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 newtonniennes 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, de 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 :

$$y_p(k) = h\left(y_p(k\text{-}1),\ \ldots,\ y_p(k\text{-}n),\ u(k\text{-}1),\ \ldots,\ u(k\text{-}m)\right)$$

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

- des modèles-hypothèse d'état :

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

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 :

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

où N est le nombre total d'instants 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 :

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

Les coefficients w_{ii} 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.

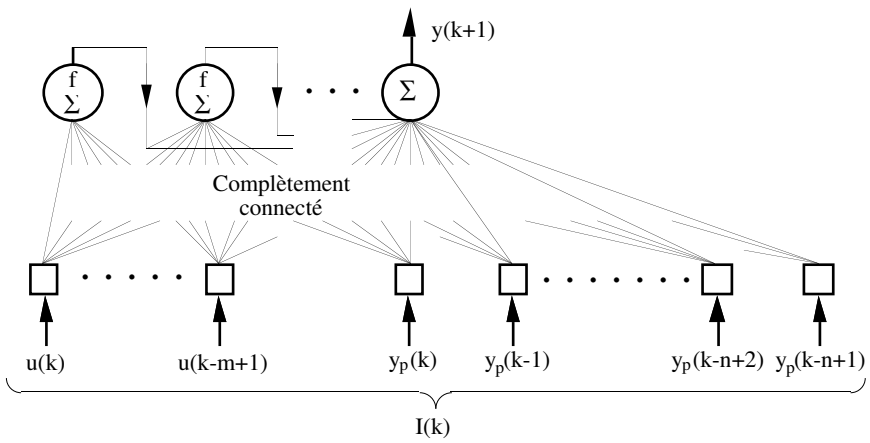


Figure 1.
Réalisation d'un prédicteur NARX à l'aide d'un réseau non bouclé complètement connecté.

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].

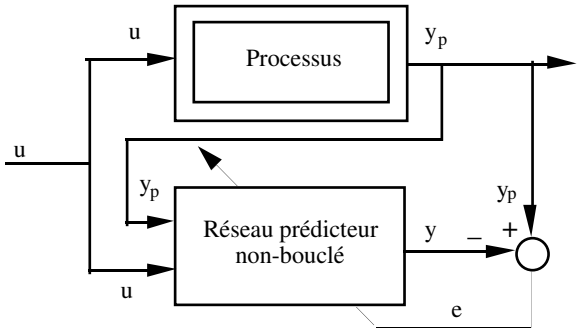
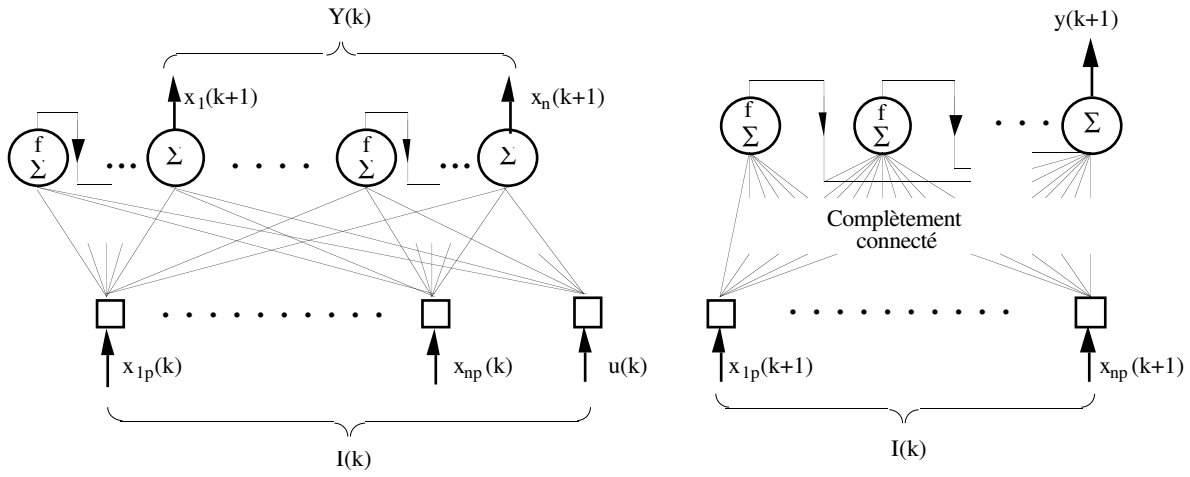


Figure 2.
Modèle-hypothèse entrée-sortie NARX :
le système d'apprentissage utilise un prédicteur non bouclé et un algorithme dirigé.

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.



Réseau non bouclé N°1 Réseau non bouclé N°2

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.

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.

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

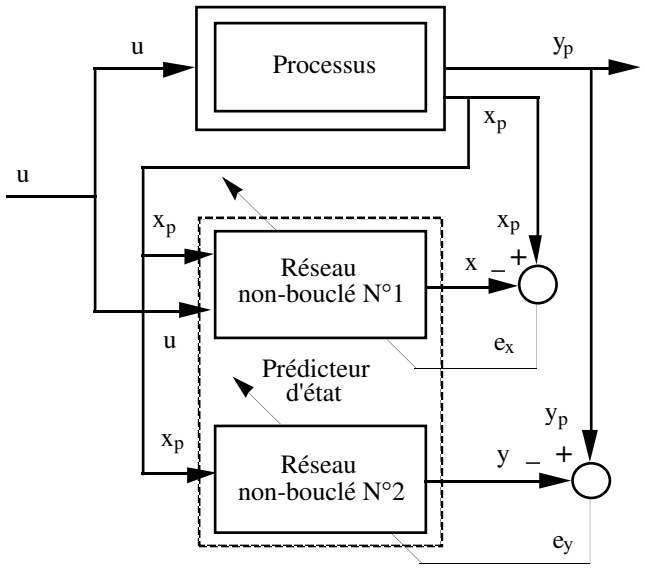


Figure 4.
Modèle-hypothèse d'état avec bruit d'état additif (ou sans perturbation) :
le système d'apprentissage utilise un prédicteur composé de deux réseaux non bouclés et un algorithme dirigé.

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.

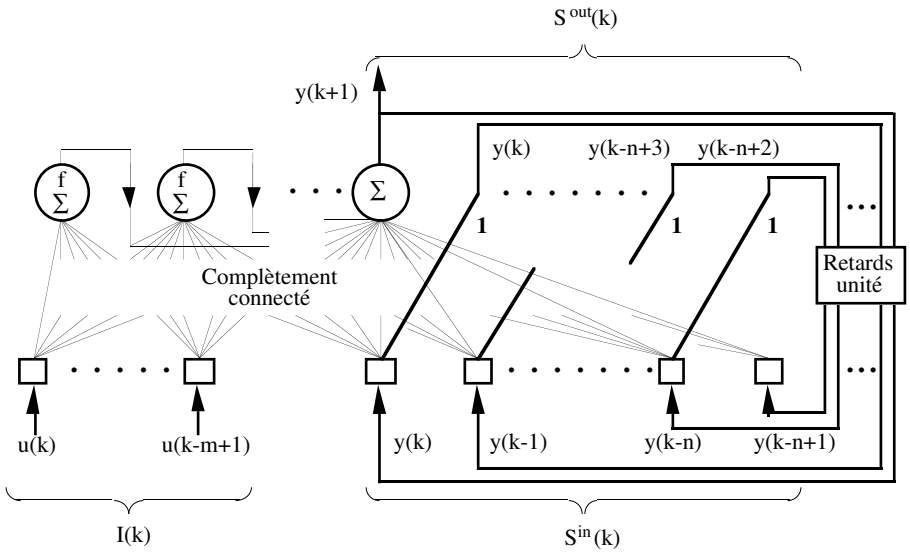


Figure 5.
Réalisation d'un prédicteur NBSX à l'aide d'un réseau bouclé complètement connecté.

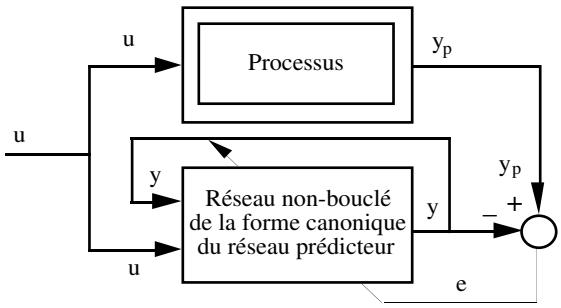


Figure 6.
Modèle-hypothèse entrée-sortie NBSX :
le système d'apprentissage utilise un prédicteur bouclé sur sa sortie et un algorithme semi-dirigé.

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 :

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

$S_i^{in}(1) = y_p(\text{-}i\text{+}1) \quad i \in [1;n] \quad (\textit{initialisation})$

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

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

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

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.

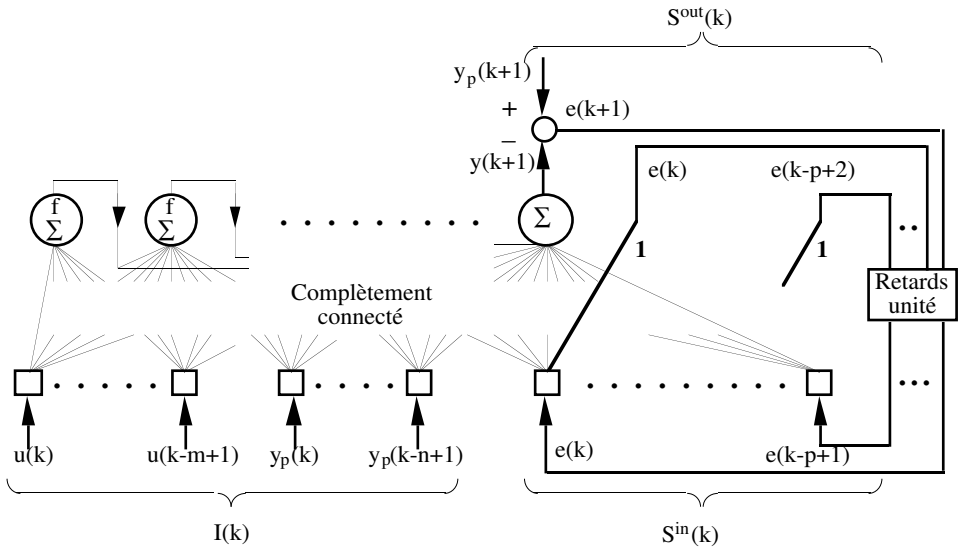


Figure 7.
Prédicteur associé à un modèle-hypothèse NARMAX réalisé à l'aide d'un réseau bouclé complètement connecté.

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 :

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

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

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

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

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

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

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

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.

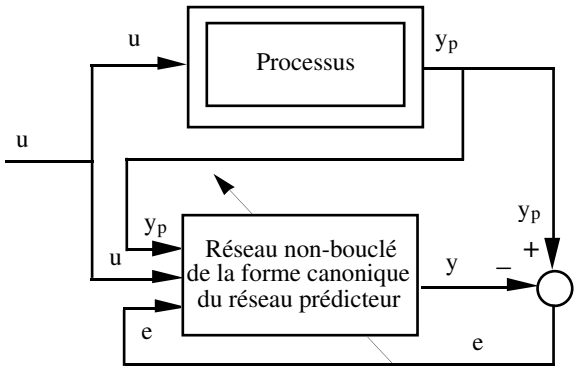


Figure 8.
Modèle-hypothèse entrée-sortie NARMAX :
le système d'apprentissage utilise un prédicteur bouclé sur l'erreur de prédiction et un algorithme semi-dirigé.

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.

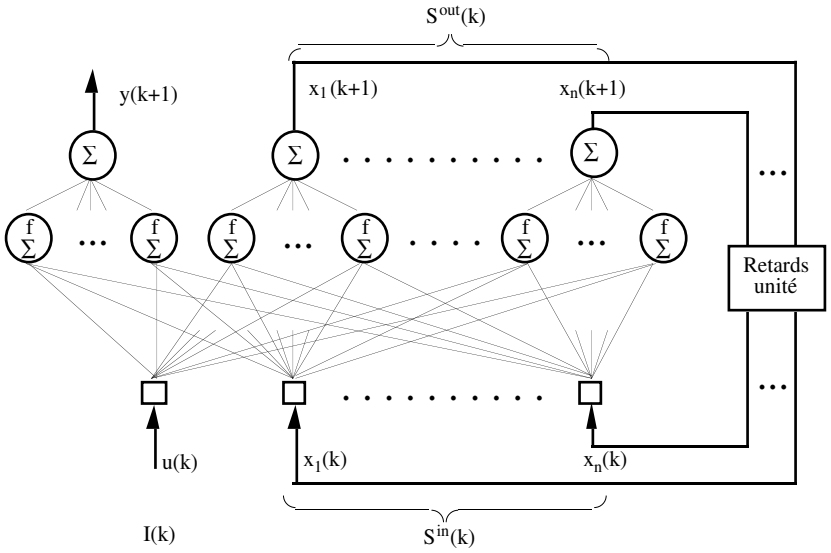


Figure 9.
Prédicteur associé à à un modèle-hypothèse d'état, lorsqu'on ne mesure pas l'état du processus,
réalisé à l'aide d'un réseau composé de n+1 sous-réseaux à une couche de neurones cachés.

