

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITÉ MOULOUD MAMMERRI DE TIZI-OUZOU



FACULTÉ DE GÉNIE ÉLECTRIQUE ET INFORMATIQUE

DEPARTEMENT D'AUTOMATIQUE

MÉMOIRE DE MAGISTER

OPTION : OPTIMISATION ET COMMANDE DES SYSTÈMES

Thème

Identification des systèmes non-linéaires à structure Wiener, Hammerstein, application au cas fractionnaire

Présenté par :

Karima HAMMAR

Ingénieur UMMTO

Soutenu le 15/11/2015 devant le jury composé de :

Président	HAMMOUCHE Kamal	Professeur	à l'U.M.M.T.0
Rapporteur	DJAMAH Tounsia	M.C.A.	à l'U.M.M.T.0
Examineur	GUERMAH Saïd	M.C.A	à l'U.M.M.T.0
Examineur	TOUAT Mohand Achour	M.C.B.	à l'U.M.M.T.0

Remerciements

Ce travail a été réalisé au laboratoire de Conception et Conduite des Systèmes de Production (L2CSP) de l'université Mouloud MAMMARI de Tizi-Ouzou. À cet effet, je tiens à présenter mes remerciements et ma reconnaissance à tout le personnel du laboratoire.

Je tiens particulièrement à remercier tout d'abord Madame Tounsia DJAMAH, Maître de conférences classe A à l'université de Tizi Ouzou, pour avoir bien voulu diriger ce travail et pour m'avoir constamment aidée tout au long de ce dernier. Qu'elle trouve ici mes très sincères remerciements et mon profond respect.

Je tiens à remercier Monsieur Kamal HAMMOUCHE professeur à l'université Mouloud MAMMARI de Tizi Ouzou, d'avoir bien voulu accepter de présider le jury. J'adresse toute ma gratitude à Monsieur Saïd GUERMAH Maître de conférences classe A l'université de Tizi Ouzou, et Monsieur Mohand Achour TOUAT Maître de conférences classe B au sein de la même université, pour l'intérêt qu'ils ont porté à ce travail en acceptant de faire partie du jury de ce mémoire.

Je tiens à exprimer ma reconnaissance et adresser mes très sincères remerciements à toutes les personnes qui ont contribué d'une manière ou d'une autre, au bon déroulement et à l'aboutissement de ce travail.

Je remercie bien évidemment toute ma famille pour son soutien permanent durant toutes ces années, je remercie mes parents d'avoir toujours été là à mes côtés, de m'avoir encouragée tout au long de mes études, mes sœurs Katia, Lamia, Souhila, mon frère Mouhamed, et mes cousins Saïd, Liliane et Samy. Enfin, j'exprime ma dernière pensée à tous mes ami(e)s qui m'ont soutenue et encouragée durant mon cursus.

Table des matières

Table des matières	vii
Liste des travaux scientifiques réalisés	ix
Liste des abréviations	xii
Table des figures	xiv
Liste des tableaux	xv
Introduction générale	1
1 Identification des systèmes non linéaires	3
1.1 Introduction	3
1.2 Structure de modèles non linéaires	4
1.2.1 Modèles de systèmes à base de blocs structurés	5
1.3 Méthodes d'identification des systèmes non-linéaires	8
1.3.1 Optimisation linéaire	9
1.3.2 Optimisation non linéaire	10
1.4 Méthodes basées sur les métaheuristiques	14
1.5 Conclusion	16
2 Identification des systèmes Hammerstein CAR	17
2.1 Introduction	17
2.2 Etat de l'art sur les méthodes d'identification des systèmes Hammerstein .	18

2.3	Identification du modèle Hammerstein CAR	19
2.4	Estimation basée sur le principe de Sur-paramétrisation	20
2.5	Estimation basée sur le principe Hiérarchique	22
2.6	Exemples de Simulation	24
2.6.1	Principe de sur-paramétrisation	24
2.6.2	Principe Hiérarchique	29
2.7	Conclusion	33
3	Identification des modèles PNLSS	35
3.1	Introduction	35
3.2	Modèles d'état non-linéaires polynomiaux PNLSS	35
3.3	Identification des systèmes PNLSS	37
3.4	Exemples de simulation	39
3.4.1	Exemple 1	39
3.4.2	Exemple 2	42
3.5	Conclusion	43
4	Identification du modèle Hammerstein fractionnaire	45
4.1	Introduction	45
4.2	Calcul fractionnaire	46
4.2.1	Définition de Grünwald-Letnikov	46
4.2.2	Modèle d'état fractionnaire	48
4.2.3	Simulation d'un modèle d'état fractionnaire	48
4.3	Modèle d'état PNLSS fractionnaire et sa simulation	49
4.4	Identification d'un système PNLSS fractionnaire	51
4.5	Exemples de simulation	53
4.5.1	Exemple 1	54
4.5.2	Exemple 2	56
4.6	Conclusion	57
	Conclusion générale et perspectives	61

Bibliographie

63

Liste des travaux scientifiques réalisés

1. **K.Hammar**, T.Djamah, and M.Bettayeb, "Fractional Hammerstein system identification using polynomial non-linear state space model", *3rd International Conference on Control, Engineering & Information Technology, CEIT'2015*, Tlemcen, Algeria, May 2015.
2. L.Serssour, T.Djamah, **K.Hammar**, and M.Bettayeb, "Wiener System Identification using Polynomial Non Linear State Space Model", *3rd International Conference on Control, Engineering & Information Technology, CEIT'2015*, Tlemcen, Algeria, May 2015.
3. **K.Hammar**, T.Djamah, and M.Bettayeb, "Hammerstein model identification using Particle Swarm Optimization", *Colloque sur l'Optimisation et les Systèmes d'Information, COSI 2015*, Oran, Algeria, June 2015.

Communications acceptées non présentées

4. **K.Hammar**, T.Djamah, and M.Bettayeb, "Non linear fractional system identification using polynomial non-linear state space model", *the ASME 2015 International Design Engineering Technical Conferences Computers and Information in Engineering Conference IDETC/CIE 2015*, Boston, Massachusetts, USA, August 2015.
5. **K.Hammar**, T.Djamah, and M.Bettayeb, "Hierarchical Particle Swarm Optimization for non-linear system identification", *10th International Conference on Soft Computing Models in Industrial and Environmental Applications, Optimization, Modeling and Control Systems (OMCSI), SOCO15-SS02*, Burgos, Spain, June 2015.

Liste des abréviations

CAR	Controlled Auto-Regression
HCAR	Hammerstein Controlled Auto-Regression
NFIR	Nonlinear Finit impulse response
NARX	Nonlinear Auto Regressive with eXternal input
NOE	Nonlinear Output Error
NARMAX	Nonlinear Auto-Regressive Moving Average with Exogeneous Input
PNLSS	Polynomial Nonlinear State-Space
PSO	Particle swarm optimization
PSOO	Particle swarm optimization based on Over-parameterization principle
PSOH	Particle swarm optimization based on Hierarchical principle
RLS	Recursive least-squares
HLS	Hierarchical least-squares
RLSO	Recursive least-squares based on Over-parameterization principle
SNR	Signal-to-Noise Ratio

Table des figures

1.1	Modèle de Hammerstein	7
1.2	Modèle de Wiener	7
1.3	Modèle Hammerstein-Wiener	7
1.4	Modèle Wiener-Hammerstein	7
1.5	Modèle Hammerstein CAR	8
1.6	Algorithme de Levenberg-Marquardt	13
2.1	Modèle Hammerstein CAR	19
2.2	Principe Hiérarchique	23
2.3	Résultats de la simulation avec RLSO, cas sans bruit	25
2.4	Résultats de la simulation avec RLSO, cas SNR=34dB	26
2.5	Résultats de la simulation avec RLSO, cas SNR=25dB	26
2.6	Résultats de la simulation avec PSOO, cas sans bruit	27
2.7	Résultats de la simulation avec PSOO, cas SNR=34dB	28
2.8	Résultats de la simulation avec PSOO, cas SNR=25dB	28
2.9	Résultats de la simulation avec PSOH exemple 1, cas sans bruit	30
2.10	Résultats de la simulation avec PSOH exemple 1, cas SNR=34dB	30
2.11	Résultats de la simulation avec PSOH exemple 1, cas SNR=25dB	31
2.12	Résultats de la simulation avec PSOH exemple 2, cas sans bruit	32
2.13	Résultats de la simulation avec PSOH exemple 2, cas SNR=34dB	32
2.14	Résultats de la simulation avec PSOH exemple 2, cas SNR=25dB	33
3.1	Système Hammerstein	36

3.2	Résultats de l'exemple 1 cas sans bruit	40
3.3	Résultats de l'exemple 1 avec SNR=34 dB	41
3.4	Résultats de l'exemple 1 avec SNR=25 dB	41
3.5	Résultats de l'exemple 2 pour le cas sans bruit	43
3.6	Résultats de l'exemple 2 avec SNR=34 dB	43
3.7	Résultats de l'exemple 2 avec SNR=25 dB	43
4.1	Système Hammerstein	50
4.2	Résultats de l'exemple 1 pour le cas sans bruit	55
4.3	Résultats de l'exemple 1 avec $SNR = 34 dB$	56
4.4	Résultats de l'exemple 1 avec SNR=25dB	56
4.5	Résultats de l'exemple 2 pour le cas sans bruit	58
4.6	Résultats de l'exemple 2 avec SNR=34dB	58
4.7	Résultats de l'exemple 2 avec SNR=25dB	58

Liste des tableaux

2.1	Résultats de la simulation avec RLSO pour différents SNR	25
2.2	Résultats de la simulation avec PSOO, cas sans bruit et bruité	27
2.3	Résultats de la simulation avec PSOH exemple 1, pour différents SNR . . .	29
2.4	Résultats de la simulation avec PSOH exemple 2, pour différents SNR . . .	32
3.1	Résultats des simulations de l'exemple 1, cas sans bruit et bruité	41
3.2	Résultats des simulations de l'exemple 2, cas sans bruit et bruité	42
4.1	Résultats d'identification d'un modèle PNLSS fractionnaire exemple 1 . . .	55
4.2	Résultats d'identification d'un modèle PNLSS fractionnaire exemple 2 . . .	57

Introduction générale

En Automatique, la modélisation et l'identification d'un système dynamique complexe sont des étapes importantes pour la simulation, la conception d'une loi de commande et le diagnostic. Différentes méthodes ont été utilisées pour l'identification des modèles linéaires [1]. Cependant, la plupart des systèmes réels montrent un comportement dynamique non-linéaire et différentes structures non-linéaires ont été proposées dans la littérature pour décrire les systèmes non-linéaires, comme par exemple les modèles boîte noire (Narmax, série de Volterra) [2], les réseaux de neurones et les modèles flous [3], ou les modèles à blocs structurés tels que les modèles de Wiener, de Hammerstein,... etc [4, 5, 6, 7, 8]. Les modèles à blocs structurés ont donné satisfaction pour modéliser les systèmes non-linéaires, ils se composent de deux blocs de base : un élément linéaire dynamique et un élément non-linéaire statique.

Actuellement, les systèmes d'ordre fractionnaire, constituent un grand champ d'investigation en Automatique. Ils permettent de modéliser de nombreux phénomènes physiques dans divers domaines des sciences de l'ingénieur [9]. Ces processus sont caractérisés par les propriétés de mémoire longue et une structure de dimension infinie, qui apparaissent dans les phénomènes de diffusion : thermique [10], acoustique [11], [12], électrochimique, [13], électromagnétique [14]...

L'objectif de ce travail est d'apporter une contribution à l'identification des systèmes Hammerstein entiers puis des systèmes Hammerstein fractionnaires.

Dans le premier chapitre, les généralités sur les structures non-linéaires sont présentées. Nous nous sommes intéressés plus particulièrement aux blocs structurés de type Hammerstein. Les différentes méthodes d'optimisation des systèmes non-linéaires sont décrites et

les différentes structures existantes pour le choix de la structure de la partie linéaire et de la partie non-linéaire de système Hammerstein sont mentionnées.

Le deuxième chapitre est consacré à l'identification des systèmes Hammerstein entiers de type Controlled Auto-Regression (CAR). Après un état de l'art sur les méthodes d'identification des systèmes Hammerstein, deux principes d'identification ont été présentés : le principe de sur-paramétrisation et le principe hiérarchique. La principale contribution de ce chapitre consiste à identifier le système Hammerstein CAR, utilisant le principe hiérarchique basé sur une méthode d'optimisation par essaim de particules (PSOH) qui est une méthode heuristique qui s'inspire du comportement social des animaux évoluant en essaim. Cette démarche est validée par la comparaison des résultats obtenus avec ceux obtenus utilisant le principe de sur-paramétrisation (PSOO).

Le troisième chapitre traite l'identification d'un modèle Hammerstein utilisant le modèle d'état non-linéaire polynomial (PNLSS pour Polynomial Non Linear State Space model) à temps discret. Le modèle PNLSS est une extension du modèle d'état linéaire au cas non-linéaire dont la partie linéaire est un modèle d'état et la partie non-linéaire est un polynôme d'ordre connu r . L'algorithme d'optimisation non-linéaire de Levenberg Marquardt est utilisé pour identifier les paramètres de la partie linéaire et ceux de la partie non-linéaire.

Le chapitre 4 est consacré à l'extension des modèles Hammerstein PNLSS pour le cas fractionnaire. Les notions préliminaires essentielles, utilisées pour la dérivation fractionnaire sont présentées. La définition de Grunwald Letnikov est utilisée pour la simulation du modèle PNLSS fractionnaire. L'algorithme de Levenberg Marquardt est développé pour le cas fractionnaire pour estimer les paramètres et l'ordre non entier. Des simulations numériques permettent de tester l'efficacité de l'algorithme développé et nous terminons par une conclusion et des perspectives de recherche.

Chapitre 1

Identification des systèmes non linéaires

1.1 Introduction

En traitement du signal, en automatique et, d'une manière générale, dans toutes les sciences appliquées, il est souvent nécessaire de décrire le comportement d'un système par un modèle mathématique. En effet, ce modèle peut servir en prédiction (afin de déterminer les sorties à des instants futurs), pour la synthèse d'une loi de commande, afin de réguler le système ou de le stabiliser, pour la simulation (dans le but de diminuer le coût et le temps des expériences),etc.

La Modélisation, consiste à déterminer la structure du modèle. Elle peut être menée soit à partir d'une analyse théorique et des lois fondamentales qui régissent le système ; de cette approche résulte un modèle dit *modèle de connaissance* ou encore *modèle "boîte blanche"*. Soit à partir d'une analyse expérimentale où la structure et le nombre de paramètres du modèle sont déduits à partir des seules données d'entrées et de sorties dont le but est de reproduire le comportement externe du système. Il en résulte *un modèle de représentation* ou *modèle "boîte noire"*.

Le modèle boîte grise provient du fait que l'utilisateur d'un système réel ne connaît a priori pas toutes les lois physiques qui régissent le comportement de ce système.

L'identification comporte quatre étapes :

- Acquisition des entrées/sorties sous protocole d'expérimentation : cette étape consiste à fixer un signal d'entrée adapté.
- Choix d'une structure : type de modèle, ordre et retard.
- Estimation des paramètres du modèle : choix d'un algorithme pour trouver le modèle en minimisant les erreurs entre les mesures et le modèle.
- Validation du modèle identifié : Réalisation de plusieurs tests de vérification.

L'objectif de ce chapitre est l'identification des systèmes dynamiques non linéaires de type Hammerstein. Après une description du principe de la modélisation mathématique, nous abordons ici différentes structures de modèles de systèmes dynamiques non linéaires, ainsi que les méthodes d'estimation paramétrique des modèles non linéaires.

1.2 Structure de modèles non linéaires

L'identification des systèmes linéaires a été très largement étudiée dans les décennies passées et différentes structures linéaires ont été utilisées. D'une manière générale, la sortie d'un système dynamique discret peut s'écrire sous la forme suivante :

$$y(k) = g(\varphi(k), \theta) + n(k) \quad (1.1)$$

Où le vecteur de régression $\varphi(k)$ peut être formé comme suit :

$$\varphi(k) = [-y(k-1) \cdots -y(k-n_y) u(k-1) \cdots u(k-n_u)]^T \quad (1.2)$$

et les paramètres n_u et n_y représentent respectivement, l'ordre de la régression sur l'entrée $u(k)$ et l'ordre de la régression sur la sortie $y(k)$, θ le vecteur de paramètres du modèle et $n(k)$ le bruit.

La fonction $g(\cdot)$ étant paramétrée par le vecteur θ , elle peut être linéaire ou non-linéaire. Dans le cas où g est linéaire, différents modèles existent : modèles ARX, ARMAX, Output Error ou Box-Jenkins.

En pratique, les systèmes réels sont non linéaires dans leur grande majorité, donc il est difficile de les caractériser dans une large plage de fonctionnement par des modèles linéaires.

Dans la référence [15], les auteurs proposent un classement, établi en fonction du vecteur de régression pour le système (1.1) :

- Modèles NFIR, dans lesquels seul $u(k-i)$ est utilisé pour construire le vecteur de régression.
- Modèles NARX, dans lesquels $u(k-i)$ et $y(k-j)$ servent à construire le vecteur de régression.
- Modèles NOE, dans lesquels $u(k-i)$ et $\hat{y}(k-j)$ servent à construire le vecteur de régression, $\hat{y}(k)$ étant la sortie estimée.
- Modèles NARMAX, dans lesquels $u(k-i)$, $y(k-j)$ et $\epsilon(k-j)$ servent à construire le vecteur de régression avec $\epsilon(k) = y(k) - \hat{y}(k)$.

avec $(1 \leq j \leq n_y)$ et $(1 \leq i \leq n_u)$.

D'autre part, les systèmes non-linéaires peuvent être représentés par un modèle d'état non-linéaire généralisé à temps discret pour n_u entrées, n_y sorties, et n_x états donné par :

$$\begin{cases} x(k+1) = f(x(k), u(k), \theta) \\ y(k) = S(x(k), u(k), \theta) \end{cases} \quad (1.3)$$

avec : $u \in \mathbb{R}^{n_u}$; $y \in \mathbb{R}^{n_y}$; $x \in \mathbb{R}^{n_x}$. Les fonctions f et S dépendent du vecteur des paramètres $\theta \in \mathbb{R}^{n_\theta}$.

Récemment les modèles à blocs structurés ont été extensivement utilisés pour la description des systèmes non-linéaires.

1.2.1 Modèles de systèmes à base de blocs structurés

Un moyen relativement simple de représenter le comportement non linéaire d'un système repose sur l'utilisation de modèles à base de blocs structurés [16] c-à-d la combinaison de deux blocs de base : un élément non linéaire statique NL et un élément linéaire dynamique L. Parmi ces modèles on distingue le modèle d'Hammerstein, le modèle de Wiener,

le modèle de Wiener-Hammerstein, le modèle Hammerstein-Wiener.

