

## **CHAPITRE V**

### **Étude de quelques exemples**

## **I. INTRODUCTION.**

Dans ce chapitre, nous présentons deux exemples de mise en œuvre des réseaux et algorithmes présentés dans ce mémoire pour la modélisation de processus. Le premier est simulé à partir d'une équation aux différences. Le second est un processus réel connu à partir d'une séquence de mesures.

Les prédicteurs que nous considérerons seront des réseaux de fonctions dorsales à sigmoïdes et des réseaux d'ondelettes fondés sur la transformée continue ; nous avons vu que les premiers permettent de réaliser des approximateurs plus parcimonieux que les modèles linéaires par rapport aux paramètres ajustables. D'autre part, il s'agit de modélisation dynamique de processus : on souhaite donc obtenir des modèles de simulation. Les prédicteurs construits à l'aide de réseaux d'ondelettes fondés sur la transformée discrète ne peuvent être candidats dans ce cas, comme nous l'avons vu dans le chapitre IV.

Nous présenterons, en premier lieu, un processus simulé. Il est en effet intéressant, d'un point de vue académique, de tester de cette manière des méthodes d'apprentissage ou des architectures de réseaux :

- le nombre d'exemples peut être arbitrairement grand,
- l'amplitude et la nature du bruit sont parfaitement connues,
- l'ordre du processus simulé est connu.

On s'affranchit ainsi des incertitudes, inévitables lorsque l'on modélise un processus réel, relatives au nombre et au choix des exemples, ainsi qu'au choix du modèle-hypothèse.

Nous présenterons ensuite la modélisation d'un processus réel, qui a été étudié en détail par d'autres auteurs.

Nous nous intéresserons essentiellement à l'apprentissage de réseaux bouclés. L'algorithme utilisé sera donc semi-dirigé (paragraphe V.1.2 du chapitre III).

Pour tous les réseaux, on effectue 50 apprentissages correspondant chacun à une initialisation différente

- des pondérations et des termes directs dans le cas des réseaux d'ondelettes (les translations et les dilatations étant initialisées suivant la technique proposée dans le paragraphe IV.2 du chapitre III)
- de tous les coefficients dans le cas d'un réseau de fonctions dorsales.

Le résultat retenu est celui présentant l'erreur la plus petite sur l'ensemble d'estimation de la performance (EQMP).

Chaque fois que nous serons amenés à comparer l'efficacité de deux algorithmes d'apprentissage, nous initialiserons les réseaux de manière identique avant application de chaque algorithme.

## II. MODÉLISATION DE PROCESSUS SIMULÉS.

### II.1 Présentation du processus simulé sans bruit.

Le processus dont nous effectuons la modélisation a été proposé dans [Urbani95] pour la validation d'une procédure de sélection de modèles neuronaux (réseaux de neurones à fonctions dorsales). Il est simulé à partir de l'équation aux différences entrée-sortie suivante :

$$y_p(n) = f(y_p(n-1), y_p(n-2), u(n-1)) = \frac{24 + y_p(n-1)}{30} y_p(n-1) \pm 0.8 \frac{u(n-1)^2}{1 + u(n-1)^2} y_p(n-2) + 0.5u(n-1) \quad (1)$$

où  $y_p(n)$  et  $u(n)$  sont respectivement la sortie mesurée du processus et l'entrée de commande à l'instant  $n$ . Pour de faibles amplitudes de l'entrée de commande, comprises dans l'intervalle  $[-0.1, 0.1]$ , l'équation aux différences ci-dessus est proche de l'équation linéaire du premier ordre suivante :

$$y_p(n) = 0.8 y_p(n \pm 1) + 0.5 u(n \pm 1) \quad (2)$$

Le comportement est alors celui d'un filtre passe-bas du 1<sup>er</sup> ordre de gain statique égal à 2.5. Lorsque l'entrée de commande est de plus grande amplitude, le comportement est non linéaire. Pour une entrée variant dans l'intervalle  $[-10, 10]$  le processus reste stable.

On choisit de modéliser ce processus simulé à l'aide de réseaux de fonctions dorsales et d'ondelettes. Pour cela, on calcule une séquence d'apprentissage et une séquence d'estimation de la performance, comprenant chacune 1000 exemples. L'entrée est une séquence de créneaux d'amplitude aléatoire comprise dans l'intervalle  $[-5, +5]$  et de durées aléatoires variant de 1 à 20 périodes d'échantillonnage. La figure 1(a) montre les séquences de l'entrée de commande pour l'apprentissage (à gauche) et pour l'estimation de la performance (à droite), et la figure 1(b) les séquences correspondantes de la sortie, utilisées pour l'apprentissage (à gauche) et l'évaluation de la performance (à droite).

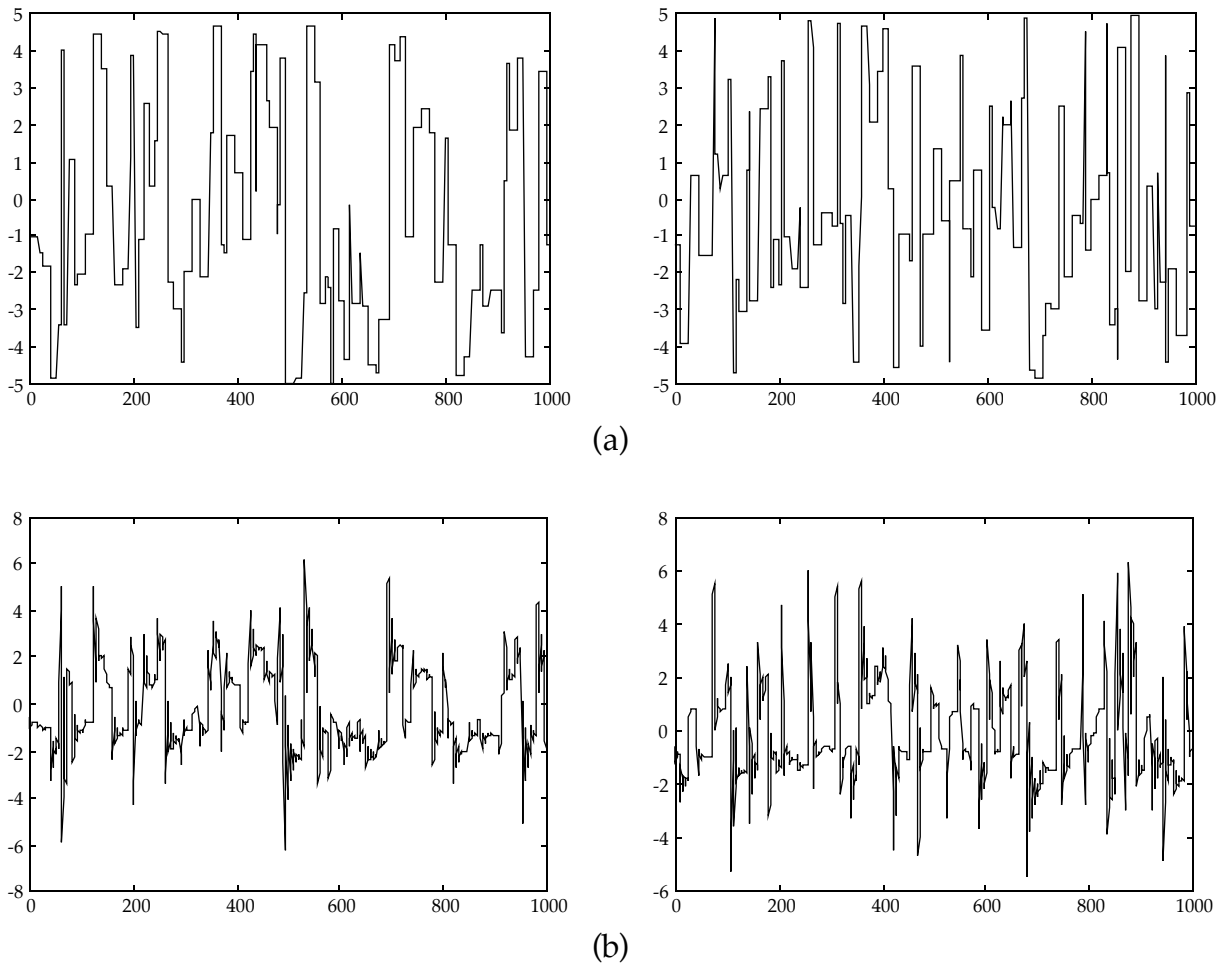


Figure 1. Séquence de l'entrée de commande (a) et séquence de la sortie calculée (b) du processus pour l'apprentissage.