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).

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

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

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

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

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.

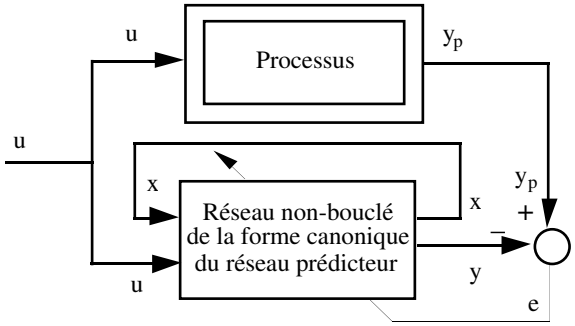


Figure 10.
Modèle-hypothèse d'état, l'état du processus n'est pas mesuré :
le système d'apprentissage utilise un prédicteur bouclé sur l'état (non imposé) et un algorithme semi-dirigé.

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 :

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

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

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

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

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

Le modèle de simulation associé est le suivant :

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

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

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

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

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

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.

Chapitre 4

EXEMPLES DE MODÉLISATION DE PROCESSUS

INTRODUCTION.

Dans les chapitres 2 et 3, nous avons décrit les principes de la modélisation de processus par réseaux de neurones, principes qui ont été appliqués à un problème industriel : la modélisation du véhicule REMI, décrite au chapitre 7 (confidentiel) du mémoire. Afin d'illustrer aussi complètement que possible le cadre général proposé, nous avons choisi d'étudier également deux processus simulés, et un processus réel, l'actionneur hydraulique d'un bras de robot :

- Le premier processus, de type entrée-sortie, est simulé par une équation aux différences non linéaire du second ordre, et illustre la modélisation en présence de perturbations aléatoires. Ce processus est utilisé au chapitre 6 au sujet de la commande à erreur de prédiction minimale.
- Le second processus est simulé avec une représentation d'état du second ordre. Nous réalisons tout d'abord l'apprentissage de prédicteurs d'état, afin de démontrer la faisabilité de cet apprentissage, et les intérêts de ces prédicteurs. Ensuite, nous utilisons des prédicteurs entrée-sortie. L'un d'eux est mis en œuvre au chapitre 6 comme modèle de simulation pour l'apprentissage de correcteurs, et comme modèle interne.
- Enfin, nous réalisons la modélisation d'un actionneur hydraulique. La modélisation de cet actionneur ayant donné lieu à plusieurs travaux récents [SJÖ93] [SJÖ94] [BEN94], nous comparons nos méthodes et les performances obtenues à celles qui sont décrites dans ces publications.

I. MODÉLISATION D'UN PROCESSUS SIMULÉ PAR UN MODÈLE ENTRÉE-SORTIE.

I.1. PRÉSENTATION.

L'objectif est ici de montrer l'influence des hypothèses faites quant à la nature des perturbations aléatoires affectant le processus à identifier. En particulier, nous montrons que, si l'hypothèse est vraie, on obtient un prédicteur très précis. Nous montrons aussi que le modèle de simulation obtenu à partir de ce prédicteur, comme nous l'avons indiqué au chapitre 3 §III.2, est une excellente approximation de la partie déterministe du processus, et peut donc servir à l'élaboration hors-ligne d'un organe de commande pour le processus.

Processus sans bruit.

Le processus est simulé par l'équation aux différences non linéaire du second ordre suivante :

$$y_p(k) = h\left(y_p(k\text{-}1),\, y_p(k\text{-}2),\, u(k\text{-}1)\right) = 50\ tanh\left[2\ 10^{-3}\left(\frac{\left(24+y_p(k\text{-}1)\right)}{3}\, y_p(k\text{-}1) - 8\,\frac{u(k\text{-}1)^2}{1+u(k\text{-}1)^2}\, y_p(k\text{-}2)\right)\right] + 0{,}5\ u(k\text{-}1)$$

La dynamique de ce processus est intéressante, car son comportement est celui (i) d'un filtre passe-bas linéaire du premier ordre pour des amplitudes de 0,1 environ, et (ii) celui d'un système du second ordre oscillatoire, linéaire pour de faibles amplitudes de la commande ($0{,}1 < |u| < 0{,}5$) , et non linéaire pour les grandes amplitudes ($0{,}5 < |u| < 5$). De plus, il n'est pas symétrique par rapport à l'origine. Nous supposons connus les arguments $y_p(k\text{-}1)$, $y_p(k\text{-}2)$, $u(k\text{-}1)$ du processus sans bruit (déterminés par exemple par la méthode de sélection de modèles présentée dans [URB94] pour le cas NARX, dont l'auteur propose le processus ci-dessus).

Séquences utilisées pour l'identification.

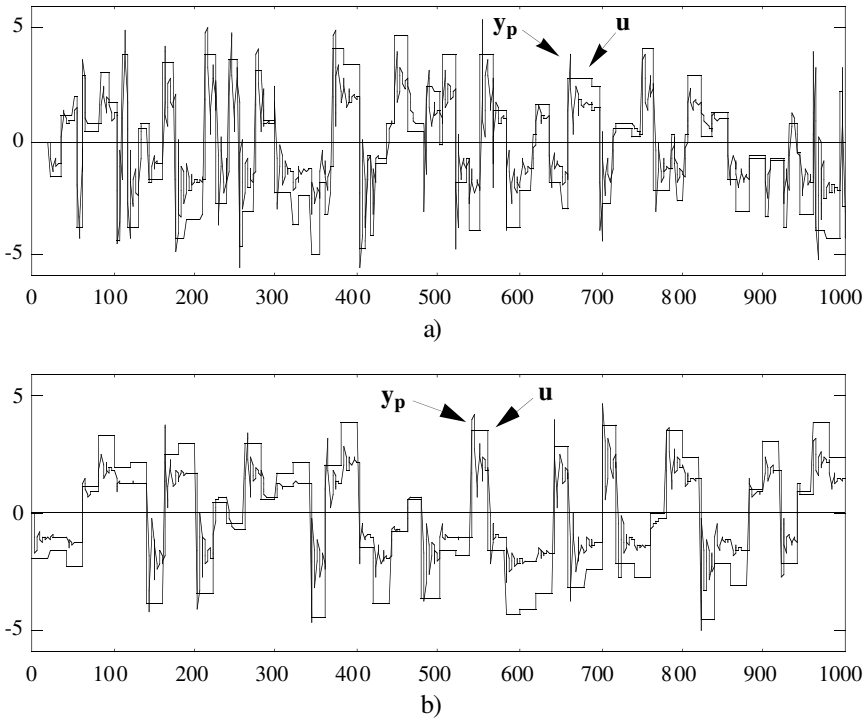


Figure 1.
a) Séquences d'apprentissage ; b) Séquences de test.

a) Séquences d'apprentissage :

La séquence de commande utilisée pour l'apprentissage des réseaux prédicteurs est constituée de créneaux d'amplitudes aléatoires entre ±5, de durées aléatoires entre 1 et 20 pas d'échantillonnage. La séquence totale comporte 1000 pas d'échantillonnage ; elle est représentée sur la figure 1a. Nous notons EQMA l'erreur quadratique moyenne sur la séquence d'apprentissage.

b) Séquences d'estimation de la performance (ou séquences de test) :

La séquence de commande pour l'estimation de la performance des prédicteurs est constituée de créneaux d'amplitudes aléatoires entre ±5, de durée constante 20 pas d'échantillonnage. La séquence

totale comporte également 1000 pas d'échantillonnage ; elle est représentée sur la figure 1b. Nous notons EQMT l'erreur quadratique moyenne sur la séquence de test.

Prédicteurs.

Les prédicteurs utilisés sont des réseaux complètement connectés, à neurones cachés dont la fonction d'activation est la tangente hyperbolique, et à neurone de sortie linéaire.

Processus avec bruit.

Soit w un bruit blanc à valeur moyenne nulle et de distribution uniforme[1] d'amplitude 0,2 (donc de variance $3{,}33\ 10^{-3}$). Nous considérons les processus avec bruit suivants :

* Processus avec perturbation additive d'état (NARX) :

$$y_p(k) = h\ \left(y_p(k\text{-}1),\ y_p(k\text{-}2),\ u(k\text{-}1)\right) + w(k)$$

* Processus avec perturbation additive de sortie (NBSX) :

$$\begin{cases} x_p(k) = h\left(x_p(k\text{-}1),\ x_p(k\text{-}2),\ u(k\text{-}1)\right) \\ y_p(k) = x_p(k) + w(k) \end{cases}$$

* Processus NARMAX :

$$y_p(k) = h\ \left(y_p(k\text{-}1),\ y_p(k\text{-}2),\ u(k\text{-}1)\right) + 0{,}5\ y_p(k\text{-}1)\ w(k\text{-}1) + w(k)$$

I.2. MODÉLISATION DU PROCESSUS SANS BRUIT.

Afin d'évaluer la difficulté du problème, nous avons préalablement procédé à l'identification de prédicteurs du processus sans bruit. Nous avons vu au chapitre 2 §I.2.1, que le prédicteur associé à un tel modèle-hypothèse peut être indifféremment bouclé ou non. *La performance qui nous intéresse étant celle d'un modèle bouclé (pour l'apprentissage d'un correcteur avec ce modèle), l'apprentissage a été effectué avec des modèles bouclés, donc en semi-dirigé*. Le tableau 1 donne les performances obtenues.

Nombre de neurones cachés	EQMA	EQMT
3	$9{,}6\ 10^{-2}$	$2{,}9\ 10^{-2}$
4	$4{,}0\ 10^{-4}$	$2{,}1\ 10^{-4}$
5	$6{,}1\ 10^{-6}$	$3{,}7\ 10^{-6}$
6	$4{,}2\ 10^{-6}$	$3{,}2\ 10^{-6}$

Tableau 1.
Processus sans bruit : résultats de l'identification de réseaux prédicteurs bouclés.

[1] Une distribution du bruit suivant une loi gaussienne eût-elle été plus réaliste ? " Tout le monde y croit fermement parce que les mathématiciens s'imaginent que c'est un fait d'observation, et les observateurs que c'est un théorème de mathématiques. " [POI02].

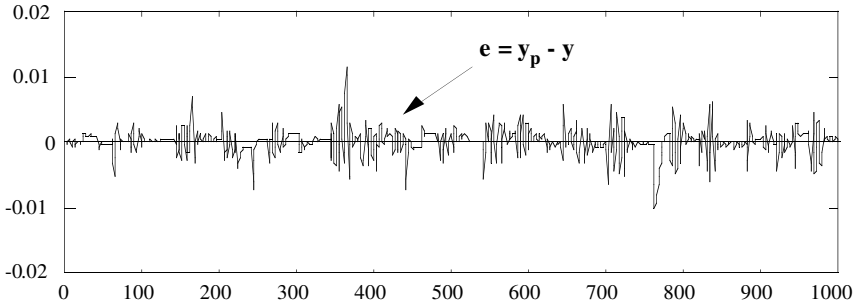


Figure 2.
Processus sans bruit : erreur de prédiction obtenue sur la séquence de test
avec le réseau de neurones bouclé à 5 neurones cachés.

L'ajout de neurones supplémentaires n'améliore pas sensiblement la précision. Ceci permet de conclure que 5 neurones cachés sont nécessaires et suffisent pour obtenir un prédicteur d'une précision satisfaisante. La figure 2 montre l'erreur du prédicteur bouclé à 5 neurones cachés, sur la séquence de test.

I.3. MODÉLISATION DU PROCESSUS AVEC BRUIT.

En présence de perturbations aléatoires, le prédicteur optimal est le prédicteur donnant une erreur de prédiction de variance minimale. Comme il s'agit d'un prédicteur à 1 pas (chapitre 2 §II.2), la variance minimale de l'erreur de prédiction obtenue avec ce prédicteur optimal est la variance de la perturbation aléatoire, qui vaut ici $3{,}33\ 10^{-3}$. Les erreurs quadratiques moyennes EQMA et EQMT doivent donc avoir une valeur voisine de celle-ci. Si l'EQMA est plus petite, il y a sur-apprentissage. Si l'EQMT est plus grande que la variance alors que la valeur de l'EQMA est correcte, c'est que l'apprentissage n'a pas été effectué dans un domaine représentatif du domaine de fonctionnement souhaité. Les précautions prises dans le choix de la séquence d'apprentissage nous ont permis d'éviter cette situation, comme le montrent les résultats du tableau 1. Les prédicteurs utilisés dans la suite possèdent 5 neurones cachés.

Afin de tester la qualité des prédicteurs obtenus en tant que modèles de simulation (cf. chapitre 3 §III.2), nous avons également évalué leur performance par rapport au processus *sans bruit*, sur la séquence de test (bien entendu, ceci n'est possible que pour un processus simulé). L'erreur quadratique moyenne ainsi obtenue est appelée EQMD (*D* pour *déterministe*). Si l'identification est réalisée dans de bonnes conditions, l'EQMD doit se rapprocher de l'EQMT obtenue par apprentissage avec le processus sans bruit (cf. tableau 1) ; en pratique, sa valeur est supérieure, ce d'autant plus que le bruit est important (10 à 100 fois pour nos exemples) et perturbe l'estimation.