Le modèle Hammerstein (figure 1.1), se compose d'un élément non linéaire statique suivi d'un bloc linéaire dynamique [17]. Le signal d'entrée passe d'abord par le modèle non-linéaire statique NL pour donner le signal de sortie intermédiaire, lui-même transformé par la partie linéaire dynamique L pour donner le signal de sortie du modèle. L'élément non-linéaire peut expliquer les non-linéarités de type actionneur et d'autres effets qui peuvent être placés à l'entrée du modèle. En dépit de leur simplicité, les modèles Hammerstein s'avèrent capables de décrire exactement une grande variété de systèmes non-linéaires par exemple : les processus chimiques [18], les amplificateurs de puissance [19], les commandes électriques [20]...

La permutation des éléments linéaires et non-linéaires dans le modèle de Hammerstein mène aux modèles de Wiener [21]. Sa structure est présentée sur la figure 1.2. Les modèles de Wiener se révèlent bien adaptés à la caractérisation, au moyen d'un modèle linéaire, du comportement dynamique d'un système dont le capteur présente un caractère non linéaire.

Dans [22], les auteurs ont montré que la majorité des systèmes non-linéaires peuvent être approximés par un modèle de Wiener avec une exactitude arbitrairement élevée. Ce fait théorique a été expérimentalement vérifié par plusieurs applications pratiques par exemple : les processus chimiques[23], les systèmes biologiques [24],...etc.

La combinaison en série d'un modèle d'Hammerstein et d'un modèle de Wiener, permet d'élaborer une nouvelle structure de type Hammerstein-Wiener (la figure 1.3). Un modèle de type Wiener-Hammerstein peut également être obtenu sur ce même principe mais en inversant l'ordre des modèles (figure 1.4). Les modèles de Hammerstein-Wiener se révèlent bien adaptés à la caractérisation d'un système dont l'actionneur et le capteur présentent un caractère non linéaire. Il a été appliqué avec succès à la modélisation de plusieurs processus physiques par exemple : les réacteurs de polymérase [25].

La plupart des travaux ont supposé que l'élément linéaire est paramétrique représenté par une fonction du transfert d'ordre connu [26] ou par une représentation d'état ; le degré de la dynamique est alors supposé connu [27]. Certains travaux ont suggéré des re-

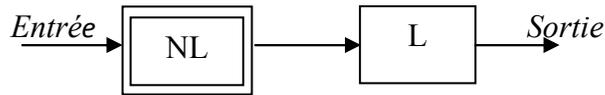


FIGURE 1.1: Modèle de Hammerstein

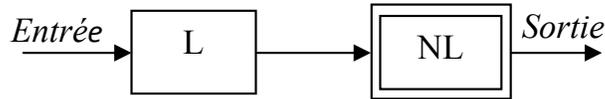


FIGURE 1.2: Modèle de Wiener

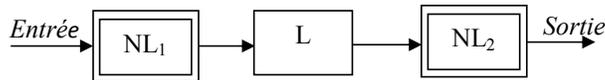


FIGURE 1.3: Modèle Hammerstein-Wiener



FIGURE 1.4: Modèle Wiener-Hammerstein

présentations non paramétriques de type réponse impulsionnelle finie (FIR) [28], réponse impulsionnelle infinie,[30], ou réponse fréquentielle [31].

Le sous-système non linéaire a été supposé polynomial et de degré connu ,[30]. Certains travaux ont supposé le développement de l'élément non linéaire en série de fonction orthogonale comme les splines cubiques dans [32], les polynômes de Laguerre dans [29], de Legendre [33]. Les non-linéarités de type hysteresis ont été considérées dans [34],[35], l'élément non linéaire de type seuils ou zone mortes a été abordé dans [36]. Dans le domaine fréquentiel, on peut citer les travaux de Bai [8].

Parmi les structures de modèle Hammerstein, on distingue le modèle Controlled Auto-Regression CAR donné en figure 1.5. La partie linéaire, est caractérisée par les deux polynômes $A(z)$ et $B(z)$, et la partie non-linéaire est un polynôme d'ordre connu. Le modèle Hammerstein CAR est donné par la relation suivante :

$$\begin{aligned} A(z) y(k) &= B(z) \tilde{u}(k) + n(k) \\ \tilde{u}(k) &= f(u(k)) \end{aligned} \tag{1.4}$$

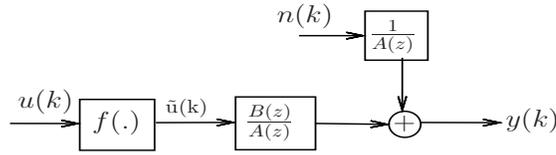


FIGURE 1.5: Modèle Hammerstein CAR

où

$u(k)$ et $y(k)$ désignent respectivement l'entrée et la sortie du système, $n(k)$ est une perturbation, et $\tilde{u}(k)$ est la sortie du bloc non linéaire (est un signal interne non mesurable), $f(u)$ est une fonction non linéaire d'une base connue avec

$$f(u(k)) = p_1 f_1(u(k)) + p_2 f_2(u(k)) + \dots + p_r f_r(u(k)) \quad (1.5)$$

$G(z) := B(z)/A(z)$ est la fonction de transfert du bloc dynamique linéaire avec :

$$\begin{aligned} A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n_a} \\ B(z) &= 1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n_b} \end{aligned} \quad (1.6)$$

où

z^{-1} représente l'opérateur retard [$z^{-1}y(k) = y(k-1)$].

L'identification de ce type de modèle fera l'objet du deuxième chapitre. Les paragraphes suivants décrivent différentes méthodes d'estimation paramétrique des modèles non linéaires.

1.3 Méthodes d'identification des systèmes non-linéaires

Dans ce paragraphe, on rappelle les caractéristiques des algorithmes d'identification les plus utilisés en identification (et en commande adaptative) des systèmes non linéaires. Les méthodes d'identification sont généralement basées sur la minimisation d'une fonctionnelle de l'écart entre la sortie mesurée $y(k)$ et de la sortie estimée $\hat{y}(k)$. Le critère J à minimiser représente l'écart quadratique entre les deux sorties indiquées :

$$J = \sum_{k=1}^K \varepsilon^2(k)$$

Où ε l'erreur de prédiction ou résidu tel que $\varepsilon(k) = y(k) - \hat{y}(k)$, K est l'horizon d'observation $(u(k), y(k))$.

1.3.1 Optimisation linéaire

Les méthodes classiques d'identification sont basées sur l'optimisation non linéaire des moindres carrés.

Soit un modèle linéaire d'écrit sous la forme :

$$y(k) = \varphi^T(k)\theta + n(k) \quad (1.7)$$

où φ est un vecteur de régression et θ le vecteur de paramètres.

Le principe de la méthode des moindres carrés est basé sur la minimisation du critère J . avec :

$$J = \sum_{k=1}^K \varepsilon^2(k) = \sum_{k=1}^K [y(k) - \varphi^T(k)\hat{\theta}]^2$$

Où θ est le vecteur de paramètres à estimer, la valeur optimale $\hat{\theta}$ minimisant ce dernier est obtenue par l'estimateur des moindres carrés :

$$\hat{\theta} = (\phi^T \phi)^{-1} \phi^T Y \quad (1.8)$$

ϕ est la matrice d'information :

$$\phi = [\varphi(1) \quad \dots \quad \varphi(K)]^T \quad (1.9)$$

Y est le vecteur de sortie :

$$Y = [y^T(1) \quad \dots \quad y^T(K)]^T \quad (1.10)$$

L'estimateur des moindres carrés conduit à un optimum unique mais il est biaisé. Pour cela, l'utilisation d'autres méthodes d'identification telles que la méthode des Moindres Carrés Récursifs est alors préconisée.

Méthode des Moindres Carrés Récursifs (RLS)

Le principe de l'identification récursive (en temps réel) consiste à calculer, à chaque instant k , le vecteur des paramètres du modèle recherché $\hat{\theta}(k)$ en fonction des paramètres estimés à l'instant précédent $\hat{\theta}(k-1)$ et des nouvelles informations acquises sur le procédé.

On considère le cas d'un système non linéaire pour lequel le modèle peut être représenté sous la forme de l'équation (1.1), le vecteur de paramètres $\hat{\theta}$ peut être mis à jour de façon récursive à chaque instant en utilisant la relation suivante :

$$\begin{cases} \hat{\theta}(k) &= \hat{\theta}(k-1) + P(k)\varphi(k)[y(k) - \varphi(k)\hat{\theta}(k-1)] \\ P^{-1}(k) &= P^{-1}(k-1) + \varphi(k)\varphi^T(k) \end{cases} \quad (1.11)$$

Où $P(k)$ le gain d'adaptation.

1.3.2 Optimisation non linéaire

Pour les modèles non linéaires, l'estimation paramétrique se fait par une méthode itérative d'optimisation non linéaire :

- méthode du Gradient,
- méthode de Gauss-Newton,
- méthode du gradient conjugué,
- méthode de Levenberg-Marquardt. . .

Méthode du gradient

Cette méthode basée sur la recherche itérative d'un vecteur des paramètres (θ) minimisant un critère J . L'équation récurrente de modification du vecteur θ est de la forme suivante :

$$\theta^{(i+1)} = \theta^{(i)} - \lambda J'_\theta; \quad J'_\theta = \left. \frac{\partial J}{\partial \theta} \right|_{\theta=\theta^{(i)}} \quad (1.12)$$

Où :

J' est le gradient du critère J .

$\theta^{(i)}$ étant le vecteur de paramètre estimé à l'itération i , et le coefficient λ est un paramètre de contrôle. Une faible valeur de λ conduit à une convergence lente vers le minimum alors qu'une valeur élevée induit des oscillations autour du minimum.

Méthode de Gauss-Newton

Cette méthode utilise des dérivées premières et secondes du critère. La mise à jour du vecteur de paramètres est obtenue à partir de l'équation de récurrence suivante [37] :

$$\theta^{(i+1)} = \theta^{(i)} - [(J''_{\theta\theta})^{-1} J'_{\theta}]_{\theta=\theta^{(i)}} \quad (1.13)$$

Le gradient peut alors être écrit :

$$J'_{\theta} = -2 \sum_{k=1}^K \varepsilon(k) \sigma(k, \theta)$$

La dérivée seconde de J (le Hessien) est donnée par :

$$J''_{\theta\theta} \approx -2 \sum_{k=1}^K \sigma(k, \theta)^T \sigma(k, \theta)$$

$\sigma(k, \theta) = \frac{\partial \hat{y}(k, \theta)}{\partial \theta}$ est la fonction de sensibilité de la sortie $\hat{y}(k)$ par rapport au paramètre θ .

L'algorithme de Gauss Newton présente l'avantage d'une rapide convergence vers l'optimum, mais s'avère instable lorsque l'initialisation n'est pas au voisinage de l'optimum. Une version améliorée de la méthode Gauss-Newton est la méthode de Levenberg-Marquardt.

Méthode de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt combine les avantages des algorithmes du gradient et de Gauss-Newton. La mise à jour de l'estimation s'effectue de la manière suivante

[38] :

$$\left\{ \begin{array}{l} \theta^{(i+1)} = \theta^{(i)} - \left\{ [J''_{\theta\theta} + \lambda I]^{-1} J'_{\theta} \right\}_{\hat{\theta}=\theta^{(i)}} \\ J'_{\theta} = -2 \left(\sum_{k=1}^K \varepsilon_k \sigma_{k/\theta} \right) \text{ le gradient} \\ J''_{\theta} = -2 \left(\sum_{k=1}^K \sigma_{k/\theta} \sigma_{k/\theta}^T \right) \text{ le Hessien} \\ \sigma_{k/\theta} = \frac{\partial \hat{y}(k, \theta)}{\partial \theta} \text{ Fonction de sensibilité de la sortie par rapport à } \theta \end{array} \right. \quad (1.14)$$

avec I la matrice identité. La valeur du paramètre de contrôle λ , appelé aussi coefficient de Marquardt doit être ajusté à chaque pas de calcul.

si $\lambda \rightarrow 0$, on retrouve l'algorithme de Gauss-Newton.

si $\lambda \rightarrow \infty$, on retrouve l'algorithme du Gradient.

Cet algorithme est basé sur le calcul du Gradient (J') et du Hessien (J''), qui utilisent des fonctions de sensibilité σ que nous définissons dans le paragraphe suivant.

Calcul des fonctions de sensibilité

Considérons le cas général d'un système non-linéaire sous forme de représentation d'état à temps discret, pour simplifier l'écriture, les signaux sont écrits en fonction du k tel que :

$$\left\{ \begin{array}{l} x(k+1) = f(x(k), u(k), \theta) \\ y(k) = g(x(k), u(k), \theta) \end{array} \right. \quad (1.15)$$

Le vecteur de paramètres à estimer comprend les éléments des matrices définissant le modèle d'état. Dans ce cas, il convient de définir deux sortes de fonctions de sensibilité :

- $\sigma_{y/\theta_j} = \frac{\partial y(k)}{\partial \theta_j}$ vecteur des fonctions de sensibilité par rapport à la sortie.

- $\sigma_{x/\theta_j} = \frac{\partial x(k)}{\partial \theta_j}$ vecteur des fonctions de sensibilité par rapport à l'état.

Avec $j = 1, \dots, n_{\theta}$ et n_{θ} le nombre de paramètres à estimer.

Ces fonctions peuvent être obtenues par différentiation du modèle (1.15) par rapport à θ_j

tel que :

$$\frac{\partial x(k+1)}{\partial \theta_j} = \sigma_{x(k+1)/\theta_j}$$

de même pour les équations de sortie :

$$\frac{\partial y(k)}{\partial \theta_j} = \sigma_{y(k)/\theta_j}$$

. Dans les travaux présentés dans les chapitres 3 et 4, nous utilisons la méthode de Levenberg-Marquardt pour la programmation non-linéaire. La figure 1.3.2 présente l'algorithme utilisé pour la réalisation de cette méthode. La modification du coefficient λ

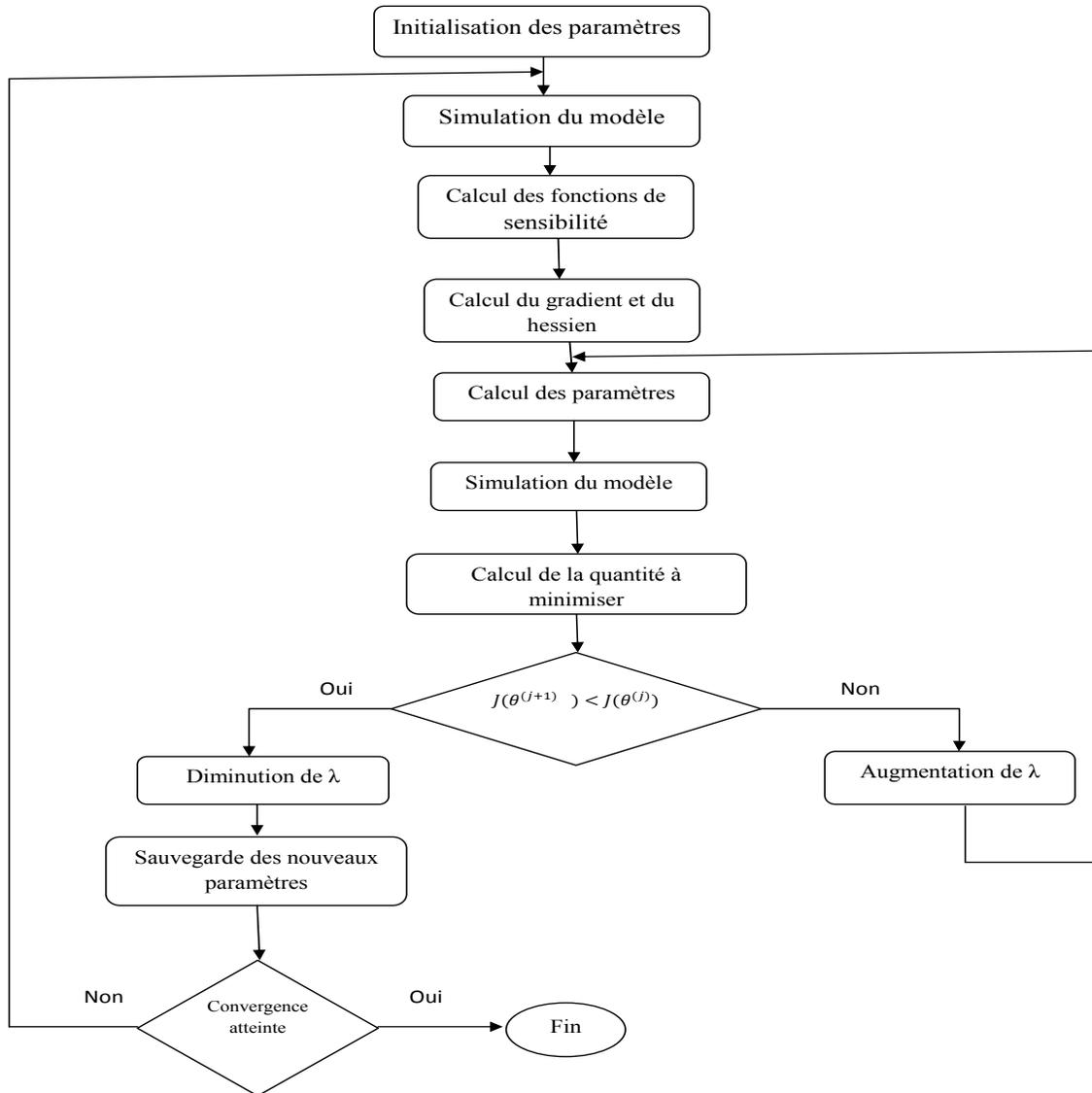


FIGURE 1.6: Algorithme de Levenberg-Marquardt

permettant un réglage de l'algorithme de Levenberg-Marquardt, est déterminée par l'évolution du critère quadratique J :

- Lorsque J augmente, on augmente alors λ pour se rapprocher de la méthode du type gradient.
- Si J diminue, on diminue λ pour se rapprocher de la méthode du type Gauss-Newton

1.4 Méthodes basées sur les métaheuristiques

Les métaheuristiques sont des méthodes approchées, leur particularité réside dans le fait que celles-ci sont adaptables à un grand nombre de problèmes sans changement majeur dans leurs algorithmes, elles sont en général stochastiques et itératives.

Ces méthodes ont des inspirations de processus naturels dans de nombreux domaines tels que : l'éthologie comme les colonies de fourmis, la physique comme le recuit simulé, de la biologie comme les algorithmes évolutionnaires, et les algorithmes d'optimisation qui utilisent les comportements sociaux et évolutifs comme la méthode d'optimisation par essaims de particules (PSO). Parmi les métaheuristiques, nous nous intéressons à l'optimisation par essaims de particules.

