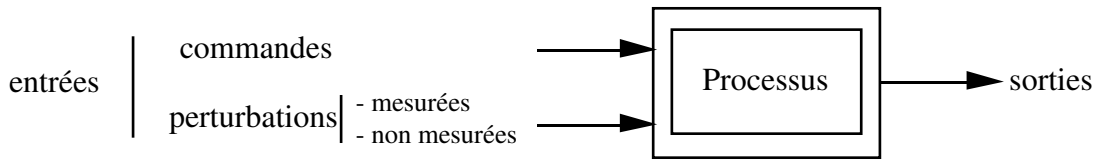
PROBLÉMATIQUE DE LA CONCEPTION D'UN SYSTÈME DE COMMANDE

LE PROCESSUS À COMMANDER.

Les processus que nous considérons sont des systèmes physiques qui évoluent au cours du temps, sous l'effet d'influences internes et externes, et sur lesquels on peut faire des observations, c'est-à-dire des mesures. Les signaux qui nous intéressent sont appelées *variables de sortie*. Les grandeurs agissant sur le processus, et donc sur ses sorties, sont appelées *variables d'entrée*. Le processus est affecté par deux types de variables d'entrée : les *commandes*, sur lesquelles on peut agir, et les *perturbations*, sur lesquelles on n'a pas d'action. Parmi ces dernières, on distingue les *perturbations mesurées* et les *perturbations non mesurées*.



Le processus à commander.

Nous nous intéressons à des *processus dynamiques*, c'est-à-dire des processus dont la valeur des sorties à un instant donné ne dépend pas uniquement des valeurs de ses entrées à ce même instant, mais aussi de leurs valeurs passées. Le concept de processus, ou de système physique, est en fait inséparable du concept de *modèle*, conçu comme *système représentatif d'un système physique*. Nous supposons que les processus qui nous intéressent peuvent être convenablement décrits par des modèles mathématiques. De tels modèles sont caractérisés par des *variables d'état*, qui constituent l'information minimale nécessaire au calcul de l'évolution des sorties, si toutes les entrées sont connues.

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

LES ÉTAPES DE LA CONCEPTION D'UN ORGANE DE COMMANDE.

Choix d'un modèle du processus.

Un modèle du processus est nécessaire à la synthèse de l'organe de commande. La modélisation consiste à rassembler les connaissances que l'on a du comportement dynamique du processus, par une analyse physique des phénomènes mis en jeu, et une analyse des données expérimentales. Ces analyses conduisent à la définition des grandeurs caractérisant le processus, c'est-à-dire ses entrées, ses variables d'état, ses sorties, et aussi les perturbations, mesurables ou non, auxquelles il est soumis. Dans le cadre de ce document, nous faisons l'hypothèse que les sorties du processus sont des fonctions déterministes d'arguments qui sont son état, les commandes, et les perturbations, qu'elles soient mesurées ou non, déterministes ou non. La structure de modèle dont on fait ainsi l'hypothèse qu'elle décrit correctement le processus est appelée *modèle-hypothèse*. En général, la démarche de modélisation conduit à plusieurs modèles-hypothèse concurrents.

Estimation des paramètres du modèle (identification).

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

Conception de l'organe de commande.

Cette étape est celle du choix de l'architecture de l'organe de commande et de la structure des éléments qui le composent, choix effectué en fonction du modèle du processus mis au point et du cahier des charges, qui spécifie les performances désirées pour le système de commande en régulation et, le cas échéant, en poursuite. L'organe de commande comprend nécessairement un élément, le *correcteur,* qui effectue le calcul de la commande à appliquer au processus à partir de la

consigne et de l'état du processus, par exemple. Il comprend en général aussi d'autres éléments : un modèle de référence, un observateur, ou encore un “ modèle interne ”. Le correcteur peut avoir la structure d’un PID, ou celle d’un correcteur par retour d’état linéaire ou non… : cette structure est choisie en fonction de la tâche, définie par le cahier des charges, que doit remplir le système de commande.

Estimation des paramètres du correcteur.

La dernière étape consiste à configurer le correcteur pour que l’organe de commande assure la tâche définie à l’étape précédente. Cette configuration est effectuée hors-ligne à l’aide du modèle : ceci caractérise les méthodes indirectes de synthèse du correcteur. Comme pour la modélisation, nous considérons des correcteurs réalisés par des réseaux de neurones, dont la structure a été fixée lors de la définition de l’organe de commande. L’estimation des coefficients du correcteur correspond à la *phase d’apprentissage* du réseau de neurones. Nous nous intéressons à la synthèse d’organes de commande *non adaptatifs*, c'est-à-dire pour lesquels l'apprentissage du correcteur est entièrement réalisé *préalablement* à son utilisation avec le processus.

En réalité, la conception d'un organe de commande est une procédure itérative, surtout lorsque celui-ci n'est pas adaptatif. Car, comme nous l’avons signalé plus haut, un réseau modèle ne peut être définitivement rejeté ou validé qu'en fonction des performances du système de commande élaboré à partir de ce modèle.

SUITE DE LA PREMIÈRE PARTIE.

La suite de cette première partie suit le fil conducteur qui vient d’être exposé. Le chapitre 2 présente les différents modèles de processus que nous utilisons. Au chapitre 3, nous traitons de l’estimation des paramètres d’un modèle donné, c’est-à-dire de l’apprentissage et de la sélection du meilleur modèle. Le chapitre 4 expose la mise en œuvre de la modélisation pour deux processus simulés, ainsi que pour l’actionneur d’un bras de robot.

Nous présentons au chapitre 5 les divers systèmes de commande que nous avons étudiés, et l’apprentissage des correcteurs utilisés par ces systèmes. Le chapitre 6 enfin, est consacré à la commande des processus simulés introduits au chapitre 4.

Chapitre 2

MODÈLES DE PROCESSUS

INTRODUCTION.

La modélisation d'un processus consiste à trouver un modèle paramétré dont le comportement dynamique approche celui du processus. Ce modèle sera utilisé pour effectuer des prédictions de la sortie du processus, ou pour l'apprentissage d'un correcteur, ou encore pour simuler le processus au sein d'un système de commande.