## II.2 Modélisation du processus simulé non bruité.

Nous allons chercher tout d'abord à modéliser le processus simulé à partir de l'équation (1) sans introduire de bruit. En l'absence de bruit, les performances sont limitées par le nombre d'exemples (ainsi que leur distribution) et par l'efficacité de l'algorithme d'apprentissage.

Dans ce cas, comme nous l'avons indiqué dans le chapitre I, on peut indifféremment effectuer l'apprentissage d'un réseau non bouclé ou celui d'un réseau bouclé. Dans la mesure où nous sommes intéressés par la conception de modèles de simulation, nous avons choisi de réaliser l'apprentissage de réseaux bouclés. En effet, un modèle de simulation est destiné à prédire des *séquences* de valeurs des sorties, donc c'est nécessairement un réseau bouclé.

D'autre part, il nous est possible de choisir un *modèle entrée-sortie* ou un *modèle d'état*. Pour ce processus, nous considérerons uniquement sa modélisation par des prédicteurs de type entrée-sortie. Ce choix est motivé par le fait que le processus est simulé à partir d'une équation aux différences de type

entrée–sortie ; un exemple de modélisation d'état est présenté dans le paragraphe III.3 à l'occasion de la modélisation d'un processus réel.

L'ordre du processus simulé étant connu, ainsi que la mémoire sur l'entrée de commande, nous considérons pour ce processus l'apprentissage d'un prédicteur ayant l'expression suivante :

$$y(n) = \psi(y(n\pm 1), y(n\pm 2), u(n\pm 1), \theta) \quad (3)$$

La fonction  $\psi$  est à approcher par un réseau de fonctions dorsales ou d'ondelettes fondé sur la transformée continue (puisque le réseau est bouclé) et  $\theta$  est le vecteur des paramètres à ajuster.

### II.2.1 Réseau prédicteur à fonctions ondelettes.

Nous considérons des réseaux d'ondelettes bouclés fondés sur la transformée continue, présentés dans le chapitre III de ce mémoire. Nous effectuons l'apprentissage de plusieurs architectures ayant un nombre d'ondelettes croissant afin de rechercher celle qui permet d'obtenir la meilleure performance (EQMP). Les apprentissages de ces réseaux sont effectués avec l'algorithme de BFGS et celui de Levenberg–Marquardt.

#### II.2.1.1 Apprentissage avec l'algorithme de BFGS.

Le tableau 1 présente les résultats obtenus lors de l'apprentissage de réseaux d'ondelettes bouclés à l'aide de l'algorithme de BFGS. Rappelons que, pour chaque architecture, 50 apprentissages ont été effectués, avec, à chaque fois, une initialisation aléatoire différente des pondérations et des termes directs (coefficients de la partie affine). Les paramètres de translation et de dilatation sont initialisés suivant la procédure présentée au chapitre III. Le résultat indiqué (pour chacune des architectures) est le meilleur obtenu sur les 50 apprentissages.

Nombre d'ondelettes	Nombre de paramètres	EQMA	Meilleure EQMP sur 50 apprentissages
1	11	$7,6 \cdot 10^{-2}$	$1,5 \cdot 10^{-1}$
2	18	$2,0 \cdot 10^{-2}$	$3,6 \cdot 10^{-2}$
3	25	$2,2 \cdot 10^{-3}$	$6,7 \cdot 10^{-3}$
4	32	$2,8 \cdot 10^{-4}$	$1,3 \cdot 10^{-3}$
5	39	$5,2 \cdot 10^{-5}$	$2,9 \cdot 10^{-4}$
6	46	$3,8 \cdot 10^{-6}$	$2,9 \cdot 10^{-5}$

Tableau 1. Résultats de la modélisation du processus simulé sans bruit avec réseaux d'ondelettes ; apprentissage à l'aide de l'algorithme de BFGS.

Au delà de 6 ondelettes, les réseaux contenant plus de fonctions ne permettent pas d'obtenir de meilleures performances.

### *II.2.1.2 Apprentissage avec l'algorithme de Levenberg–Marquardt.*

Comme précédemment, nous nous intéressons à l'apprentissage de plusieurs architectures de réseaux d'ondelettes mais cette fois en utilisant l'algorithme de Levenberg–Marquardt pour l'ajustement des paramètres. Ainsi que nous l'avons indiqué au chapitre III, le calcul du gradient se fait dans le sens direct. L'initialisation des paramètres est la même que pour l'algorithme de BFGS.

Nombre d'ondelettes	Nombre de paramètres	EQMA	Meilleure EQMP sur 50 apprentissages
1	11	$7,6 \cdot 10^{-2}$	$1,5 \cdot 10^{-1}$
2	18	$2,1 \cdot 10^{-2}$	$4,3 \cdot 10^{-2}$
3	25	$2,5 \cdot 10^{-3}$	$5,8 \cdot 10^{-3}$
4	32	$9,7 \cdot 10^{-4}$	$2,4 \cdot 10^{-3}$
5	39	$5,0 \cdot 10^{-5}$	$2,5 \cdot 10^{-4}$
6	46	$3,4 \cdot 10^{-5}$	$1,3 \cdot 10^{-4}$
7	53	$2,2 \cdot 10^{-6}$	$2,1 \cdot 10^{-5}$

Tableau 2. Résultats de la modélisation du processus simulé sans bruit avec réseaux d'ondelettes ; apprentissage à l'aide de l'algorithme de Levenberg–Marquardt.

Ces résultats sont équivalents à ceux obtenus avec l'algorithme de BFGS et représentés sur le tableau 1. Notons que dans le cas de l'utilisation de l'algorithme de Levenberg–Marquardt, il est nécessaire d'utiliser un réseau de 7 ondelettes pour atteindre une performance de  $10^{-5}$ .

### **II.2.2 Réseau prédicteur à fonctions dorsales.**

Afin d'évaluer les performances des réseaux d'ondelettes fondés sur la transformée continue par rapport à celles que l'on peut obtenir avec une classe de réseaux possédant de bonnes propriétés de parcimonie, on se propose d'effectuer la modélisation de ce processus à l'aide de prédicteurs fondés sur des réseaux de fonctions dorsales. On choisit pour fonction d'activation la fonction sigmoïde (qui est la brique élémentaire des réseaux de neurones conventionnels) et l'on effectue l'apprentissage de réseaux bouclés dont la partie statique est constituée d'une couche de fonctions dorsales et d'un neurone de sortie linéaire (figures 4 et 7 du chapitre II).