L'optimisation par essaim de particule, ou Particle Swarm Optimization (PSO) en Anglais, est une méthode populaire pour résoudre des problèmes d'optimisation complexes, où la population s'appelle maintenant un essaim et chaque individu s'appelle une particule. Elle a été proposée par Kennedy et Eberhart en 1995 [39], cette méthode est inspirée du comportement social des animaux évoluant en essaim [39, 40], comme les oiseaux évoluant en groupe, les bancs de poissons,... etc. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs, les algorithmes à essaims de particules recherchent des solutions pour un problème d'optimisation.

Dans un système de PSO, les particules se déplacent à l'intérieur d'un espace de recherche, pendant le déplacement, chaque particule ajuste sa position selon sa meilleure position, et de celle de ses voisines.

Pour appliquer PSO, il faut définir un espace de recherche constitué de particules et d'une fonction coût (fonction 'objectif') à optimiser. Le principe de l'algorithme est de déplacer ces particules afin qu'elles trouvent l'optimum, le déplacement d'une particule

est influencé par les trois types de comportement : suivre son propre chemin, sa tendance à revenir vers sa meilleure position atteinte et sa tendance à aller vers son meilleur voisin. Chaque particule i est caractérisé par une position X_i et une vitesse V_i

L'algorithme PSO est donné par les étapes suivantes :

- Étape 1. Génération d'une population initiale de Q particules avec des positions et des vitesses aléatoires. Chaque particule a une position initiale X_i^0 et une vitesse initiale V_i^0 , avec $i = 1 \dots, Q$.
- Étape 2. Construction du modèle à identifier.
- Étape 3. Exécution de la procédure d'identification et l'estimation du vecteur de paramètres inconnus, avec l'évaluation du critère de performance ou d'une erreur quadratique J .
- Étape 4. Mise à jour de la meilleure position $p_{best_i}^l$ de chaque particule et celle du groupe g_{best}^l .
- Étape 5. Mise à jour de la position et de la vitesse de chaque particule par l'équation (1.16).

Au départ de l'algorithme, les particules de l'essaim sont initialisées de manière aléatoire/régulière dans l'espace de recherche du problème. À chaque itération, la position et la vitesse sont mises à jour suivant la formule citée ci-dessous :

$$\begin{cases} V_i^{l+1} &= \omega V_i^l + \alpha_1 rand_1(p_{best_i}^l - X_i^l) + \alpha_2 rand_2(g_{best}^l - X_i^l) \\ X_i^{l+1} &= X_i^l + V_i^{l+1} \end{cases} \quad (1.16)$$

ω est le coefficient d'inertie, α_1 et α_2 sont deux constantes, appelées coefficients d'accélération. Avec $\alpha_1=0.7$ et $\alpha_2 = (2/0.97725)\alpha_1$, et le coefficient d'inertie ω est calculé comme suit :

$$\omega = 0.9 - (0.55/l_{max})l \quad (1.17)$$

avec l l'itération courante et l_{max} le nombre maximal d'itérations.

1.5 Conclusion

Dans ce chapitre, nous avons présenté les principes de la modélisation "boîte noire", les étapes de la conception d'un tel modèle, ainsi que quelques structures paramétrées utilisables, et les algorithmes qu'il convient de mettre en œuvre pour l'ajustement des paramètres. Le chapitre suivant sera consacré à l'identification d'un système Hammerstein de type CAR et à la mise en œuvre des deux principes d'identification utilisant l'algorithme d'optimisation par essaim de particule PSO : le principe de sur-paramétrisation et le principe hiérarchique.

Chapitre 2

Identification des systèmes

Hammerstein CAR

2.1 Introduction

Dans ce chapitre, le problème d'identification des systèmes Hammerstein CAR (HCAR) est abordé. Le but de l'identification est l'estimation des paramètres du modèle de la dynamique linéaire et ceux de la partie non linéaire. Dans ce but, deux principes d'identification peuvent être utilisés le principe de sur-paramétrisation et le principe hiérarchique, deux méthodes sont associées au principe de sur-paramétrisation la première est celle des moindres carrés récurrents, et la deuxième est l'optimisation par essaim de particule PSO, cette dernière méthode est associée aussi au principe hiérarchique.

Le chapitre débute par un état de l'art des méthodes d'identification des systèmes Hammerstein, le système à identifier est décrit dans la section 2.3. La section 2.4 aborde le principe de sur-paramétrisation, le principe d'identification hiérarchique est présenté dans la section 2.5. La fin de ce chapitre est consacrée à des exemples d'identification illustrant les performances de PSO et les limites des RLS utilisant le principe de sur-paramétrisation. Finalement, une comparaison entre les deux principes à base d'algorithme PSO illustre l'efficacité de PSO Hiérarchique.

2.2 Etat de l'art sur les méthodes d'identification des systèmes Hammerstein

Les premiers travaux sur l'identification des systèmes Hammerstein ont été réalisés par Narendra et Gallman (1966)[16] ; Depuis, plusieurs méthodes d'identifications ont été proposées. Dans [16] [41], les auteurs ont abordé la méthode d'identification itérative, cette dernière peut donner de bons résultats mais elle peut être divergente [42]. En 2002, Bai a utilisé la méthode d'identification aveugle [8]. Dans [43][44][45] [46] [47], les auteurs ont implémenté le principe de Sur-paramétrisation (en Anglais *Over-parameterization*) ce principe sera développé dans la section 2.4.

En 2008, Giri et al. [48] ont abordé le problème d'identification des systèmes Hammerstein en présence des non-linéarités de type hysteresis-relais et, ils ont utilisé l'algorithme des moindres carrés récursifs.

En plus des contributions mentionnées ci-dessus, beaucoup de travaux sur l'identification des systèmes Hammerstein existent dans la littérature. Par exemple, Vörös a étudié le problème d'identification des paramètres des systèmes de Hammerstein avec des non-linéarités discontinues en utilisant le principe de séparation des termes clé *the key term separation principal*[43].

Le principe d'identification hiérarchique a été abordé par Chen et al (2014) [49], les auteurs ont développé l'algorithme (HLS) des moindres carrés hiérarchiques pour l'estimation des paramètres du système Hammerstein de type Controlled Auto Regression (H-CAR), l'étude de la convergence a montré que l'algorithme (HLS) converge plus rapidement par rapport à l'algorithme des moindres carrés récursifs donné dans [50].

En 2010, Wang et al [51] ont utilisé l'optimisation par essaim de particule pour identifier le système Hammerstein, dans le même contexte, en 2014 Wang et al [52] ont estimé les paramètres d'un processus thermique à base d'un modèle Hammerstein.

Dans notre travail, le modèle Hammerstein CAR est considéré.

2.3 Identification du modèle Hammerstein CAR

Soit le modèle Hammerstein CAR (HCAR) de la figure (2.1) décrit par la relation suivante :

$$\begin{aligned} A(z)y(k) &= B(z)\tilde{u}(k) + n(k) \\ \tilde{u}(k) &= f(u(k)) \end{aligned} \quad (2.1)$$

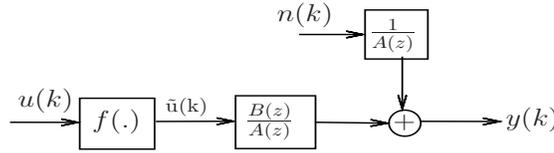


FIGURE 2.1: Modèle Hammerstein CAR

La partie non-linéaire $f(u)$ est une fonction non linéaire d'une base connue avec

$$f(u(k)) = p_1 f_1(u(k)) + p_2 f_2(u(k)) + \dots + p_r f_r(u(k)) \quad (2.2)$$

Les deux polynômes $A(z)$ et $B(z)$ de la partie linéaire sont donnés comme suit :

$$\begin{aligned} A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n_a} \\ B(z) &= b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n_b} \end{aligned} \quad (2.3)$$

où

z^{-1} représente l'opérateur retard [$z^{-1}y(k) = y(k-1)$].

En effet, en substituant 2.2 et 2.3 dans 2.1 on obtient successivement :

$$\begin{aligned} y(k) &= -\sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=1}^{n_b} b_i \tilde{u}(k-i) + n(k) \\ &= -\sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=1}^{n_b} b_i \sum_{j=1}^r p_j f_j(u(k-i)) + n(k) \\ &= -\sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=1}^{n_b} \sum_{j=1}^r p_j b_i f_j(u(k-i)) + n(k) \\ &= -\sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=1}^{n_b} [p_1 b_i f_1(u(k-i)) + p_2 b_i f_2(u(k-i)) + \dots + p_r b_i f_r(u(k-i))] + n(k) \\ &= -\sum_{i=1}^{n_a} a_i y(k-i) + p_1 b_1 f_1(u(k-1)) + p_1 b_2 f_1(u(k-2)) + \dots + p_1 b_{n_b} f_1(u(k-n_b)) \\ &\quad + p_2 b_1 f_2(u(k-1)) + p_2 b_2 f_2(u(k-2)) + \dots + p_2 b_{n_b} f_2(u(k-n_b)) + \dots + p_r b_1 f_r(u(k-1)) \\ &\quad + p_r b_2 f_r(u(k-2)) + \dots + p_r b_{n_b} f_r(u(k-n_b)) + n(k). \end{aligned} \quad (2.4)$$

L'objectif de l'identification est de déterminer les vecteurs de paramètres a et b de système CAR et p de la partie non linéaire avec :

$$\begin{aligned} a &= [a_1, a_2, \dots, a_{n_a}]^T \in \mathbb{R}^{n_a} \\ b &= [b_1, b_2, \dots, b_{n_b}]^T \in \mathbb{R}^{n_b} \\ p &= [p_1, p_2, \dots, p_r]^T \in \mathbb{R}^r \end{aligned} \quad (2.5)$$

L'équation (2.4) peut être réécrite sous la forme suivante [16], [47], [53] :

$$y(k) = \varphi^T(k)a + b^T F(k)p + n(k) \quad (2.6)$$

avec $\varphi(k)$ le régresseur, défini par

$$\varphi(k) = [-y(k-1) - y(k-2) \dots - y(k-n_a)]^T \in \mathbb{R}^{n_a} \quad (2.7)$$

$F(k)$ est la matrice d'information telle que :

$$F(k) = \begin{bmatrix} f_1(u(k-1)) & f_2(u(k-1)) & \dots & f_r(u(k-1)) \\ f_1(u(k-2)) & f_2(u(k-2)) & \dots & f_r(u(k-2)) \\ \vdots & \vdots & & \vdots \\ f_1(u(k-n_b)) & f_2(u(k-n_b)) & \dots & f_r(u(k-n_b)) \end{bmatrix}$$

$$F(k) = [\mathbf{f}(u(k-1)) \mathbf{f}(u(k-2)) \dots \mathbf{f}(u(k-n_b))]^T \quad (2.8)$$

Sans perte de généralité on normalise l'un des gains de $\frac{B(z)}{A(z)}$ ou $f(u(k))$. On prend le premier coefficient de $f(\cdot)$ égal à 1 c-à-d $p_1 = 1$ [8].

Nous présentons par la suite deux principes d'identification, le premier est le principe de sur-paramétrisation et le second est le principe Hiérarchique.

2.4 Estimation basée sur le principe de Sur-paramétrisation

Le principe d'identification présenté ici, relève des travaux de [47, 46, 45, 44, 43, 54, 50], il consiste d'utiliser l'expression de de la partie non linéaire comme somme des fonctions de base, de sorte que, le vecteur final de paramètres est un produit entre les paramètres de

la non linéarité statique et la dynamique linéaire. En d'autres termes, les paramètres et les produits inconnus des paramètres semblent être linéaires dans le vecteur de paramètres, et alors n'importe quel algorithme linéaire peut être appliqué.

L'équation (2.6) peut être écrite sous la forme linéaire suivante :

$$y(k) = \xi(k)^T \theta + n(k) \quad (2.9)$$

avec θ et $\xi(k)$ désignent, respectivement, le vecteur de paramètres et le vecteur d'observation.

$$\theta = [a \quad p_1 b_1 \quad p_1 b_2 \quad \cdots \quad p_1 b_{n_b} \quad p_2 b_1 \quad \cdots \quad p_r b_1 \quad \cdots \quad p_r b_{n_b}] \in \mathbb{R}^{n_\theta} \quad (2.10)$$

$$\theta = [a \quad b \quad p_2 b \quad \cdots \quad p_r b] \in \mathbb{R}^{n_\theta} \quad (2.11)$$

$$n_\theta = n_a + n_b r \quad (2.12)$$

$$\xi(k) = [\varphi(k)^T, \mathbf{f}(u(k-1)), \mathbf{f}(u(k-2)), \cdots, \mathbf{f}(u(k-n_b))]^T \in \mathbb{R}^{n_\theta} \quad (2.13)$$

Le problème de l'estimation se formalise, maintenant, par la recherche d'un vecteur de paramètres $\hat{\theta}$ minimisant un critère quadratique donné par l'équation suivante :

$$J = \sum_{k=1}^K [y(k) - \xi(k)^T \hat{\theta}]^2 \quad (2.14)$$

L'estimation de θ fera l'objet du paragraphe suivant.

Auparavant, arrêtons-nous sur le problème de passage des paramètres composés $b_i p_j$ ($i = 1, 2, \dots, n_b$ et $j = 1, 2, \dots, r$) vers les paramètres qui nous intéressent réellement, à savoir b_1, b_2, \dots, b_{n_b} et p_1, p_2, \dots, p_r . L'équation (2.11) suggère alors les estimés $\hat{a} = [\hat{a}_1 \quad \hat{a}_2 \quad \cdots \quad \hat{a}_{n_a}]$ comme les premiers n_a éléments de $\hat{\theta}$; $\hat{b} = [\hat{b}_1 \quad \hat{b}_2 \quad \cdots \quad \hat{b}_{n_b}]$ les n_b élément suivant de $\hat{\theta}$ c-à-d :

$$\hat{a}_i = \hat{\theta}(i), \quad i = 1, 3, \dots, n_a \quad (2.15)$$

$$\hat{b}_i = \hat{\theta}(n_a + i), \quad i = 1, 3, \dots, n_b \quad (2.16)$$

Les estimés \hat{p}_j du vecteur p , $j = 2, 3 \dots, r$ peuvent être déduits de l'équation suivante :

$$\hat{p}_j = \frac{\hat{\theta}(n_a + j n_b - n_b + i)}{\hat{b}_i}, \quad j = 2, 3, \dots, r; \quad i = 1, 2, \dots, n_b \quad (2.17)$$

De l'équation (2.11), nous remarquons que pour chaque p_j nous avons n_b estimés \hat{p}_j , la valeur moyenne peut être calculé comme estimé de \hat{p}_j comme suit :

$$\hat{p}_j = \frac{1}{r} \sum_{k=1}^r \frac{\hat{\theta}(n_a + kn_b - n_b + i)}{\hat{b}_i}, \quad i = 1, 2, \dots, n_b \quad (2.18)$$

Le vecteur θ intervient linéairement dans l'équation (2.9) en utilisant à cet effet un estimateur de type moindres carrés récursifs dont la forme a été présentée dans le sous paragraphe 1.3.1 et un algorithme d'optimisation par essaim de particules PSO développé au paragraphe 1.4.

L'inconvénient de ce principe, est l'augmentation de nombre de paramètres à estimer de $n_\theta = n_a + n_b + r$ à $n_\theta = n_a + n_b r$, ce qui alourdit le temps de calcul, Pour y remédier à cet inconvénient le principe hiérarchique est préconisé. Il transforme un système à grande dimension en plusieurs sous-systèmes, avec de petites dimensions, plus faciles à identifier, ce principe est décrit dans la section suivante.

2.5 Estimation basée sur le principe Hiérarchique

Le principe d'identification hiérarchique [55], [56], [49], [58], [57], consiste à décomposer le système Hammerstein CAR -HCAR- en deux sous systèmes avec avec de petites dimensions, et peu de variables. Le premier contient le vecteur de paramètres de la partie linéaire $\theta_1 = [a \quad b]^T$, et le second contient le vecteur de paramètres de la partie non linéaire $\theta_2 = p$ (figure 2.5). Ainsi le système (2.6) peut être réécrit comme suit :

$$\begin{aligned} S1 : y(k) &= \phi^T(p, k)\theta_1 + n(k) \\ S2 : y_1(t) &= \Psi^T(b, k)\theta_2 + n(k) \end{aligned} \quad (2.19)$$

Les vecteurs d'observation $\phi(p, k)$ et $\Psi(b, k)$ sont donnés comme suit :

$$\phi(p, k) = \begin{bmatrix} \varphi((k) \\ F(k)p \end{bmatrix} \in \mathfrak{R}^{n_a+n_b} \quad \Psi(b, k) = F^T(k)b \in \mathbb{R}^r \quad (2.20)$$

Cependant, les sous-systèmes sont liés entre eux, c-à-d, le premier sous-système contient le

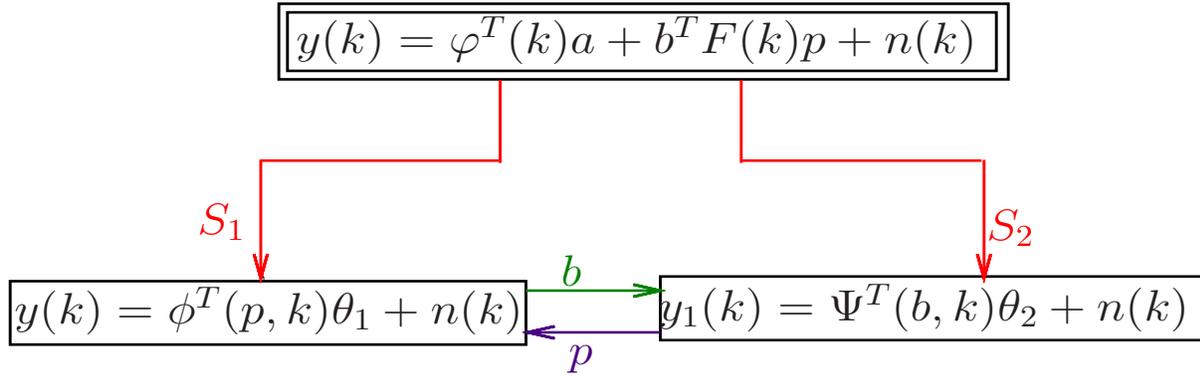


FIGURE 2.2: Principe Hiérarchique

vecteur inconnu de paramètres p du deuxième sous-système, alors que le seconde contient le vecteur inconnu de paramètres b du premier sous système. Par conséquent le problème des calculs itératifs est résolu en calculant les estimés de θ_1 à l'itération l avec le paramètre inconnu θ_2 , remplacé par son estimé, $\hat{\theta}_2$, à l'itération $(l - 1)$ et calculant l'estimation de θ_2 à l'itération l et le vecteur de paramètres θ_1 du premier système, remplacé avec son estimé, $\hat{\theta}_1$ calculé à l'itération $(l - 1)$.