La première phase d'une modélisation consiste à rassembler les connaissances que l'on a du comportement dynamique du processus (d'après des expériences et/ou une analyse théorique des phénomènes physiques mis en jeu), ce qui conduit à faire plusieurs hypothèses de structures de modèles susceptibles de décrire ce comportement. Ces structures de modèles, appelées *modèles-hypothèse*, sont caractérisées par le nombre et la nature de leurs variables d'entrée (entrées de commandes ou perturbatrices), d'état et de sortie, et éventuellement par les relations entre ces variables. Par exemple, si l'on a des connaissances précises sur le processus, ces relations peuvent être l'expression de lois physiques. Si ces connaissances sont rudimentaires, on est conduit à choisir des modèles de type " boîte noire " ; on peut également combiner les deux approches au sein d'un même modèle. On est ainsi conduit à un ensemble de modèles-hypothèse concurrents. Chacun d'eux est défini à partir d'une ou de plusieurs fonctions inconnues (ou partiellement connues) d'arguments déterminés, qui vont être réalisées par des fonctions paramétrées, ici des réseaux de neurones.

La seconde phase de la modélisation, aussi dite identification, consiste à estimer les paramètres du modèle. Pour cela, on met en œuvre un système d'apprentissage constitué d'un prédicteur de la sortie du processus associé au modèle-hypothèse, et un algorithme d'apprentissage. L'estimation des paramètres du modèle est effectuée en minimisant une fonction de coût définie à partir de l'écart entre les sorties mesurées du processus (séquences d'apprentissages) et les valeurs prédites. La qualité de cette estimation dépend du modèle-hypothèse choisi, de la richesse des séquences d'apprentissage et de l'efficacité de l'algorithme utilisé.

À l'issue de l'identification relative à chaque modèle-hypothèse, on choisit le modèle neuronal donnant la meilleure performance pour l'utilisation prévue (prédiction, simulation, commande).

Le présent chapitre est consacré à la description des modèles-hypothèse considérés dans ce mémoire, et à la définition du prédicteur associé à un modèle-hypothèse donné, qui est l'élément principal du système d'apprentissage. Le chapitre 3 aborde le problème spécifique de l'estimation des paramètres (identification).

I. MODÈLES-HYPOTHÈSE SANS BRUIT.

I.1. MODÈLES D'ÉTAT ET MODÈLES ENTRÉE-SORTIE.

Dans ce paragraphe, nous précisons les types de modèles-hypothèse sans bruit, c'est-à-dire sans perturbation non mesurée, que nous allons considérer. Les entrées de ces modèles-hypothèse sont donc constituées uniquement des commandes et des perturbations mesurées. Pour la modélisation, ces deux types d'entrées sont désignés globalement sous le nom de commande. La distinction sera réintroduite ultérieurement pour la commande du processus.

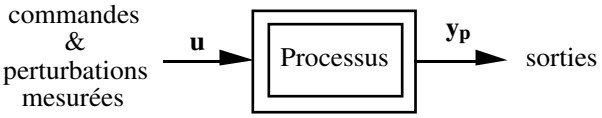


Figure 1.
Modélisation d'un processus sans perturbation non mesurée.

Nous considérons deux classes de modèles-hypothèse :

- la classe des *modèles 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 $x_p \in \mathbb{R}^{n_x}$ est le vecteur d'état, $u \in \mathbb{R}^{n_u}$ est l'entrée de commande, et $y_p \in \mathbb{R}^{n_y}$ est la sortie ; f et g sont des fonctions non linéaires. Nous supposons que tout processus peut être décrit par un modèle-hypothèse d'état.

- la classe des *modèles entrée-sortie* :

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

où $u \in \mathbb{R}^{n_u}$ est l'entrée de commande, et $y_p \in \mathbb{R}^{n_y}$ est la sortie ; h est une fonction non linéaire. Cette classe de modèles est moins vaste que la précédente, mais les modèles entrée-sortie sont souvent plus faciles à mettre en œuvre que les modèles d'état, en particulier si l'état n'est pas mesuré.

Pour simplifier la présentation, nous prendrons $n_u=n_y=1$, et $n_x=n$ (mono-entrée/mono-sortie).

Choix d'un modèle-hypothèse.

Nous faisons l'hypothèse de modèles d'état ou entrée-sortie essentiellement en fonction des possibilités d'analyse physique du processus. Deux cas sont envisageables :

a) Une modélisation physique approfondie peut être effectuée.

Cette modélisation conduit à l'hypothèse d'un modèle d'état, dont les variables d'état ont une signification physique.

Si les variables d'état sont mesurées, les données utilisées pour la modélisation sont la séquence des entrées, et les séquences des variables d'état et des sorties mesurées (séquences d'apprentissage). On peut estimer la fonction f de l'équation d'état à l'aide d'un modèle entrée-sortie, dont les sorties sont les variables d'état du modèle d'état ; l'équation d'observation (fonction g) est modélisée séparément.

Si les variables d'état ne sont pas mesurées, les séquences d'apprentissage sont alors la séquence des entrées et la séquence des sorties mesurées. Deux démarches sont possibles :

- on peut chercher à modéliser le comportement entrée-sortie du processus à l'aide d'un modèle d'état, mais dont l'état n'est pas imposé ;
- on peut faire l'hypothèse plus restrictive d'un modèle entrée-sortie.

Notons que dans ce dernier cas, la meilleure solution, si elle est praticable, est quand même de mettre des capteurs pour mesurer les variables d'état. C'est-à-dire que les nécessités de la commande doivent être prises en considération par l'automaticien dès la conception du processus.

b) Le processus est trop complexe pour être modélisé physiquement.

En pratique, cette situation est équivalente au cas où une modélisation physique est possible, mais où l'état n'est pas mesuré. En effet, les mêmes démarches sont envisageables :

- tenter de modéliser le comportement global du processus à l'aide d'un modèle d'état, sans imposer l'état.
- faire l'hypothèse plus restrictive d'un modèle-hypothèse entrée-sortie.

La mise au point d'un modèle entrée-sortie ou d'un modèle d'état dans ces dernières conditions repose donc surtout sur les résultats expérimentaux. Cependant, comme nous l'avons indiqué dans l'introduction, il peut être très utile, voire indispensable dans le cas de processus complexes, de tirer profit de connaissances *a priori*, même partielles, sur les phénomènes physiques dont le processus est le siège, pour établir la structure de ces modèles. En particulier, ces connaissances peuvent être utilisées pour évaluer l'ordre du modèle.