### II.2.2.1 Apprentissage avec l'algorithme de BFGS.

Comme précédemment, nous effectuons l'apprentissage de plusieurs architectures en augmentant à chaque fois le nombre de neurones. Le tableau 3 illustre les meilleurs résultats obtenus pour chacune des architectures.

Nombre de sigmoïdes	Nombre de paramètres.	EQMA	Meilleure EQMP sur 50 apprentissages
1	9	$1,1 \cdot 10^{-1}$	$1,8 \cdot 10^{-1}$
2	14	$7,1 \cdot 10^{-2}$	$1,0 \cdot 10^{-1}$
3	19	$1,1 \cdot 10^{-3}$	$8,4 \cdot 10^{-3}$
4	24	$3,9 \cdot 10^{-4}$	$2,3 \cdot 10^{-3}$
5	29	$4,5 \cdot 10^{-6}$	$1,8 \cdot 10^{-5}$
6	34	$4,2 \cdot 10^{-6}$	$1,6 \cdot 10^{-5}$

Tableau 3. Résultats de la modélisation du processus simulé sans bruit avec réseaux de sigmoïdes et algorithme de BFGS.

Là encore, une augmentation du nombre de fonctions n'améliore pas la performance, au-delà de 6 neurones cachés.

Les résultats, concernant l'EQMP, portés dans les tableaux 1 et 3, sont représentés graphiquement sur la figure 2 qui illustre l'évolution de l'EQMP en fonction du nombre de fonctions pour les deux types de réseaux. On ne constate pas de différence significative entre les deux types de réseaux.

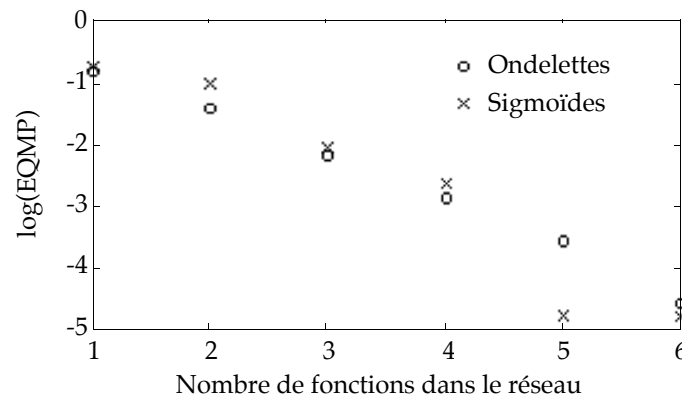


Figure 2. Évolution de la performance en fonction de l'architecture du réseau (BFGS).

### II.2.2.2 Apprentissage avec l'algorithme de Levenberg–Marquardt.

On effectue également l'apprentissage de ces réseaux bouclés de sigmoïdes à l'aide de l'algorithme de Levenberg–Marquardt. Les remarques faites dans le

chapitre III concernant la nécessité de faire le calcul du gradient de la fonction de coût dans le sens direct s'appliquent également aux réseaux de fonctions dorsales.

Le tableau 4 illustre les meilleurs résultats obtenus pour chacune des architectures.

Nombre de sigmoïdes.	Nombre de paramètres.	EQMA	Meilleure EQMP sur 50 apprentissages
1	9	$1,1 \cdot 10^{-1}$	$1,8 \cdot 10^{-1}$
2	14	$7,2 \cdot 10^{-2}$	$1,0 \cdot 10^{-1}$
3	19	$1,8 \cdot 10^{-3}$	$6,0 \cdot 10^{-3}$
4	24	$5,7 \cdot 10^{-4}$	$2,0 \cdot 10^{-3}$
5	29	$6,4 \cdot 10^{-6}$	$1,7 \cdot 10^{-5}$
6	34	$3,5 \cdot 10^{-6}$	$1,5 \cdot 10^{-5}$
7	39	$1,8 \cdot 10^{-6}$	$8,5 \cdot 10^{-6}$

Tableau 4. Résultats de la modélisation du processus simulé sans bruit avec réseaux de sigmoïdes ; apprentissage avec l'algorithme de Levenberg–Marquardt.

Les résultats, concernant l'EQMP, portés dans les tableaux 2 et 4, sont représentés graphiquement sur la figure 3 qui illustre l'évolution de l'EQMP en fonction du nombre de fonctions pour les deux types de réseaux. Comme précédemment, on ne constate pas de différence significative entre les deux types de réseaux.

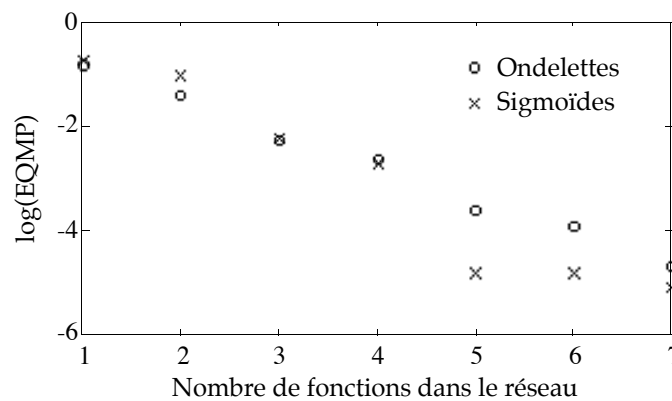


Figure 3. Évolution de la performance en fonction de l'architecture du réseau (Levenberg-Marquardt).

### II.3 Modélisation du processus simulé avec bruit.

Nous nous proposons à présent de modéliser un processus bruité. Étant donné que l'on simule le processus, on a la possibilité de choisir la manière avec laquelle agit le bruit. Nous simulerons l'existence d'un bruit additif de sortie, puis celle d'un bruit additif d'état ; dans chacun des cas, nous ferons le choix du

modèle-hypothèse qui correspond au type de bruit qui est effectivement mis en œuvre dans les simulations, car nous nous intéressons, dans ce travail, à l'influence du choix des fonctions utilisées pour construire le modèle ; l'influence du choix du modèle-hypothèse (et notamment l'effet d'un choix erroné) a été étudiée dans [Nerrand94].

### II.3.1 Modélisation du processus simulé avec bruit additif de sortie.

Lorsque l'on fait l'hypothèse d'un bruit additif en sortie (Output Error), le prédicteur optimal associé est bouclé. On considère un prédicteur d'ordre 2, dont la mémoire sur l'entrée de commande est de 1 période d'échantillonnage. Ce prédicteur peut être approché par un réseau de fonctions réalisant le modèle suivant:

$$y(n) = \psi(y(n\pm 1), y(n\pm 2), u(n\pm 1), \theta) \quad (4)$$

Si la séquence d'apprentissage est suffisamment riche et représentative du comportement processus, si l'algorithme d'apprentissage est efficace, et si la taille du réseau est suffisante pour approcher la partie déterministe du processus avec une bonne précision, alors les EQMA et EQMP obtenues doivent être égales à la variance du bruit (c'est à dire à la partie non prédictible du comportement du processus).

Nous simulons le processus avec un bruit pseudo-blanc additif en sortie de distribution uniforme et de variance  $10^{-2}$ . Les résultats de la modélisation sans bruit montrent qu'un réseau constitué de cinq fonctions permet d'effectuer une bonne approximation de la partie déterministe du processus (performance inférieure à la variance du bruit).

Nous avons donc effectué l'apprentissage de réseaux constitués de cinq fonctions (sigmoïdes ou ondelettes). Le tableau 5 illustre les meilleurs résultats obtenus sur 20 apprentissages.