Définissons les critères quadratiques pour les sous-systèmes S_1 et S_2 de (2.19) :

$$\begin{aligned} J_1(\hat{\theta}_1) &= \sum_{k=1}^N [y(k) - \phi^T(p, k)\hat{\theta}_1]^2 \\ J_2(\hat{\theta}_2) &= \sum_{k=1}^N [y_1(k) - \Psi^T(b, k)\hat{\theta}_2]^2 \end{aligned} \quad (2.21)$$

Ces deux principes seront utilisés pour l'optimisation de la méthode PSO heuristique afin de déterminer les paramètres de modèle HCAR. Pour le principe hiérarchique, la population est divisée en deux sous populations, y compris la position initiale et la vitesse de chaque particule.

Soit P_{1best}^l et P_{2best}^l respectivement la position initiale de la première sous-population et la deuxième sous-population, et nous calculons g_{1best}^l selon la première fonction fitness de la première sous-population et g_{2best}^l pour la deuxième fitness de la seconde sous-population. Afin de tester la validité et la précision de l'algorithme PSOH, deux simulations numériques sont réalisées sur un système HCAR, dans le prochain paragraphe.

2.6 Exemples de Simulation

2.6.1 Principe de sur-paramétrisation

Ce principe est utilisé avec la méthode classique récursive moindre carré puis la méthode heuristique PSO.

L'objectif de ce paragraphe, est d'illustrer les performances de l'algorithme PSO à base du principe de sur-paramétrisation PSOO, et de comparer les résultats obtenus avec celle obtenus avec l'algorithme RLS développé au chapitre 1, en utilisant le principe de sur-paramétrisation (RLSO).

Soit un système Hammerstein HCAR décrit par l'équation suivante :

$$A(z)y(k) = B(z)\tilde{u}(k) + n(k)$$

Les polynômes $A(z)$ et $B(z)$ de la partie linéaire sont donnés comme suit :

$$\begin{aligned} A(z) &= 1 + a_1z^{-1} + a_2z^{-2} = 1 - 1.60z^{-1} + 0.80z^{-2} \\ B(z) &= b_1z^{-1} + b_2z^{-2} = z^{-1} + 0.65z^{-2} \end{aligned} \tag{2.22}$$

La partie non-linéaire est un polynôme d'ordre 3

$$\begin{aligned} f(u(k)) &= p_1u(k) + p_2u^2(k) + p_3u^3(k) \\ &= u(k) + 0.50u^2(k) + 0.25u^3(k) \end{aligned} \tag{2.23}$$

à savoir $\theta = [a_1 \ a_2 \ b_1 \ b_2 \ p_2 \ p_3]$, mais avec le principe de sur-paramétrisation le vecteur de paramètres est : $\theta = [a_1 \ a_2 \ b_1 \ b_2 \ b_1p_2 \ b_2p_2 \ b_1p_3 \ b_2p_3]$ et $n_\theta = 8$

L'entrée $u(k)$ est une excitation persistante de moyenne nulle et de variance égale à un et $n(k)$ est un bruit blanc de moyenne nulle. Les simulations ont été réalisées pour le cas sans bruit puis en présence de bruit avec différent rapports signal sur bruit (SNR=34 dB et SNR=25dB).

Application avec les moindres carrés récursifs

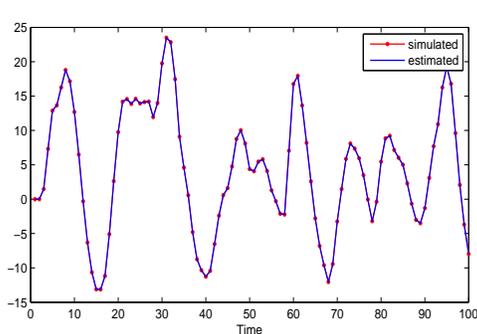
La méthode des moindres carrés récursif développée au chapitre 1, est utilisée pour identifier les paramètres du modèle HCAR utilisant le principe de sur-paramétrisation, les paramètres de RLSO sont initialisés comme suit :

$$P(0) = 10^{-6}I_{n_\theta} \text{ et } \hat{\theta} = 0 \text{ avec } n_\theta = 8$$

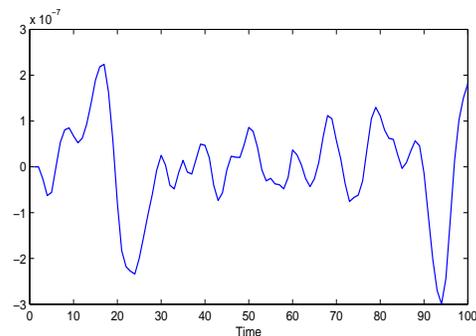
Le tableau 2.1 donne les résultats des simulations, pour différents SNR et les Figures 2.3(a) et 2.4(a) montrent une parfaite adéquation entre la sortie estimée et la sortie mesurée, respectivement pour le cas sans bruit et SNR=34 dB, tandis que les erreurs sont représentées en Figure 2.3(b) et en Figure 2.4(b) de même que les paramètres estimés sont très proches des paramètres exacts, pour le cas SNR=25dB les paramètres sont biaisés et la sortie estimée est comparée avec la sortie mesurée en Figure 2.5(a) et la Figure 2.5(b) représente l'erreur de prédiction.

TABLE 2.1: Résultats de la simulation avec RLSO pour différents SNR

SNR	a_1	a_2	b_1	b_2	p_2	p_3	J
<i>Sans bruit</i>	-1.600	0.800	1.000	0.650	0.500	0.250	$6.434e - 013$
<i>34dB</i>	-1.600	0.800	1.011	0.635	0.503	0.253	0.039
<i>25dB</i>	-1.465	0.674	0.864	1.055	0.483	0.242	590.938
<i>Valeurs exactes</i>	-1.600	0.800	1.000	0.650	0.500	0.25	-

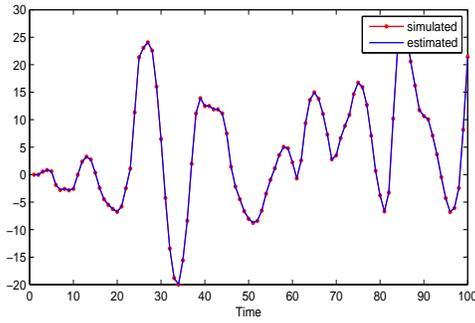


(a) Sortie estimée et sortie mesurée

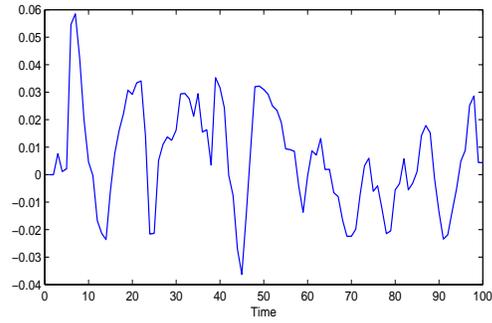


(b) Erreur de prédiction

FIGURE 2.3: Résultats de la simulation avec RLSO, cas sans bruit

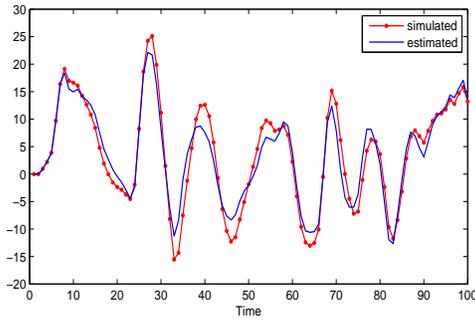


(a) Sortie estimée et sortie mesurée

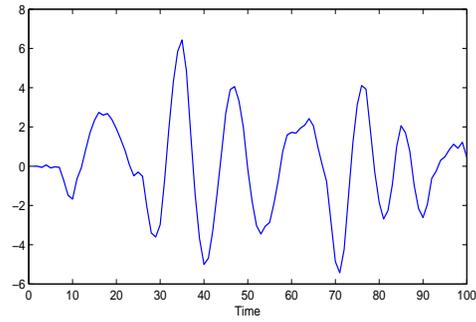


(b) erreur de prédiction

FIGURE 2.4: Résultats de la simulation avec RLSO, cas SNR=34dB



(a) Sortie estimée et sortie mesurée



(b) erreur de prédiction

FIGURE 2.5: Résultats de la simulation avec RLSO, cas SNR=25dB

Application avec PSO

La méthode évolutionnaire PSO est basée sur le principe de sur-paramétrisation PSOO (PSO Over-parameterization). L'algorithme PSOO a été exécuté pour une population de taille 100 et un nombre d'itérations $l_{max} = 300$. Les paramètres de l'algorithme PSO donnés au chapitre 1 sont choisis comme suit :

$\alpha_1=0.7$ et $\alpha_2 = (2/0.97725)\alpha_1$, et le coefficient d'inertie ω est calculé par l'équation suivante :

$$\omega = 0.9 - (0.55/l_{max})l \quad (2.24)$$

avec l l'itération courante et l_{max} le nombre maximal d'itérations.

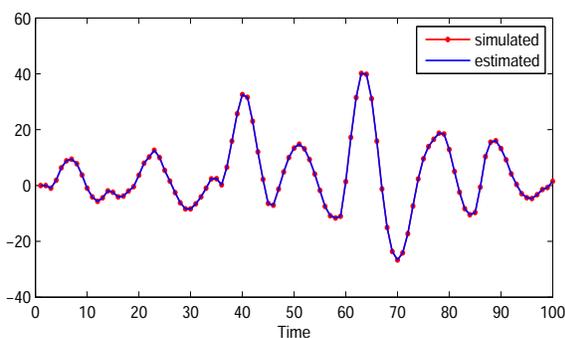
Le tableau (2.2) illustre les résultats de simulation, en utilisant la méthode PSOO, les résultats montrent que dans le cas sans bruit, les paramètres estimés sont les valeurs exactes des paramètres du système avec ($J \simeq 10^{-19}$), et la valeur du critère quadratique

J augmente avec l'augmentation du bruit. L'erreur de prédiction pour le cas sans bruit,

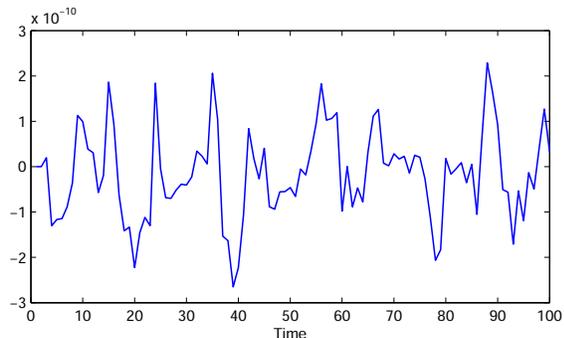
TABLE 2.2: Résultats de la simulation avec PSO, cas sans bruit et bruité

SNR	a_1	a_2	b_1	b_2	p_2	p_3	J
<i>Sans bruit</i>	-1.600	0.800	1.000	0.650	0.500	0.250	$9.989e - 19$
<i>34dB</i>	-1.600	0.800	1.001	0.648	0.500	0.250	0.0028
<i>25dB</i>	-1.603	0.801	1.093	0.577	0.490	0.242	5.688
<i>Valeurs exactes</i>	-1.600	0.800	1.000	0.650	0.500	0.250	-

représenté en Figure (2.6(b)) est nulle (\approx de 10^{-10}); ainsi, la Figure 2.6(a) montre une adéquation parfaite entre la sortie estimée et la sortie mesurée. En présence de bruit $SNR = 34dB$, et $SNR = 25dB$, la sortie du modèle estimée est comparée aux données respectivement en Figure 2.7(a) et en Figure 2.8(a), elles montrent une adéquation parfaite, alors que les erreurs de prédictions sont représentées en Figure 2.7(b) et Figure 2.8(b).



(a) Sortie estimée et sortie mesurée



(b) erreurs de prédiction

FIGURE 2.6: Résultats de la simulation avec PSO, cas sans bruit

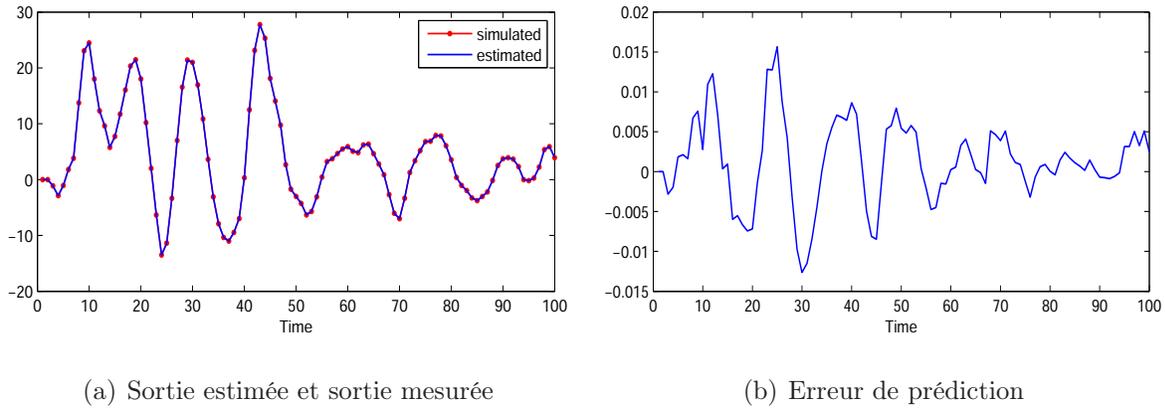


FIGURE 2.7: Résultats de la simulation avec PSO, cas SNR=34dB

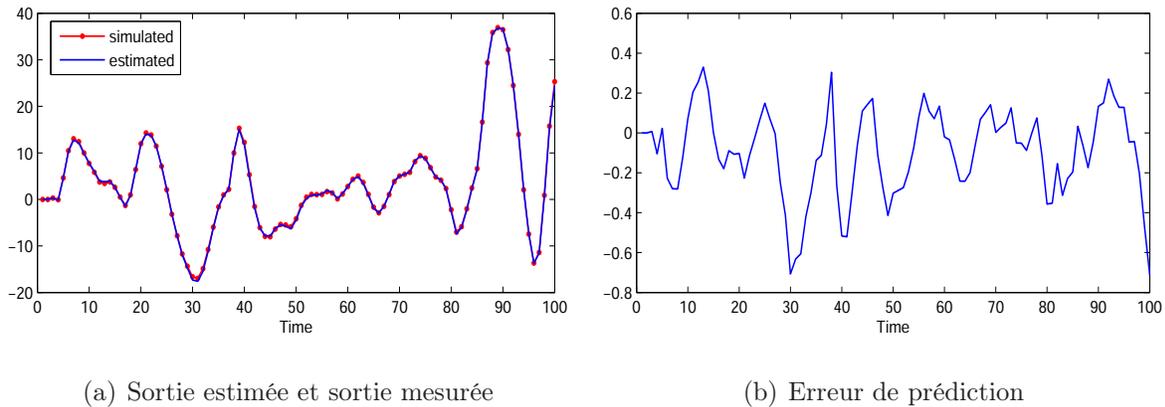


FIGURE 2.8: Résultats de la simulation avec PSO, cas SNR=25dB

La comparaison des résultats obtenus en appliquant le principe de sur-paramétrisation combiné soit avec la méthode RLS soit avec PSO montrent que PSO estime les paramètres du système avec une meilleure précision que RLSO surtout en présence de bruit. Ce qui confirme que PSO est un approximateur universel pour les systèmes non-linéaires. L'inconvénient de ce principe est que, la dimension du vecteur θ à estimer est couramment très grande, qui nécessite un volume de calcul important, afin de résoudre ce problème, le principe d'identification hiérarchique est habituellement adopté. Il transforme un système à grande dimension en plusieurs sous-systèmes, avec de petites dimensions, plus faciles à identifier.

2.6.2 Principe Hiérarchique

Afin d'illustrer les performances de l'algorithme PSO basé sur le principe hiérarchique (PSOH), l'identification est appliquée à deux modèles de Hammerstein avec différente complexité. Dans le premier exemple la partie linéaire est d'ordre 2 ($n_a = n_b = 2$) et la partie non linéaire est d'ordre 2 ($r = 2$) et dans le deuxième exemple $n_a = n_b = 2$, et $r = 3$. L'entrée est une excitation persistante de moyenne nulle et de variance égal à un et $n(k)$ est un bruit blanc de moyenne nulle. L'algorithme de PSOH a été exécuté pour une taille d'essaim 100 et un certain nombres d'itérations $l_{max} = 300$. Les simulations ont été effectuées pour le cas sans-bruit puis pour différent rapports signal sur bruit $SNR = 34dB$ et $SNR = 25dB$.

Exemple 1

Le système est décrit par les equations suivantes :

$$\begin{aligned}
 A(z)y(k) &= B(z) \tilde{u}(k) + n(k) \\
 A(z) &= 1 + a_1z^{-1} + a_2z^{-2} = 1 - 1.00z^{-1} + 0.50z^{-2} \\
 B(z) &= b_1z^{-1} + b_2z^{-2} = z^{-1} + 0.30z^{-2} \\
 f(u(k)) &= p_1u(k) + p_2u^2(k) = u(k) + 0.25u^2(k)
 \end{aligned} \tag{2.25}$$

Le tableau 2.3 donne les résultats de simulation pour $SNR = 34 dB$ et $SNR = 25 dB$ et pour le cas sans bruit. Les résultats obtenus montrent que l'erreur est très faible.

TABLE 2.3: Résultats de la simulation avec PSOH exemple 1, pour différents SNR

SNR	a_1	a_2	b_1	b_2	p_2	J
<i>Sans bruit</i>	-1.000	0.500	1.000	0.300	0.250	$5.011e - 030$
<i>34dB</i>	-0.999	0.499	1.000	0.300	0.249	$6.340e - 006$
<i>25dB</i>	-1.002	0.490	0.982	0.283	0.264	0.141
<i>Valeurs exactes</i>	-1.000	0.500	1.00	0.300	0.250	-

L'erreur de prédiction pour le cas sans bruit, représenté en Figure 2.9(b) est nulle (\approx

de 10^{-15}); alors que, la sortie estimée coïncide avec les données d'identification en Figure 2.9(a).

Dans le cas de données bruitées, avec $SNR = 34dB$, et $SNR = 25dB$, la sortie estimée est comparée avec les données d'identification respectivement en Figure 2.10(a) et Figure 2.11(a). Ces résultats illustrent une adéquation parfaite.

L'erreur de prédiction représentée en Figure 2.10(b) pour un $SNR = 34 dB$, est très faible (\approx de 10^{-6}) et la Figure 2.11(b) représente l'erreur pour un $SNR = 25 dB$.