Notations.

Dans le cas d'un modèle-hypothèse d'état, nous notons $x_p \in \mathbb{R}^n$ l'état du processus et $y_p \in \mathbb{R}$ sa sortie, $x \in \mathbb{R}^n$ l'état du prédicteur et $y \in \mathbb{R}$ sa sortie, et $u \in \mathbb{R}$ la commande. Dans le cas d'un modèle-hypothèse entrée-sortie, nous notons $y_p \in \mathbb{R}$ la sortie du processus et $y \in \mathbb{R}$ celle du prédicteur, et $u \in \mathbb{R}$ la commande.

a) La notation :

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

désigne un réseau prédicteur non bouclé où ψ_{RN} est la fonction réalisée par le réseau muni des coefficients C (voir figure 2).

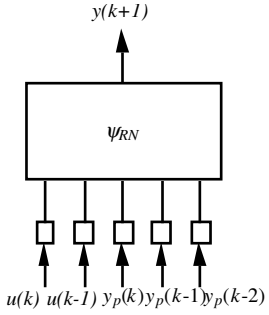


Figure 2.
Exemple de réseau non bouclé avec : n=3 ; m=2.

b) La notation :

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

désigne de façon compacte un réseau prédicteur bouclé d'ordre n, dont les n variables d'état sont la sortie et n-1 valeurs retardées de celle-ci. φ_{RN} est la fonction réalisée par la partie non bouclée du réseau munie des coefficients C (les connexions à coefficients fixés égaux à 1 pour la propagation des n-1 variables d'état sont implicites dans cette écriture, voir annexe I). Un tel réseau est appelé *réseau bouclé entrée-sortie* (voir figure 3).

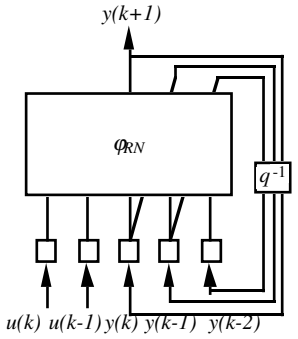


Figure 3.
Exemple de réseau bouclé entrée-sortie avec : n=3 ; m=2.

c) La notation :

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

désigne un réseau prédicteur bouclé d'ordre p, dont les p variables d'état sont les p valeurs successives de l'erreur $e(k) = y_p(k) - y(k)$. φ_{RN} est la fonction réalisée par la partie non bouclée du réseau munie des coefficients C (les connexions à coefficients fixés égaux à 1 pour la propagation des p-1 variables d'état sont implicites dans cette écriture). Un tel réseau est appelé *réseau bouclé entrée-erreur* (voir figure 4).

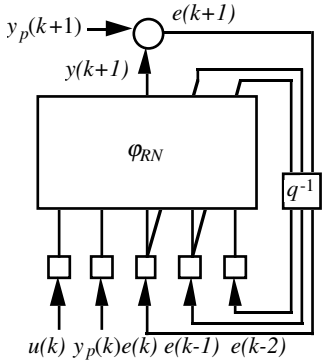


Figure 4.
Exemple de réseau bouclé entrée-erreur avec : n=1 ; m=1 ; p=3.

d) La notation :

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

désigne un réseau prédicteur bouclé dont les variables d'état sont les activités de neurones d'état distincts des neurones de sortie, donc pour lesquels il n'y a pas de valeurs désirées. φ_{RN} et ψ_{RN} sont les fonctions réalisées par la partie non bouclée du réseau munie des coefficients C. Un tel réseau est appelé *réseau d'état*. (voir figure 5a).

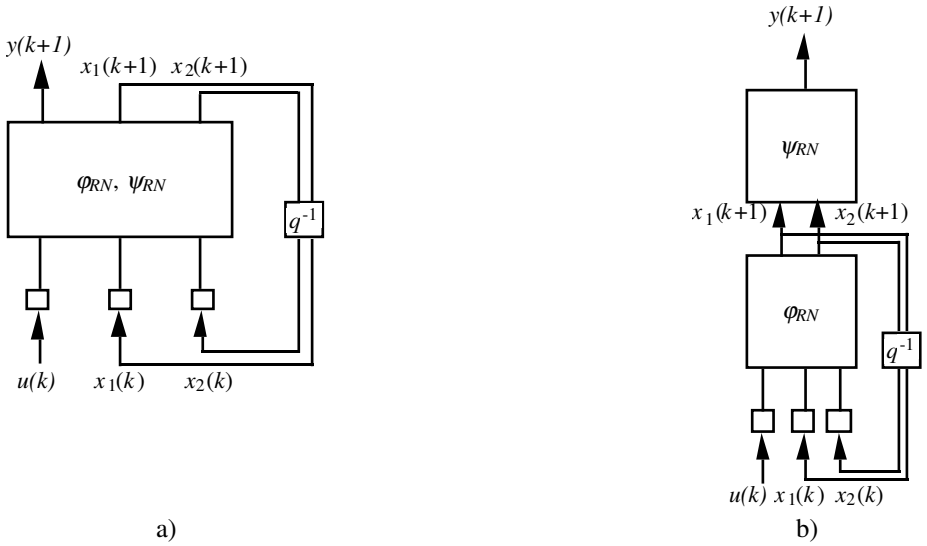


Figure 5.
Exemple de réseau prédicteur d'état avec : n=2.

Lorsque l'on impose une dépendance exclusive de y(k+1) en les variables d'état x(k+1), on note le réseau (voir figure 5b) :

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

I.2. PRÉDICTEURS ASSOCIÉS AUX MODÈLES-HYPOTHÈSE SANS BRUIT.

Définition du prédicteur associé au modèle-hypothèse

Dans le cas où le modèle-hypothèse est complètement déterministe, la notion de prédicteur associé est évidente : c'est un prédicteur qui, si l'hypothèse est vraie, permet de prédire *sans erreur* la sortie du processus.

Pour la clarté de l'exposé, nous commençons par la présentation des modèles-hypothèse entrée-sortie. Comme nous l'avons vu au §I.1, dans le cas d'un modèle-hypothèse d'état, si l'état défini par la modélisation est mesuré, la modélisation est effectuée à l'aide d'un modèle entrée-sortie. Sinon, la modélisation peut être menée d'une manière spécifique que nous décrivons. Enfin, nous montrons qu'il est presque toujours légitime de faire l'hypothèse d'une représentation entrée-sortie, et donc de réaliser la modélisation correspondante, mais au prix d'une augmentation du nombre des arguments du prédicteur associé (§I.2.2.3).

