

Chapitre 3

ESTIMATION DES PARAMÈTRES D'UN MODÈLE

INTRODUCTION.

Le chapitre précédent fournit le prédicteur théorique associé à un modèle-hypothèse donné ; ceci permet, pour chaque modèle envisagé, de définir un ensemble de réseaux de neurones candidats pour l'estimation du prédicteur théorique. En effet, celui-ci définit la structure bouclée ou non, et les arguments de la ou des fonctions que doit réaliser un prédicteur neuronal candidat. Or sa définition complète comporte celle de son architecture : nombre et type de neurones, connectivité, et valeurs des coefficients. Pour un type de réseau donné, chaque candidat est caractérisé par le nombre de ses neurones. L'estimation des coefficients de chacun, ou identification, est obtenue par apprentissage. Nous traitons dans ce chapitre du choix de séquences d'apprentissage appropriées et de l'algorithme d'apprentissage adéquat. Le meilleur candidat de l'hypothèse considérée est sélectionné à l'issue d'une procédure itérative. Cette procédure, appliquée à toutes les hypothèses, conduit à la sélection du meilleur réseau de neurones. Pour terminer, nous précisons les modalités d'utilisation du prédicteur obtenu pour l'application à laquelle il est destiné (prédiction, simulation, commande).

I. SÉQUENCES D'APPRENTISSAGE.

Ce paragraphe donne quelques indications sur le choix des séquences d'apprentissage.

I.1. SÉQUENCE DES ENTRÉES DE COMMANDE.

* *Contraintes sur les entrées de commande.*

Elles portent sur l'amplitude et le type de signaux de commande que le processus est susceptible de recevoir pendant son fonctionnement. Les amplitudes maximales sont en général faciles à déterminer, car leur ordre de grandeur correspond aux valeurs de saturation des actionneurs, qui peuvent être estimées physiquement (puissance maximale que peut délivrer un moteur, pression maximale d'un circuit de freinage ou d'un mécanisme hydraulique..., cf. chapitre 7).

En ce qui concerne le type de signaux à utiliser, un principe général est que les signaux utilisés pour l'identification doivent être de même nature que ceux qui seront calculés par l'organe de commande pendant l'utilisation du processus. L'idéal serait d'utiliser pour l'identification le correcteur même qui sera synthétisé à l'aide du prédicteur identifié ! *Si le bruit est négligeable*, une bonne démarche consiste à effectuer les expériences *en asservissement*, avec un correcteur simple (par

exemple linéaire). Cette démarche, ou identification *en boucle fermée*, permet d'explorer le domaine de fonctionnement désiré, en imposant une séquence de consigne correspondant au cahier des charges. C'est ainsi que nous avons procédé pour la modélisation du véhicule REMI (cf. chapitre 7). Une autre démarche, couramment utilisée, consiste à explorer "au mieux" le domaine de fonctionnement, par exemple avec des créneaux de commande (riches en fréquences), d'amplitudes et de durées diverses. Notons que cette dernière solution n'est pas praticable s'il existe des contraintes sur des sorties ou des variables internes du processus, et en particulier si le processus est instable.

** Fréquence d'échantillonnage.*

Les réseaux de neurones étant des modèles non linéaires, il n'est pas possible de passer simplement d'un modèle discret, valable à une fréquence d'échantillonnage donnée, à un autre : il est donc nécessaire d'effectuer l'identification à la fréquence qui sera utilisée pour la commande du processus. Si des contraintes diverses, par exemple le temps de calcul de la commande, nécessitent le choix d'une fréquence d'asservissement plus basse que la fréquence utilisée pour l'identification, il faut procéder à une nouvelle identification.

I.2. SÉQUENCES D'APPRENTISSAGE ET ESTIMATION DE LA PERFORMANCE.

Comme nous l'avons précisé au chapitre 1, l'apprentissage consiste à ajuster la (les) fonction(s) du prédicteur à un ensemble de points définis par des séquences d'apprentissage, ceci en minimisant une fonction de coût. Il y a surajustement (overfitting) lorsque l'apprentissage conduit à annuler quasiment la fonction de coût sans que pour autant la(les) fonction(s) réalisée(s) par la partie non bouclée du réseau prédicteur approche(nt) celle(s) du modèle-hypothèse auquel le prédicteur est associé. Pour des séquences d'apprentissage données, le surajustement se produit si le réseau prédicteur possède trop de coefficients, c'est-à-dire définit une famille de fonctions trop riche.

Pour sélectionner un réseau candidat pour une hypothèse donnée, il est donc nécessaire de répartir les données disponibles en une séquence d'apprentissage et une séquence d'estimation de la performance, dite séquence de test. Une séquence de test, de même type (issue de la même population) que la séquence d'apprentissage, conduit à une meilleure estimation de la variance de l'erreur de prédiction (erreur quadratique moyenne de test, notée EQMT) que celle obtenue avec la séquence d'apprentissage (notée EQMA). L'évolution de EQMA et de EQMT lorsque l'on augmente le nombre de neurones cachés, permet de détecter un surajustement, et de sélectionner le meilleur réseau de neurones parmi les candidats : dès qu'ajouter un neurone supplémentaire au réseau fait augmenter l'EQMT, même si l'EQMA continue à décroître, le nombre optimal de neurones est atteint. Cette méthode, à condition d'utiliser des méthodes d'optimisation performantes (par exemple les méthodes quasi-newtoniennes proposées en annexe I §II.3), permet de sélectionner le réseau le plus parcimonieux.