	EQMA	EQMP
Réseaux d'ondelettes	$1,02 \cdot 10^{-2}$	$1,05 \cdot 10^{-2}$
Réseaux de sigmoïdes	$1,03 \cdot 10^{-2}$	$1,07 \cdot 10^{-2}$

Tableau 5. Résultats de la modélisation avec bruit de sortie.

Les deux types de réseaux permettent d'obtenir de façon quasiment identique une fonction de coût en fin d'apprentissage, et une performance, très proches de la variance du bruit.

### II.3.2 Modélisation du processus simulé avec bruit d'état additif.

Nous considérons à présent la modélisation du processus simulé avec un bruit d'état additif (Equation Error) de variance  $10^{-2}$ . Le prédicteur optimal associé est non bouclé. On effectue donc l'apprentissage de réseaux réalisant le modèle suivant :

$$y(n) = \psi(y_p(n\pm 1), y_p(n\pm 2), u(n\pm 1), \theta) \quad (5)$$

On utilise des architectures de réseaux constitués de 5 fonctions (sigmoïdes ou ondelettes) et pour chaque architecture on retient la meilleure performance obtenue sur 20 apprentissages. Les résultats sont illustrés sur le tableau 6.

	EQMA	EQMP
Réseaux d'ondelettes	$9,68 \cdot 10^{-3}$	$1,00 \cdot 10^{-2}$
Réseaux de sigmoïdes	$9,67 \cdot 10^{-3}$	$1,05 \cdot 10^{-2}$

Tableau 6. Résultats de la modélisation avec bruit d'état.

Comme dans le cas d'un bruit de sortie additif, réseaux d'ondelettes et de fonctions dorsales aboutissent à des prédicteurs optimaux avec des précisions quasiment identiques.

### II.4 Conclusion.

Dans la première partie de ce chapitre, nous avons effectué une modélisation de type entrée-sortie d'un processus simulé mono-entrée-mono-sortie d'ordre 2. Nous avons utilisé, dans des conditions identiques, des réseaux d'ondelettes fondés sur la transformée continue et des réseaux de fonctions dorsales sigmoïdes. Lors de la modélisation sans bruit puis avec brut, les deux types de réseaux montrent des performances équivalentes, que les apprentissages soient effectués à l'aide de l'algorithme de BFGS ou à l'aide de l'algorithme de Levenberg-Marquardt.

## III. MODÉLISATION D'UN PROCESSUS RÉEL.

Les simulations précédentes ont permis d'étudier les performances de réseaux de neurones et de réseaux d'ondelettes en fonction du nombre de fonctions présentes dans ces réseaux, toutes choses égales et connues par ailleurs (ordre du modèle, mémoire sur l'entrée de commande, variance du bruit). Dans la plupart des applications réelles, on ignore :

- l'ordre nécessaire pour le modèle (c'est-à-dire la valeur du paramètre  $N_S$  défini au chapitre I),
- la mémoire sur l'entrée de commande (c'est-à-dire la valeur du paramètre  $N_e$  défini au chapitre I)

- la nature et la variance du bruit.

Lorsque l'on cherche à modéliser un processus réel, il est donc nécessaire, d'une part, d'essayer plusieurs modèles-hypothèses et de retenir celui qui semble le mieux adapté, et, d'autre part, de recourir à des techniques de sélection de modèles [Urbani95] pour trouver des valeurs satisfaisantes de  $N_s$  et  $N_w$ .

### III.1 Présentation du processus.

Le processus dont nous nous proposons de faire la modélisation dans ce qui suit est l'actionneur hydraulique d'un bras de robot articulé. La sortie d'intérêt  $y_p(n)$  est la pression d'huile de l'actionneur qui détermine la position du bras. L'entrée de commande qui agit sur la pression est l'ouverture d'une vanne  $u(n)$ . Les données relatives à ce processus ont été fournies par l'Université de Linköping ; ce processus a fait l'objet de modélisations "boîtes noires" de la part de plusieurs équipes [Sjöberg95].

Le processus est connu par une séquence de 1024 couples d'entrées et de sorties  $\{u(k), y_p(k)\}$  mesurées. La première moitié de ces données (c'est à dire 512 points) est utilisée comme séquence d'apprentissage et la seconde moitié comme séquence pour l'estimation de la performance.

La figure 4 illustre la séquence de l'entrée de commande (a) et celle de la sortie mesurée (b) dont nous disposons.

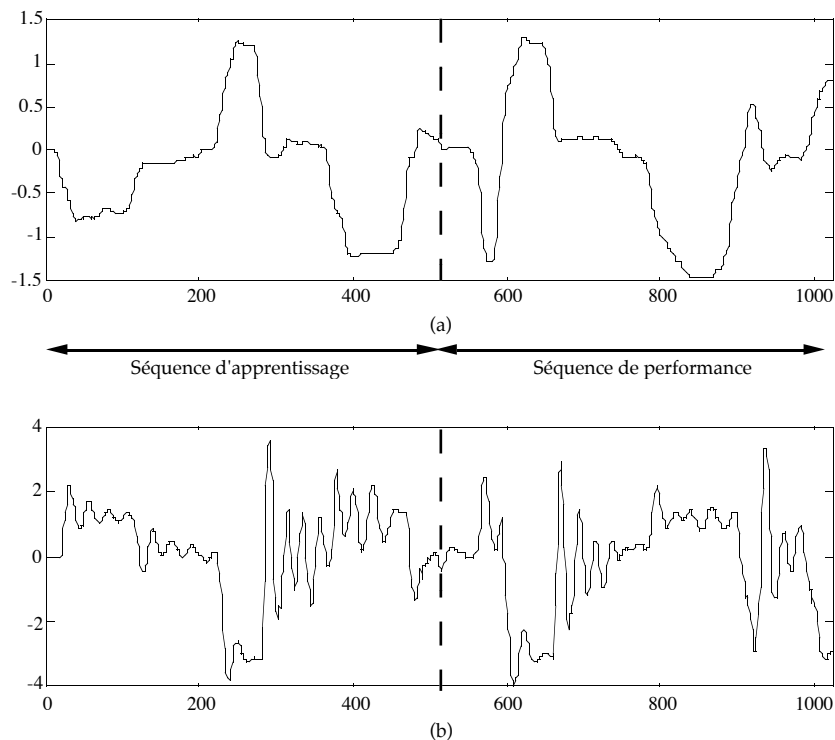


Figure 4. Séquences de l'entrée de commande (a) et de la sortie (b).

On se propose dans la suite d'effectuer une modélisation de ce processus avec des prédicteurs entrée–sortie et d'état fondés sur des réseaux d'ondelettes et de neurones à fonctions sigmoïdes.

### III.2 Modélisation entrée–sortie.

Tout d'abord, nous nous proposons d'effectuer une modélisation entrée–sortie du processus. Comme toute modélisation, la première étape consiste à choisir un modèle-hypothèse. Le prédicteur optimal pour l'apprentissage associé à une hypothèse "bruit d'état" est non bouclé. Une étude antérieure de ce même processus[Rivals95b] a montré que les modèles construits à partir de l'hypothèse "bruit d'état" ont de mauvaises performances. Nous avons donc opté pour l'apprentissage de prédicteurs bouclés.

La deuxième question à résoudre concerne le choix de l'ordre du modèle et de la mémoire sur l'entrée de commande. Nous avons adopté une démarche qui consiste à considérer d'abord le modèle le plus simple, puis à le rendre plus complexe et à retenir celui qui présente la meilleure performance. Dans ce cas, nous partons d'un prédicteur avec  $N_S = 2$  (le caractère oscillatoire de la réponse suggère que le modèle est au moins du second ordre) et  $N_e = 1$ .