I.2.1. Modélisation entrée-sortie.

Nous appelons ces modèles NDARMA (Non Linéaire Déterministe AutoRégressif à Moyenne Ajustée). Ces modèles sont de la forme :

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

Deux prédicteurs sont possibles :

* le prédicteur associé non bouclé :

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

* le prédicteur associé bouclé :

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

avec des conditions initiales correctes.

h est donc la fonction inconnue que l'on désire approcher à l'aide d'un réseau de neurones. Lors de la phase d'identification, le système d'apprentissage peut indifféremment utiliser les prédicteurs neuronaux suivants :

* un prédicteur neuronal non bouclé :

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

* un prédicteur neuronal bouclé entrée-sortie :

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

Si l'hypothèse est vraie, si la fonction h est dans la famille définie par le réseau de neurones, si l'ensemble d'apprentissage est suffisamment riche, et si l'algorithme est performant, alors la prédiction de la sortie du processus y_p sera très bonne dans le domaine des entrées du réseau défini par l'ensemble d'apprentissage.

I.2.2. Modélisation d'état.

Nous faisons l'hypothèse qu'il existe une relation de la forme :

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

Nous envisageons successivement l'hypothèse où l'état est mesuré, et celle où il ne l'est pas. Pour des raisons que nous explicitons, la deuxième hypothèse conduit à un modèle difficile à manipuler pour la commande du processus. Nous établissons donc ensuite à quelle condition il existe un modèle-hypothèse entrée-sortie équivalent au modèle-hypothèse d'état (§I.2.2.3), mieux adapté à la synthèse et à la mise en œuvre de systèmes de commande. Ce problème a été abordé notamment dans [LEV92] et [NAR92], dont nous utilisons les résultats.

I.2.2.1. L'état est mesuré.

Si l'état est mesuré, il est possible de prédire l'état et la sortie du processus. Deux prédicteurs sont associés au modèle-hypothèse d'état :

* le prédicteur associé non bouclé :

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

* le prédicteur associé bouclé entrée-sortie :

$$\begin{cases} \boldsymbol{x}(\boldsymbol{k}+1) = \boldsymbol{f}\left(\boldsymbol{x}(\boldsymbol{k}), \boldsymbol{u}(\boldsymbol{k})\right) \\ \boldsymbol{y}(\boldsymbol{k}+1) = \boldsymbol{g}\left(\boldsymbol{x}(\boldsymbol{k}+1)\right) \end{cases}$$

avec des conditions initiales correctes.

Le système d'apprentissage utilise donc un premier réseau non bouclé ψ^{f}_{RN} pour approcher la fonction f :

$$x(k+1) = \psi^{f}_{RN}\left(x_p(k), u(k); C_f\right)$$

ou un réseau bouclé entrée-sortie φ^{f}_{RN} :

$$x(k+1) = \varphi^{f}_{RN}\left(x(k), u(k); C_f\right)$$

Il utilise également un second réseau non bouclé ψ^{g}_{RN} pour approcher la fonction g de l'équation d'observation :

$$y(k) = \psi^{g}_{RN}\left(x_p(k); C_g\right)$$

Les séquences d'apprentissage sont constituées de la séquence des entrées {u(k)}, et des séquences des variables d'état $\{x_p(k)\}$ et des sorties mesurées $\{y_p(k)\}$ du processus.

I.2.2.2. L'état n'est pas mesuré.

Si l'état n'est pas mesuré, seule la modélisation du comportement entrée-sortie du processus peut être réalisée. Comme nous l'avons vu au §I.2.2.1, un prédicteur d'état bouclé associé au modèle-hypothèse d'état est évidemment :

$$\begin{cases} \boldsymbol{x}(\boldsymbol{k}+1) = \boldsymbol{f}\left(\boldsymbol{x}(\boldsymbol{k}), \boldsymbol{u}(\boldsymbol{k})\right) \\ \boldsymbol{y}(\boldsymbol{k}+1) = \boldsymbol{g}\left(\boldsymbol{x}(\boldsymbol{k}+1)\right) \end{cases}$$

où x est l'état du prédicteur bouclé.

Le comportement entrée-sortie du processus peut donc théoriquement toujours être identifié à l'aide d'un réseau d'état de la forme :

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

On peut en effet approcher les fonctions f et g avec la précision désirée par la partie non bouclée du réseau φ_{RN} et par le réseau ω_{RN}. Cependant, les séquences d'apprentissage étant constituées de la séquence des entrées {u(k)}, et de la séquence des sorties mesurées $\{y_p(k)\}$ du processus, le même comportement entrée-sortie peut être obtenu avec d'autres couples de fonctions. En effet, pour toute transformation inversible $x = \Phi(x_p)$ de l'état, les prédicteurs de la forme suivante sont équivalents :

$$\begin{cases} x(k+1) = f'\left(x(k), u(k)\right) \\ y(k+1) = g'\left(x(k+1)\right) \end{cases} \text{ avec } \begin{cases} f'\left(. , .\right) = \boldsymbol{\Phi}\left(f\left(\boldsymbol{\Phi}^{-1}(.), .\right)\right) \\ g'\left(.\right) = g\left(\boldsymbol{\Phi}^{-1}(.)\right) \end{cases}$$

L'identification au moyen d'un prédicteur neuronal d'état est donc susceptible de conduire à un modèle dont le comportement entrée-sortie est proche de celui du processus, mais rien n'impose à l'état du réseau de neurones d'être identique à celui du processus (on sait seulement qu'il doit exister entre les états x et x_p une transformation Φ respectant les relations ci-dessus).

Remarque pratique.

Le prédicteur associé peut également s'écrire :

$$\begin{cases} x(k+1) = f\big(x(k), u(k)\big) \\ y(k+1) = g\big(f\big(x(k), u(k)\big)\big) \end{cases}$$

Il est donc également possible d'utiliser un réseau de neurones de la forme générale :

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

Dans la pratique, nous préférons cette forme à la précédente, qui n'impose pas l'utilisation de deux sous-réseaux distincts (voir les notations du §I.1, et les exemples des chapitres 3 et 4).