Remarque importante.

Dans la littérature ([SJÖ94] par exemple), la séquence de test est souvent utilisée pour arrêter l'apprentissage d'un réseau donné : dès qu'une itération d'apprentissage augmente l'EQMT, celui-ci est arrêté. Dans notre travail, *la séquence de test sert exclusivement à estimer la performance des réseaux après apprentissage, et n'est en aucun cas utilisée pour interrompre l'apprentissage.* L'apprentissage n'est arrêté que lorsqu'un minimum est atteint : l'arrêt est décidé en fonction de la valeur de la fonction de coût et de celle de la norme de son gradient. Si le nombre de neurones est augmenté de façon incrémentale comme nous l'indiquons ci-dessus, on obtient forcément le prédicteur le plus parcimonieux réalisant la meilleure performance sans surajustement.

II. ALGORITHMES D'APPRENTISSAGE.

Si le modèle-hypothèse est juste, si le prédicteur neuronal possède bien la structure du prédicteur associé, et si le choix des séquences d'apprentissage est conforme aux principes que nous venons d'énoncer, il est encore nécessaire d'utiliser l'algorithme d'apprentissage adéquat. Celui-ci est en fait fixé par les affectations des entrées externes I et des sorties Y du prédicteur, et, s'il est bouclé, des ses entrées d'état S^{in} et de ses sorties d'état S^{out} . Les calculs de la fonction de coût, de son gradient et des modifications des coefficients, s'effectuent selon la présentation de l'annexe I. Comme au chapitre 2, nous considérons les modèles-hypothèse mono-entrée/mono-sortie suivants :

- des modèles-hypothèse entrée-sortie :

$$y_p(k) = h(y_p(k-1), \dots, y_p(k-n), u(k-1), \dots, u(k-m))$$

où $y_p \in \mathbb{R}$, $u \in \mathbb{R}$, et h est une fonction non linéaire inconnue.

- des modèles-hypothèse d'état :

$$\begin{cases} x_p(k+1) = f(x_p(k), u(k)) \\ y_p(k) = g(x_p(k)) \end{cases}$$

où $y_p \in \mathbb{R}$, $u \in \mathbb{R}$, $x_p \in \mathbb{R}^n$, et f et g sont des fonctions non linéaires inconnues.

La fonction de coût s'écrit :

- dans le cas d'un prédicteur de la sortie du processus :

$$J(C) = \frac{1}{N} \sum_{k=1}^N (y_p(k) - y(k))^2 = \frac{1}{N} \sum_{k=1}^N (e(k))^2$$

où N est le nombre total d'instantanés de l'ensemble d'apprentissage, constitué de plusieurs séquences correspondant à des initialisations différentes de l'état du processus. Pour simplifier la présentation des algorithmes, et sans perte de généralité, nous supposons la séquence d'apprentissage obtenue à partir d'un seul état initial.

- dans le cas d'un prédicteur de l'état (mesuré) du processus :

$$J(C) = \frac{1}{N} \sum_{k=1}^N \sum_{i=1}^n w_{ii} (x_{ip}(k) - x_i(k))^2 = \frac{1}{N} \sum_{k=1}^N \sum_{i=1}^n w_{ii} (e_i(k))^2$$

Les coefficients w_{ij} sont les coefficients de pondération de la fonction de coût, choisie ici diagonale. Pour les modèles-hypothèse et les prédicteurs associés considérés dans ce travail, le choix de la pondération n'est pas crucial.

II.1. PRÉDICTEURS NON BOUCLÉS.

Nous distinguons ici le cas de l'identification de prédicteurs associés à des modèles-hypothèse entrée-sortie, qui constitue un problème d'identification mono-sortie, du cas de l'identification de prédicteurs associés à des modèles-hypothèse d'état lorsque l'on mesure l'état du processus, qui est un problème d'identification multi-sortie (puisque les composantes de l'état sont considérées comme des " sorties " pour l'apprentissage - on dispose d'une valeur désirée pour chacune d'elles).

II.1.1. Prédicteur associé à un modèle-hypothèse entrée-sortie.

Un modèle-hypothèse NARX impose un prédicteur non bouclé (chapitre 2 §II.2.1.1). Un tel prédicteur peut, par exemple, être réalisé à l'aide du réseau complètement connecté de la figure 1.