#### III.2.1 Réseau prédicteur à fonctions ondelettes.

Nous commençons par présenter les résultats obtenus pour des prédicteurs fondés sur des réseaux d'ondelettes bouclés, comme ceux présentés dans le chapitre III. Comme pour les processus simulés, nous effectuons l'apprentissage de plusieurs réseaux en augmentant le nombre d'ondelettes.

##### III.2.1.1 Apprentissage avec l'algorithme de BFGS.

Le tableau 7 présente les résultats obtenus pour des apprentissages utilisant l'algorithme de BFGS. Comme pour le processus simulé, on effectue 50 apprentissages pour chacune des architectures avec à chaque fois une initialisation différente ; les résultats présentés correspondent aux apprentissages présentant les meilleures EQMP.

Nombre d'ondelettes.	Nombre de paramètres.	EQMA	Meilleure EQMP sur 50 apprentissages
1	11	0,25	0,30
2	18	0,11	0,13
3	25	0,13	0,15

Tableau 7. Résultats de la modélisation du processus réel avec réseaux d'ondelettes ; apprentissage à l'aide de l'algorithme de BFGS.

Pour plus de 3 ondelettes, la performance du modèle se dégrade. Le réseau présentant la meilleure performance est donc celui qui est constitué de deux ondelettes. C'est d'ailleurs, à notre connaissance, le meilleur résultat publié, relatif à un prédicteur entrée-sortie de ce processus [Rivals95b, Pucar95, Sjöberg95].

### III.2.1.2 Apprentissage avec l'algorithme de Levenberg–Marquardt.

Dans les mêmes conditions que précédemment (mêmes séquences, mêmes architectures de réseaux, mêmes initialisations), on effectue des apprentissages de ces réseaux d'ondelettes en utilisant l'algorithme de Levenberg–Marquardt. Pour chaque réseau l'apprentissage présentant la meilleure EQMP sur les 50 est présenté sur le tableau 8.

Nombre d'ondelettes	Nombre de paramètres	EQMA	Meilleure EQMP sur 50 apprentissages
1	11	0,22	0,39
2	18	0,094	0,19
3	25	0,084	0,20
4	32	0,046	0,24

Tableau 8. Résultats de la modélisation du processus réel avec réseaux d'ondelettes ; apprentissage à l'aide de l'algorithme de Levenberg–Marquardt.

Comme avec l'algorithme de BFGS, le réseau de 2 ondelettes présente la meilleure performance.

La figure suivante, qui montre l'évolution de l'EQMA et de l'EQMP en fonction du nombre d'ondelettes, met en évidence que pour une architecture comprenant 3 ondelettes ou plus, on observe un phénomène de surajustement.

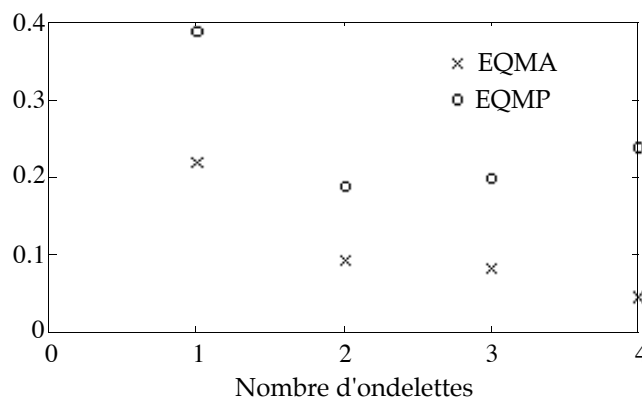


Figure 5. Évolution de la meilleure EQMP sur 50 apprentissages et de l'EQMA correspondante (apprentissage avec l'algorithme de Levenberg–Marquardt).

### III.2.1.3 Fréquence d'occurrence du meilleur résultat.

Les résultats présentés dans les tableaux ci-dessus sont, pour chaque réseau, le meilleur apprentissage obtenu sur 50. La fréquence d'occurrence du meilleur résultat, parmi tous les essais effectués, peut constituer un élément de choix entre plusieurs algorithmes. Pour le réseau de 2 ondelettes, nous avons représenté, sur la figure 6, les histogrammes d'apparition des différentes valeurs de la fonction de coût en fin d'apprentissage (a) et de l'estimation de la performance (b) obtenues parmi les 50 apprentissages en utilisant l'algorithme de BFGS. La figure 7 illustre ces deux histogrammes dans le cas de l'utilisation de l'algorithme de Levenberg-Marquardt.

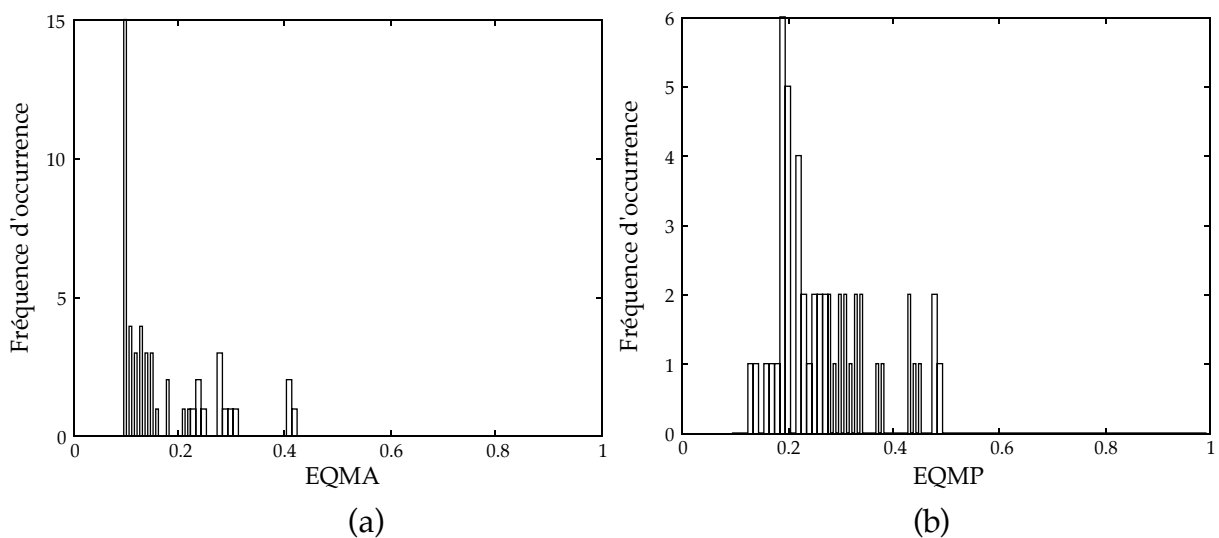


Figure 6. Histogrammes des EQMA (a) et des EQMP (b) de réseaux d'ondelettes avec apprentissage à l'aide de l'algorithme de BFGS.