L'utilisation d'un modèle d'état est intéressante si le but de la modélisation est uniquement de décrire le comportement entrée-sortie du processus. Cependant, si l'on souhaite utiliser le prédicteur d'état pour élaborer un correcteur par retour d'état pour le processus, le correcteur ne pourra être utilisé qu'au sein d'un système de commande utilisant le même prédicteur pour prédire l'état, c'est-à-dire un système avec *modèle interne*. Or il n'est pas toujours possible de mettre en œuvre ces systèmes de commande avec des réseaux de neurones, en particulier si l'inverse du prédicteur est instable (cf. chapitre 5). Pour ne pas limiter les possibilités de commande, il est donc souhaitable de se ramener à une représentation entrée-sortie du processus. Nous étudions maintenant cette alternative.

I.2.2.3. Existence d'un modèle entrée-sortie équivalent.

Le but de ce paragraphe est de donner les conditions pour que l'hypothèse d'existence d'un modèle entrée-sortie soit vraie, et de caractériser cette représentation entrée-sortie. Les conditions pour qu'un modèle d'état non linéaire possède une représentation entrée-sortie dans un domaine donné de son espace d'état sont intimement liées à l'observabilité du modèle. En effet, écrivons le comportement de sa sortie sur l'intervalle [k, k+m-1], on obtient :

$$\begin{cases} y_p(k) = g\big(x_p(k)\big) \\ y_p(k+1) = g\big(f\big(x_p(k), u(k)\big)\big) \\ \dots \\ y_p(k+m\text{-}1) = g\big(f\big(\dots\big(f\big(x_p(k), u(k)\big), u(k+1)\big)\dots, u(k+m\text{-}2)\big)\big) \end{cases} \qquad (1)$$

Dans le cas d'un modèle linéaire d'ordre n, observable, on peut résoudre ce système pour m=n, et l'état $x_p(k)$ peut donc être calculé à partir de n-1 valeurs de l'entrée (u(k), …, u(k+n-2)), et de n valeurs de la sortie ($y_p(k)$, …, $y_p(k+n-1)$). Tout modèle d'état linéaire observable possède donc une représentation entrée-sortie. Dans le cas non linéaire, il est clair que le système (1) ne peut être résolu que si les fonctions f et g vérifient certaines conditions.

[LEV92] propose des conditions d'existence de représentations entrée-sortie *globales*, reposant sur la propriété d'observabilité générique[1]. Un modèle d'état non linéaire est *génériquement*

---

[1] Des conditions suffisantes concernant l'existence de représentations entrée-sortie *locales* sont données dans [LEO85a], établies à partir de la matrice de Hankel du système. Dans le domaine des réseaux de neurones, un autre

observable s'il existe un entier p tel que presque toute séquence d'entrées et de sorties de longueur supérieure ou égale à p détermine l'état de manière unique à partir de (1)[2]. En s'appuyant sur les résultats d'Ayeles [AYE81], il commence par montrer que presque tout système d'état est génériquement observable avec p=2n+1 (n est l'ordre du système, supposé connu, ou sa limite supérieure)[3]. Pour un tel système, il établit l'existence d'un observateur global de la forme :

$$x_p(k) = \Phi_{glo}\left(\quad y_p(k+2n), \ldots, y_p(k), u(k+2n), \ldots, u(k)\right)$$

Supposons que nous sommes dans ce cas. La sortie du processus à l'instant k+2n+1 s'écrit :

$$y_p(k+2n+1) = g\left(f\left(\ldots\left(f\left(x_p(k), u(k)\right), u(k+1)\right)\ldots, u(k+2n)\right)\right)$$

En remplaçant $x_p(k)$ par son expression dans $y_p(k+2n+1)$, on déduit de l'existence de l'observateur celle d'une représentation entrée-sortie globale de la forme :

$$y_p(k+2n+1) = \Psi_{glo}\left(\quad y_p(k+2n), \ldots, y_p(k), u(k+2n), \ldots, u(k)\right)$$

ou encore, en réarrangeant les indices :

$$y_p(k) = \Psi_{glo}\left(\quad y_p(k-1), \ldots, y_p(k-2n-1), u(k-1), \ldots, u(k-2n-1)\right)$$

et donc celle d'un réseau de neurones approchant Ψ_{glo} avec la précision désirée.

Ces résultats nous autorisent donc à considérer que les conditions nécessaires à l'existence d'une représentation entrée-sortie globale pour un modèle-hypothèse d'état sont souvent remplies, et donc à remplacer l'hypothèse d'un modèle d'état par celle d'un modèle entrée-sortie de la forme :

$$\boldsymbol{y_p(k) = h\left(y_p(k-1), \ldots, y_p(k-p), u(k-1), \ldots, u(k-p)\right)}$$

où p est un entier compris entre n_x, l'ordre du modèle-hypothèse d'état, et $2n_x+1$.

Le système d'apprentissage doit utiliser un réseau de neurones dont la structure est celle des prédicteurs associés, soit :

- un réseau prédicteur non bouclé :

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

- un réseau prédicteur bouclé entrée-sortie :

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

où p est un entier que l'on fera varier entre n_x, l'ordre du modèle-hypothèse d'état, et $2n_x+1$.

L'inconvénient de la modélisation entrée-sortie par rapport à la modélisation d'état est de nécessiter un prédicteur possédant un plus grand nombre d'arguments, et donc de coefficients

---

résultat est donné dans [LEV92] ; il repose sur les propriétés d'observabilité du modèle linéarisé autour d'un point d'équilibre.

[2] Presque toute séquence : au sens où l'ensemble des séquences de longueur supérieure ou égale à p qui permettent de reconstituer l'état est dense dans l'ensemble des séquences de longueur supérieure ou égale à p.

[3] Presque tout système : au sens où , si g est une fonction de Morse C^∞ (c'est-à-dire ne possède que des points critiques isolés), l'ensemble des fonctions f C^∞ pour lesquelles le système :

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

possède la propriété d'observabilité générique avec p=2n+1, est dense dans C^∞.

ajustables, ce qui pose un problème si les séquences d'apprentissage sont peu riches. Nous montrons sur un tel exemple, réel (modélisation d'un bras de robot, chapitre 4 §III), que la modélisation d'état peut dans ce cas s'avérer bien plus efficace que la modélisation entrée-sortie, même en augmentant l'ordre du modèle entrée-sortie.