Pour les processus NARX et NBSX, les conséquences du choix d'une hypothèse, vraie ou fausse, sur la qualité du prédicteur a été montrée dans [NER92a] ; notre apport sera d'analyser les résultats dans le cas d'une hypothèse fausse, en nous appuyant sur la fonction d'autocorrélation de l'erreur. Nous modélisons ensuite le processus NARMAX, modélisation dont on ne trouve guère

d'exemples à l'aide de réseaux de neurones (sauf dans [BIL92], pour le cas particulier d'un processus très simple affecté d'un bruit MA, voir chapitre 2 §II.2.1).

Enfin, nous ne simulons pas une démarche " réaliste " dans les choix successifs des modèles-hypothèse : nous formulons en général d'abord l'hypothèse vraie, puis les hypothèses fausses.

I.3.1. Processus avec perturbation additive d'état (NARX).

Le processus obéit à l'équation :

$$y_p(k) = h\left(y_p(k\text{-}1),\ y_p(k\text{-}2),\ u(k\text{-}1)\right) + w(k)$$

I.3.1.1. Hypothèse vraie NARX.

On fait ici l'hypothèse vraie NARX. Le prédicteur associé à l'hypothèse n'est pas bouclé (chapitre 2 §II.2.1.1). Pour réaliser ce prédicteur, on utilise un réseau de neurones non bouclé de la forme :

$$y(k+1) = \psi_{RN}\left(y_p(k),\ y_p(k\text{-}1),\ u(k);\ C\right)$$

L'algorithme d'apprentissage est dirigé. Les résultats obtenus en fin d'apprentissage sont donnés par le tableau 2.

EQMA	EQMT	*EQMD*
$3{,}2\ 10^{-3}$	$3{,}4\ 10^{-3}$	*$3{,}4\ 10^{-4}$*

Tableau 2.
Processus NARX : résultats de l'identification avec un réseau prédicteur NARX.

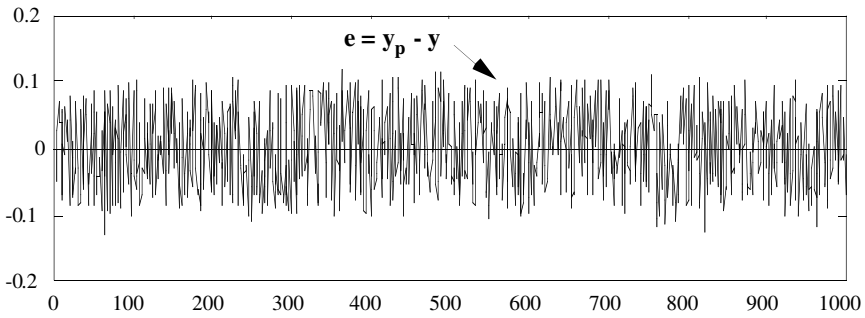


Figure 3.
Processus NARX : erreur de prédiction obtenue avec un réseau prédicteur NARX.

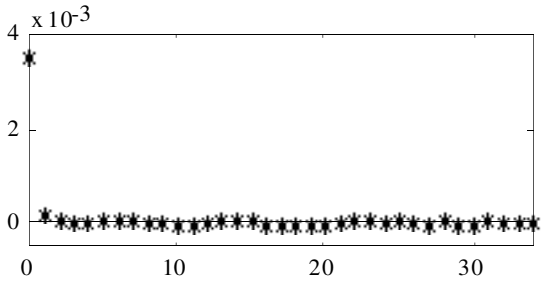


Figure 4.
Processus NARX : fonction d'autocorrélation de l'erreur de prédiction obtenue avec le réseau prédicteur NARX.

La figure 3 représente l'erreur de prédiction obtenue sur l'ensemble de test. Cette erreur présente bien les mêmes caractéristiques que la perturbation aléatoire (distribution, corrélation). En particulier, la fonction d'autocorrélation de l'erreur est bien celle d'un bruit blanc, c'est-à-dire nulle sauf en zéro, où elle prend la valeur de la variance du signal (cf. figure 4).

I.3.1.2. Hypothèse fausse NBSX.

Faisons à présent l'hypothèse fausse NBSX. Le prédicteur associé à cette hypothèse est bouclé d'ordre 2 sur la sortie du prédicteur (chapitre 2 §II.2.1.2). Pour identifier ce prédicteur, on utilise donc un réseau de neurones bouclé de la forme :

$$y(k+1) = \varphi_{RN}\left(y(k),\, y(k\text{-}1),\, u(k);\, C\right)$$

L'algorithme d'apprentissage est semi-dirigé. Les résultats obtenus en fin d'apprentissage sont donnés par le tableau 3 suivant.

EQMA	EQMT	*EQMD*
$7{,}4\ 10^{-3}$	$9{,}2\ 10^{-2}$	*$9{,}8\ 10^{-4}$*

Tableau 3.
Processus NARX : résultats de l'identification d'un réseau prédicteur NBSX

La variance de l'erreur de prédiction n'est que légèrement plus importante que lorsque l'hypothèse est vraie. L'EQMD montre d'ailleurs que le comportement déterministe est assez bien approché.

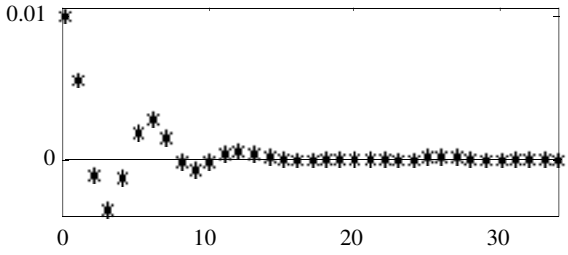


Figure 5.
Processus NARX : fonction d'autocorrélation de l'erreur de prédiction obtenue avec le réseau prédicteur NBSX

Cependant, la fonction d'autocorrélation de l'erreur est donnée sur la figure 5. Elle permet de conclure que l'erreur de prédiction n'est pas un bruit blanc et de rejeter l'hypothèse, par comparaison avec la fonction d'autocorrélation du prédicteur NARX précédent.

Interprétation de l'allure de la fonction de corrélation.

Interprétons l'allure de la fonction d'autocorrélation à l'aide d'une étude en linéaire. Soit le processus ARX suivant :

$$A(q)\ y_p(k) = B(q)\ u(k) + w(k)$$

Le prédicteur non bouclé associé au processus ARX est :

$$y(k) = \left(1 - A(q)\right) y_p(k) + B(q)\ u(k)$$

Supposons que l'on utilise le prédicteur linéaire suivant :

$$A(q)\ y(k) = B(q)\ u(k)$$

Les coefficients de ce prédicteur sont bien ceux du processus, mais il est bouclé, alors que le prédicteur associé au processus ne l'est pas. Exprimons l'erreur de prédiction en soustrayant ces deux expressions :

$$e(k) = y_p(k) - y(k) = \frac{1}{A(q)}\ w(k)$$

La fonction d'autocorrélation de l'erreur est celle d'un filtre AR dont l'ordre est celui du processus.

De même, dans le cas non linéaire, si l'identification du processus NARX est menée avec un prédicteur NBSX, mais que la fonction réalisée par le prédicteur est malgré tout proche de celle du modèle-hypothèse, alors l'erreur de prédiction sera essentiellement due au fait que le prédicteur est bouclé, et elle est analogue à celle que l'on obtient en linéaire.

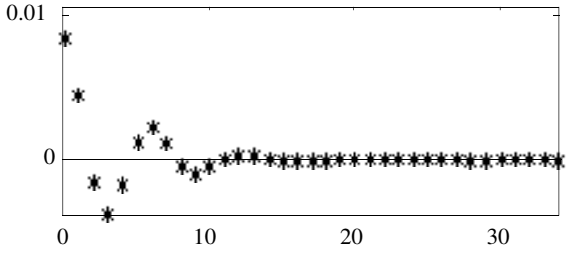


Figure 6.
Processus NARX : fonction d'autocorrélation de l'erreur de prédiction obtenue avec le réseau prédicteur NARX, mais utilisé bouclé.

La figure 6 montre justement l'autocorrélation de l'erreur obtenue avec le prédicteur NARX appris au paragraphe précédent (la partie non bouclée de ce prédicteur réalise donc la bonne fonction), mais utilisé bouclé. Cette autocorrélation ressemble beaucoup à celle d'une séquence AR du second ordre (le processus non linéaire est d'ordre 2). Elle est presque identique à celle de la figure 5.

I.3.2. Processus avec perturbation additive de sortie (NBSX).

Le processus obéit à l'équation :

$$\begin{cases} x_p(k) = h\left(x_p(k\text{-}1),\ x_p(k\text{-}2),\ u(k\text{-}1)\right) \\ y_p(k) = x_p(k) + w(k) \end{cases}$$

I.3.2.1. Hypothèse vraie NBSX.

On fait tout d'abord l'hypothèse vraie. Le prédicteur associé à cette hypothèse est bouclé d'ordre 2 sur la sortie du prédicteur (chapitre 2 §II.2.1.2). On utilise donc un réseau bouclé :

$$y(k+1) = \varphi_{RN}\left(y(k),\ y(k\text{-}1),\ u(k);\ C\right)$$

L'algorithme d'apprentissage est semi-dirigé. Les résultats obtenus sont donnés par le tableau 4.

EQMA	EQMT	*EQMD*
$3{,}3\ 10^{-3}$	$3{,}3\ 10^{-3}$	*$6{,}3\ 10^{-5}$*

Tableau 4.
Processus NBSX : résultats de l'identification d'un réseau prédicteur NBSX.

Le modèle de simulation associé au prédicteur identifié sera utilisé au chapitre 6 pour la commande. Nous notons ce modèle de simulation, dont les coefficients sont fixés, de la façon suivante :

$$y(k) = \varphi_{RN}^{Sim1}\left(y(k\text{-}1),\, y(k\text{-}2),\, u(k\text{-}1)\right)$$

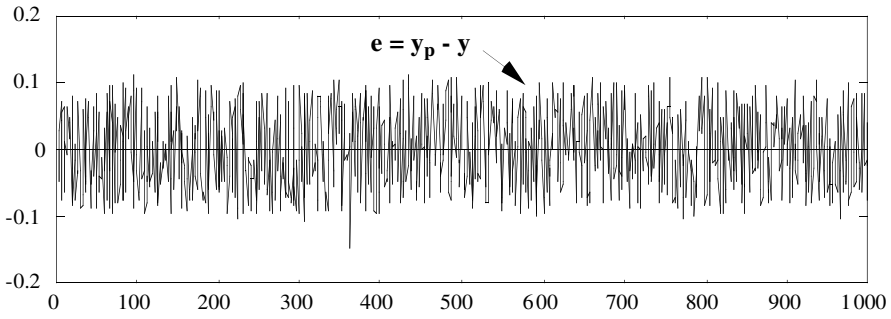


Figure 7.
Processus NBSX : erreur de prédiction obtenue avec le réseau prédicteur NBSX.

La figure 7 représente l'erreur de prédiction obtenue en fin d'apprentissage sur l'ensemble de test. Cette erreur présente bien les mêmes caractéristiques que la perturbation aléatoire de sortie (bruit blanc de valeur moyenne nulle et de distribution uniforme d'amplitude 0,2).

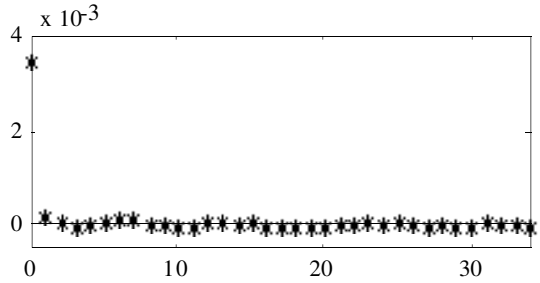


Figure 8.
Processus NBSX : fonction d'autocorrélation de l'erreur de prédiction obtenue avec le réseau prédicteur NBSX.

La fonction d'autocorrélation de l'erreur, donnée en figure 8, caractérise bien un bruit blanc.

I.3.2.2. Hypothèse fausse NARX.

Faisons maintenant l'hypothèse fausse NARX. Le prédicteur associé à l'hypothèse n'est pas bouclé (chapitre 2 §II.2.1.1). Pour identifier ce prédicteur, nous utilisons donc un réseau de neurones non bouclé de la forme :

$$y(k+1) = \psi_{RN}\left(y_p(k),\, y_p(k\text{-}1),\, u(k);\, C\right)$$

L'algorithme utilisé est donc dirigé. Les résultats obtenus sont donnés par le tableau 5.

EQMA	EQMT	*EQMD*
$1{,}0\ 10^{-2}$	$2,\ 0\ 10^{-2}$	$1{,}2\ 10^{-2}$

Tableau 5.
Processus NBSX : résultats de l'identification d'un réseau prédicteur NARX.