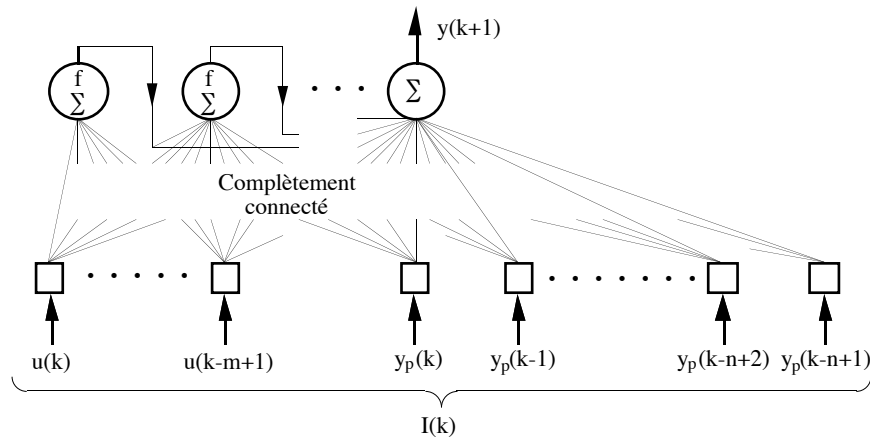


Figure 1.

Réalisation d'un prédicteur NARX à l'aide d'un réseau non bouclé complètement connecté.

Le réseau de la figure 1 définit la copie du réseau à l'instant k . Ce réseau, non bouclé, n'a pas d'état (N_S , dimension de S^{in} et S^{out} , est nul). Aux entrées externes sont affectées les composantes du vecteur $I(k)$, de dimension $m+n$, qui sont les suivantes :

$$I_i(k) = u(k-i+1) \quad i \in [1; m], k \in [1; N]$$

$$I_i(k) = y_p(k+m-i+1) \quad i \in [m+1; m+n], k \in [1; N]$$

$$Y(k) = y(k+1) \quad k \in [1; N]$$

où N est la taille de la séquence d'apprentissage. On suppose les séquences d'apprentissage numérotées de telle manière que u et y_p soient définies pour les indices ci-dessus.

Le système d'apprentissage utilisant ce réseau est représenté schématiquement sur la figure 2. Le prédicteur est dit " dirigé " par le processus, car l'état de celui-ci est imposé à ses entrées externes à chaque instant de la fenêtre d'apprentissage. Les entrées externes sont indépendantes des coefficients

du réseau. On est donc en présence de N copies indépendantes du réseau dont chacune fournit en sortie une erreur intervenant dans la fonction de coût. Chacune de ces erreurs est l'entrée d'un réseau de rétro-propagation, et le gradient est la somme des N gradients partiels calculés par rétro-propagation. Cet algorithme est encore appelé "teacher-forcing" dans la littérature [JOR85].

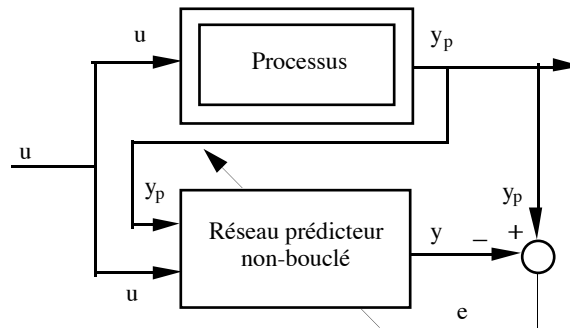


Figure 2.
Modèle-hypothèse entrée-sortie NARX :
le système d'apprentissage utilise un prédicteur non bouclé et un algorithme dirigé.

II.1.2. Prédicteur associé à un modèle-hypothèse d'état.

L'identification du prédicteur associé à un modèle-hypothèse d'état affecté d'un bruit d'état additif lorsque l'état du processus est mesuré impose aussi un prédicteur entrée-sortie non bouclé (chapitre 2 §II.2.2.1). Pour réaliser ce prédicteur, nous avons vu que l'on peut utiliser deux réseaux de neurones distincts, l'un réalisant la prédiction de l'état, et l'autre l'équation d'observation. Par exemple, le prédicteur associé au problème peut être réalisé à l'aide du réseau composé de n sous-réseaux complètement connectés, et du réseau complètement connecté de la figure 3.