II. MODÈLES-HYPOTHÈSE AVEC BRUIT.

L'expérience montre que des modèles déterministes comme ceux que nous venons de présenter sont insuffisants pour décrire la plupart des processus réels, parce que l'on ne mesure jamais tous les signaux qui les influencent, et en raison de l'incertitude qui affecte toute mesure. Nous présentons dans ce paragraphe des modèles qui rendent mieux compte de l'incertitude qui affecte la connaissance de tout processus physique.

II.1. PERTURBATIONS NON MESURÉES.

On distingue deux classes de perturbations non mesurées : les perturbations déterministes et les perturbations de type bruit.

Perturbation déterministe.

Une perturbation déterministe peut être modélisée par une entrée (échelon, rampe, sinus,…) qui survient ou se modifie à des instants aléatoires. Par exemple, l'encrassement d'un appareil, ou le changement de la masse de carburant d'un véhicule, peuvent être modélisés par une entrée de type rampe. Dans ce cas, on peut concevoir un système d'identification adaptatif, qui sera l'un des éléments d'un système de commande adaptatif.

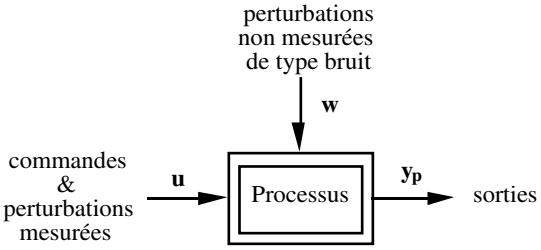


Figure 6.
Modélisation d'un processus avec perturbations non mesurées de type bruit.

Perturbation de type bruit.

Nous appelons perturbation de type bruit toute entrée modélisée par une séquence de variables aléatoires. Par exemple, toute mesure est entachée d'incertitude. On peut ne pas prendre du tout en considération ce caractère incertain, et utiliser des modèles déterministes de variables déterministes, comme nous l'avons fait dans les paragraphes précédents. On peut, à l'autre extrême, considérer

toute variable d'entrée, d'état ou de sortie d'un modèle comme une réalisation de processus aléatoire. Nous nous intéressons à une catégorie de modèles intermédiaire, des modèles déterministes dont certaines entrées sont des séquences de variables aléatoires.

Dans le cadre de ce mémoire, nous ne considérons pour la modélisation que des perturbations non mesurées de type bruit. Les perturbations non mesurées déterministes doivent être compensées par l'organe de commande (voir chapitre 5 §II).

II.2. PRÉDICTEURS ASSOCIÉS AUX MODÈLES-HYPOTHÈSE AVEC BRUIT.

Définition d'un prédicteur associé à un modèle hypothèse avec bruit.

Le prédicteur associé fournit l'espérance mathématique de la sortie du processus à l'instant k+1 conditionnée par les mesures jusqu'à l'instant k. Ce prédicteur théorique minimise la variance de l'erreur de prédiction. Le prédicteur réel (un réseau de neurones), dont les paramètres sont estimés en minimisant l'erreur de prédiction quadratique moyenne, calcule une estimation de cette espérance mathématique conditionnelle. Nous dirons encore que le prédicteur associé est optimal, au sens de la variance de l'erreur de prédiction.

Exemple : prédicteur associé au modèle-hypothèse linéaire ARMAX.

Soit le modèle-hypothèse ARMAX (Auto-Régressif à Moyenne Ajustée avec entrée eXogène) :

$$A(q)\, y_p(k) = B(q)\, u(k) + C(q)\, w(k) = q^{-d}\, B'(q)\, u(k) + C(q)\, w(k)$$

où A est un polynôme monique[4] de degré n, B un polynôme de degré $m = d+m'$, et C un polynôme monique de degré p ; {w(k)} est une séquence de variables aléatoires indépendantes (bruit blanc) à valeur moyenne nulle. Goodwin [GOO84] montre que, si les zéros de C sont à l'intérieur du cercle unité, la prédiction optimale à d pas y(k+d) satisfait :

$$C(q)\, y(k+d) = G(q)\, y_p(k) + F(q)\, B'(q)\, u(k)$$

où :

$$y_p(k+d) - y(k+d) = F(q)\, w(k+d)$$

et où F (monique) et G sont les uniques polynômes de degrés respectifs d-1 et n-1 tels que :

$$C(q) = F(q)\, A(q) + q^{-d}\, G(q)$$

La prédiction à un pas, donc avec d=1, conduit à F(q) = 1, et donc à :

$$y_p(k+1) - y(k+1) = w(k+1).$$

N. B. Cette forme prédicteur du modèle ARMAX n'est qu'une écriture équivalente à la forme prédicteur du filtre de Kalman [GOO84].

---

[4] Un polynôme en q^{-1} ***monique*** est de la forme : $\mathbf{1} + a_1 q^{-1} + a_2 q^{-2} \ldots$

Cet exemple met en lumière deux faits importants :

a) La variance de l'erreur de prédiction obtenue avec le prédicteur associé est bien minimale, puisque c'est celle du bruit blanc (qui est non prédictible par définition).

b) Faisons apparaître la parenté du *prédicteur à 1 pas* avec le modèle-hypothèse :

On a $F(q) = 1$ et $G(q) = q\left(C(q) - A(q)\right)$, soit : $C(q)\, y(k+1) = q\,(C(q) - A(q))\, y_p(k) + B'(q)\, u(k)$.

En réarrangeant les termes, il vient :

$y(k+1) = (1 - A(q))\, y_p(k+1) + B'(q)\, u(k) + (C(q) - 1)\, e(k+1)$

Quel que soit le retard d du processus, le calcul des paramètres du prédicteur à 1 pas associé au modèle-hypothèse fournit donc directement les coefficients des polynômes A, B et C, c'est-à-dire les paramètres du modèle-hypothèse lui-même, puisque ce dernier s'écrit :

$$y_p(k+1) = \left(1 - A(q)\right)\, y_p(k+1) + B'(q)\, u(k) + \left(C(q) - 1\right)\, w(k+1) + w(k+1)$$

De même, pour la modélisation d'un processus non linéaire, si le prédicteur neuronal du système d'apprentissage a la structure du prédicteur à 1 pas associé au modèle-hypothèse, si ses coefficients sont modifiés de manière à minimiser une estimation de la variance de l'erreur de prédiction, si l'hypothèse est vraie, si le prédicteur neuronal est de taille suffisante, si les séquences d'apprentissage sont bien choisies (i.e. si l'estimation de la variance est bonne), et enfin si l'algorithme est performant, alors le réseau prédicteur donnera une bonne approximation de la (des) fonction(s) du modèle-hypothèse. *L'utilisation de réseaux de neurones permet donc d'étendre à une large classe de modèles non linéaires la forme prédicteur du filtre de Kalman.*

II.2.1 Modélisation entrée-sortie avec bruit.