La variance de l'erreur de prédiction est très importante, ce qui trahit une très mauvaise modélisation. Ceci est confirmé par une valeur importante de l'EQMD, beaucoup plus importante que celle obtenue lors de l'identification d'un processus NARX avec un prédicteur NBSX.

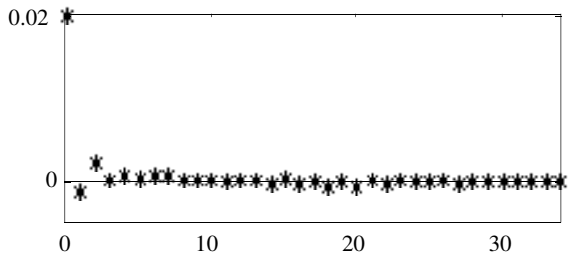


Figure 9.
Processus NBSX : fonction d'autocorrélation de l'erreur de prédiction obtenue avec le réseau prédicteur NARX.

La fonction d'autocorrélation, donnée sur la figure 9, a presque les caractéristiques de celle d'un bruit blanc. Il faut donc se fonder aussi bien sur la valeur de la variance de l'erreur de prédiction que sur son autocorrélation pour comparer les hypothèses. On observera souvent ce type de résultats lors de l'identification d'un processus NBSX avec un prédicteur NARX. Expliquons pourquoi, ici encore à l'aide des résultats que l'on peut établir en linéaire.

Interprétation de l'allure de la fonction d'autocorrélation.

Soit le processus linéaire NBSX suivant :

$$A(q)\ y_p(k) = B(q)\ u(k) + A(q)\ w(k)$$

Le prédicteur associé à ce processus est bouclé :

$$A(q)\ y(k) = B(q)\ u(k)$$

Supposons cependant que l'on utilise le prédicteur non bouclé suivant, ayant les mêmes coefficients que le processus :

$$y(k) = \big(1 - A(q)\big)\ y_p(k) + B(q)\ u(k)$$

L'erreur de prédiction est obtenue en soustrayant ces deux expressions :

$$e(k) = A(q)\ w(k)$$

La fonction d'autocorrélation est donc celle d'un filtre MA. Si le processus est d'ordre n, elle possède donc 2n+1 valeurs non nulles (donc n+1 si l'on ne considère la fonction, paire, que sur $\mathbb{N}$).

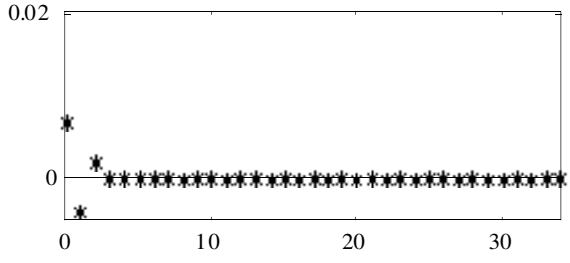


Figure 10.
Processus NBSX : fonction d'autocorrélation de l'erreur de prédiction obtenue avec le réseau prédicteur NBSX, mais utilisé non bouclé.

De même, en non linéaire, on obtient des résultats analogues. La figure 10 représente la fonction d'autocorrélation de l'erreur obtenue en utilisant le prédicteur associé NBSX obtenu au paragraphe précédent (la partie non bouclée du prédicteur réalise la bonne fonction), mais sans le boucler. Cette fonction d'autocorrélation a les mêmes caractéristiques que celles d'une séquence AR. En raison des erreurs de modélisations dues à l'utilisation d'un mauvais prédicteur, les fonctions d'autocorrélation des figures 9 et 10 sont toutefois différentes.

Ceci montre que l'identification d'un processus à l'aide d'un prédicteur non bouclé (NARX) d'un processus qui exigerait un prédicteur bouclé (NBSX) conduit souvent à une erreur très peu corrélée. Il faut donc impérativement effectuer d'autres hypothèses, et comparer les prédicteurs obtenus pour chacune d'elles, non seulement sur la base de la corrélation de l'erreur, mais aussi sur la variance de celle-ci. *Dans la réalité, il se peut aussi qu'aucune hypothèse ne soit vraie, et il est alors recommandé, comme nous y avons insisté au chapitre 3 §III.2, de comparer les prédicteurs selon les performances de leurs modèles de simulation associés, afin de ne pas favoriser le prédicteur NARX.*

I.3.3. Processus NARMAX.

Le processus obéit à l'équation :

$$y_p(k) = h\left(y_p(k\text{-}1),\ y_p(k\text{-}2),\ u(k\text{-}1)\right) + 0{,}5\ y_p(k\text{-}1)\ w(k\text{-}1) + w(k)$$

I.3.3.1. Hypothèse vraie NARMAX.

Faisons tout d'abord l'hypothèse vraie. Le prédicteur associé à l'hypothèse NARMAX est bouclé sur l'erreur, d'ordre 1 (chapitre 2 §III.1.3). Pour l'identifier, il faut donc utiliser un réseau bouclé de la forme :

$$y(k\text{+}1) = \varphi_{RN}\left(y_p(k),\ y_p(k\text{-}1),\ u(k),\ e(k);\ C\right)$$

Le tableau 9 donne les résultats obtenus en fin d'apprentissage :

EQMA	EQMT	*EQMD*
$3{,}9\ 10^{-3}$	$3{,}5\ 10^{-3}$	*$4{,}1\ 10^{-4}$*

Tableau 9.
Processus NARMAX : résultats de l'identification d'un réseau prédicteur NARMAX.

En toute rigueur, la fonction à approcher étant plus complexe, il aurait fallu prendre un réseau comportant plus de neurones cachés. Les résultats ci-dessus étant quasiment optimaux (EQMT de $3{,}54\ 10^{-3}$ alors que la valeur théorique est de $3{,}33\ 10^{-3}$), nous n'avons pas jugé nécessaire d'agrandir le réseau, pour cette présentation. De plus, le prédicteur utilisé comme modèle de simulation a d'excellentes performances.

La figure 12 représente l'erreur de prédiction obtenue en fin d'apprentissage sur l'ensemble de test. Cette erreur présente bien encore les mêmes caractéristiques que la perturbation aléatoire. Il est absolument nécessaire, pour obtenir un tel résultat, de disposer d'algorithmes d'optimisation efficaces, tels que ceux présentés en annexe I.

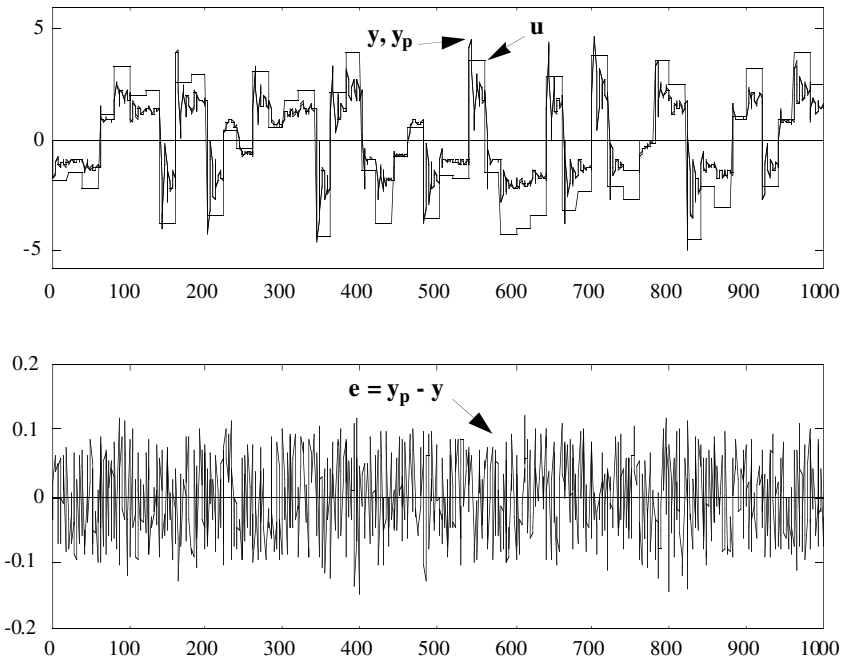


Figure 12.
Processus NARMAX : commande, sorties du processus et du réseau prédicteur, erreur de prédiction, obtenues avec le réseau prédicteur NARMAX.

I.3.3.2. Hypothèse fausse NBSX.

Si l'on fait l'hypothèse fausse qu'il s'agit d'un processus NBSX, et que l'on identifie le prédicteur associé à l'aide d'un réseau de même structure, on obtient les résultats du tableau 10.

EQMA	EQMT	*EQMD*
$1,5\ 10^{-2}$	$1,5\ 10^{-2}$	*$1,3\ 10^{-3}$*

Tableau 10.
Processus NARMAX : résultats de l'identification d'un réseau prédicteur NBSX.

Le prédicteur ainsi obtenu a encore une performance très mauvaise pour la prédiction (valeur de l'EQMT très supérieure à la valeur théorique $3,33\ 10^{-3}$). Comme le montre la figure 13a, l'erreur de prédiction n'a plus les caractéristiques de la perturbation aléatoire. Sa distribution n'est pas uniforme, son amplitude est beaucoup plus importante, et la corrélation est apparente (elle serait révélée par des tests statistiques). Utilisé comme modèle de simulation, ce prédicteur est également peu satisfaisant.

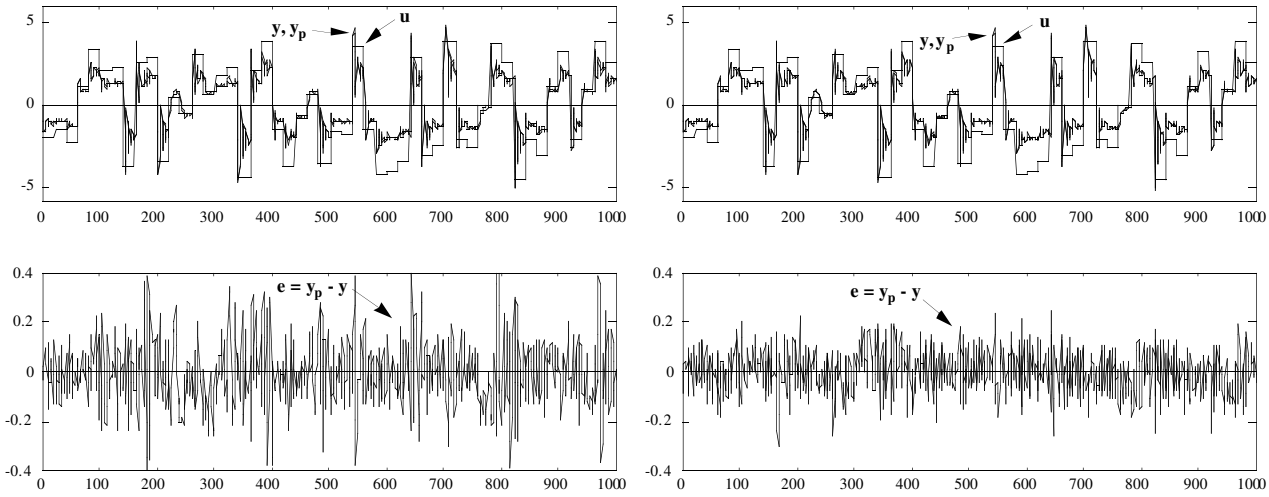
I.3.3.3. Hypothèse fausse NARX.

Si l'on fait l'hypothèse fausse qu'il s'agit d'un processus NARX, et que l'on identifie le prédicteur associé à l'aide d'un réseau de même structure, on obtient les résultats du tableau 11.

EQMA	EQMT	*EQMD*
$6,7\ 10^{-3}$	$7,0\ 10^{-2}$	*$4,2\ 10^{-3}$*

Tableau 11.
Processus NARMAX : résultats de l'identification d'un réseau prédicteur NARX.

Le réseau ainsi obtenu n'est toujours pas optimal comme prédicteur, quoique meilleur que le précédent. Ceci est normal, puisque le bruit est un bruit d'état (bien que coloré). Ici encore, comme le montre la figure 13b, l'erreur de prédiction n'a plus la distribution de la perturbation aléatoire et trahit l'erreur faite sur l'hypothèse. Utilisé comme modèle de simulation, il est particulièrement imprécis.



a) Réseau prédicteur NBSX. b) Réseau prédicteur NARX.

Figure 13.
Processus NARMAX : commande, sorties du processus et du réseau prédicteur, erreur de prédiction.

II. MODÉLISATION D'UN PROCESSUS SIMULÉ PAR UN MODÈLE D'ÉTAT.

II.1. PRÉSENTATION.

Le choix du modèle d'état pour cette présentation a été guidé par plusieurs préoccupations :
a) Tout d'abord, celle de montrer l'utilité de modèles d'état pour réaliser des prédicteurs, souvent très économes en nombre de neurones (parcimonieux).
b) Ensuite, comme pour l'exemple précédent, nous avons le souci de montrer l'influence des hypothèses concernant les perturbations aléatoires affectant le processus, que le prédicteur choisi soit de type état ou entrée-sortie.
c) Enfin, pour ne pas multiplier les exemples de processus, nous voulons réaliser la commande de ce même processus au chapitre 6, et démontrer avec lui les propriétés de robustesse des systèmes de commande qui sont présentés au chapitre 5. Pour cela, il est nécessaire de construire un processus non linéaire qui reste analysable et permette l'interprétation des résultats de la commande.