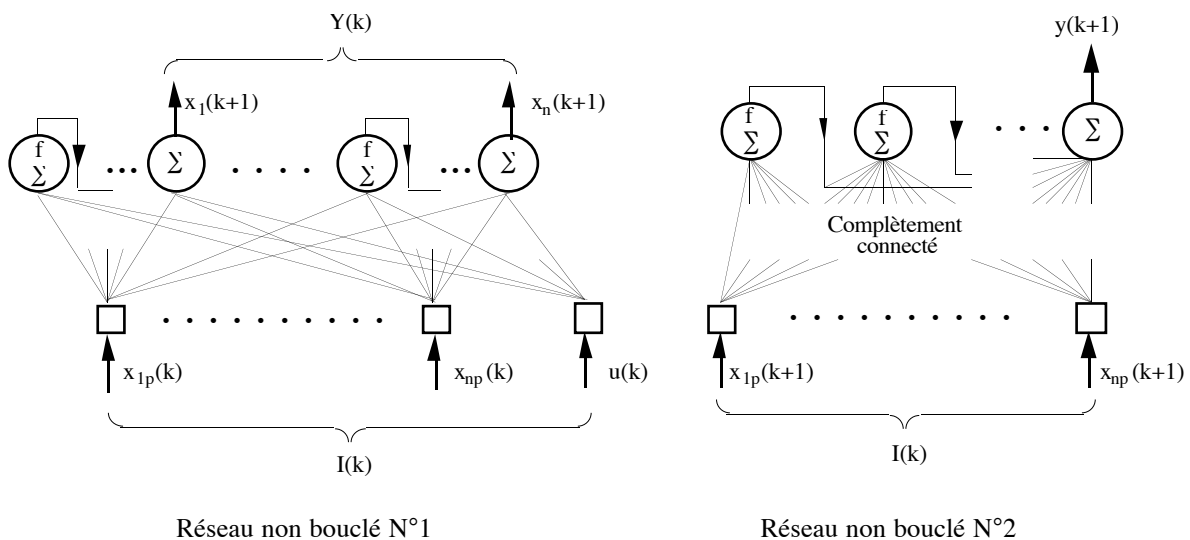


Figure 3.

Réalisation d'un prédicteur associé à un modèle-hypothèse d'état affecté d'un bruit d'état, à l'aide d'un réseau composé de n sous-réseaux complètement connectés (réseau N°1), et d'un réseau complètement connecté (réseau N°2).

Réseau N°1 :

$$I_i(k) = x_{i p}(k) \quad i \in [1;n], k \in [1;N]$$

$$I_{n+1}(k) = u(k) \quad k \in [1;N]$$

$$Y_i(k) = x_i(k+1) \quad k \in [1;N]$$

Ce réseau est un prédicteur non bouclé.

Le système d'apprentissage utilisant ces deux réseaux est représenté sur la figure 4.

Réseau N°2 :

$$I_i(k) = x_{i p}(k+1) \quad i \in [1;n], k \in [1;N]$$

$$Y(k) = y(k+1) \quad k \in [1;N]$$

Ce réseau non bouclé réalise une transformation algébrique de ses entrées.

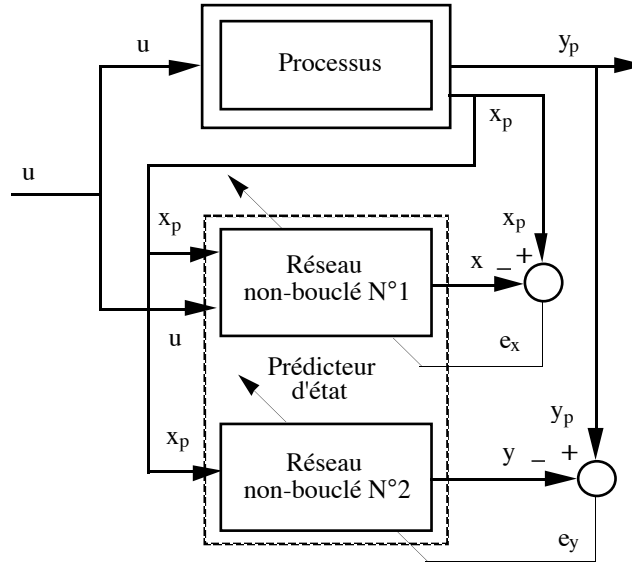


Figure 4.

Modèle-hypothèse d'état avec bruit d'état additif (ou sans perturbation) :
le système d'apprentissage utilise un prédicteur composé de deux réseaux non bouclés et un algorithme dirigé.

L'apprentissage des deux réseaux est dirigé par les variables d'état mesurées du processus. On est encore une fois en présence de N copies indépendantes de chacun des réseaux prédicteurs dont chacune fournit en sortie une erreur, e_x de dimension n et e_y de dimension 1, intervenant dans les 2 fonctions de coût. Comme précédemment, chacune de ces erreurs est l'entrée d'un réseau de rétro-propagation, et le gradient est la somme des N gradients partiels calculés par rétro-propagation. Pour éviter le problème de la pondération de la fonction de coût définie pour le réseau 1, on peut également utiliser n réseaux distincts.

II.2. PRÉDICTEURS BOUCLÉS.

Dans ce paragraphe, nous présentons les systèmes d'apprentissage pour l'identification de prédicteurs bouclés. Nous étudions séparément les prédicteurs associés à des modèles-hypothèse entrée-sortie (les composantes de l'état du réseau correspondent à des sorties successives du processus) et ceux qui sont associés à des modèles-hypothèse d'état (il n'y a pas de valeurs désirées pour leurs variables d'état).

II.2.1. Prédicteur associé à un modèle-hypothèse entrée-sortie.

a) Modèle-hypothèse bruit de sortie additif (NBSX).

L'identification d'un prédicteur NBSX impose l'utilisation d'un réseau de neurones bouclé (chapitre 2 §II.2.1.2). Un tel prédicteur peut être réalisé, par exemple, à l'aide du réseau complètement connecté de la figure 5.