Soit {w(k)} une séquence de variables aléatoires indépendantes (bruit blanc) à valeur moyenne nulle, intervenant dans le modèle-hypothèse d'une manière que nous allons préciser. Le modèle-hypothèse entrée-sortie avec bruit le plus général que nous considérons est le modèle NARMAX, (Non linéaire Auto-Régressif à Moyenne Ajustée avec entrée eXogène), extension non linéaire du modèle ARMAX. Cependant, des cas particuliers de ce modèle sont extrêmement utiles, et les prédicteurs associés faciles à manipuler. L'étude théorique des différents modèles-hypothèse et des prédicteurs associés a déjà été effectuée pour les réseaux de neurones dans [NER92a&b]. Cependant, il n'existe pas à notre connaissance de mise en œuvre de prédicteurs associés à des modèles généraux NARMAX, mise en œuvre beaucoup plus délicate que celle des cas particuliers. Des exemples de processus NARMAX et de l'apprentissage des prédicteurs associés sont donnés dans [LEO87], mais pour des modèles (et des processus !) polynômiaux. [BIL92] présente la modélisation neuronale d'un processus NARMAX dans le cas particulier d'un bruit MA (voir §II.2.1.3), mais n'utilise pas le prédicteur associé. Pour combler cette lacune, nous spécifions ici le prédicteur qui s'impose pour la modélisation NARMAX, et en donnons des exemples au chapitre 4.

Les différents modèles-hypothèse avec bruit que nous présentons sont classés *selon le type de prédicteur neuronal* qu'il faut mettre en œuvre pour réaliser le prédicteur associé, et *dans l'ordre de la complexité croissante de ce prédicteur.*

II.2.1.1. Modélisation avec bruit d'état additif NARX.

Ce modèle (Non linéaire Auto-Régressif avec entrée eXogène) est le modèle non linéaire avec bruit qui conduit au prédicteur le plus simple. Dans le cas linéaire, noté ARX, ce modèle est encore appelé " equation error " [LJU87]. Il s'écrit :

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

Le prédicteur associé est non bouclé :

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

Ce prédicteur est effectivement celui qui minimise la variance de l'erreur de prédiction puisque :
$y_p(k+1) - y(k+1) = w(k+1)$

Le système d'apprentissage doit donc utiliser un réseau de neurones prédicteur ayant la structure du prédicteur associé, soit un réseau non bouclé de la forme :

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

Si l'hypothèse est vraie et si les autres conditions d'une bonne identification sont réunies (séquences d'apprentissage représentatives, algorithme adéquat et performant), alors la fonction réalisée par le réseau de neurones sera une bonne approximation de h.

II.2.1.2. Modélisation avec bruit de sortie additif NBSX.

Le modèle NBSX (Non linéaire avec Bruit de Sortie additif et entrée eXogène) peut représenter un processus affecté d'un bruit de mesure additif non corrélé. Dans le cas linéaire, que nous notons BSX, ce modèle est appelé " output error " [LJU87]. Il s'écrit de la manière suivante :

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

Le prédicteur associé est un prédicteur d'ordre n, dont l'état est constitué de la sortie et des n-1 valeurs précédentes de celle-ci :

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

Ce prédicteur est optimal puisque si l'on suppose les erreurs de prédictions précédentes égales au bruit, on a :

$$\begin{cases} y_p(k) - y(k) = w(k) \\ \ldots \\ y_p(k\text{-}n+1) - y(k\text{-}n+1) = w(k\text{-}n+1) \end{cases} \quad avec \quad \begin{cases} y_p(k) = x_p(k) + w(k) \\ \ldots \\ y_p(k\text{-}n+1) = x_p(k\text{-}n+1) + w(k\text{-}n+1) \end{cases} \quad \Rightarrow \begin{cases} y(k) = x_p(k) \\ \ldots \\ y(k\text{-}n+1) = x_p(k\text{-}n+1) \end{cases}$$

alors :

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

et l'on a bien :
$y_p(k+1) - y(k+1) = y_p(k+1) - x_p(k+1) = w(k+1)$

Le système d'apprentissage doit ici utiliser un réseau de neurones prédicteur bouclé entrée-sortie d'ordre n :

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

Ici encore, si l'hypothèse est vraie, et si toutes les conditions d'une bonne identification sont réunies, alors la fonction réalisée par la partie non bouclée du réseau de neurones sera une bonne approximation de h.

II.2.1.3. Modélisation NARMAX.

Dans le domaine du Traitement du Signal et de l'Automatique, les modèles précédents sont souvent insuffisants pour décrire correctement les processus. Le modèle-hypothèse non linéaire le plus général que nous considérons est le modèle NARMAX (Non linéaire Auto-Régressif à Moyenne Ajustée avec entrée eXogène). Ce modèle est une extension au non linéaire du modèle linéaire ARMAX [LEO85b] [CHE89] [CHE90a&b]. Il est de la forme :

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

Le prédicteur associé est d'ordre p, ses variables d'état étant les p erreurs de prédiction passées :

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

où $e(k) = y_p(k) - y(k)$

Ce prédicteur est optimal puisque si l'on suppose les erreurs de prédiction passées égales au bruit :

$e(k) = w(k), \ldots, e(k\text{-}p+1) = w(k\text{-}p+1)$

alors l'erreur de prédiction à k+1 est elle aussi égale au bruit :

$e(k+1) = y_p(k+1) - y(k+1) = w(k+1)$

Le système d'apprentissage doit donc utiliser un réseau de neurones prédicteur bouclé d'ordre p donc les variables d'état sont les p erreurs de prédiction passées (réseau bouclé entrée-erreur) :

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

De même, si l'hypothèse est vraie, et si toutes les conditions d'une bonne identification sont remplies, alors la fonction réalisée par la partie non bouclée du réseau de neurones est une bonne approximation de h.