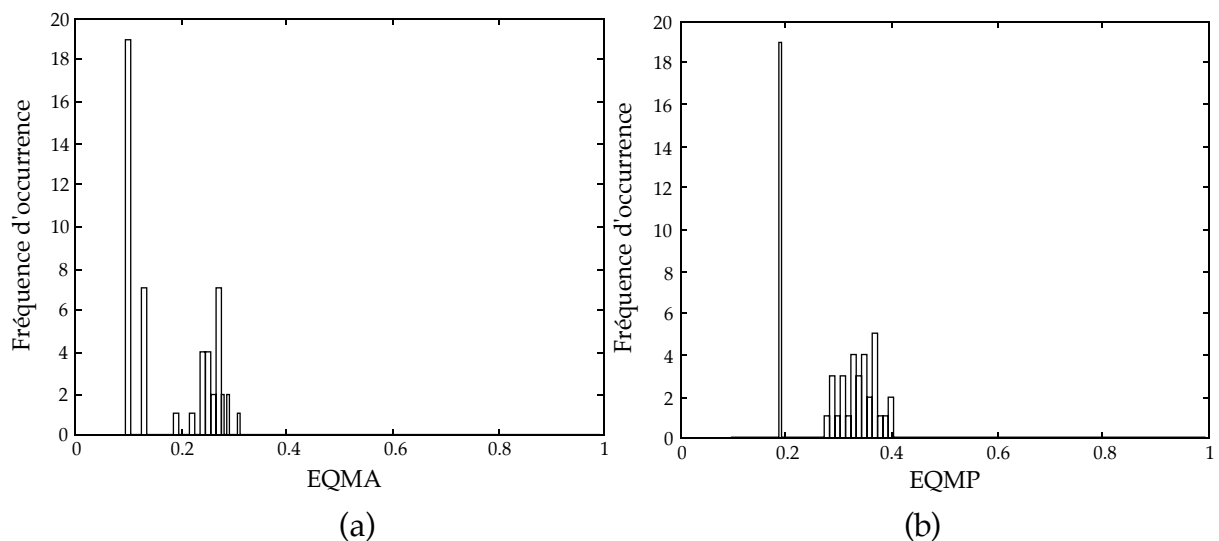


Figure 7. Histogrammes des EQMA (a) et des EQMP (b) de réseaux d'ondelettes avec apprentissage à l'aide de l'algorithme de Levenberg-Marquardt.

On observe que, du point de vue considéré ici, l'algorithme de Levenberg Marquardt possède, pour cet exemple qui met en jeu un petit nombre d'ondelettes, une efficacité supérieure à celle de la méthode de BFGS : une des valeurs de l'EQMP (= 0.19) ainsi que l'EQMA (= 0.094) qui lui correspond sont obtenues avec une plus grande fréquence que les autres, et il s'agit précisément du meilleur apprentissage retenu dans le tableau 8 pour le réseau à 2 ondelettes. D'autre part, les coefficients de tous les réseaux présentant cette performance sont identiques, ce qui prouve qu'il s'agit bien du même minimum de la fonction de coût. Cette comparaison est également effectuée, pour des réseaux à fonctions dorsales sigmoïdes, dans le paragraphe III.2.2.3.

### III.2.2 Réseau prédicteur à fonctions dorsales.

Nous nous intéressons également à la modélisation de ce processus à l'aide de réseaux de fonctions sigmoïdes. D'autres travaux [Rivals95b] ont abordé cette modélisation. Nous la reprenons ici à l'aide de nos outils (notamment avec l'algorithme de Levenberg–Marquardt nécessitant le calcul du gradient dans le sens direct).

#### III.2.2.1 Apprentissage avec l'algorithme de BFGS.

Dans les mêmes conditions que les réseaux d'ondelettes (mêmes séquences d'apprentissage et d'évaluation de la performance, 50 apprentissages par architecture, et utilisation de l'algorithme de BFGS), nous obtenons les résultats présentés dans le tableau 9.

Nombre de sigmoïdes	Nombre de paramètres	EQMA	EQMP
1	9	0,20	0,3
2	14	0,13	0,17
3	19	0,15	0,14
4	24	0,085	0,16

Tableau 9. Résultats de la modélisation du processus réel avec réseaux de sigmoïdes ; apprentissage à l'aide de l'algorithme de BFGS.

La meilleure performance est donc obtenue avec un réseau de 3 neurones à fonctions sigmoïdes. Augmenter le nombre de neurones améliore l'EQMA mais pas la performance.

### III.2.2.2 Apprentissage avec l'algorithme de Levenberg–Marquardt.

Le tableau suivant illustre les meilleures performances obtenues avec des réseaux de fonctions sigmoïdes et des apprentissages utilisant l'algorithme de Levenberg–Marquardt.

Nombre de sigmoïdes	Nombre de paramètres	EQMA	EQMP
1	9	0,23	0,38
2	14	0,11	0,20
3	19	0,092	0,15
4	24	0,086	0,15

Tableau 10. Résultats de la modélisation du processus réel avec réseaux de sigmoïdes ; apprentissage à l'aide de l'algorithme de Levenberg–Marquardt.

Les architectures à 3 et 4 neurones cachés réalisent une performance égale. En retenant la plus parcimonieuse (réseau à trois neurones), on retrouve donc la même que dans le cas d'un apprentissage avec l'algorithme de BFGS. Les deux algorithmes permettent donc d'aboutir à des modèles de précision équivalente.

### III.2.2.3 Fréquence d'occurrence du meilleur résultat.

Dans le paragraphe III.2.2.3 de ce chapitre, nous avons comparé "l'efficacité" des deux algorithmes du point de vue de la fréquence d'occurrence de la meilleure solution (que l'on retient) et ceci dans le cas de modèles fondés sur des réseaux d'ondelettes. Nous effectuons à nouveau cette comparaison, avec cette fois les résultats obtenus sur les modèles fondés sur des réseaux de fonctions sigmoïdes.

La figure 8 illustre les histogrammes du nombre d'apparitions des différents critères d'apprentissages (a) et des performances (b) obtenus dans le cas de l'algorithme de BFGS. La figure 9 illustre ces deux histogrammes dans le cas de l'utilisation de l'algorithme de Levenberg–Marquardt.

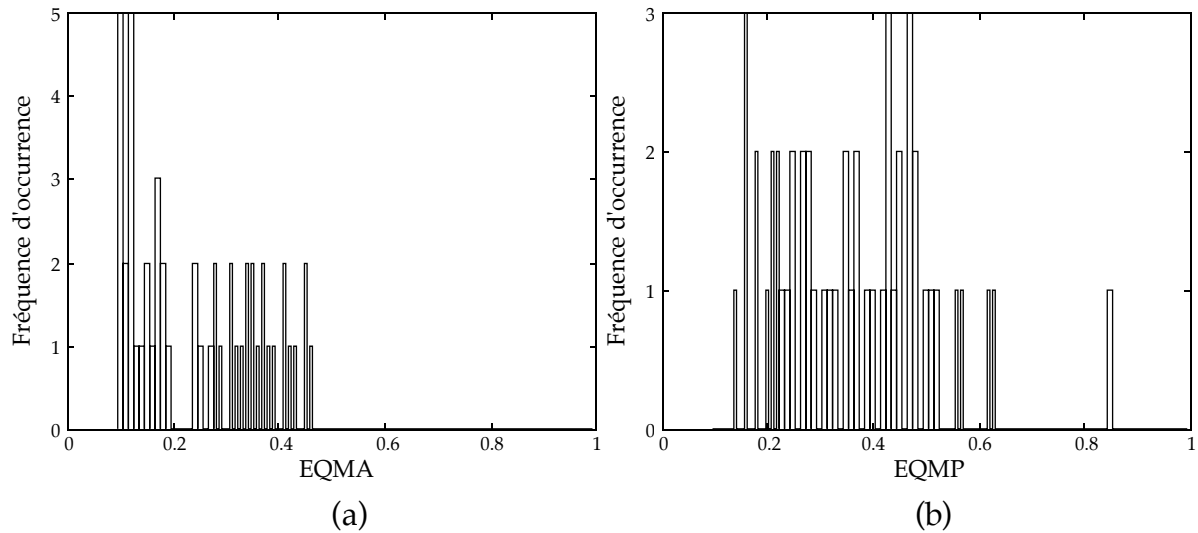


Figure 8. Histogrammes des EQMA (a) et des EQMP (b) de réseaux de fonctions dorsales avec apprentissage à l'aide de l'algorithme de BFGS.