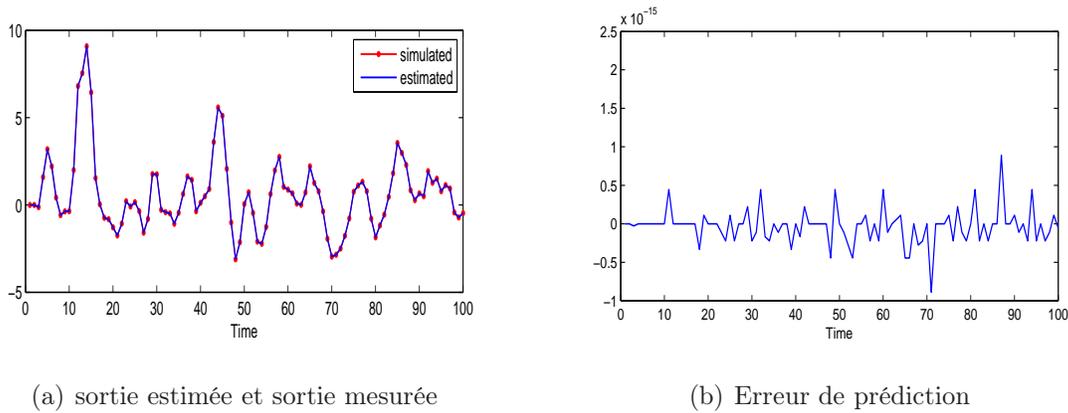


FIGURE 2.9: Résultats de la simulation avec PSOH exemple 1, cas sans bruit

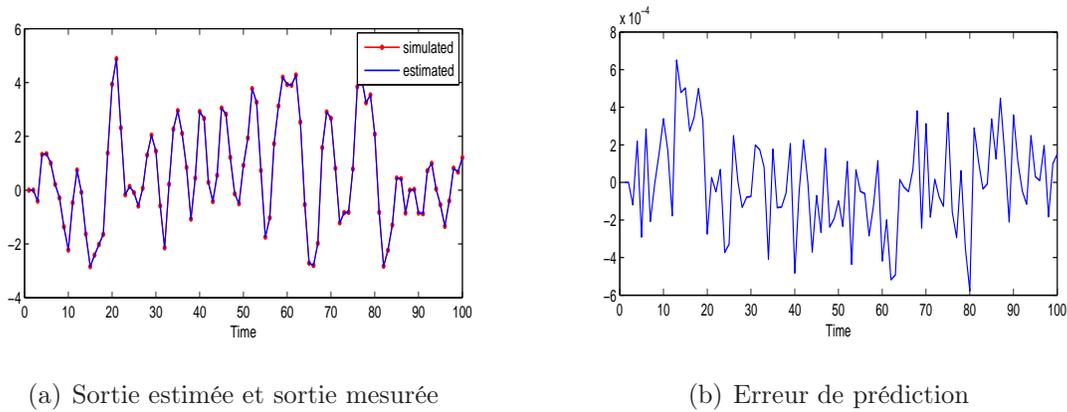


FIGURE 2.10: Résultats de la simulation avec PSOH exemple 1, cas $SNR=34dB$

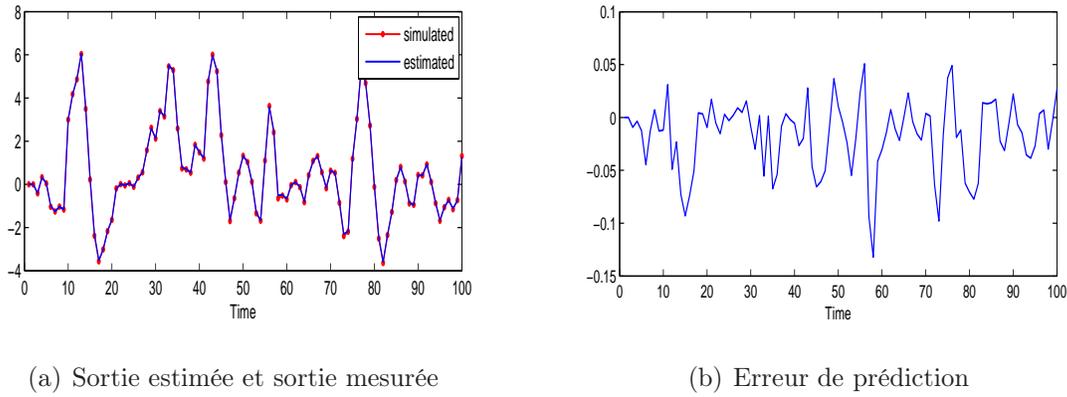


FIGURE 2.11: Résultats de la simulation avec PSOH exemple 1, cas SNR=25dB

Dans l'exemple 2, la méthode PSOH est appliquée à l'exemple de la section 2.6.1, étudié avec la méthode PSOO et une comparaison sera discutée.

Exemple 2

Les polynômes $A(z)$ et $B(z)$ de la partie linéaire sont donnés comme suit :

$$\begin{aligned} A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} = 1 - 1.60z^{-1} + 0.80z^{-2} \\ B(z) &= b_1 z^{-1} + b_2 z^{-2} = z^{-1} + 0.65z^{-2} \end{aligned} \quad (2.26)$$

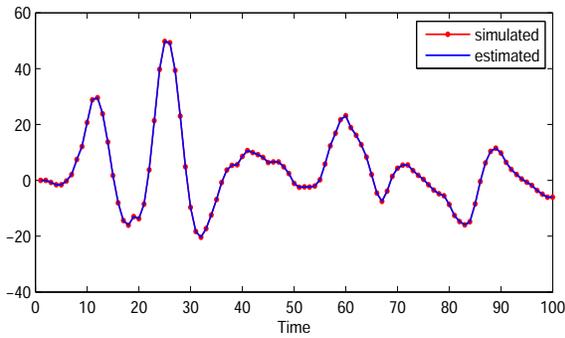
La partie non-linéaire est un polynôme d'ordre 3

$$\begin{aligned} f(u(k)) &= p_1 u(k) + p_2 u^2(k) + p_3 u^3(k) \\ &= u(k) + 0.50u^2(k) + 0.25u^3(k) \end{aligned} \quad (2.27)$$

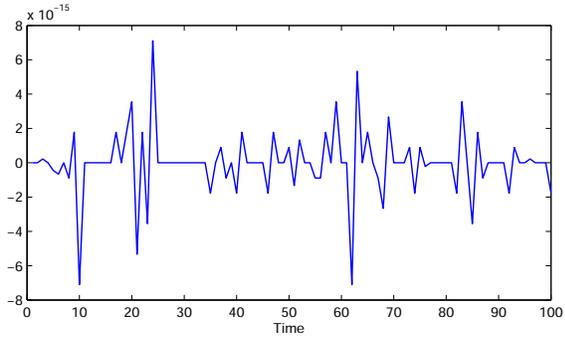
Le Tableau (2.4) donne les résultats de simulation en utilisant l'algorithme PSOH. On remarque une bonne estimation du système même pour un niveau élevé de bruit . Pour le cas sans bruit, les figures relatives sont représentées en Figure 2.12(a) et Figure 2.12(b). Pour $SNR = 34dB$ et $SNR = 25dB$ les sorties sont données respectivement en Figure 2.13(a) et Figure 2.14(a), tandis que les erreurs sont représentées en Figure 2.13(b) et Figure 2.14(b).

TABLE 2.4: Résultats de la simulation avec PSOH exemple 2, pour différents SNR

SNR	a_1	a_2	b_1	b_2	p_2	p_3	J
<i>Sans bruit</i>	-1.600	0.800	1.00	0.650	0.500	0.250	$3.491e - 28$
<i>34dB</i>	-1.600	0.800	1.000	0.651	0.499	0.249	$4.216e - 04$
<i>25dB</i>	1.593	0.792	1.010	0.582	0.506	0.275	2.026
<i>Valeurs exactes</i>	-1.600	0.800	1.000	0.650	0.500	0.250	-

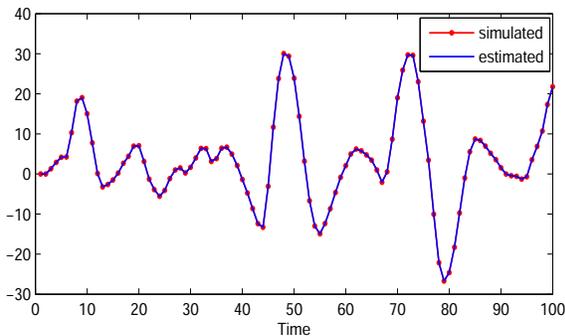


(a) Sortie estimée et sortie mesurée

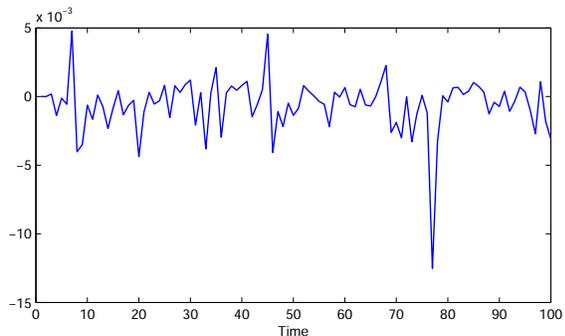


(b) Erreur de prédiction

FIGURE 2.12: Résultats de la simulation avec PSOH exemple 2, cas sans bruit



(a) Sortie estimée et sortie mesurée



(b) Erreur de prédiction

FIGURE 2.13: Résultats de la simulation avec PSOH exemple 2, cas SNR=34dB

La méthode PSOH estime les paramètres de la partie non-linéaire plus précisément et converge plus rapidement que la méthode PSOO, de plus elle donne des résultats satisfaisants même en présence d'un haut niveau de bruit. Ceci confirme le fait que PSO peut estimer n'importe quel système non-linéaire avec une bonne exactitude et le principe hiérarchique montre son efficacité.

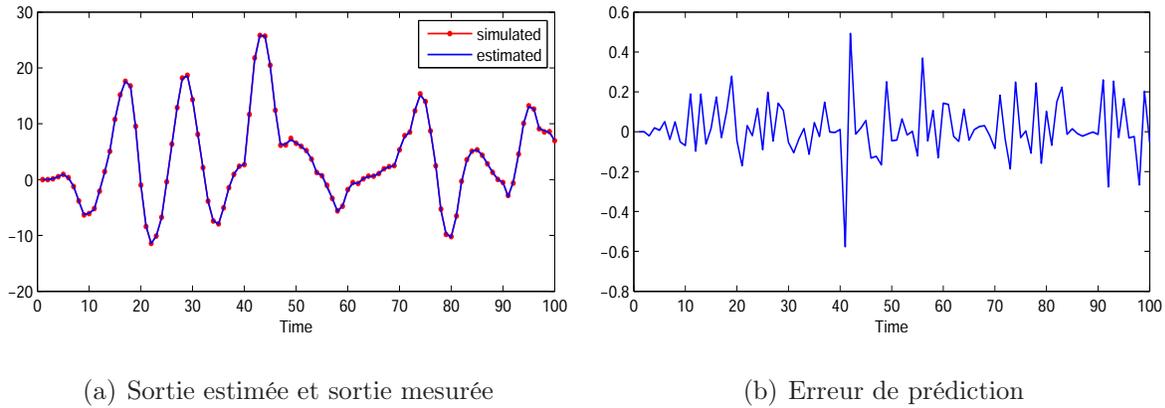


FIGURE 2.14: Résultats de la simulation avec PSOH exemple 2, cas SNR=25dB

2.7 Conclusion

Dans ce chapitre, nous avons abordé le problème d'identification des systèmes non-linéaires Hammerstein de type CAR. Deux principes ont été considérés : celui de type sur-paramétrisation et celui de type hiérarchique.

Dans la première étape, le principe de sur-paramétrisation est considéré, à cet effet on a estimé les paramètres de la partie linéaire et le produit inconnu de paramètres avec l'algorithme RLSO puis avec PSOO. Les simulations effectuées, illustrent les performances de l'algorithme PSO.

Cependant, le principe de sur-paramétrisation présente l'inconvénient de la redondance des paramètres, ce qui provoque un temps de calcul important ; plus précisément ; avec le principe de sur-paramétrisation le nombre de paramètres à estimer est : $n_{\theta} = n_a + n_b r$ qui est plus grand que $n_{\theta} = n_a + n_b + r$ du système réel pour $n_b, r > 2$. Pour y remédier à cet inconvénient le principe hiérarchique est considéré dans la deuxième étape. La contribution principale de ce chapitre est d'associer le principe hiérarchique à l'algorithme PSO (PSOH).

Les simulations ont permis de mettre en évidence la capacité de la méthode PSOH même en présence de bruit.

Récemment les modèles Polynomial Non Linear State Space model (PNLSS) ont été utilisés pour représenter les systèmes non-linéaires. Ils reposent sur un modèle d'état qui permet de modéliser ainsi bien les systèmes monovariables que multivariables.

Le chapitre 3 présente le modèle PNLSS appliqué au modèle Hammerstein et une

méthode d'identification basée sur l'algorithme de Levenberg Marquardt sera développée.

Chapitre 3

Identification des modèles PNLSS

3.1 Introduction

Dans ce chapitre, nous nous intéressons aux modèles Hammerstein de type modèle d'état non-linéaire polynomial ou "Polynomial Non Linear State Space model (PNLSS)" en Anglais [59]. Le principal avantage de ce dernier est sa capacité de décrire un système complexe avec un nombre réduit de paramètres. Le PNLSS prend en compte la partie linéaire et la partie non-linéaire. L'identification est réalisée par une approche à erreur de sortie associée à l'algorithme d'optimisation non-linéaire Levenberg Marquardt.

3.2 Modèles d'état non-linéaires polynomiaux PNLSS

On rappelle ici le modèle d'état non-linéaire généralisé à temps discret avec n_u entrées, n_y sorties, n_a états :

$$\begin{cases} x(k+1) = f(x(k), u(k), \theta) \\ y(k) = h(x(k), u(k), \theta) \end{cases} \quad (3.1)$$

avec : $u \in \mathbb{R}^{n_u}$; $y \in \mathbb{R}^{n_y}$; $x \in \mathbb{R}^{n_a}$. Les fonctions f et h dépendent du vecteur des paramètres $\theta \in \mathbb{R}^{n_\theta}$.

Dans ce mémoire, nous considérons les non-linéarités de type polynôme, dans ce cas le modèle (3.1) est appelé *Polynomial Nonlinear State-space Model* (PNLSS) [59], il est

défini comme suit [7], [2], [60] :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases} \quad (3.2)$$

Les coefficients du sous-système linéaire sont donnés par les matrices A et B de l'équation d'état, et C et D de l'équation de sortie. Les vecteurs $\zeta(k)$ et $\eta(k)$ contiennent des monômes non-linéaires en $x(k)$ et $u(k)$ de degré deux à un degré choisi r (avec $r \geq 2$). Les coefficients associés à ces termes non-linéaires sont donnés par les matrices E et F , où les monômes de degré un sont inclus dans la partie linéaire de la structure du modèle PNLSS.

Le modèle PNLSS a l'avantage de décrire une très grande classe de systèmes non-linéaires, dans [7], [63], [2], les auteurs ont montré que certains modèles non-linéaires à structure bloc (Wiener, Hammerstein, Wiener-Hammerstein) peuvent être décrits par ce modèle.

Dans ce travail, le système Hammerstein est considéré, il consiste en un bloc non-linéaire statique suivi d'un bloc linéaire dynamique de type modèle d'état, comme il est représenté en figure 3.1.

avec $\tilde{u}(k)$ la sortie du bloc non-linéaire (l'entrée du bloc linéaire). Le bloc dynamique

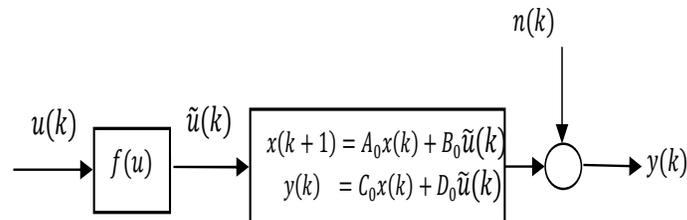


FIGURE 3.1: Système Hammerstein

linéaire du modèle Hammerstein, en absence du bruit $n(k)$ est décrit par le modèle d'état suivant : (A_0, B_0, C_0, D_0)

$$\begin{cases} x(k+1) = A_0 x(k) + B_0 \tilde{u}(k) \\ y(k) = C_0 x(k) + D_0 \tilde{u}(k) \end{cases} \quad (3.3)$$

avec $y(k)$ la sortie du système, le signal intermédiaire $\tilde{u}(k)$ est la sortie du bloc non-linéaire

Le bloc non-linéaire est décrit par un polynôme non-linéaire d'ordre connu r avec les coefficients (p_1, p_2, \dots, p_r) [61][41][62]

$$\tilde{u}(k) = f(u(k)) = \sum_{i=1}^r p_i u^i(k) \quad (3.4)$$

La substitution de l'équation (3.4) dans (3.3) donne lieu au modèle PNLSS :

$$\begin{cases} x(k+1) = A_0 x(k) + B_0 \sum_{i=1}^r p_i u^i(k) \\ y(k) = C_0 x(k) + D_0 \sum_{i=1}^r p_i u^i(k) \end{cases} \quad (3.5)$$

Le nouveau modèle d'état(3.2) peut être déduit avec des matrices définies comme suit :

$$A = A_0 \quad B = p_1 B_0 \quad C = C_0 \quad D = p_1 D_0 \quad (3.6)$$

avec p_1 le coefficient associé au monôme $u(k)$ du degré un. Les vecteurs $\zeta(k)$ et $\eta(k)$ dans ce cas (un modèle Hammerstein) sont égaux, et contiennent le monôme dans $u(k)$ de degré 2 à r , qui peuvent être écrits sous la forme : $\zeta(k) = \eta(k) = u(k)_{\{r\}}$

Où

$$u(k)_{\{r\}} = [u^2(k) \quad u^3(k) \quad \dots \quad u^r(k)]$$

Les coefficients associés à ces termes non linéaires sont donnés par les matrices E et F comme suit :

$$\begin{aligned} E &= \begin{bmatrix} p_2 B_0 & p_3 B_0 & \dots & p_r B_0 \end{bmatrix} \\ F &= \begin{bmatrix} p_2 D_0 & p_3 D_0 & \dots & p_r D_0 \end{bmatrix} \end{aligned} \quad (3.7)$$

Sans perte de généralité, le premier coefficient de $f(\cdot)$ peut être choisi égal à 1, $p_1 = 1$ [8],[45].

3.3 Identification des systèmes PNLSS

L'objectif de l'identification du modèle PNLSS (3.2) consiste à estimer les coefficients des matrices A, B, E, C, D, F . Le vecteur θ à estimer est donné par :

$$\theta = [A, \quad B, \quad C, \quad D, \quad E, \quad F] \quad (3.8)$$

Pour réduire le nombre de paramètres à estimer, sans perte de généralité, la forme canonique commandable peut être considérée.