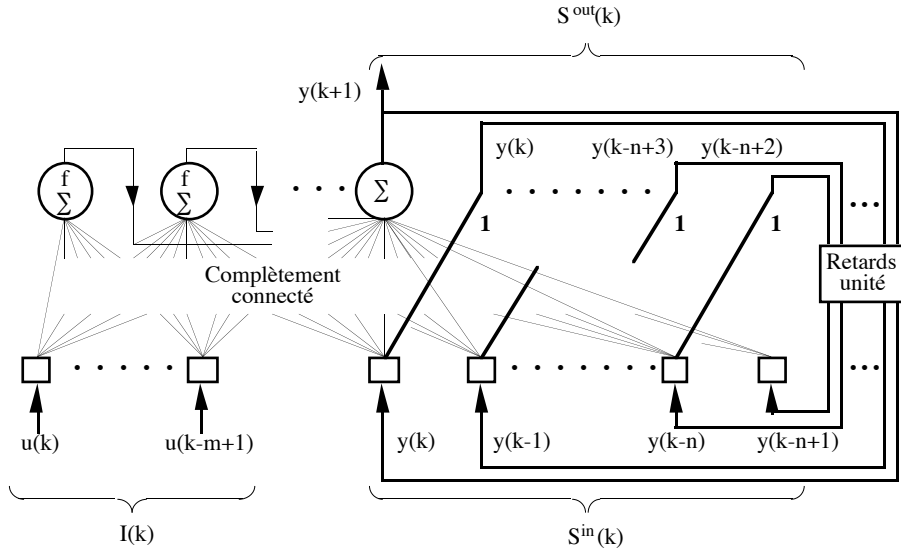


Figure 5.
Réalisation d'un prédicteur NBSX à l'aide d'un réseau bouclé complètement connecté.

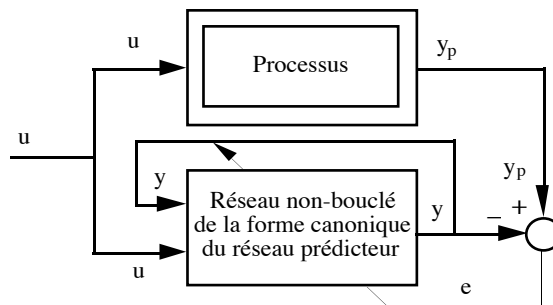


Figure 6.
Modèle-hypothèse entrée-sortie NBSX :
le système d'apprentissage utilise un prédicteur bouclé sur sa sortie et un algorithme semi-dirigé.

L'état du prédicteur correspond à celui du modèle-hypothèse et est donc constitué de n sorties successives du prédicteur. Les valeurs passées de la commande sont affectées aux entrées externes.

On a :

$$\begin{aligned}
 I_i(k) &= u(k-i+1) \quad i \in [1; m], k \in [1; N] \\
 S_i^{in}(1) &= y_p(-i+1) \quad i \in [1; n] \quad (\text{initialisation}) \\
 S_i^{in}(k) &= S_i^{out}(k-1) \quad i \in [1; n], k \in [2; N] \\
 S_i^{out}(k) &= S_{i-1}^{in}(k) \quad i \in [2; n], k \in [1; N] \\
 Y(k) &= S_1^{out}(k) = y(k+1) \quad k \in [1; N]
 \end{aligned}$$

$Y(k)$ est la seule valeur réellement calculée par le réseau : les autres variables d'état sont des sorties décalées dans le temps. Les entrées d'état de la première copie doivent être fixées par le concepteur. Le choix le plus raisonnable consiste à leur affecter les valeurs précédentes des sorties du processus.

Le système d'apprentissage utilisant ce prédicteur est représenté schématiquement sur la figure 6. Le prédicteur est dit “ *semi-dirigé* ” car les valeurs de ses entrées d'état ne sont imposées qu'au début de la fenêtre de la fonction de coût. Cette fois, les entrées d'état dépendent des coefficients du réseau, sauf pour la première copie. Pour le calcul du gradient, la rétro-propagation des N erreurs intervenant dans la fonction de coût doit donc être effectuée sur le réseau composé du dépliement spatial des N copies en cascade : l'algorithme de calcul de la fonction de coût et de son gradient est semi-dirigé.

b) Modèle-hypothèse NARMAX.

L'identification d'un prédicteur NARMAX impose l'utilisation d'un réseau de neurones bouclé (chapitre 2 §II.2.1.3). Un tel prédicteur peut par exemple être réalisé à l'aide du réseau complètement connecté de la figure 7.