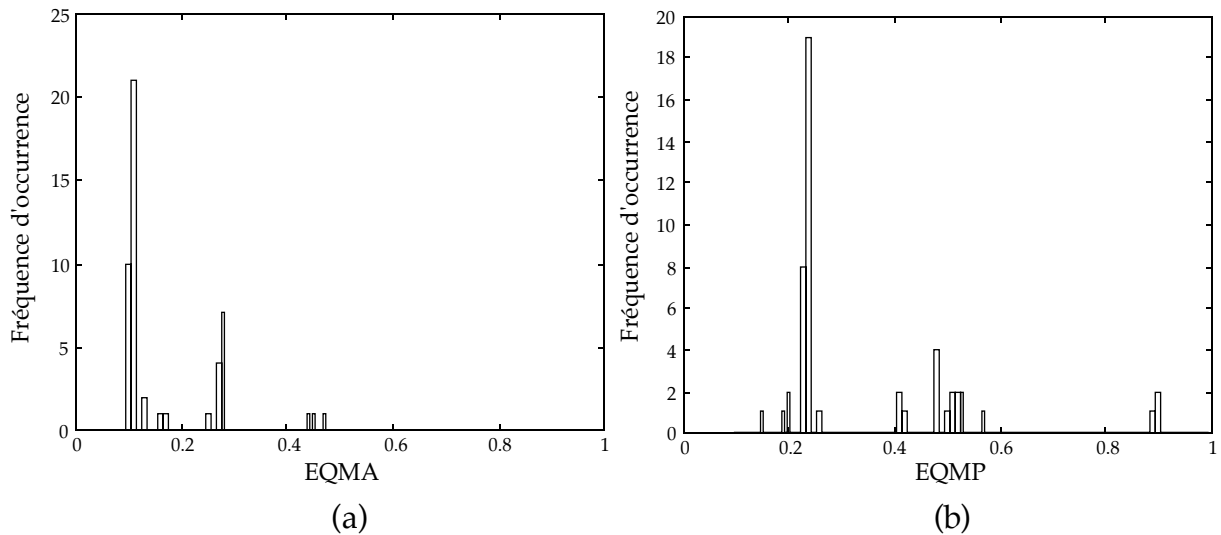


Figure 9. Histogrammes des EQMA (a) et des EQMP (b) de réseaux de fonctions dorsales avec apprentissage à l'aide de l'algorithme de Levenberg–Marquardt.

On fait ici une constatation semblable à celle que nous avons faite pour les réseaux d'ondelettes : une des valeurs de l'EQMP ( $= 0.24$ ) ainsi que l'EQMA correspondante ( $= 0.11$ ) sont obtenues beaucoup plus fréquemment que les autres lorsque l'on utilise l'algorithme de Levenberg-Marquardt. Néanmoins, ce n'est pas, cette fois, la valeur minimale de l'EQMP. D'autre part, une comparaison des coefficients des réseaux possédant cette performance montre qu'ils sont dans la plupart des cas identiques.

### III.2.3 Conclusion de la modélisation entrée–sortie.

Nous avons effectué une modélisation entrée-sortie d'un processus réel à l'aide de réseaux de neurones à sigmoïdes et d'ondelettes. Nous avons opté pour

l'apprentissage de prédicteurs bouclés afin d'obtenir des modèles de simulation du processus. Cela revient à considérer un modèle hypothèse de type bruit de sortie (Output Error).

Du point de vue des performances obtenues, les deux types de réseaux utilisés dans les mêmes conditions aboutissent généralement à des modèles de précision très proche.

Une comparaison des deux algorithmes que nous utilisons sur l'exemple de la modélisation du processus réel, sur la base de la fréquence d'occurrence de la meilleure solution trouvée a montré que l'algorithme de Levenberg–Marquardt possède un avantage sur la méthode de BFGS. Cette tendance a été observée de façon similaire sur les réseaux d'ondelettes et sur les réseaux de fonctions sigmoïdes. Il n'est évidemment pas possible de généraliser ce résultat à partir de ce seul exemple, qui a pour caractéristique de porter sur un réseau comprenant un très petit nombre d'ondelettes, mais il pourrait être intéressant de mener une comparaison plus systématique des deux algorithmes sous cet angle.

### III.3 Modélisation d'état.

Nous nous proposons à présent de modéliser ce processus avec une représentation d'état à variables d'état libres (non mesurées). Nous conservons un ordre 2 ( $N_s = 2$ ) et une mémoire de 1 sur l'entrée de commande ( $N_e = 1$ ).

L'état n'étant pas mesuré, le prédicteur ne peut être que bouclé. Ce prédicteur est optimal dans le cas où le processus est non bruité, ou si l'on est en présence d'un bruit additif en sortie. Le prédicteur ainsi construit n'est pas un estimateur de l'état : son seul rôle est de modéliser le comportement entrée–sortie du processus.

Nous effectuons l'apprentissage de réseaux réalisant les fonctions suivantes :

$$\begin{cases} x_1(k+1) = \psi_{11}(x_1(k), x_2(k), u(k)) \\ x_2(k+1) = \psi_{12}(x_1(k), x_2(k), u(k)) \\ y(k+1) = \psi_2(x_1(k), x_2(k), u(k)) \end{cases} \quad (6)$$

Les fonctions  $\psi_{11}$ ,  $\psi_{12}$  et  $\psi_2$  sont des fonctions réalisées à l'aide d'un réseau d'ondelettes (figure 13 du chapitre III) ou d'un réseau de fonctions dorsales (figure 8 du chapitre II).

Les apprentissages ont été effectués à l'aide de l'algorithme de BFGS. En effet, nous avons vu dans le chapitre III que le calcul du gradient dans le sens direct, nécessaire pour la mise en œuvre de l'algorithme de Levenberg–Marquardt, est coûteux, d'un point de vue numérique, pour des réseaux d'état.

### III.3.1 Réseau prédicteur d'état à fonctions d'ondelettes.

Nous effectuons l'apprentissage de plusieurs architectures de réseaux d'ondelettes, dans les mêmes conditions que précédemment.

L'initialisation des réseaux d'ondelettes dans le cas de l'apprentissage d'un prédicteur d'état bouclé présente une difficulté particulière : en effet, les états n'étant pas connus a priori, le domaine des entrées d'état n'est pas connu. Or l'application de la procédure d'initialisation proposée pour les réseaux d'ondelettes fondés sur la transformée continue (chapitre III) nécessite la connaissance de ces domaines.

On peut néanmoins remarquer que, en début d'apprentissage, les valeurs des variables d'état en sortie du réseau sont au maximum égales aux pondérations, lesquelles sont uniformément distribuées dans l'intervalle  $[-10^{-2}, 10^{-2}]$ . Sous réserve que la sortie soit centrée en zéro, la procédure d'initialisation valable pour les réseaux de type entrée-sortie reste donc applicable pour des réseaux d'état.

Le tableau 11 illustre les résultats obtenus pour des réseaux constitués de 1 et 2 ondelettes.

Nombre d'ondelettes.	Nombre de paramètres.	EQMA	Meilleure EQMP sur 50 apprentissages
1	19	0,38	0,42
2	28	0,091	0,15

Tableau 11. Résultats de la modélisation du processus réel avec réseaux d'état de fonctions ondelettes.