La partie linéaire (3.5) est donnée comme suit :

$$\left\{ \begin{array}{l} x(k+1) = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ -a_1 & -a_2 & \dots & -a_{n_a-1} & -a_{n_a} \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tilde{u}(k) \\ y(k) = \begin{bmatrix} c_1 & c_2 & \dots & c_{n_a} \end{bmatrix} x(k) + [d] \tilde{u}(k) \end{array} \right. \quad (3.9)$$

Dans ce cas, le vecteur des paramètres θ à estimer réduit à :

$$\theta = [a_1 \dots a_n \quad p_2 \dots p_{r-1} \quad c_1 \dots c_{n_a} \quad d] \quad (3.10)$$

et les coefficients des matrices E et F peuvent être déduits; le nombre de paramètres à estimer dans le cas de la forme commandable est : $n_\theta = 2n_a + r$.

Soit $\tilde{\theta}$ l'estimation du vecteur de paramètres exact θ , la valeur optimale de $\tilde{\theta}$ est obtenue en minimisant le critère quadratique :

$$J = \sum_{k=1}^K \varepsilon^2(k) \text{ avec } \varepsilon(k) = y(k) - \hat{y}(k) \text{ le résidu.}$$

L'algorithme de Marquardt est utilisé pour estimer θ itérativement [38].

$$\left\{ \begin{array}{l} \theta^{(i+1)} = \theta^{(i)} - \left\{ [J''_{\theta\theta} + \lambda I]^{-1} J'_\theta \right\}_{\hat{\theta}=\theta^{(i)}} \\ J'_\theta = -2 \left(\sum_{k=1}^K \varepsilon(k) \sigma_{k/\theta} \right) \text{ le Gradient} \\ J''_\theta = -2 \left(\sum_{k=1}^K \sigma_{k/\theta} \sigma_{k/\theta}^T \right) \text{ le Hessien} \\ \sigma_{k/\theta} = \frac{\partial \hat{y}(k, \theta)}{\partial \theta} \text{ Fonction de sensibilité de la sortie par rapport à } \theta \end{array} \right. \quad (3.11)$$

L'optimisation non linéaire est basée sur le calcul des fonctions de sensibilité σ , le développement de ces dernières est présenté dans le paragraphe suivant.

Calcul des fonctions de sensibilité

La différenciation de l'équation d'état de (3.2) par rapport à chaque composante θ_j de $\tilde{\theta}$ avec $j = 1, \dots, n_\theta$ permet d'obtenir le modèle des fonctions de sensibilité paramétriques

tel que :

$$\left\{ \begin{array}{l} \frac{\partial x(k+1)}{\partial \theta_j} = \frac{\partial A}{\partial \theta_j} x(k) + A \frac{\partial x(k)}{\partial \theta_j} + \frac{\partial B}{\partial \theta_j} u(k) + B \frac{\partial u(k)}{\partial \theta_j} + E \frac{\partial \zeta(k)}{\partial \theta_j} + \frac{\partial E}{\partial \theta_j} \zeta(k) \\ \frac{\partial y(k)}{\partial \theta_j} = \frac{\partial C}{\partial \theta_j} x(k) + C \frac{\partial x(k)}{\partial \theta_j} + \frac{\partial D}{\partial \theta_j} u(k) + D \frac{\partial u(k)}{\partial \theta_j} + F \frac{\partial \eta(k)}{\partial \theta_j} + \frac{\partial F}{\partial \theta_j} \eta(k) \end{array} \right. \quad (3.12)$$

$$\left\{ \begin{array}{l} \sigma_{x(k+1)/\theta_j} = A \sigma_{x(k)/\theta_j} + \begin{bmatrix} \frac{\partial A}{\partial \theta_j} & \frac{\partial B}{\partial \theta_j} & \frac{\partial E}{\partial \theta_j} \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \\ \zeta(k) \end{bmatrix} \\ \sigma_{y(k)/\theta_j} = C \sigma_{x(k)/\theta_j} + \begin{bmatrix} \frac{\partial C}{\partial \theta_j} & \frac{\partial D}{\partial \theta_j} & \frac{\partial F}{\partial \theta_j} \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \\ \zeta(k) \end{bmatrix} \end{array} \right. \quad (3.13)$$

3.4 Exemples de simulation

La méthode d'identification présentée dans ce chapitre est appliquée à deux modèles Hammerstein de type PNLSS avec différentes complexités. Dans le premier exemple la partie linéaire est d'ordre deux ($n_a = 2$) et la partie non linéaire est d'ordre deux ($r = 2$), dans le deuxième exemple la partie linéaire est d'ordre deux ($n_a = 2$) et la partie non linéaire est d'ordre trois ($r = 3$).

L'entrée est une excitation persistante de moyenne nulle et de variance égal à un et $n(k)$ est un bruit blanc de moyenne nulle.

Les simulations ont été effectuées pour le cas sans-bruit puis pour différents rapports signal sur bruit $SNR = 34 \text{ dB}$ et $SNR = 25 \text{ dB}$.

3.4.1 Exemple 1

Le système est décrit par les équations PNLSS suivantes :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases} \quad (3.14)$$

Avec

$$A = \begin{bmatrix} 0 & 1 \\ -0.1 & -0.4 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.1 & 0.5 \end{bmatrix} \quad D = \begin{bmatrix} 0.1 \end{bmatrix}$$

Le sous système non-linéaire est décrit par le polynôme suivant :

$$f(u(k)) = u(k) + 0.25u^2(k)$$

Le tableau (3.4.1) illustre les résultats des simulations de l'exemple 1 pour différents SNR et pour le cas sans bruit. Les résultats obtenus montrent que dans le cas sans bruit les paramètres estimés sont les valeurs exactes des paramètres du système.

L'erreur de prédiction pour le cas sans bruit, représentée en figure 3.2(b) est nulle ($\approx 10^{-16}$), ainsi la sortie du système estimée et mesurée en figure 3.2(a) montre une parfaite adéquation.

En présence du bruit $SNR = 34dB$, et $SNR = 25dB$, la sortie du modèle estimée est comparée aux données respectivement en figure 3.3(a) et en figure 3.4(a), elles montrent une parfaite adéquation. L'erreur de prédiction pour $SNR = 34dB$ est représentée en figure 3.3(b), elle est de l'ordre de 10^{-16} et la figure 3.4(b) représente l'erreur pour $SNR = 25dB$.

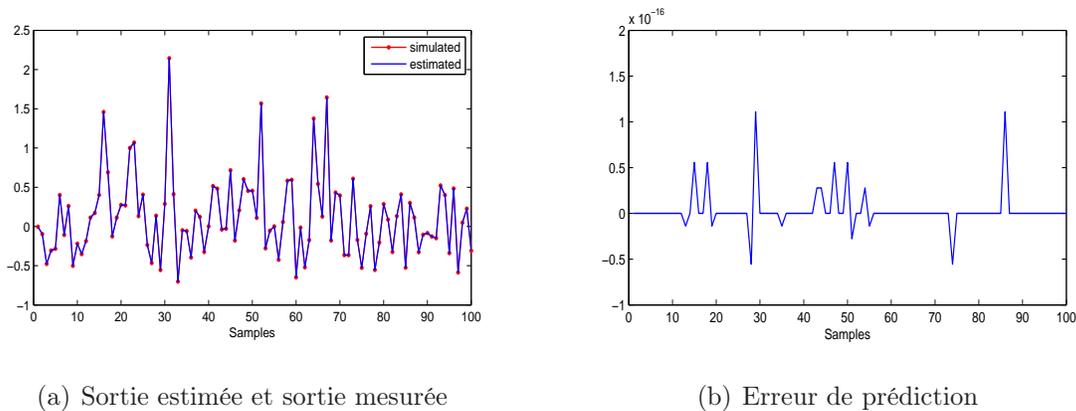
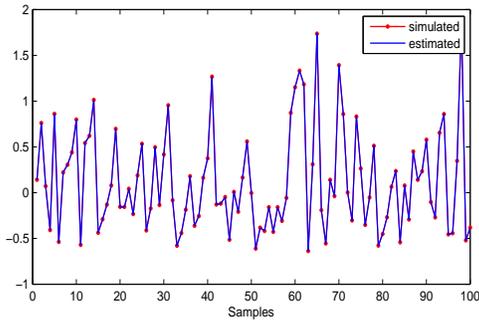


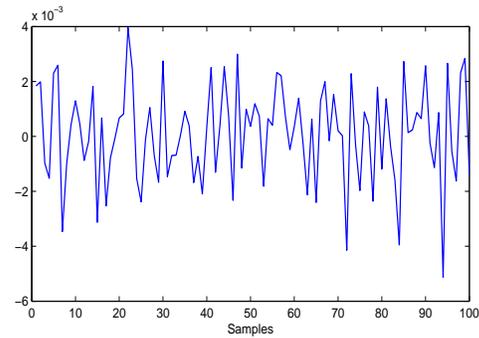
FIGURE 3.2: Résultats de l'exemple 1 cas sans bruit

TABLE 3.1: Résultats des simulations de l'exemple 1, cas sans bruit et bruité

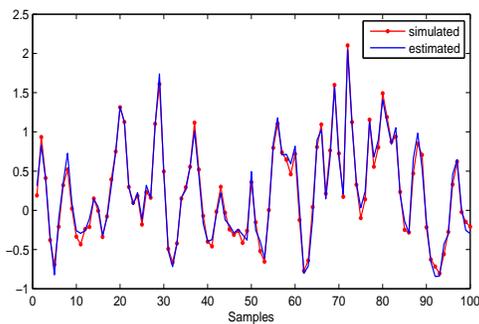
SNR	<i>Sans – bruit</i>	$34dB$	$25dB$	<i>Valeurs exactes</i>
a_1	-0.100	-0.101	-0.0823	-0.100
a_2	-0.400	-0.400	0.364	-0.400
p_2	0.250	0.25	0.241	0.250
c_1	0.100	0.100	0.126	0.100
c_2	0.500	0.499	0.502	0.500
d	0.100	0.100	0.101	0.100
J	$4.699e - 032$	$3.100e - 04$	0.695	—



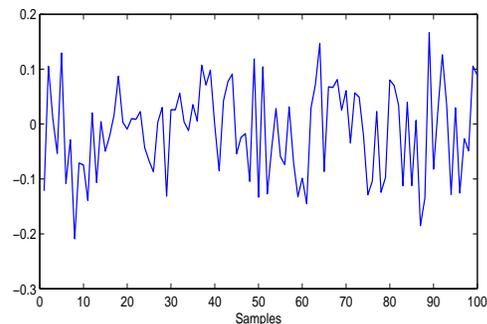
(a) Sortie estimée et sortie mesurée



(b) Erreur de prédiction

FIGURE 3.3: Résultats de l'exemple 1 avec $SNR=34$ dB

(a) Sortie estimée et sortie mesurée



(b) Erreur de prédiction

FIGURE 3.4: Résultats de l'exemple 1 avec $SNR=25$ dB

3.4.2 Exemple 2

Dans ce cas, $n_a = 2$, et $r = 3$. Le système PNLSS est donné par :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases} \quad (3.15)$$

Avec

$$A = \begin{bmatrix} 0 & 1 \\ -0.1 & -0.3 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ C = \begin{bmatrix} 0.25 & 0.45 \end{bmatrix} \quad D = \begin{bmatrix} 0.1 \end{bmatrix}$$

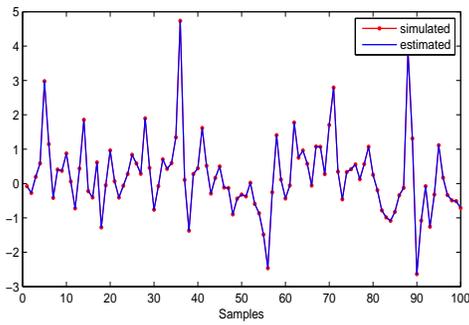
Le sous-système non-linéaire est décrit par le polynôme du troisième degré suivant :

$$f(u(k)) = u(k) + 0.35u^2(k) + 0.4u^3(k)$$

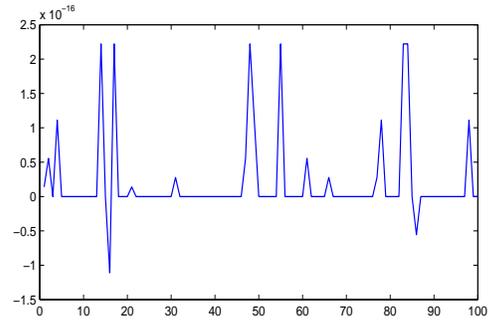
Les simulations sont effectuées en absence du bruit et pour des données bruitées avec $SNR = 34dB$ et $SNR = 25dB$. Les résultats du tableau (3.4.2) sont très proches des paramètres exacts. Les figures 3.5(a), 3.6(a), et 3.7(a) montrent que les sorties mesurées et estimées sont très proches.

TABLE 3.2: Résultats des simulations de l'exemple 2, cas sans bruit et bruité

SNR	<i>Sans – bruit</i>	$34dB$	$25dB$	<i>Valeurs exactes</i>
a_1	-0.100	-0.099	-0.104	-0.100
a_2	-0.300	-0.300	-0.304	-0.300
p_2	0.350	0.349	0.334	0.350
p_3	0.400	0.398	0.369	0.400
c_1	0.250	0.251	0.267	0.250
c_2	0.450	0.450	0.480	0.450
d	0.100	0.100	0.103	0.100
J	$3.724e - 031$	$8.942e - 04$	1.673	–

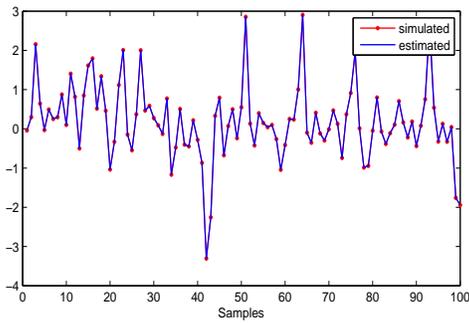


(a) Sortie estimée et sortie mesurée

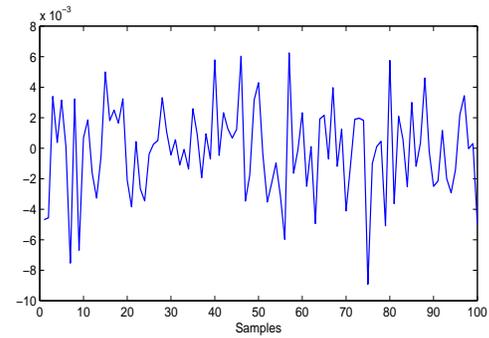


(b) Erreur de prédiction

FIGURE 3.5: Résultats de l'exemple 2 pour le cas sans bruit

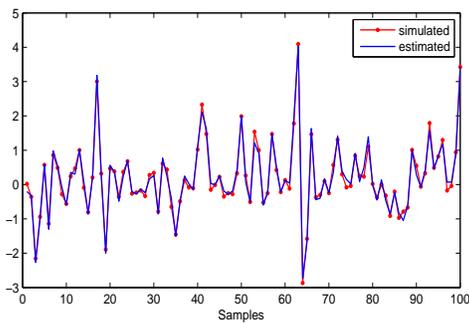


(a) Sortie estimée et sortie mesurée

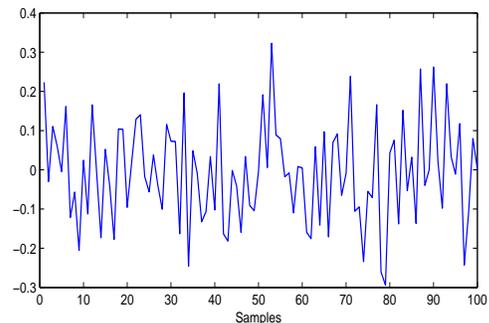


(b) Erreur de prédiction

FIGURE 3.6: Résultats de l'exemple 2 avec SNR=34 dB



(a) Sortie estimée et sortie mesurée



(b) Erreur de prédiction

FIGURE 3.7: Résultats de l'exemple 2 avec SNR=25 dB

3.5 Conclusion

Dans ce chapitre, nous avons abordé le problème d'identification des systèmes non-linéaires Hammerstein de type PNLSS. Le modèle est obtenu à partir d'un modèle d'état

linéaire montée en série avec une partie non-linéaire de type polynôme. L'objectif d'identification était d'estimer les paramètres linéaires et non-linéaires du système PNLSS. Nous avons utilisé une approche à erreur de sortie. Enfin, nous avons illustré par voie de simulation l'efficacité de la méthode de Levenberg Marquard qui permet d'identifier le système PNLSS même en présence d'un pourcentage important de bruit. Les résultats présentés dans ce chapitre peuvent être étendus au cas du système PNLSS d'ordre fractionnaire.

Chapitre 4

Identification du modèle Hammerstein fractionnaire

4.1 Introduction

Les systèmes fractionnaires ont suscité une grande attention ces dernières décennies en raison de leur flexibilité à permettre une modélisation plus précise des systèmes complexes dans divers domaines des sciences et de la technologie.

L'idée du calcul fractionnaire remonte à 1695, dans une lettre que l'Hôpital a écrit à Leibniz, s'interrogeant sur le concept de la différenciation de l'ordre non entier. Depuis, plusieurs mathématiciens célèbres tels que Euler, Laplace, Fourier, Abel et Laurent ont travaillé dans l'idée des opérateurs différentiels fractionnaires. Néanmoins, ce n'est que seulement plus tard au dix-neuvième siècle et dûe principalement aux contributions de Liouville, de Grünwald, de Letnikov et de Riemann qu'une théorie complète appropriée a été formalisée.

Le calcul fractionnaire présente l'avantage de décrire plusieurs phénomènes physiques, par exemple, les phénomènes de la diffusion de la chaleur [10], la diffusion de matières par l'eau dans le sol [64]. Il est aussi fortement utilisé dans l'industrie comme : Les processus chimiques [65], le traitement de signal [66], La viscoélasticité [67], la théorie de chaos [68].

Dans le chapitre précédent nous nous sommes intéressés à l'identification des systèmes non-linéaires Hammerstein PNLSS à temps discret dont la partie linéaire est supposée de type entier. Nous avons constaté que les résultats de l'identification des paramètres de modèle Hammerstein sont très satisfaisants. Par ailleurs, nous nous sommes posés la question suivante : Que seront les résultats de l'identification lorsque la partie linéaire n'est pas entière mais fractionnaire ?

Dans ce chapitre, nous essayons de répondre à cette question, surtout nous considérons le cas où la partie linéaire est un modèle d'état d'ordre fractionnaire et la partie non-linéaire est un polynôme ce qui aboutit à un modèle PNLSS fractionnaire.