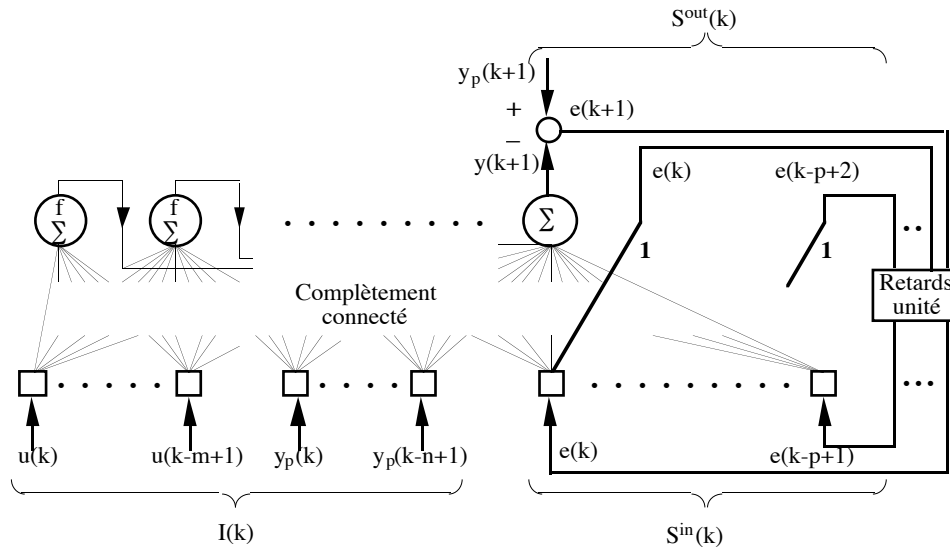


Figure 7.

Prédicteur associé à un modèle-hypothèse NARMAX réalisé à l'aide d'un réseau bouclé complètement connecté.

L'état du prédicteur n'est pas celui du modèle-hypothèse : il est constitué des p dernières erreurs du prédicteur. Les entrées externes I sont les commandes et les sorties mesurées du processus :

$$I_i(k) = u(k-i+1) \quad i \in [1; m], k \in [1; N]$$

$$I_i(k) = y_p(k-i-m+1) \quad i \in [m+1; m+n], k \in [1; N]$$

$$S_i^{in}(1) = 0 \quad i \in [1; p] \quad (\text{initialisation})$$

$$S_i^{in}(k) = S_i^{out}(k-1) \quad i \in [1; p], k \in [2; N]$$

$$S_i^{out}(k) = S_{i-1}^{in}(k) \quad i \in [2; p], k \in [1; N]$$

$$Y(k) = y(k+1) \quad k \in [1; N]$$

$$S_1^{out}(k) = y_p(k+1) - y(k+1) \quad k \in [1; N]$$

Les entrées d'état de la première copie doivent ici encore être fixées par le concepteur. Ces entrées d'état représentent cette fois l'erreur de prédiction. Un choix raisonnable consiste à leur affecter la valeur zéro. Le système d'apprentissage utilisant ce prédicteur est représenté sur la figure 8.

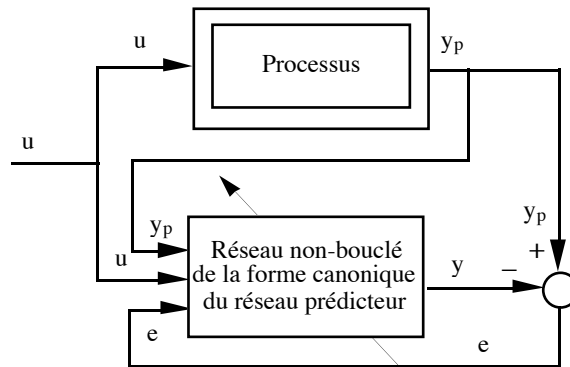


Figure 8.

Modèle-hypothèse entrée-sortie NARMAX :

le système d'apprentissage utilise un prédicteur bouclé sur l'erreur de prédiction et un algorithme semi-dirigé.

Les p entrées d'état dépendent des coefficients du réseau, sauf pour la première copie. Pour le calcul du gradient, la rétro-propagation des N erreurs intervenant dans la fonction de coût doit donc être effectuée sur le réseau composé du dépliement spatial des N copies en cascade : l'algorithme est semi-dirigé.

II.2.2. Prédicteur associé à un modèle-hypothèse d'état.

L'identification du prédicteur associé à un modèle-hypothèse d'état lorsque l'état du processus n'est pas mesuré impose l'utilisation d'un réseau de neurones bouclé (chapitre 2 §I.2.2.2 et §II.2.2.2). Ce prédicteur peut, par exemple, être réalisé à l'aide du réseau composé de $n+1$ sous-réseaux à une couche de neurones cachés de la figure 9.