Processus sans bruit.

Les équations d'état et d'observation du processus sont les suivantes :

$$\begin{cases} x_{1p}(k+1) = a_1\, x_{1p}(k) + a_2\, x_{2p}(k) + (b_1 + 2\, b_2)\, u(k) \\ x_{2p}(k+1) = \dfrac{x_{1p}(k)}{1 + 0{,}01\, x_{2p}(k)^2} + \dfrac{(-\, b_2)}{a_2}\, u(k) \\ y_p(k) = 4\, tanh\left(\dfrac{x_{1p}(k)}{4}\right) \end{cases}$$

avec $a_1 = 1{,}145$; $a_2 = -\,0{,}549$; $b_1 = 0{,}222$; $b_2 = 0{,}181$.

Ces équations ont été construites à partir du modèle linéaire suivant :

$$y_p(k) = a_1\, y_p(k\text{-}1) + a_2\, y_p(k\text{-}2) + b_1\, u(k\text{-}1) + b_2\, u(k\text{-}2)$$

Cette équation résulte de la discrétisation exacte du second ordre linéaire de pulsation ω_n=3, d'amortissement ξ=0,4 et de gain K=1, avec une période T=0,25 s.

Le comportement du processus en réponse à des échelons de commande est représenté sur la figure 14. Autour de zéro, le processus a une dynamique linéaire oscillante, et un gain unité. Le dénominateur de la deuxième variable d'état lui donne un caractère de plus en plus amorti et plus lent pour des amplitudes supérieures à 1 ; il diminue aussi le gain, ce qui est encore accentué par la tangente hyperbolique de l'équation d'observation. Le zéro du processus linéaire a été modifié pour le rendre positif, afin d'éviter des oscillations parasites lors de la commande.

Séquences utilisées pour l'identification.

Le processus étant plus simple que le précédent, et notre but étant de nous rapprocher de conditions expérimentales qui ne permettent pas en général de réaliser une identification " parfaite ", ces séquences sont choisies moins riches que celles utilisées pour l'identification du processus du §I.

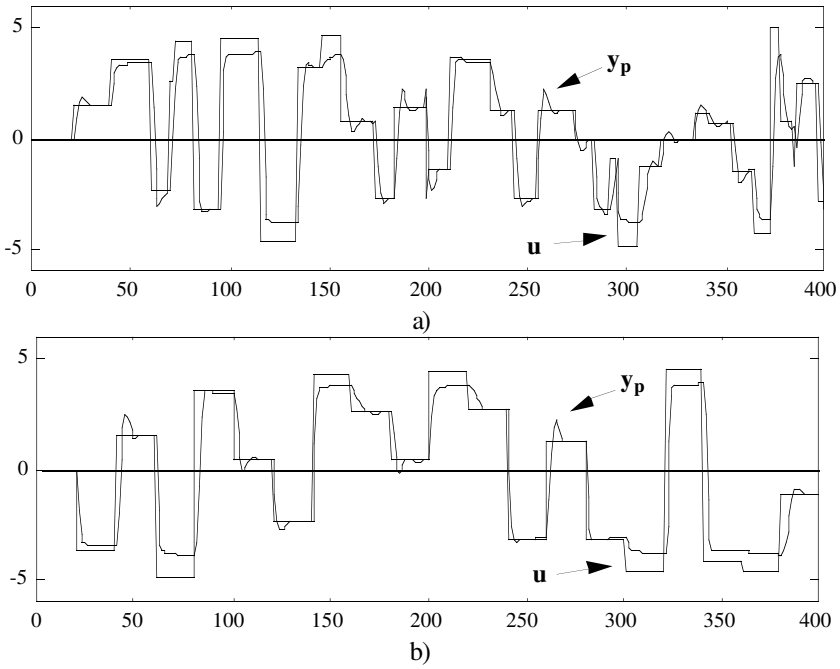


Figure 14.
a) Séquences d'apprentissage ; b) Séquences de test.

a) Séquences d'apprentissage :

La séquence de commande utilisée pour l'apprentissage des réseaux prédicteurs est constituée d'une série de créneaux d'amplitudes aléatoires entre ±5, de durées aléatoires entre 1 et 20 pas d'échantillonnage. La séquence totale comporte 400 pas d'échantillonnage : elle est représentée sur la figure 14a. EQMA est encore l'erreur quadratique moyenne obtenue sur la séquence d'apprentissage.

b) Séquences de test :

La séquence de commande utilisée pour le test des prédicteurs est constituée d'une série de créneaux d'amplitudes aléatoires entre ±5, de durée 20 pas d'échantillonnage. La séquence totale comporte 400 pas d'échantillonnage : elle est représentée sur la figure 14b. EQMT est encore l'erreur quadratique moyenne obtenue sur la séquence de test.

Prédicteurs.

Les prédicteurs utilisés sont des réseaux complètement connectés, à neurones cachés dont la fonction d'activation est la tangente hyperbolique, et à neurone de sortie linéaire.

Processus avec bruit.

Nous effectuons une étude moins exhaustive que pour l'exemple précédent, en partie en raison du fait que l'on ne mesure pas l'état. Soit encore w un bruit blanc à valeur moyenne nulle et de distribution uniforme d'amplitude 0,2 (donc de variance $3{,}33\ 10^{-3}$). Nous considérons les processus perturbés suivants :

- processus avec perturbation additive de sortie (NBSX) :

$$\begin{cases} x_{1p}(k+1) = a_1\, x_{1p}(k) + a_2\, x_{2p}(k) + (b_1 + 2\, b_2)\, u(k) \\ x_{2p}(k+1) = \dfrac{x_{1p}(k)}{1 + 0{,}01\, x_{2p}(k)^2} + \dfrac{(-\, b_2)}{a_2}\, u(k) \\ y_p(k) = 4\, tanh\left(\dfrac{x_{1p}(k)}{4}\right) + w(k) \end{cases}$$

- processus avec perturbation additive d'état (noté ici BE) :

$$\begin{cases} x_{1p}(k+1) = a_1\, x_{1p}(k) + a_2\, x_{2p}(k) + (b_1 + 2\, b_2)\, u(k) \\ x_{2p}(k+1) = \dfrac{x_{1p}(k)}{1 + 0{,}01\, x_{2p}(k)^2} + \dfrac{(-\, b_2)}{a_2}\, u(k) + w(k) \\ y_p(k) = 4\, tanh\left(\dfrac{x_{1p}(k)}{4}\right) \end{cases}$$

II.2. MODÉLISATION D'ÉTAT.

Comme on ne mesure pas l'état, que le processus soit perturbé ou non, on utilise un réseau prédicteur d'état (bouclé) de la forme :

$$\begin{cases} x(k+1) = \varphi_{RN}(x(k),\, u(k);\, C) \\ y(k+1) = \psi_{RN}(x(k),\, u(k);\, C) \end{cases}$$

II.2.1. Processus sans bruit.

Des performances excellentes sont obtenues avec de très petits réseaux (voir tableau 12).

Nombre de neurones cachés	EQMA	EQMT
1 (16 coefficients)	$5{,}5\ 10^{-4}$	$4{,}7\ 10^{-4}$
2 (24 coefficients)	$1{,}6\ 10^{-7}$	$3{,}8\ 10^{-7}$

Tableau 12.
Processus d'état sans bruit : résultats de l'identification de réseaux prédicteurs d'état.

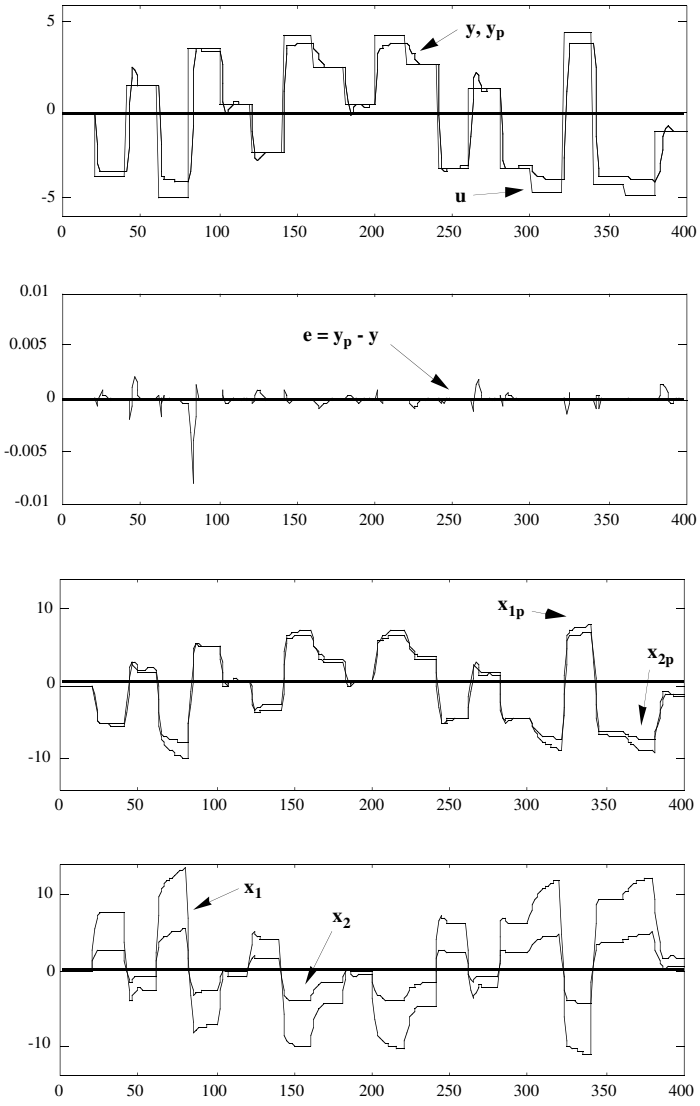


Figure 15.
Processus sans bruit : résultats obtenus avec le réseau prédicteur d'état à deux neurones cachés.

Les résultats de l'identification sont représentés sur la figure 15. Cette figure montre bien que les variables d'état utilisées par le réseau, x_1 et x_2, ne sont en général pas identiques à celles du processus x_{1p} et x_{2p}. L'ajout de neurones supplémentaires n'apporte pas d'amélioration sensible sur la séquence de test. Les identifications suivantes sont ainsi réalisées à l'aide d'un réseau prédicteur d'état possédant deux neurones cachés.

II.2.2. Processus NBSX.

En présence d'une perturbation aléatoire de sortie, le prédicteur d'état permet théoriquement d'identifier le prédicteur optimal. Les résultats du tableau 13 le confirment.

EQMA	EQMT	*EQMD*
$3,2\ 10^{-3}$	$3,5\ 10^{-3}$	*$1,2\ 10^{-4}$*

Tableau 13.
Processus d'état NBSX : résultats de l'identification d'un réseau prédicteur d'état.

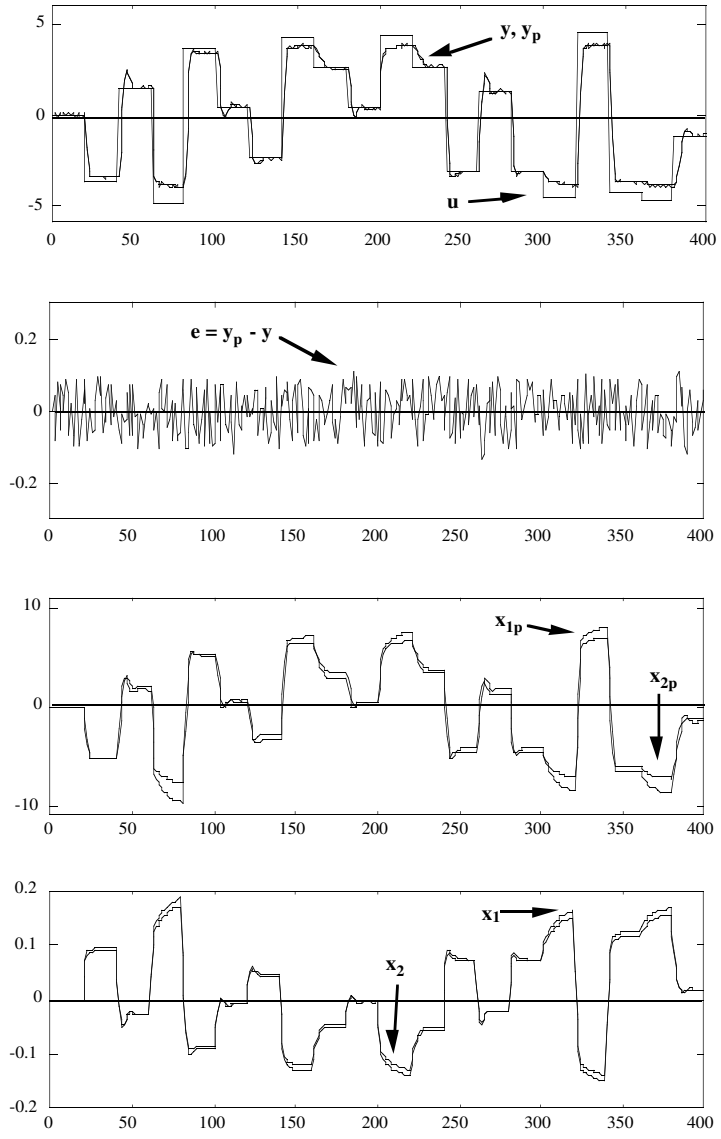


Figure 16.
Processus d'état NBSX : résultats obtenus avec un réseau prédicteur d'état (bouclé).