Un cas particulier de l'hypothèse NARMAX est le cas d'un bruit corrélé additif ARMA (Auto-Régressif à moyenne Ajustée) :

$$\boldsymbol{A(q)\, y_p(k) = h\left(y_p(k\text{-}1), \ldots, y_p(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}m)\right) + C(q)\, w(k)}$$

où A et C sont deux polynômes moniques de degrés respectifs n et p. Le prédicteur associé est :

$$\boldsymbol{y(k+1) = h\left(y_p(k), \ldots, y_p(k\text{-}n+1), u(k), \ldots, u(k\text{-}m+1)\right) + \left(1 - A(q)\right) y_p(k+1) + \left(C(q) - 1\right) e(k+1)}$$

Le réseau prédicteur utilisé pour l'apprentissage sera simplifié en imposant une dépendance linéaire de sa sortie par rapport à l'erreur de prédiction. Il faut utiliser un réseau composé de deux sous-réseaux, ou de deux réseaux distincts :

$$y(k+1) = \psi^{nl}_{RN}\left(y_p(k), \ldots, y_p(k\text{-}n+1), u(k), \ldots, u(k\text{-}m+1); C_{nl}\right) + \varphi^{lin}_{RN}\left(e(k+1); C_{lin}\right)$$

où le réseau bouclé φ^{lin}_{RN} est composé d'un seul neurone linéaire, destiné à estimer les coefficients de C(q)–1. Le réseau non bouclé ψ^{nl}_{RN} doit réaliser la somme de la fonction h et de (1–A(q)) y_p(k+1) (il est donc conseillé d'utiliser un réseau complètement connecté). Le même type de réseau sera utilisé dans le cas de la modélisation d'un bruit coloré MA (A(q)=1), le prédicteur associé étant :

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

Le prédicteur associé au modèle-hypothèse non linéaire affecté d'un bruit additif ARMA est d'ailleurs donné dans [GOO84] sous une forme équivalente[5].

II.2.2. Modélisation d'état avec bruit.

II.2.2.1. L'état est mesuré.

Si l'état est mesuré, quelle que soit la nature du bruit, il est possible de se ramener à deux modèles-hypothèse entrée-sortie, comme dans le cas déterministe. Le système d'apprentissage doit alors utiliser un réseau de neurones ayant la structure du prédicteur associé à l'équation d'état, et un réseau statique destiné à modéliser l'équation de sortie. Les séquences d'apprentissage sont constituées de la séquence de commande, de la séquence des variables d'état mesurées et de la séquence des sorties mesurées sur le processus.

II.2.2.2. L'état n'est pas mesuré.

II.2.2.2.1. Modélisation d'état NBSX.

Ce modèle-hypothèse se prête bien à la description d'un modèle d'état avec bruit de mesure :

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

Un prédicteur optimal associé est :

$$\begin{cases} \boldsymbol{x(k+1) = f\left(x(k), u(k)\right)} \\ \boldsymbol{y(k+1) = g\left(x(k+1)\right)} \end{cases}$$

En effet, si l'on suppose que $x(k) = x_p(k)$, on a $x(k+1) = x_p(k+1)$, et :

$e(k+1) = y_p(k+1) - y(k+1) = w(k+1)$

Comme dans le cas déterministe, tous les prédicteurs dont l'état est une transformation de x_p vérifiant les conditions du §I.2.2.2 sont également optimaux.

Le système d'apprentissage doit donc utiliser un réseau de neurones prédicteur bouclé sur l'état non imposé (réseau d'état) de la forme :

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

ou bien encore :

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

---

[5] L'auteur ne fait jamais apparaître l'erreur de prédiction dans l'expression du prédicteur associé ; le prédicteur non linéaire avec bruit ARMA est ainsi donné sous la forme :

$$y(k+1) = h\left(y_p(k), \ldots, y_p(k-n+1), u(k), \ldots, u(k-m+1)\right) + \left(A(q) - C(q)\right) y_p(k+1) + \left(C(q) - 1\right) y(k+1)$$

II.2.2.2.2. Modélisation d'état avec autre bruit.

Si le bruit n'est pas purement un bruit de sortie, le prédicteur associé au modèle-hypothèse a nécessairement pour arguments les variables d'état du processus. Ces variables n'étant pas mesurées, le système d'apprentissage, qui ne peut qu'utiliser un réseau de neurones prédicteur bouclé sur l'état comme au paragraphe précédent, conduira à un prédicteur sous-optimal, et donc à un moins bon modèle. La meilleure solution est dans ce cas de faire l'hypothèse NARX ou NARMAX (voir §II.2.2.3 pour la validité de cette hypothèse), et d'effectuer l'identification avec un prédicteur neuronal ayant la structure du prédicteur associé.

II.2.2.3. Existence d'un modèle entrée-sortie équivalent.

Des conditions de l'existence d'une représentation entrée-sortie *locale* de type NARMAX pour des modèles d'états discrets affectés de perturbations aléatoires sont données par Leontaritis et Billings dans [LEO85b] et sont les seules à notre connaissance pour des modèles non linéaires généraux. Il reste donc possible de supposer ces conditions réunies dans le domaine considéré de l'espace d'état, et de faire l'hypothèse NARMAX. Le système d'apprentissage devra donc utiliser le prédicteur associé à l'hypothèse NARMAX.

CONCLUSION.

Dans ce chapitre, nous avons déterminé la structure du prédicteur associé à un modèle-hypothèse donné qui doit être utilisé par le système d'apprentissage, ceci aussi bien pour des modèles entrée-sortie que pour des modèles d'état, que l'état soit mesuré ou non. Nous avons aussi donné brièvement les conditions du succès de l'apprentissage du prédicteur.

Le chapitre 3 suivant précise ces conditions en examinant la mise en œuvre du système d'apprentissage complet qui utilise : un prédicteur associé au modèle-hypothèse, et dont les coefficients sont à estimer, des séquences et un algorithme d'apprentissage. Nous abordons également le problème de la sélection d'un prédicteur parmi les prédicteurs associés à différentes hypothèses. Nous définissons enfin les utilisations possibles du prédicteur retenu (prédiction, simulation, élaboration d'un correcteur pour le processus).