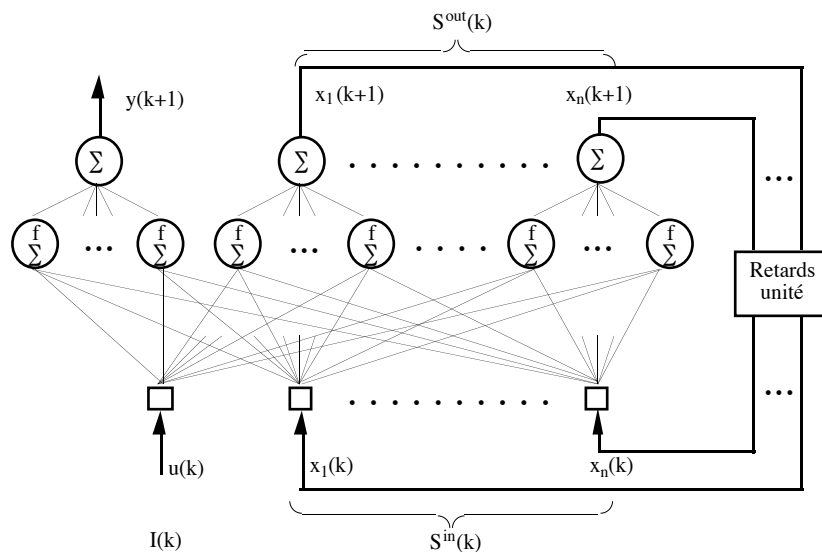


Figure 9.

Prédicteur associé à un modèle-hypothèse d'état, lorsqu'on ne mesure pas l'état du processus, réalisé à l'aide d'un réseau composé de $n+1$ sous-réseaux à une couche de neurones cachés.

Comme nous le verrons au chapitre 4 suivant, il n'est pas forcément nécessaire de fragmenter le réseau en sous-réseaux pour chaque variable d'état (surtout si les fonctions à réaliser par le réseau non bouclé de la forme canonique sont simples).

$$I_1(k) = u(k) \quad k \in [1; N]$$

$$S_i^{in}(1) = 0 \quad i \in [1; n] \quad (\text{initialisation})$$

$$S_i^{in}(k) = S_i^{out}(k-1) \quad i \in [1; n], k \in [2; N]$$

$$Y_1(k) = y(k+1) \quad k \in [1; N]$$

Pour la première copie, des valeurs arbitraires (0 par exemple, des valeurs particulières si l'on dispose d'informations) doivent être affectées à toutes les entrées d'état du prédicteur. Des erreurs d'initialisation sont inévitables, puisque l'on ne mesure pas l'état. Le système d'apprentissage utilisant le prédicteur est représenté schématiquement sur la figure 10.

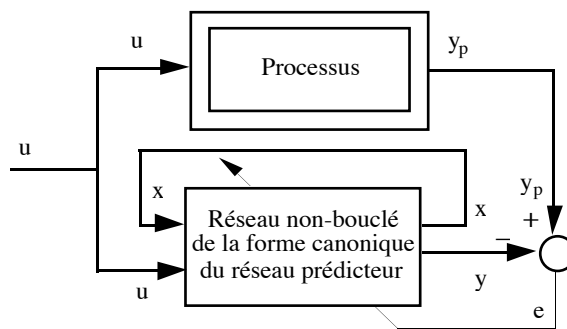


Figure 10.

Modèle-hypothèse d'état, l'état du processus n'est pas mesuré :
le système d'apprentissage utilise un prédicteur bouclé sur l'état (non imposé) et un algorithme semi-dirigé.

Le prédicteur est encore dit semi-dirigé car les valeurs de ses entrées d'état ne sont imposées qu'au début de la séquence d'apprentissage. Comme pour un prédicteur entrée-sortie bouclé, le calcul du gradient doit être effectué sur le réseau composé du dépliement spatial des N copies en cascade : l'algorithme de calcul de la fonction de coût et de son gradient est semi-dirigé.

III. SÉLECTION ET UTILISATIONS DU PRÉDICTEUR.

III.1. SÉLECTION DU PRÉDICTEUR.

Pour modéliser un processus, il est donc nécessaire de :

- formuler un ou plusieurs modèles-hypothèse compatibles avec les connaissances *a priori* que l'on a du processus (sur la nature de ses entrées, de ses sorties, éventuellement de son ordre, de son retard...); déterminer les prédicteurs théoriques associés à chacun de ces modèles-hypothèse ;
- pour chacune des hypothèses, mettre en œuvre le système d'apprentissage adéquat (séquences et algorithme d'apprentissage) pour les réseaux prédicteurs candidats. À partir de la performance de chaque candidat sur la séquence de test, déterminer le meilleur candidat associé à l'hypothèse.
- sélectionner la meilleure hypothèse, à partir des meilleurs candidats de chacune d'elles.

La validation finale du prédicteur est effectuée dans le cadre de son utilisation.

III.2. UTILISATIONS DU PRÉDICTEUR.

Le prédicteur obtenu, c'est-à-dire essentiellement la fonction ψ_{RN} réalisée par la partie non bouclée du réseau (ou φ_{RN} , ou les fonctions φ_{RN} et ψ_{RN} selon les cas) peut être utilisé comme modèle de simulation pour la commande, comme prédicteur, ou comme simulateur du processus.

Modèle de simulation pour la commande.

La mise au point d'un système de commande nécessite un modèle de simulation du processus (cf. chapitre 5). Dans ce travail, nous nous limitons à des modèles dont les entrées de bruit sont mises à zéro. De tels modèles sont en effet nécessaires :

- pour effectuer l'apprentissage d'un correcteur ;
- pour simuler le processus au sein d'un système de commande, en particulier dans un système de commande avec modèle interne.