Dans ce chapitre, nous rappellerons les éléments de base du calcul fractionnaire et la simulation d'un modèle d'état fractionnaire à temps discret. L'extension du modèle Hammerstein PNLSS au cas fractionnaire est décrite, son identification basée sur une méthode à erreur de sortie est développée. La fin de ce chapitre est consacrée à des exemples d'identification illustrant l'efficacité de la méthode implémentée.

4.2 Calcul fractionnaire

4.2.1 Définition de Grünwald-Letnikov

Le calcul d'ordre fractionnaire a une grande histoire, et plusieurs mathématiciens ont contribué à des degrés divers à son développement au fil des siècles. Son développement pour les systèmes dynamiques est basé sur la généralisation des opérateurs intégration et de différentiation en un seul opérateur fondamental D_t^α , où α est un nombre non entier (reel...).

$$D_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & \text{pour } \alpha > 0 \\ 1 & \text{pour } \alpha = 0 \\ I_t^\alpha & \text{pour } \alpha < 0 \end{cases} \quad (4.1)$$

Les notations D_t^α et I_t^α représentent l'opérateur de la dérivation et d'intégration.

Les dérivées et les intégrales fractionnaires apparaissent dans le contrôle des systèmes

dynamiques ou le contrôleur est décrit par une equation différentielle fractionnaire.

Bien qu'il existe plusieurs définitions de généralisation de la dérivée d'ordre entier à un ordre non entier, les définitions les plus utilisées sont : les définitions de Riemann-Liouville, Grünwald- Letnikov, et Caputo.

Dans ce chapitre, on s'intéresse à la définition de Grünwald- Letnikov à temps discret.

L'idée de la dérivée au sens de Grünwald, est de généraliser la définition de la dérivée entière d'une fonction temporelle x à des dérivées d'ordre réel $\alpha \in \mathbb{R}_+^*$.

Soit $t = kh$ le temps d'échantillonnage pour $k = 0, 1, 2, \dots$, avec h est la période d'échantillonnage. Dans ce chapitre, nous supposons, sans perte de généralité, que $h = 1$.

En effet la différence finie d'ordre 1 est donnée comme suit :

$$\Delta^1 x(k+1) = x(k+1) - x(k) \quad (4.2)$$

L'équation (4.2) représente l'approximation discrète d'Euler de la dérivée d'ordre entier $\frac{dx(t)}{dt}$.

La généralisation à un ordre $\alpha \in \mathbb{R}_+^*$ de la dérivée entière est donnée [72] :

$$\Delta^\alpha x(kh) = \frac{1}{h^\alpha} \sum_{j=0}^k (-1)^j \binom{\alpha}{j} x((k-j)h) \quad (4.3)$$

Avec :

Δ l'opérateur de dérivation fractionnaire discret avec le temps initial égal à zero.

$\binom{\alpha}{j}$ est le binôme de Newton généralisé à des nombres réels :

$$\binom{\alpha}{j} = \begin{cases} 1 & \text{pour } j = 0 \\ \frac{\alpha(\alpha-1)\dots(\alpha-j+1)}{j!} & \text{pour } j > 0 \end{cases} \quad (4.4)$$

Pour la dérivée d'ordre 1, mis à part $j = 0$ et $j = 1$, les coefficients de la pondération $(-1)^j \binom{\alpha}{j}$ sont nuls. La dérivée entière fournit donc une caractérisation locale de la fonction. En revanche, pour des ordres α non entiers, les coefficients de pondération ne s'annulent pas. La valeur à chaque instant est alors une combinaison linéaire de toutes les valeurs de la fonction $x((k-j)h)$, $j = 0, 1, \dots, k$. La dérivée non entière d'une fonction à un instant t prend en compte les valeurs de cette fonction à tous les temps passés, donnant ainsi une

caractérisation globale de cette fonction. Par conséquent, les systèmes non entiers sont souvent assimilés à des systèmes à mémoire longue.

4.2.2 Modèle d'état fractionnaire

Un système fractionnaire peut être décrit par trois modèles : équation différentielle, fonction de transfert généralisée, et modèle d'état fractionnaire. Dans notre travail, nous considérons un système défini par un modèle d'état fractionnaire tel que :

$$\begin{cases} D^\alpha x(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (4.5)$$

Avec : $u \in \mathbb{R}$, $y \in \mathbb{R}$, $\alpha \in \mathbb{R}^+$ et $D^\alpha = [D^{\alpha_1}x \quad D^{\alpha_2}x \quad \dots \quad D^{\alpha_n}x]^T$. Dans le cas de système d'ordre fractionnaire commensurable tous les états x sont dérivés à un même ordre non entier α : $D^\alpha = [D^\alpha x \quad D^\alpha x \quad \dots \quad D^\alpha x]^T$.

Ainsi, le modèle d'état fractionnaire à temps discret est donné comme suit [71][72] :

$$\begin{cases} \Delta^\alpha x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases} \quad (4.6)$$

Dans notre travail, le modèle d'état fractionnaire à temps discret est considéré.

4.2.3 Simulation d'un modèle d'état fractionnaire

Considérons le modèle d'état discret décrit par les équations suivantes :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases} \quad (4.7)$$

Avec $u \in \mathbb{R}$ et $y \in \mathbb{R}$.

La dérivée d'ordre fractionnaire de l'équation (4.7) est donnée comme suit [71] [72]

$$\begin{cases} \Delta^\alpha x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases} \quad (4.8)$$

De l'équation (4.3) on déduit :

$$\Delta^\alpha x(k+1) = x(k+1) - \sum_{j=1}^{k+1} (-1)^j \binom{\alpha}{j} x(k-j+1) \quad (4.9)$$

La substitution de l'équation 4.8 dans (4.9) conduit à :

$$\begin{aligned} \Delta^\alpha x(k+1) &= Ax(k) + Bu(k) \\ x(k+1) &= \Delta^\alpha x(k+1) - \sum_{j=1}^{k+1} (-1)^j \binom{\alpha}{j} x(k-j+1) \\ y(k) &= Cx(k) \end{aligned} \quad (4.10)$$

Soit $\beta(j)$ des coefficients définit comment suit :

$$\beta(j) = \frac{1}{h^\alpha} (-1)^j \binom{\alpha}{j}$$

La relation de récurrence peut être déduite :

$$\begin{cases} \beta(0) = 1 \\ \beta(j) = \beta(j-1) \frac{(j-1)(\alpha-1)}{j} \text{ for } j = 1, \dots, k \end{cases} \quad (4.11)$$

La substitution de (4.11) dans (4.10) conduit à :

$$\begin{cases} x(k+1) = \Delta^\alpha x(k+1) - \sum_{j=1}^{k+1} \beta(j) x(k-j+1) \\ y(k) = Cx(k) \end{cases} \quad (4.12)$$

Pour des réalisations pratiques le nombre d'échantillons pris en considération est réduit à un nombre L, appelé la longueur de la mémoire de système.

4.3 Modèle d'état PNLSS fractionnaire et sa simulation

On rappelle ici le modèle Hammerstein basé sur un modèle d'état non-linéaire PNLSS entier à temps discret [7, 2] donné en figure 4.1 :

$$\begin{cases} x(k+1) = Ax(t) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases} \quad (4.13)$$

Avec

$u(k)$ et $y(k)$ sont respectivement l'entrée et la sortie du système PNLSS, le signal intermédiaire $\tilde{u}(k)$ est l'entrée du bloc linéaire (la sortie du bloc non-linéaire).

La partie linéaire est décrite par une représentation d'état de matrices (A_0, B_0, C_0, D_0) :

$$\begin{cases} x(k+1) = A_0 x(k) + B_0 \tilde{u}(k) \\ y(k) = C_0 x(k) + D_0 \tilde{u}(k) \end{cases} \quad (4.14)$$

La partie non-linéaire est un polynôme d'ordre r :

$$\tilde{u}(k) = \sum_{i=1}^r p_i u^i(k) \quad (4.15)$$

Les matrices A et B de l'équation d'état et C et D de l'équation de sortie du modèle PNLSS (4.13) sont données comme suit :

$$A = A_0 \quad B = p_1 B_0 \quad C = C_0 \quad D = p_1 D_0 \quad (4.16)$$

Les vecteurs $\zeta(k)$ et $\eta(k)$ dans le cas d'un système Hammerstein sont égaux

$$\zeta(t) = \eta(t) = u(t)_{\{r\}} \quad (4.17)$$

Les matrices E et F sont données comme suit :

$$\begin{cases} E = [p_2 B_0 & p_3 B_0 & \dots & p_r B_0] \\ F = [p_2 D_0 & p_3 D_0 & \dots & p_r D_0] \end{cases} \quad (4.18)$$

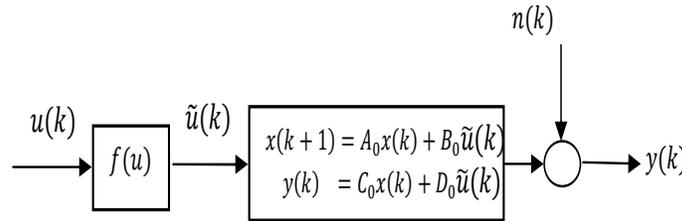


FIGURE 4.1: Système Hammerstein

L'avantage majeur de PNLSS est sa capacité de décrire un système complexe avec un nombre réduit de paramètres. Dans ce cas, les équations PNLSS développées pour un modèle Hammerstein à temps continu d'ordre entier peuvent être étendu pour le cas d'ordre fractionnaire :

$$\begin{cases} D^\alpha x(t) = Ax(t) + Bu(t) + E\zeta(t) \\ y(t) = Cx(t) + Du(t) + F\zeta(t) \end{cases} \quad (4.19)$$

Où $D^\alpha x(t)$ est la dérivée d'ordre fractionnaire d'ordre α de $x(t)$. Dans ce chapitre, nous considérons le cas d'un système non-linéaire fractionnaire à temps discret défini par :

$$\begin{cases} \Delta^\alpha x(k+1) = Ax(k) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\zeta(k) \end{cases} \quad (4.20)$$

où α est l'ordre fractionnaire et Δ est l'opérateur de différence fractionnaire discret définis dans [71].

Le modèle fractionnaire (4.20) sera simulé utilisant (4.12).

4.4 Identification d'un système PNLSS fractionnaire

Le problème d'identification des paramètres devient très difficile pour les systèmes d'ordre fractionnaire, il consiste à estimer non seulement les matrices du modèle PNLSS mais aussi le vecteur des ordres non entiers. On assume que le modèle PNLSS (4.21) est donné comme suit :

$$\begin{cases} \Delta^\alpha x(k+1) = Ax(k) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\zeta(k) \end{cases} \quad (4.21)$$

L'objectif d'identification du modèle PNLSS consiste à estimer les coefficients des matrices du modèle PNLSS A, B, E, C, D, F , et le vecteur d'ordre fractionnaire α .

Dans ce chapitre, nous considérons le cas commensurable où tout les états sont dérivés en même ordre fractionnaire α . Pour réduire le nombre de paramètres à estimer, sans perte de généralité, la forme canonique commandable peut être considérée, dans ce cas, le vecteur de paramètres θ a estimé se réduit à :

$$\theta = [a_1 \dots a_{n_a} \quad p_2 \dots p_{r-1} \quad c_1 \dots c_{n_a} \quad d \quad \alpha] \quad (4.22)$$

et les coefficients des matrices E et F peuvent être déduits.

La sortie du modèle étant non-linéaire en θ , une méthode à erreur de sortie basée sur une optimisation non linéaire est utilisée [69, 70]. L'estimation est basé sur l'algorithme de Levenberg-Marquardt (LM) [38] qui est étendu au modèle PNLSS fractionnaire, et le

nouveau modèle des fonctions de sensibilité, dans ce cas, est développé.

L'ensemble des données est composé de K observations $[u(k); y(k)]$

La valeur optimale de $\hat{\theta}$ est obtenu par la minimisation de critère quadratique :

$$J(\theta) = \sum_{k=1}^K \varepsilon(k)^2 \quad \text{avec} \quad \varepsilon(k) = y(k) - \hat{y}(k) \quad (4.23)$$

L'algorithme de Levenberg-Marquardt est utilisé pour estimer $\hat{\theta}$ itérativement.

Où

$$\left\{ \begin{array}{l} \theta^{(i+1)} = \theta^{(i)} - \left\{ [J''_{\theta\theta} + \lambda I]^{-1} J'_{\theta} \right\}_{\hat{\theta}=\theta^{(i)}} \\ J'_{\theta} = -2 \left(\sum_{k=1}^K \varepsilon_k \sigma_{k/\theta} \right) \text{ le Gradient} \\ J''_{\theta} = -2 \left(\sum_{k=1}^K \sigma_{k/\theta} \sigma_{k/\theta}^T \right) \text{ le Hessien} \\ \sigma_{k/\theta} = \frac{\partial \hat{y}(k, \theta)}{\partial \theta} \text{ Fonction de sensibilité de la sortie par rapport à } \theta \end{array} \right. \quad (4.24)$$

Cet algorithme est basé sur des fonctions de sensibilité nécessaires pour le calcul du Gradient et du Hessien [1] .

Le paragraphe suivant présente la méthode de calcul des fonctions de sensibilité pour le cas fractionnaire.

Un modèle d'état multivariable d'ordre non entier est développé dans [69] pour calculer les fonctions de sensibilités par rapport $\tilde{\theta}$. La différentiation de l'équation d'état du modèle PNLSS par rapport à chaque composante θ_j de $\tilde{\theta}$ avec $j = 1, \dots, n_{\theta}$, donne :

$$\left\{ \begin{array}{l} \Delta^{\alpha} \left[\sigma_{x/\tilde{\theta}} \right] = A_s \sigma_{x/\tilde{\theta}} + \left[\begin{array}{ccc} \frac{\partial A}{\partial \tilde{\theta}} & \frac{\partial B}{\partial \tilde{\theta}} & \frac{\partial E}{\partial \tilde{\theta}} \end{array} \right] \left[\begin{array}{c} x \\ u \\ \zeta \end{array} \right] \\ \\ \sigma_{y/\tilde{\theta}} = C_s \sigma_{x/\tilde{\theta}} + \left[\begin{array}{ccc} \frac{\partial C}{\partial \tilde{\theta}} & \frac{\partial D}{\partial \tilde{\theta}} & \frac{\partial F}{\partial \tilde{\theta}} \end{array} \right] \left[\begin{array}{c} x \\ u \\ \zeta \end{array} \right] \end{array} \right. \quad (4.25)$$

Avec $n_{\theta} = 2n_a + r + 1$ le nombre de paramètres du vecteur $\tilde{\theta}$.

Par simplification d'écriture nous écrivons (4.25) sous forme :

$$\begin{cases} \Delta^\alpha X_s = A_s X_s + B_s U_s \\ Y_s = C_s X_s + D_s U_s \end{cases} \quad (4.26)$$

Avec

$$\begin{cases} A_s = \text{Diagonal block } [A], \\ C_s = \text{Diagonal block } [C], \end{cases} \quad (4.27)$$

$$\begin{cases} B_s = \begin{bmatrix} \frac{\partial A}{\partial \theta} & \frac{\partial B}{\partial \theta} & \frac{\partial E}{\partial \theta} \end{bmatrix} \\ D_s = \begin{bmatrix} \frac{\partial C}{\partial \theta} & \frac{\partial D}{\partial \theta} & \frac{\partial F}{\partial \theta} \end{bmatrix} \end{cases} \quad (4.28)$$

$$\begin{cases} Y_s = [\sigma_{y/\theta_1} \ \sigma_{y/\theta_2} \ \cdots \ \sigma_{y/\theta_{n_\theta}}]^T \\ X_s = \sigma_{x/\hat{\theta}} = [\sigma_{x/\theta_1} \ \sigma_{x/\theta_2} \ \cdots \ \sigma_{x/\theta_{n_\theta}}]^T \\ U_s = [x_1 \ x_2 \ \cdots \ x_{n_a} \ u \ \zeta]^T \end{cases} \quad (4.29)$$

Pour la dernière composante du vecteur θ , à savoir l'ordre α , sa sensibilité est déterminée numériquement :

$$y(k, \alpha + \delta\alpha) - y(k, \alpha) \approx \delta\alpha \frac{\partial y}{\partial \alpha} = \delta\alpha \sigma_{y/\alpha} \quad (4.30)$$

Pour une variation suffisamment faible de $\delta\alpha$ une bonne approximation de la sensibilité $\sigma_{y/\alpha}$ est obtenue.

Afin de tester l'efficacité de l'algorithme développé ci dessus, des simulations numériques sont réalisées sur un système PNLSS fractionnaire, dans le cas sans bruit et en présence de bruit, dans le prochain paragraphe .

4.5 Exemples de simulation

La méthode d'identification présentée dans ce chapitre est appliquée à deux modèles Hammerstein de type PNLSS avec différentes complexités. Dans le premier exemple la partie linéaire est d'ordre deux ($n_a = 2$), la partie non linéaire est d'ordre deux ($r = 2$), et l'ordre fractionnaire $\alpha = 0.5$ dans le deuxième exemple ($n_a = 2$, $r = 3$), et $\alpha = 0.3$.

L'entrée $u(k)$ est une excitation persistante de moyenne nulle et de variance égal à un, et $n(k)$ est un bruit blanc de moyenne nulle.

L'identification est réalisée dans un contexte sans bruit, puis en présence de bruit $SNR = 34 dB$ et $SNR = 25 dB$.

4.5.1 Exemple 1

Dans cet exemple, la partie linéaire est d'ordre deux ($n_a = 2$) et la partie non-linéaire est d'ordre deux ($r = 2$). Le système est décrit par les équations PNLSS suivantes :

$$\begin{cases} \Delta^\alpha x(k+1) = Ax(k) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases} \quad (4.31)$$

avec $\alpha = 0.5$ et les matrices de système sont données comme suit :

$$A = \begin{bmatrix} 0 & 1 \\ -0.25 & -0.1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ C = \begin{bmatrix} 0.1 & 0.5 \end{bmatrix} \quad D = \begin{bmatrix} 1 \end{bmatrix}$$

Le sous système non-linéaire est décrit par le polynôme suivant :

$$f(u(t)) = u(t) + 0.2u^2(t)$$

Le tableau 4.1 donne les résultats de simulation de l'exemple 1 pour différents SNR : $SNR = 34 dB$, $SNR = 25 dB$ et pour le cas sans bruit. Les résultats obtenus montrent que les erreurs sont très faibles et les ordres fractionnaires du système sont estimés avec une bonne précision pour tout les cas. L'erreur de prédiction pour le cas sans bruit représentée en figure 4.2(b) est nulle (\approx de 10^{-16}); ainsi, la sortie estimée coïncide avec la sortie mesurée en figure 4.2(a).