Comme le montre la figure 16, les variables d'état du prédicteur sont encore différentes. Très souvent, ce sont approximativement des multiples de celles du processus (cf. chapitre 2 §I.2.2.2), lorsque l'on fait la bonne hypothèse toutefois.

II.2.3. Processus BE.

En présence d'un bruit d'état, le prédicteur d'état (bouclé) ne permet théoriquement pas d'identifier le prédicteur optimal. Les résultats du tableau 14 le confirment.

EQMA	EQMT	*EQMD*
$1{,}2\ 10^{-3}$	$1{,}2\ 10^{-3}$	*$6{,}5\ 10^{-4}$*

Tableau 14.
Processus d'état BE : résultats de l'identification d'un réseau prédicteur d'état (bouclé).

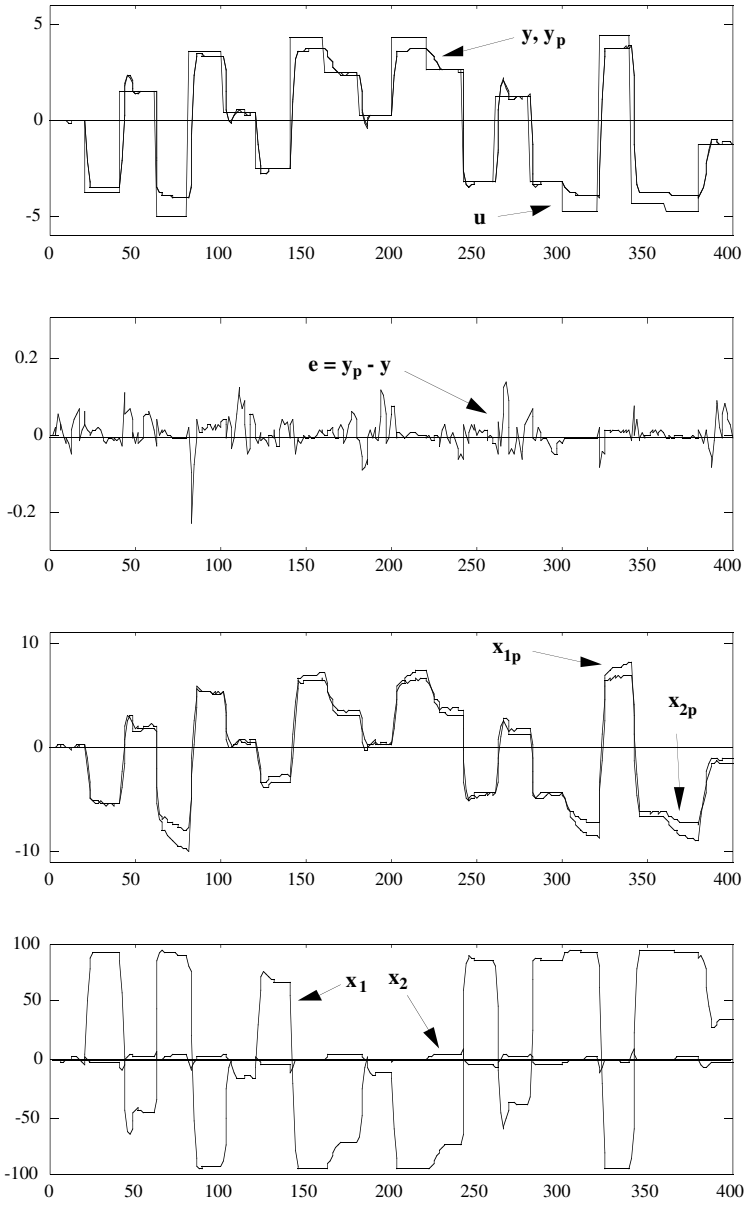


Figure 17.
Processus d'état BE : résultats obtenus avec un réseau prédicteur d'état (bouclé).

L'erreur de prédiction n'est pas un bruit blanc (cf. figure 17). On constate ici que les variables d'état du prédicteur ne sont plus des multiples de celles du processus. Cependant, la comparaison au processus sans bruit (EQMD) montre que le prédicteur identifié est un bon modèle de simulation.

II.3. MODÉLISATION ENTRÉE-SORTIE.

II.3.1. Processus sans bruit.

On peut par exemple utiliser un prédicteur entrée-sortie bouclé (algorithme semi-dirigé) :

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

Avec un prédicteur bouclé, on obtient les résultats donnés du tableau 12.

Nombre de neurones cachés	EQMA	EQMT
4 (35 coefficients)	$9{,}2\ 10^{-4}$	$7{,}1\ 10^{-4}$
5 (45 coefficients)	$2{,}1\ 10^{-4}$	$3{,}1\ 10^{-4}$
…	…	…
10 (110 coefficients)	$1{,}1\ 10^{-6}$	$1{,}2\ 10^{-4}$

Tableau 12.
Processus d'état sans bruit : résultats de l'identification de réseaux prédicteurs d'état.

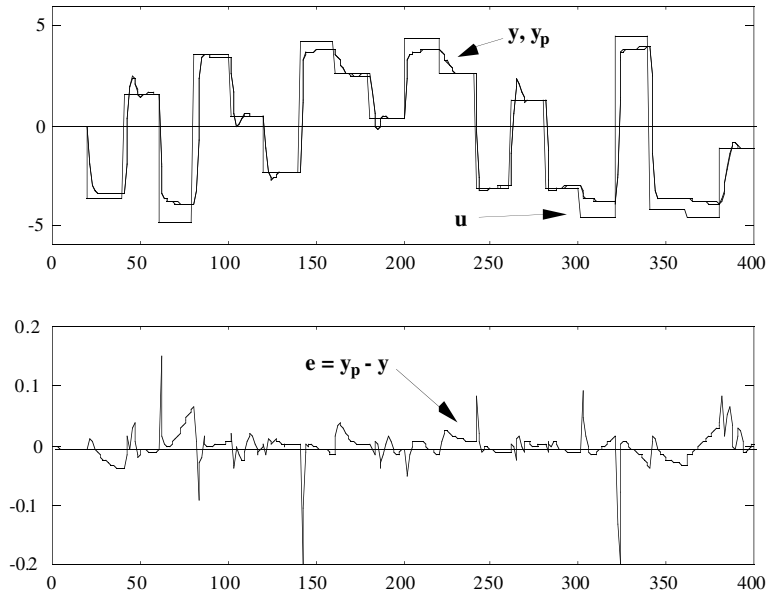


Figure 18.
Processus d'état sans bruit : résultats obtenus avec le réseau prédicteur bouclé entrée-sortie possédant 5 neurones cachés.

L'identification par un réseau prédicteur entrée-sortie se révèle peu économe en nombre de neurones. Avec un prédicteur non bouclé, les résultats sont encore moins bons. Augmenter l'ordre du réseau reste également sans effet appréciable (nous avons vu au chapitre 2 §I.2.2.3 que cela peut être nécessaire pour obtenir une représentation entrée-sortie équivalente au modèle d'état).

Les résultats correspondant au prédicteur bouclé à 5 neurones cachés sont montrés sur la figure 18. Dans un souci de réalisme, en particulier parce que les prédicteurs identifiés doivent servir au chapitre 6 pour la commande, nous nous limitons à cette taille de réseau pour la suite.

II.3.2. Processus NBSX.

Le processus est simulé par :

$$\begin{cases} x_1(k+1) = a_1\, x_1(k) + a_2\, x_2(k) + (b_1 + 2\, b_2)\, u(k) \\ x_2(k+1) = \dfrac{x_1(k)}{1 + 0{,}01\, x_2(k)^2} + \dfrac{(-\, b_2)}{a_2}\, u(k) \\ y_p(k) = 4\, tanh\left(\dfrac{x_1(k)}{4}\right) + w(k) \end{cases}$$

II.3.2.1. Hypothèse vraie NBSX.

S'il existe une représentation entrée-sortie du processus sans bruit, alors il doit exister un prédicteur entrée-sortie bouclé optimal. On peut donc espérer l'identifier à l'aide d'un réseau prédicteur bouclé, ce que confirment les résultats du tableau 17.

EQMA	EQMT	*EQMD*
$3{,}3\ 10^{-3}$	$4{,}5\ 10^{-3}$	*1,2 10-4*

Tableau 17.
Processus d'état NBSX : résultats de l'identification d'un réseau prédicteur entrée-sortie bouclé.

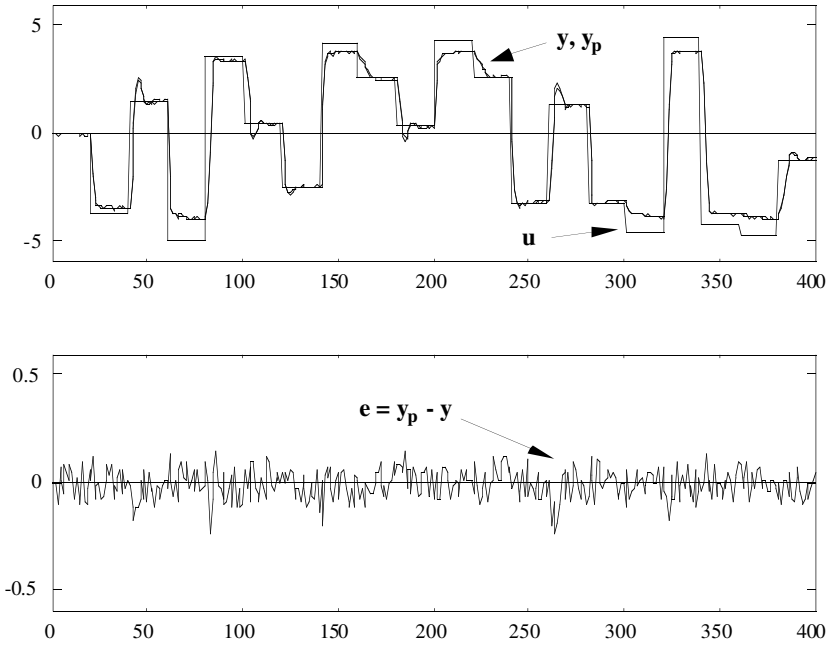


Figure 19.
Processus d'état NBSX : résultats obtenus avec un réseau prédicteur entrée-sortie bouclé.

Malgré (ou peut-être grâce à ?) la petitesse du réseau, les résultats sont quasi optimaux (valeur de l'EQMT proche de la valeur théorique de $3{,}33\ 10^{-3}$). Ceci se traduit par l'obtention d'un modèle de simulation assez satisfaisant (EQMD). Les résultats sont présentés sur la figure 19.

II.3.2.2. Hypothèse fausse.

En revanche, si l'on fait l'hypothèse d'un bruit d'état dominant et que l'on tente l'identification à l'aide d'un prédicteur entrée-sortie non bouclé (algorithme dirigé), les résultats sont mauvais (cf. tableau 18).

EQMA	EQMT	*EQMD*
$6{,}2\ 10^{-3}$	$9{,}1\ 10^{-3}$	$6{,}5\ 10^{-4}$

Tableau 18.
Processus d'état NBSX : résultats de l'identification d'un réseau prédicteur entrée-sortie non bouclé.

La figure 20 montre les résultats correspondants. Utilisé comme modèle de simulation, le prédicteur obtenu est particulièrement médiocre.

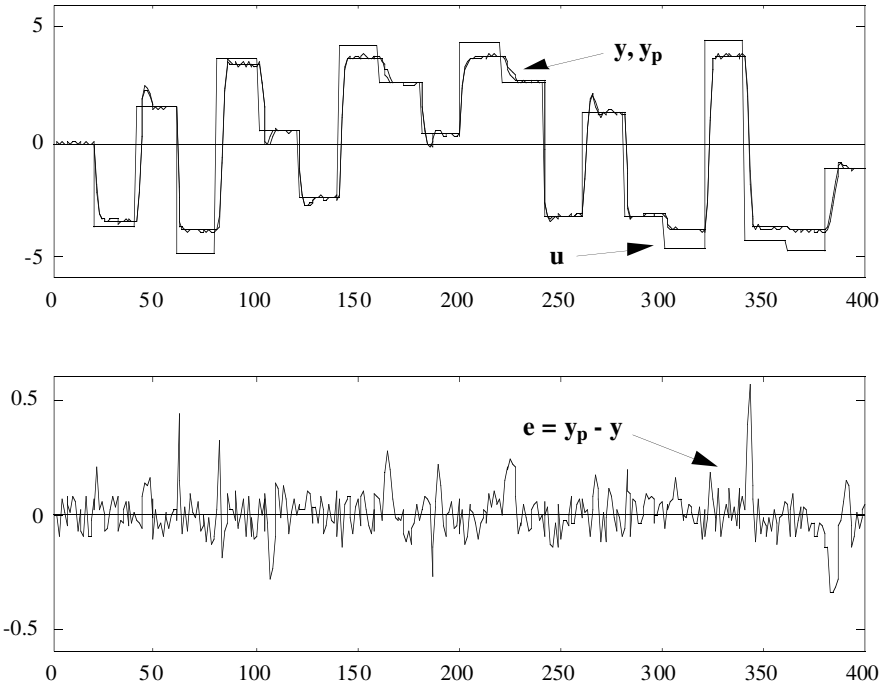


Figure 20.
Processus d'état NBSX : résultats obtenus avec un réseau prédicteur entrée-sortie non bouclé.