Les modèles de simulation correspondant aux divers modèles-hypothèse étudiés sont obtenus de la manière suivante à partir des prédicteurs optimaux identifiés :

a) Prédicteur neuronal associé à un modèle-hypothèse NARX :

$$y(k+1) = \psi_{RN}(y_p(k), \dots, y_p(k-n+1), u(k), \dots, u(k-m+1); C)$$

Pour obtenir le modèle de simulation associé, il suffit de boucler le réseau :

$$y(k) = \varphi_{RN}(y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-m); C)$$

b) Prédicteur neuronal associé à un modèle-hypothèse NARMAX :

$$y(k+1) = \psi_{RN}(y_p(k), \dots, y_p(k-n+1), u(k), \dots, u(k-m+1), e(k), \dots, e(k-p); C)$$

Le modèle de simulation associé est le suivant :

$$y(k) = \varphi_{RN}(y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-m), 0, \dots, 0; C)$$

c) Prédicteur neuronal associé à un modèle-hypothèse NBSX :

$$y(k+1) = \varphi_{RN}(y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1); C)$$

Il est utilisé tel quel comme modèle de simulation du processus :

$$y(k) = \varphi_{RN}(y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-m); C)$$

d) De même, un prédicteur neuronal bouclé associé à un modèle d'état est utilisé tel quel comme modèle de simulation. Un prédicteur neuronal associé à un modèle d'état composé de deux réseaux non bouclés est utilisé à cette fin en bouclant le réseau des équations d'état.

Remarque importante.

Dans la pratique, il est fréquent que certaines entrées ne soit pas mesurées (température d'un moteur thermique, paramètres de la combustion), ou que des perturbations non mesurées agissent sur le processus. Ce cas ne correspond à aucune des hypothèses du chapitre 2, et corollairement, *aucun prédicteur associé* ne fournit une erreur de prédiction qui est un bruit blanc. La sélection du prédicteur

ne doit alors pas s'appuyer sur cette erreur, car elle favorise le prédicteur NARX, *non bouclé*, qui donne presque toujours une erreur de prédiction plus faible et moins corrélée que celle d'un prédicteur *bouclé* NARMAX ou NBSX (voir l'explication du chapitre 4 §I.3.1 et §I.3.2). *Il est donc fortement recommandé, dans une telle situation, de sélectionner le prédicteur en fonction de la performance du modèle de simulation (bouclé) qui lui est associé.*

Prédicteur.

Des systèmes de commande, tels que les systèmes à variance minimale, nécessitent l'utilisation, au sein même du système de commande, du prédicteur optimal [GOO84]. C'est ici une utilisation immédiate du prédicteur obtenu. Nous donnons un exemple d'une telle utilisation au chapitre 6 §II.

Simulateur.

Nous avons montré au chapitre 2 §II.2 que la fonction intervenant dans le prédicteur optimal (le réseau non-bouclé de sa forme canonique) permet d'estimer le modèle-hypothèse, et donc de réaliser un simulateur du processus. Celui-ci peut être utilisé notamment pour simuler le processus en fonctionnement "normal" et détecter les anomalies en comparant l'état du simulateur à celui du processus (voir [PLO94] pour une telle utilisation du simulateur neuronal d'une colonne à distiller).

CONCLUSION.

La procédure de modélisation qui vient d'être proposée conduit à des réseaux conciliant universalité et parcimonie. En ce qui concerne l'universalité, des non linéarités très simples telles que des saturations peuvent ainsi être très difficiles à modéliser par d'autres moyens [NOR88]. En revanche, nous verrons que deux saturations en cascade peuvent être parfaitement identifiées à l'aide d'un réseau de neurones à deux neurones cachés sigmoïdaux (chapitre 7). Du point de vue de la parcimonie, contrairement à d'autres modèles universels tels que les modèles polynômiaux, le nombre de coefficients d'un réseau de neurones à sigmoïdes n'augmente pas nécessairement de façon importante avec la complexité de la fonction et la dimension de l'espace d'entrée.

En contrepartie, la sensibilité de la sortie d'un réseau de neurones par rapport à un coefficient particulier ne s'exprime pas de manière élémentaire (par rapport à celle d'un modèle linéaire, polynômial, ou encore bilinéaire), et rend complexe l'analyse du comportement dynamique d'un réseau de neurones bouclé.

Insistons enfin sur le caractère générique des algorithmes d'apprentissage que nous avons présentés : ils ne dépendent ni de la complexité, ni du caractère bouclé ou non des prédicteurs utilisés. Alors que dans la "culture linéaire", on cherche souvent par des techniques *ad-hoc* à se ramener à une méthode de type moindres carrés, justifiée dans sa version ordinaire pour le prédicteur ARX seulement, l'identification par réseaux de neurones non récursive (et même récursive [NER92]) telle qu'elle a été exposée aux chapitres 2 et 3 présente une grande homogénéité.