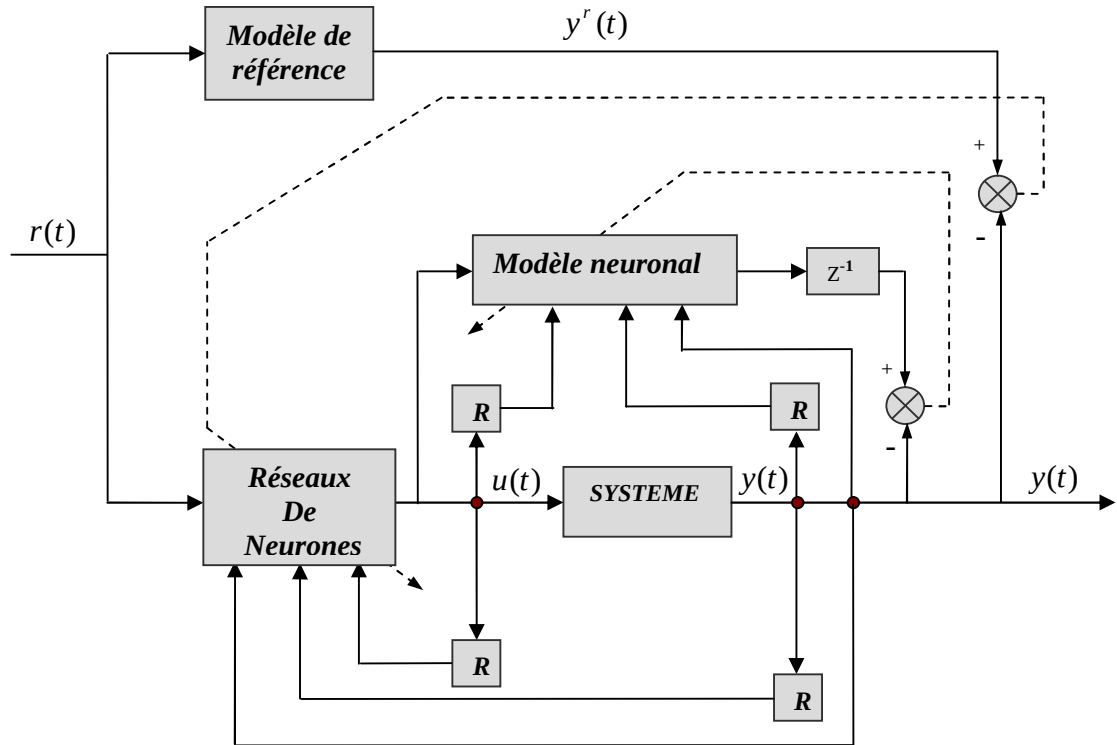


Figure (II.9) Structure de contrôle indirect

II.3 Conclusion

Dans ce chapitre on a étudié l'utilisation des réseaux de neurones pour la modélisation du comportement dynamique direct et inverse des systèmes, ainsi que les structures de contrôle neuronal lesquelles les modèles neuronaux directs et inverses jouent des rôles très importants.