Cependant, c'est à partir de ce prédicteur que nous établissons un modèle de simulation pour l'apprentissage de correcteurs au chapitre 6, afin de mettre en évidence et de comparer les propriétés de robustesse des différentes méthodes de commande vis-à-vis de défauts de modélisation. Nous notons ce modèle de simulation (bouclé) :

$$y(k) = \varphi_{RN}^{Sim2}\big(y(k\text{-}1),\ y(k\text{-}2),\ u(k\text{-}1),\ u(k\text{-}2)\big)$$

où φ_{RN}^{Sim2} est la fonction réalisée par la partie non bouclée du réseau prédicteur avec les coefficients fixés aux valeurs obtenues en fin d'apprentissage.

III. MODÉLISATION D'UN ACTIONNEUR HYDRAULIQUE.

III.1. PRÉSENTATION.

La position d'un bras de robot est commandée par un actionneur hydraulique. La position du bras dépend de la pression d'huile dans l'actionneur, pression commandée par l'ouverture d'une vanne. Les variations de l'ouverture de la vanne, c'est-à-dire la séquence de commande {u(k)}, et la pression d'huile correspondante, c'est-à-dire la séquence de sortie $\{y_p(k)\}$, sont montrées sur la figure 21. Ce fichier de données[2] contient 1024 points de mesure : la première moitié d'entre eux est utilisée pour l'apprentissage, la seconde pour l'estimation de la performance

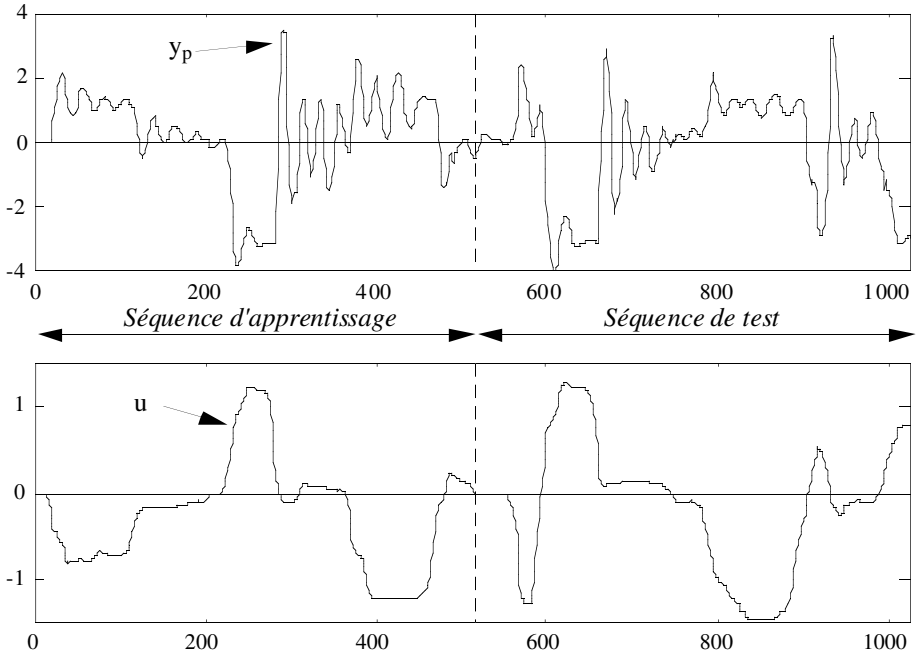


Figure 21.
Séquences d'apprentissage et de test pour la modélisation d'un bras de robot.

L'examen des données montre que les séquences d'apprentissage et de test n'explorent qu'approximativement le même domaine de fonctionnement (signaux de sortie et de commande de même type et de même amplitude). On note qu'aux instants 600 et 850 environ de la séquence de test, l'amplitude de la commande dépasse les amplitudes maximales atteintes sur la séquence d'apprentissage. De plus, pour des entrées similaires, les oscillations du processus sont beaucoup plus entretenues sur la séquence d'apprentissage que sur la séquence de test.

---

[2] Ces données proviennent de la Division of Oil Hydraulics and Pneumatics, Dept. of Mechanical Eng., Linköping University, et nous ont été aimablement communiquées par P.-Y. Glorennec.

Nous avons tout d'abord envisagé des modèles entrée-sortie, puis des modèles d'état.

III.2. MODÉLISATION ENTRÉE-SORTIE.

III.2.1. Modèle-hypothèse NARX.

Nous avons considéré des modèles NARX :

$$y_p(k) = h\left(y_p(k\text{-}1),\ ...,\ y_p(k\text{-}n),\ u(k\text{-}1),\ ...u(k\text{-}m)\right) + w(k)$$

où w est un bruit blanc, pour plusieurs valeurs de n et de m. Les modèles de simulation associés aux meilleurs prédicteurs NARX (c'est-à-dire ces prédicteurs *utilisés bouclés*) ont de très mauvaises performances, aussi bien sur la séquence d'apprentissage que sur la séquence de test (ils sont même parfois instables). Ceci signifie soit que l'hypothèse NARX n'est pas vraie, soit que la part déterministe du comportement dynamique du processus est mal modélisée, par exemple parce que l'on ne dispose pas de toutes les entrées du processus. Quoi qu'il en soit, les modèles de simulation associés aux prédicteur NARX sont inutilisables.

III.2.2. Modèle-hypothèse NBSX.

Le modèle-hypothèse NBSX est le suivant :

$$\begin{cases} x_p(k) = h\left(x_p(k\text{-}1),\ ...,\ x_p(k\text{-}n),\ u(k\text{-}1),\ ...u(k\text{-}m)\right) \\ y_p(k) = x_p(k) + w(k) \end{cases}$$

où w est un bruit blanc. Le prédicteur associé est de la forme :

$$y(k+1) = h\left(y(k),\ \ldots,\ y(k\text{-}n+1),\ u(k),\ \ldots,\ u(k\text{-}m+1)\right)$$

Nous avons effectué l'apprentissage de prédicteurs neuronaux de la structure du prédicteur associé, pour plusieurs valeurs de m et de n :

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

Ces prédicteurs sont bouclés, l'apprentissage est donc semi-dirigé.

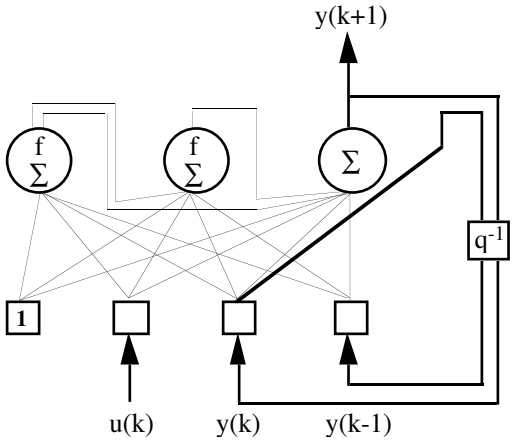
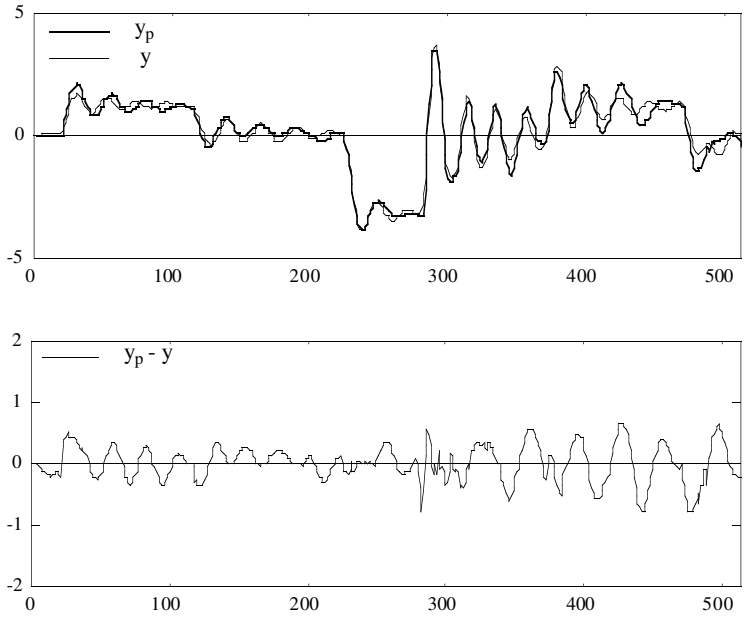
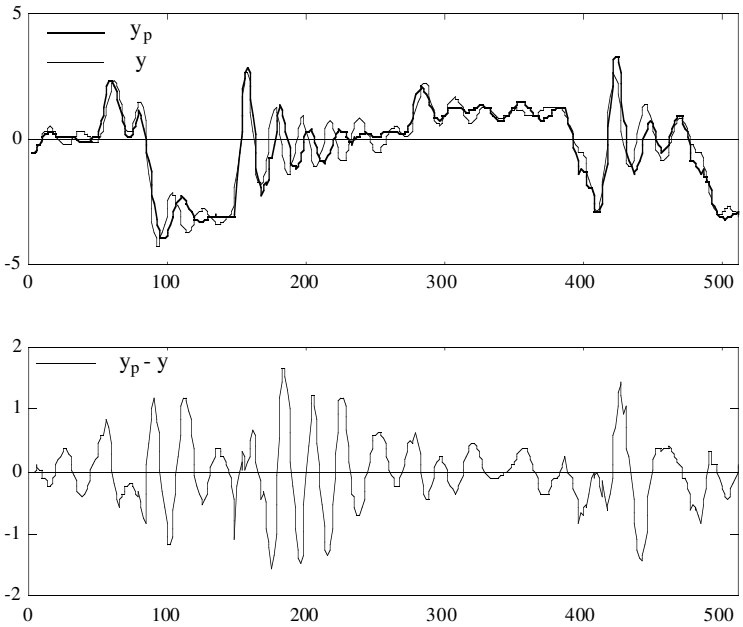


Figure 22.
Prédicteur entrée-sortie bouclé.

Les meilleurs résultats sont obtenus avec n=2 et m=1, et 2 neurones cachés (nous utilisons des réseaux complètement connectés à neurones cachés tangente hyperbolique, et neurone de sortie linéaire). Ce prédicteur est représenté sur la figure 22 (il possède 15 coefficients ajustables). Le passage à 3 neurones cachés (ou plus) diminue l'EQMA, mais augmente l'EQMT.



a) Séquence d'apprentissage : EQMA = 0,085.



b) Séquence de test : EQMT = 0,305.

Figure 23.
Performance du prédicteur entrée-sortie bouclé (NBSX).

Les résultats obtenus avec le prédicteur précédent sur les séquences d'apprentissage et de test sont représentés sur la figure 23. L'EQMA et l'EQMT sont de 0,085 et 0,305 respectivement. L'erreur n'a pas les caractéristiques d'un bruit blanc : l'hypothèse NBSX avec n=2 et m=1 n'est donc pas non plus vraie. Néanmoins, un prédicteur inexact NBSX (apprentissage semi-dirigé) est nécessairement

un meilleur modèle de simulation qu'un prédicteur inexact NARX (apprentissage dirigé) que l'on boucle pour simuler le processus.

III.3. MODÉLISATION D'ÉTAT.

Il apparaît nettement que la plus grande part de l'erreur de prédiction obtenue avec le prédicteur précédent n'est pas due à un bruit aléatoire (on distingue même une composante sinusoïdale), mais à *un défaut de modélisation déterministe*. Nous avons donc mis en œuvre des prédicteurs plus généraux, les prédicteurs d'état. Ces prédicteurs sont associés au modèle-hypothèse d'état suivant :

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

où $x \in \mathbb{R}^n$ est l'état du modèle-hypothèse. Les prédicteurs considérés sont également optimaux pour des modèles-hypothèse avec bruit de sortie. Si le bruit affecte l'état, il est impossible de mettre en œuvre le prédicteur d'état non bouclé associé, puisque les variables d'état ne sont pas mesurées (elles sont même inconnues). Mais, comme nous venons de le préciser, notre but est ici d'améliorer la modélisation de la part déterministe du comportement du processus. Les prédicteurs neuronaux utilisés sont donc de la forme :

$$\begin{cases} x(k+1) = \varphi_{RN}\left(x(k),\, u(k)\right) \\ y(k+1) = \psi_{RN}\left(x(k),\, u(k)\right) \end{cases}$$

où $x \in \mathbb{R}^n$ est l'état du prédicteur. Ces prédicteurs sont bouclés, et l'apprentissage semi-dirigé. Nous avons effectué l'apprentissage de tels prédicteurs pour n=2 et n=3. Le meilleur prédicteur obtenu est du second ordre et possède deux neurones cachés à tangente hyperbolique complètement connectés ; les neurones d'état et le neurone de sortie sont linéaires, et ne sont pas connectés entre eux, ni à l'entrée constante (le prédicteur possède 24 coefficients ajustables). Ce prédicteur est représenté sur la figure 25.