La meilleure performance est donc obtenue avec un réseau de 2 ondelettes (comme pour le modèle de type entrée-sortie). Cette performance est très proche de celle obtenue avec un modèle entrée-sortie (voir tableau 7) mais nous ne sommes pas arrivés à améliorer cette dernière bien que les réseaux d'état constituent une représentation plus générale que les réseaux entrée-sortie. Ce phénomène peut s'expliquer par une taille insuffisante de la séquence d'apprentissage. En effet, dans les résultats obtenus en modélisation entrée-sortie, on remarque que les performances se dégradent souvent pour des réseaux de plus de 25 coefficients ; d'autre part, il faut au moins deux fonctions non linéaires pour modéliser correctement ce processus. Étant donné les réseaux d'état d'ondelettes que nous utilisons, ces deux conditions (un réseau de moins de 25 coefficients et constitué d'au moins deux fonctions) ne peuvent être remplies simultanément.

### III.3.2 Réseau prédicteur d'état à fonctions dorsales.

Nous considérons à présent des modèles d'état constitués de réseaux de neurones à une couche de fonctions sigmoïdes et un neurone de sortie linéaire. On peut faire la même remarque concernant le nombre de paramètres. Un réseau d'état à deux fonctions sigmoïdes contient 26 coefficients ajustables. Dans le cas d'un réseau entrée–sortie, ce nombre de paramètres aboutit généralement à un surajustement comme le montrent les résultats présentés plus haut.

Le tableau 12 donne le meilleur résultat obtenu pour des architectures constitués de 1 à 3 fonctions sigmoïdes.

Nombre de sigmoïdes	Nombre de paramètres.	EQMA	Meilleure EQMP sur 50 apprentissages
1	19	0,24	0,34
2	26	0,091	0,18
3	33	0,058	0,18

Tableau 12. Résultats de la modélisation du processus réel avec réseaux d'état de fonctions sigmoïdes.

A partir d'un réseau de 2 neurones, on obtient un critère d'apprentissage meilleur que celui réalisé par un réseau d'entrée–sortie constitué par le même nombre de fonctions. En revanche, la performance n'est pas améliorée. Ceci tend à confirmer l'hypothèse concernant l'existence d'un phénomène de surajustement pour des réseaux de plus de 24 coefficients.

### III.3.3 Réseau prédicteur d'état à fonctions dorsales dont la sortie est l'un des états.

On se propose d'utiliser pour la modélisation du processus réel des modèles d'état particuliers dont la sortie est considérée comme un état. Ce type de réseau a été introduit à la fin du chapitre III et le calcul du gradient de la fonction de coût dans le cas de réseaux d'ondelettes est présenté en annexe de ce mémoire.

Dans la suite, nous présentons les résultats obtenus en utilisant de tels réseaux fondés sur des neurones à fonctions dorsales sigmoïdes. Ils réalisent des modèles de la forme :

$$\begin{cases} y(k+1)=x_1(k+1) = \psi_1(x_1(k), x_2(k), u(k)) \\ x_2(k+1) = \psi_2(x_1(k), x_2(k), u(k)) \end{cases} \quad (7)$$

Le tableau 13 illustre les meilleurs résultats sur 50 apprentissages effectués pour chaque réseau de 1 à 3 neurones cachés. L'utilisation des réseaux contenant plus de neurones n'améliore pas la performance.

Nombre de sigmoïdes	Nombre de paramètres.	EQMA	Meilleure EQMP sur 50 apprentissages
1	14	0,2	0,28
2	20	0,12	0,15
3	26	0,071	0,117

Tableau 13. Résultats de la modélisation du processus réel avec réseaux d'état de fonctions sigmoïdes dont la sortie est un état.

Le réseau à 3 neurones présente la meilleure performance que nous ayons obtenue pour la modélisation de ce processus réel, pour tous types de réseaux et de modèles. D'autre part, ce réseau d'état à 3 fonctions sigmoïdes contient 26 paramètres ajustables soit moins qu'un réseau à deux états libres ayant le même nombre de neurones cachés. La figure suivante montre l'architecture de ce réseau :

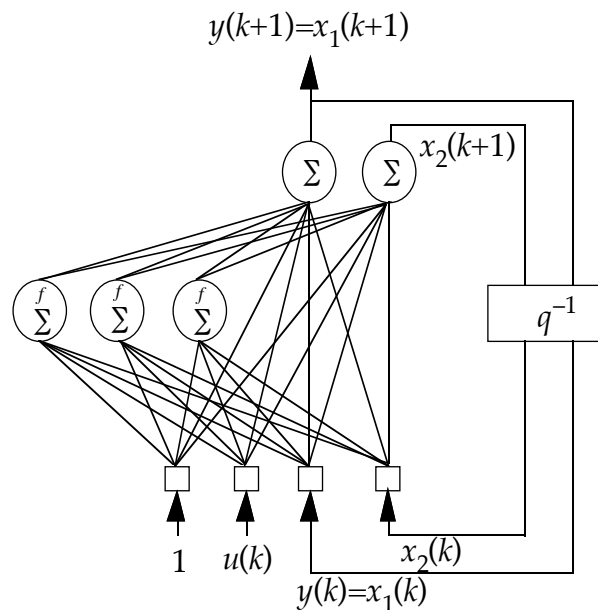


Figure 10. Réseau d'état à 3 fonctions sigmoïdes dont la sortie est un état.

### III.3.4 Conclusion de la modélisation d'état.

Du point de vue de la représentation mathématique, les réseaux d'état constituent des modèles plus généraux que ceux de type entrée–sortie.

Nous avons modélisé un processus réel à l'aide de prédicteurs d'état fondés sur des réseaux d'ondelettes et de fonctions dorsales sigmoïdes. Les résultats obtenus montrent que, pour un même nombre de fonctions, un apprentissage avec un réseau d'état présente une meilleure précision que celui avec un réseau entrée–sortie. En revanche, les performances ne sont pas meilleures.

Sur la base des observations effectuées sur les résultats de la modélisation entrée–sortie, il est très probable que ceci est dû à un phénomène de surajustement. En effet, pour un même nombre de fonctions, les réseaux d'état présentent plus de paramètres ajustables que les réseaux entrée–sortie. Les résultats obtenus dans le paragraphe précédent avec un réseau d'état ayant moins de paramètres à nombre de neurones égal (ceci est possible en choisissant la sortie comme un état) sont cohérents avec cette hypothèse.

#### **IV. CONCLUSION.**

Nous avons étudié la modélisation de deux processus à l'aide de réseaux d'ondelettes (fondés sur la transformée continue) et de réseaux de neurones bouclés. Les résultats obtenus montrent que les deux types de réseaux, pour un même nombre de fonctions, permettent d'obtenir des modèles de performances très souvent équivalentes.

D'autre part, nous avons effectué une comparaison des deux algorithmes d'apprentissage utilisés dans ce mémoire. Cette comparaison est fondée sur la fréquence d'occurrence d'une solution parmi plusieurs apprentissages effectués pour une même architecture de réseau. Il apparaît que, pour les exemples étudiés dans ce mémoire, l'algorithme de Levenberg-Marquardt possède une meilleure robustesse vis-à-vis de l'initialisation aléatoire des paramètres des réseaux.

La modélisation à l'aide d'un réseau d'état dont la sortie est un état a permis d'améliorer les performances obtenues pour le processus réel avec des réseaux entrée–sortie et d'état. Le résultat obtenu avec un réseau de trois sigmoïdes est égal à celui réalisé dans [Rivals95b] avec un réseau de neurones d'état complètement connecté à deux sigmoïdes.