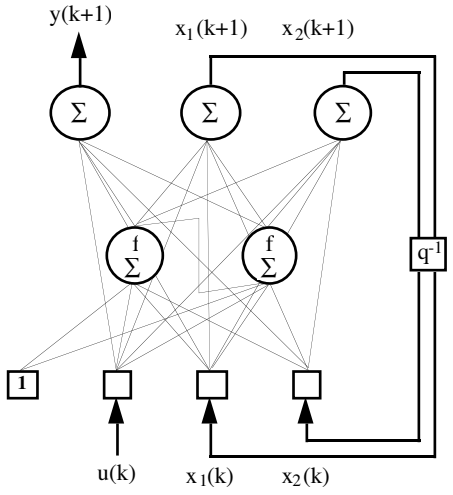
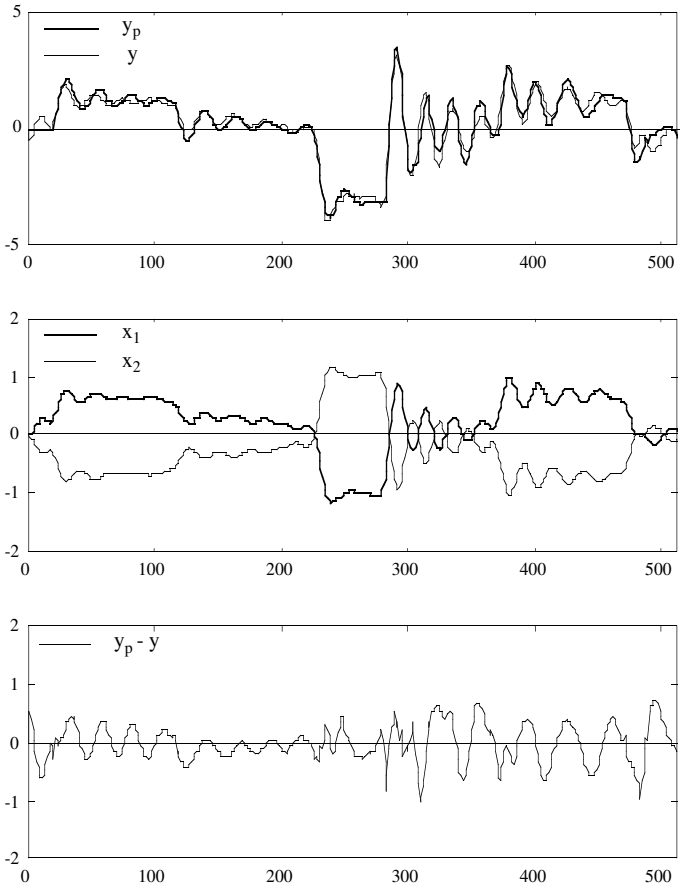
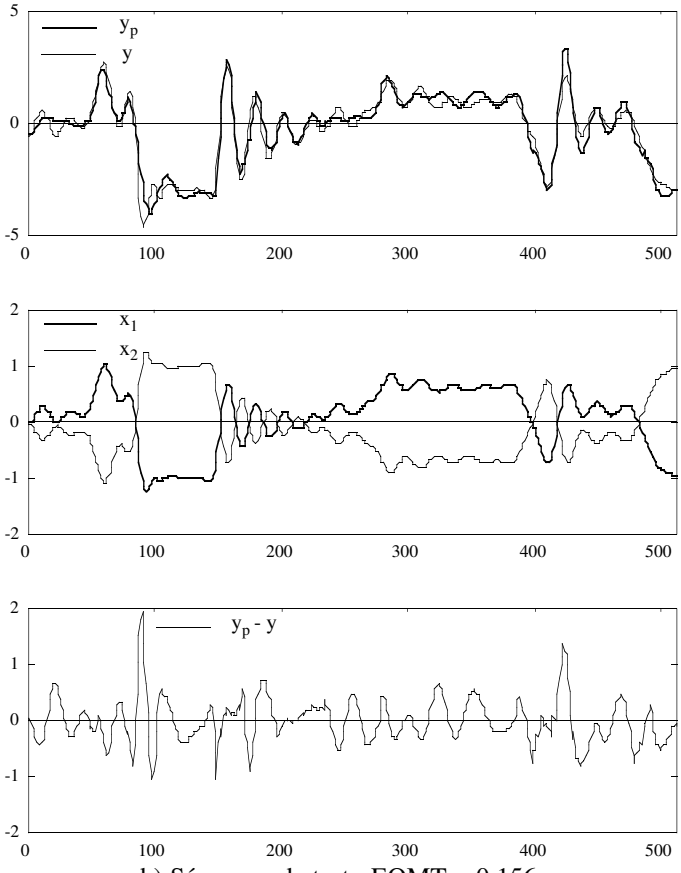


Figure 25.
Prédicteur d'état.



a) Séquence d'apprentissage : EQMA = 0,100.



b) Séquence de test : EQMT = 0,156.

Figure 26.
Performance du prédicteur d'état bouclé.

Les résultats obtenus avec le prédicteur d'état sont représentés sur la figure 26. Ils sont excellents : l'EQMA et l'EQMT sont respectivement de 0,100 et 0,156. Comme en entrée-sortie, le passage à 3 neurones cachés (ou plus) diminue l'EQMA, mais augmente l'EQMT.

La figure 27 montre l'évolution de l'EQMA et de l'EQMT au cours de l'apprentissage. L'apprentissage utilise une méthode de gradient à pas asservi pendant 200 itérations, puis une méthode quasi newtonnienne, jusqu'à ce qu'un minimum soit atteint (cf. chapitre 3 §I.2).

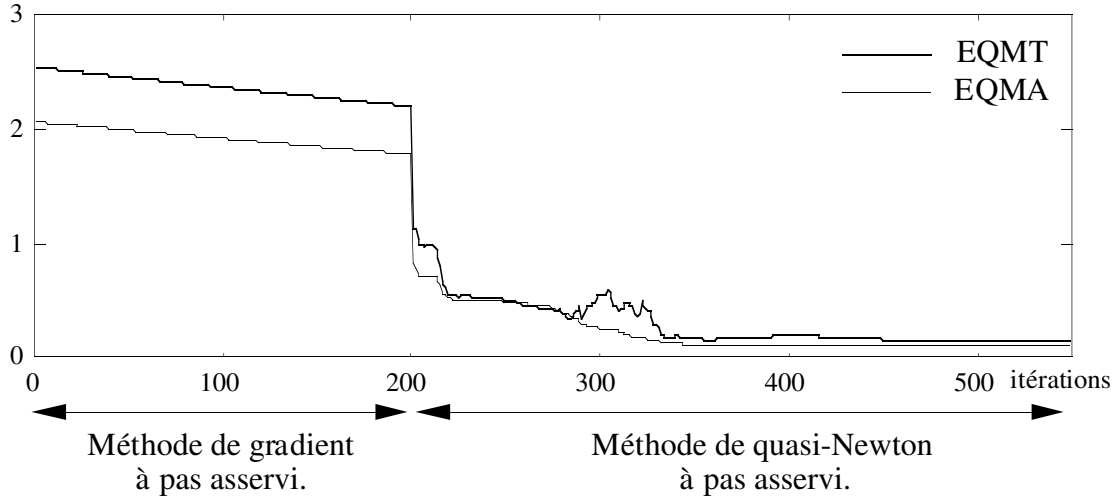


Figure 27.
Évolution de l'EQMA et de l'EQMT au cours de l'apprentissage.

On observe que l'EQMT passe par un maximum (pour 300 itérations) avant d'atteindre le minimum final (550 itérations). Cet exemple montre donc que, d'une manière générale, il n'est pas fondé d'arrêter l'apprentissage dès que l'EQMT se met à croître.

III.4. INTERPRÉTATION ET COMPARAISONS.

Les résultats obtenus à l'aide des meilleurs prédicteurs candidats sont rassemblés sur le tableau 19.

	EQMA	EQMT
Prédicteur entrée-sortie bouclé n=2, m=1, 2 neurones cachés (15 coefficients ajustables)	0,085	0,305
Prédicteur d'état bouclé n=2, 2 neurones cachés (24 coefficients ajustables)	0,100	0,156

Tableau 19.
Récapitulation des performances des meilleurs prédicteurs candidats.

Interprétation.

On observe qu'un prédicteur d'état d'ordre 2 à deux neurones cachés conduit à la meilleure performance, et qu'aucun prédicteur entrée-sortie n'approche cette performance. Tentons d'interpréter ceci.

Supposons qu'il existe un modèle d'état d'ordre 2 décrivant bien le processus. Nous avons vu au chapitre 2 §I.2.2.3, que le comportement d'un modèle d'état peut ne pas être réalisable par un modèle entrée-sortie possédant les mêmes entrées. Si l'on est dans les conditions énoncées dans ce paragraphe, pour un modèle d'état d'ordre n, 2n+1 valeurs précédentes de la sortie et 2n+1 valeurs précédentes de la commande peuvent être nécessaires (soit en tout 10 entrées pour le réseau de neurones si n=2). Or si l'on augmente le nombre d'entrées, on augmente nécessairement aussi le nombre de coefficients du réseau et donc le risque de surajustement. Ceci est d'autant plus vrai que les séquences d'apprentissage et de test contiennent peu de points (ici 512). Nous avons effectivement observé le phénomène de surajustement en ajoutant des entrées supplémentaires (EQMA continue à diminuer tandis que EQMT augmente). Ceci explique que la performance du prédicteur d'état, qui possède peu d'entrées et donc peu de coefficients ajustables, ne puisse être égalée par des réseaux prédicteurs entrée-sortie.

Rappelons nous néanmoins les observations du §III.1 au sujet des séquences d'apprentissage et de test, qui n'explorent pas le même domaine de fonctionnement : toute interprétation en termes de surajustement doit donc être nuancée.

Comparaisons avec les résultats d'autres équipes.

[SJÖ94] réalise l'apprentissage dirigé d'un prédicteur NARX avec n=3, m=2 et 10 neurones cachés. Le test est effectué avec le prédicteur obtenu bouclé (modèle de simulation associé au prédicteur), et fournit un EQMT de 0,585. Une performance analogue est obtenue dans les mêmes conditions dans [BEN94] à l'aide d'un modèle à ondelettes.

[SJÖ93] réalise un EQMT de 0,465 à l'aide du modèle de simulation associé à un prédicteur NARX avec n=2, m=2 et 8 neurones cachés.

Enfin, P.-Y. Glorennec (communication privée) obtient un EQMT de 0,196 à l'aide d'un modèle de simulation " neuroflou " avec n=3, m=2.

Les résultats que nous avons obtenus à l'aide d'un prédicteur d'état à deux neurones cachés, bien meilleurs que les précédents (EQMT de 0,1562), démontrent donc l'intérêt de ces prédicteurs très généraux, ainsi que la faisabilité de leur apprentissage.

CONCLUSION.

Ce chapitre illustre la mise en œuvre pratique de réseaux de neurones pour la modélisation de processus, pour tous les types de modèles-hypothèse présentés au chapitre 2, avec la méthodologie (choix des séquences d'apprentissage et de test, algorithme d'apprentissage) exposée au chapitre 3.

L'exemple du processus entrée-sortie a mis en évidence l'importance du choix du prédicteur et de l'algorithme d'apprentissage pour la modélisation d'un processus avec bruit. Dans le cas où le choix effectué était correct, nous avons modélisé avec succès un processus de type NARMAX fortement non linéaire. Nous avons aussi expliqué quantitativement, dans les cas plus simples NARX et NBSX, l'influence d'une hypothèse fausse sur l'autocorrélation de l'erreur.

L'exemple du processus d'état a mis en lumière la capacité des réseaux prédicteurs d'état à modéliser de façon parcimonieuse les processus les plus généraux. En effet, même si l'existence d'une représentation entrée-sortie est assurée, un prédicteur entrée-sortie est souvent beaucoup moins économe (possède plus de coefficients) qu'un prédicteur d'état réalisant la même performance. Enfin, dans le but d'utiliser le prédicteur pour la commande du processus, nous avons réalisé la modélisation du processus d'état avec bruit à l'aide d'un prédicteur entrée-sortie.

De surcroît, la modélisation d'un processus réel, un actionneur hydraulique, avec d'excellentes performances, confirme l'intérêt pratique des prédicteurs d'état.