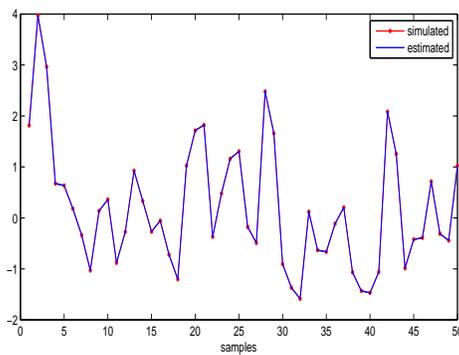
En présence de bruit $SNR = 34 dB$, et $SNR = 25 dB$, la sortie estimée est comparée avec la sortie mesurée en Figure 4.3(a) et en Figure 4.4(a), elles montrent une parfaite adéquation. L'erreur de prédiction pour un $SNR = 34 dB$ est représentée en Figure 4.3(b), elle est d'ordre très faible (\approx de 10^{-3}) et la Figure 4.4(b) représente l'erreur pour

TABLE 4.1: Résultats d'identification d'un modèle PNLSS fractionnaire exemple 1

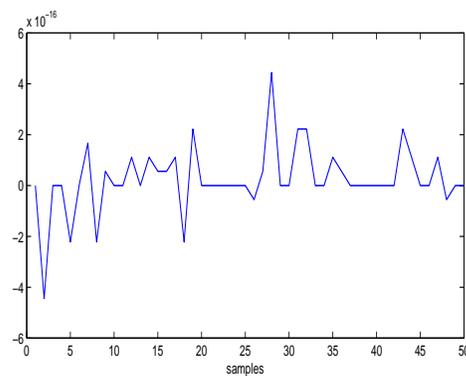
SNR	<i>Sans bruit</i>	$34dB$	$25dB$	<i>Valeurs exactes</i>
a_1	-0.250	-0.250	-0.225	-0.250
a_2	-0.100	-0.098	-0.169	-0.100
p_2	0.200	0.200	0.185	0.200
c_1	0.100	0.099	0.161	0.100
c_2	0.500	0.499	0.512	0.500
d	1.000	1.001	0.949	1.000
α	0.5	0.50	0.54	0.5
J	$8.6282e - 031$	$4.7e - 4$	0.916	—

un $SNR = 25 dB$.

Les résultats obtenus montrent que la méthode estime les paramètres et les ordres fractionnaires avec une bonne précision même en présence de bruit ; ainsi l'efficacité de l'estimateur est confirmée.

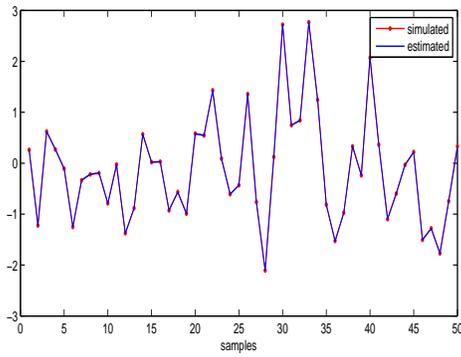


(a) Sortie estimée et sortie mesurée

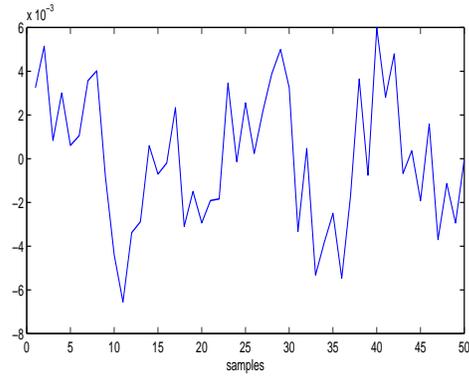


(b) Erreur de prédiction

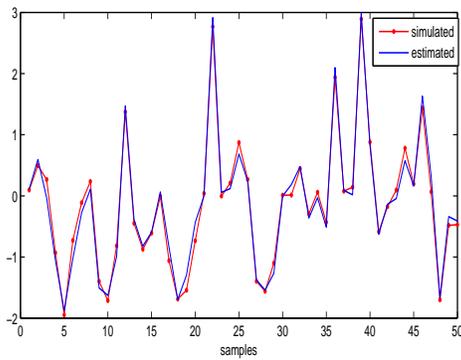
FIGURE 4.2: Résultats de l'exemple 1 pour le cas sans bruit



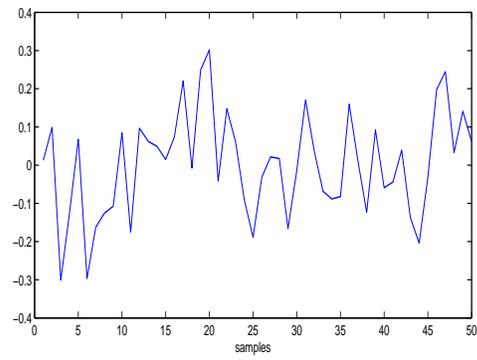
(a) Sortie estimée et sortie mesurée



(b) Erreur de prédiction

FIGURE 4.3: Résultats de l'exemple 1 avec $SNR = 34 \text{ dB}$ 

(a) Sortie estimée et sortie mesurée



(b) Erreur de prédiction

FIGURE 4.4: Résultats de l'exemple 1 avec $SNR=25\text{dB}$

4.5.2 Exemple 2

Dans ce cas, la complexité du système est augmenté avec $n_a = 2$ et $r = 3$, l'ordre fractionnaire $\alpha = 0.3$ et les matrices du système sont données comme suit :

$$A = \begin{bmatrix} 0 & 1 \\ -0.2 & -0.4 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.1 & 0.5 \end{bmatrix} \quad D = \begin{bmatrix} 0.1 \end{bmatrix}$$

Le bloc non-linéaire est décrit par le polynôme suivant

$$f(u(t)) = u(t) + 0.25u^2(t) + 0.5u^3(t)$$

Les simulations sont réalisées pour le cas sans bruit et pour des rapports signal sur bruit (SNR) $SNR = 34 \text{ dB}$, et $SNR = 25 \text{ dB}$. Le tableau 4.2 donne les résultats des simulations, ils montrent que les paramètres sont estimés avec une précision satisfaisante. Les

TABLE 4.2: Résultats d'identification d'un modèle PNLSS fractionnaire exemple 2

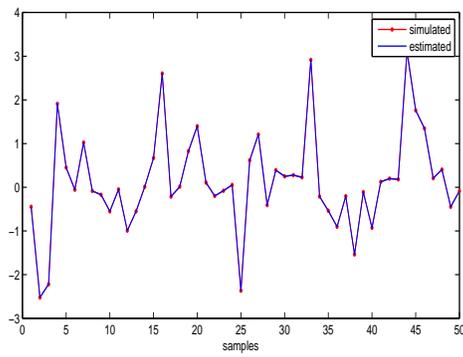
SNR	<i>Sans bruit</i>	34dB	25dB	<i>Valeurs exactes</i>
a_1	-0.200	-0.199	-0.235	-0.200
a_2	-0.400	-0.3965	-0.4732	-0.400
p_2	0.250	0.250	0.223	0.250
p_3	0.500	0.499	0.528	0.500
c_2	0.100	0.0987	0.134	0.100
c_3	0.500	0.5010	0.490	0.500
d	0.100	0.099	0.113	0.100
α	0.300	0.32	0.28	0.3
J	$1.145e - 30$	$7.544e - 4$	0.916	-

Figure (4.5(a)) et (4.6(a)) montrent une parfaite adéquation entre les sorties simulées et les sorties estimées respectivement pour le cas sans bruit et pour un $SNR = 34 \text{ dB}$; les erreurs de prédictions sont données en Figure (4.5(b)) et (4.6(b)) qui sont nulles (respectivement 10^{-10} et 10^{-3}).

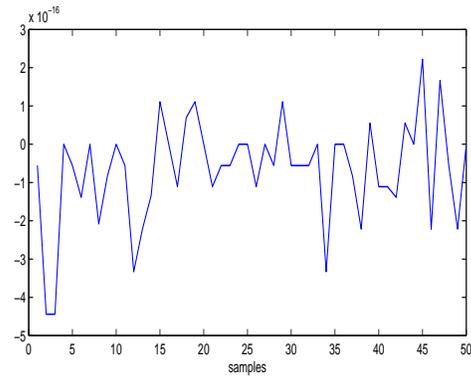
Pour un niveau de bruit $SNR = 25 \text{ dB}$, la sortie estimée correspond à la sortie mesurée en Figure (4.7(a)) et l'erreur de prédiction est représentée en Figure (4.7(b)).

4.6 Conclusion

Ce chapitre est consacré à l'extension des modèles Hammerstein entier au cas fractionnaire. Après un rappel sur les éléments de bases du calcul fractionnaire et la simulation d'un modèle d'état fractionnaire, le modèle PNLSS est utilisé pour décrire la structure

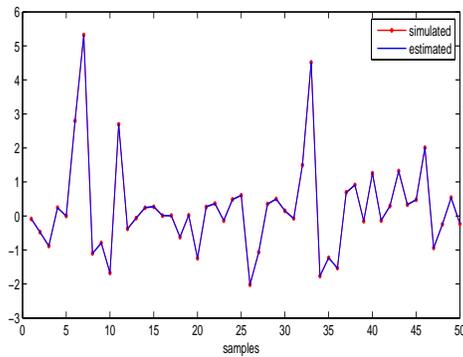


(a) Sortie estimée et sortie mesurée

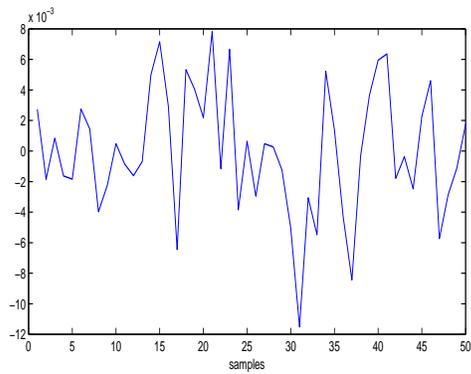


(b) Erreur de prédiction

FIGURE 4.5: Résultats de l'exemple 2 pour le cas sans bruit

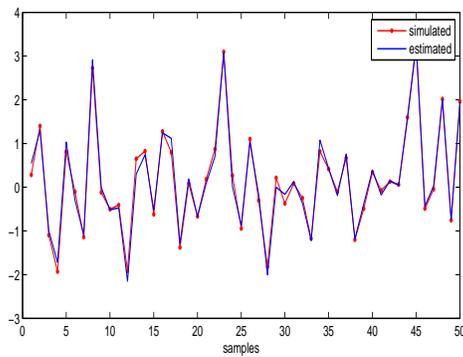


(a) Sortie estimée et sortie mesurée

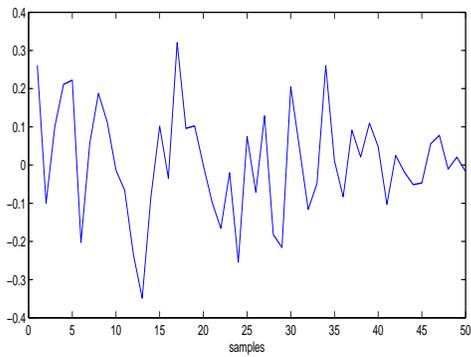


(b) Erreur de prédiction

FIGURE 4.6: Résultats de l'exemple 2 avec SNR=34dB



(a) Sortie estimée et sortie mesurée



(b) Erreur de prédiction

FIGURE 4.7: Résultats de l'exemple 2 avec SNR=25dB

du système Hammerstein fractionnaire ; la partie linéaire est représentée par un modèle d'état d'ordre fractionnaire tandis que la partie non-linéaire est un polynôme d'ordre r .

La contribution principale de ce chapitre consiste à généraliser le modèle PNLSS au cas fractionnaire à temps discret ; et le simuler utilisant la définition de Grunwald Letnikov. La méthode d'identification développée est basée sur une méthode à erreur de sortie , celle-ci est utilisée pour estimer les paramètres et l'ordre non entier du modèle PNLSS, et permet d'étendre l'algorithme de Marquardt au cas non entier ; le modèle de fonctions de sensibilité est développé, c'est un système multivariable fractionnaire.

Afin d'illustrer les performances de l'algorithme utilisé, l'identification a été appliquée à deux modèles Hammerstein de type PNLSS fractionnaire avec différentes complexités. Les résultats de simulation obtenus montrent l'efficacité de la méthode pour l'identification d'un modèle d'état non-linéaire polynomial PNLSS fractionnaire.

Conclusion générale

L'objectif de ce travail est d'apporter une contribution à l'identification des systèmes non linéaires, le cas Hammerstein entier a été considéré dans une première étape, puis le cas fractionnaire dans une deuxième étape.

Nous avons entamé ce travail par un rappel sur quelques structures non-linéaires et les principales méthodes d'identification. Notre intérêt s'est porté sur les modèles à blocs structurés en raison de leur simplicité pour représenter le comportement non-linéaire d'un système. Les différentes structures de ce modèle sont rappelées.

La deuxième partie de ce mémoire est orientée vers l'aspect identification des systèmes Hammerstein entier de type Controlled Auto-Regression CAR. Après un état de l'art des différentes méthodes d'identification des systèmes Hammerstein, deux principes d'identification sont testés en utilisant deux méthodes d'identifications RLS et PSO pour estimer les parties linéaires et non-linéaires. L'implémentation des algorithmes et leur simulation sont réalisées dans le but de tester la précision de ces méthodes.

La troisième partie est consacrée au modèle Hammerstein, décrit par un modèle d'état non-linéaire polynomial PNLSS à temps discret. La partie linéaire est représentée par un modèle d'état et la partie non-linéaire est un polynôme d'ordre connu. Sons identification est basée sur une méthode à erreur de sortie basée sur technique d'optimisation non-linéaire. Les simulations ont mis en évidence l'efficacité de la méthode associée à l'algorithme d'optimisation non-linéaire Levenberg Marquardt.

La dernière partie de notre travail est dédiée à l'extension du modèle PNLSS Hammerstein discret au cas non entier dont la partie linéaire est un modèle d'état d'ordre fractionnaire. On rappelle tout d'abord quelques notions mathématiques principales du

calcul fractionnaire, la définition de Grunwald Letnikov est utilisée pour simuler le modèle PNLSS fractionnaire. La méthode d'identification développée est basée sur l'algorithme d'optimisation non linéaire de Levenberg Marquardt que nous avons étendu au cas fractionnaire, en développant les fonctions de sensibilité du modèle non-linéaire fractionnaire. Les résultats obtenus montrent l'efficacité de la méthode de Levenberg Marquardt qui permet d'identifier le système PNLSS fractionnaire même en présence d'un pourcentage important de bruit.

Quant aux perspectives de recherche, elles s'inscrivent directement dans la continuité des travaux exposés dans ce mémoire. La première perspective consiste à appliquer les méthodes développées pour le cas d'un modèle Wiener. La deuxième perspective concerne l'application de la méthode d'optimisation PSO à un modèle Hammerstein fractionnaire.

Bibliographie

- [1] L.Ljung, *System identification : theory for the user*, Prentice Hall, London, 1987
- [2] A.Van Mulders, M.Volckaert, M.Diehl, and J.Schoukens, "Two non-linear optimization methods for black box identification compared", *Proc. 15th IFAC symposium on system identification, SYSID09*, pp. 1086-1091, 2009.
- [3] G.Dreyfus, *Neural Networks : Methodology And Applications*, Springer Verlag, 2005.
- [4] F. Giri, and E.W.Bai, *Block-oriented non-linear system identification*, 1st ed. Springer, 2010.
- [5] F.Ding, and T.Chen, "Identification of Hammerstein nonlinear ARMAX systems", *Automatica*, vol. 41, no. 9, pp. 1479-1489, 2005.
- [6] M.Schoukens,R. Pintelon, and Y.Rolain, "Parametric identification of parallel hammerstein systems", *IEEE Transactions on Instrumentation and Measurement*, doi :10.1109/TIM.2138370, 2011.
- [7] J.Paduart, L.Lauwers, R.Pintelon, and J.Schoukens, "Identification of a Wiener Hammerstein system using the polynomial nonlinear state space approach". *Proc. 15th IFAC symposium on system identification, SYSID09*, pp. 1080-1085, 2009.
- [8] E. W.Bai, "A blind approach to the Hammerstein–Wiener model identification", *Automatica*, vol. 38 no. 6, pp. 967-979, 2002.
- [9] A.Oustaloup, *La Derivation non entiere : Theorie, synthese et application*, 1995. Paris, Hermes edition.
- [10] O.Cois, "Systèmes linéaires non entiers et identification par modèle non entier : Application en thermique", Thèse de doctorat, Université de Bordeaux, France, 2002.

- [11] D.Matignon, " Représentations en variables d'état de guides d'ondes avec dérivation fractionnaire", Thèse de doctorat, Université de Paris-Sud, Orsay, France, 1994.
- [12] Z.E.A.Fellah , S.Berger, S.Lauriks, and C.Depollier, "Verification of kramerskonig relationship in porous materials having a rigid frame", *Journal of sound and vibration*, Vol. 270, pp. 865-885, 2004.
- [13] Sabatier J., M. Aoun, A. Oustaloup, G. Gregoire, F. Ragot, and P. Roy, "Estimation of lead acid battery state of charge with a novel fractional model", 2006.
- [14] Y.Ferdi , J.P.Herbeuval , A.Charef, and B. R.Boucheham, "wave detection using fractional digital differentiation", *ITBM-RBM*, Vol. 24, pp. 273-280, 2003.
- [15] Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P., Hjalmarsson, H., and A.Juditsky, "Nonlinear black-box modeling in system identification : a unified overview", *Automatica*, vol. 31, no. 12 :1691-1724, 1995.
- [16] K. S. Narendra, and P. G. Gallman, "An iterative method for the identification of nonlinear systems using a Hammerstein model," *IEEE Trans. Automat. Contr.*, vol. AC-11, pp. 546-550, 1966.
- [17] A.Hammerstein, "Nichtlineare integralgleichung nebst anwendungen", *Acta Mathematica*, vol. 54, pp. 117-176, 1930.
- [18] E.Dempsey, and D.Westwick, "Identification of Hammerstein models with cubic spline nonlinearities", *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 237-245, 2004.
- [19] J.Kim, and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory", *IEE Electronics Letters*, vol. 37, pp. 1417-1418, 2001.
- [20] A. Balestrino, A.Landi, M.Ould-Zmirli, and L.Sani, "Automatic nonlinear auto-tuning method for Hammerstein modeling of electrical drives", *IEEE Transactions on Industrial Electronics*, vol. 48, pp.645-655, 2001.
- [21] N.Wiener, *Nonlinear problems in random theory*, Wiley, New York, 1958.

- [22] S.Boyd, and L.O. Chua, "Fading memory and the problem of approximating non-linear operators with Volterra series", *IEEE Transactions on Circuits and Systems*, vol. 32, pp. 1150-1161, 1985.
- [23] A.Kalafatis, L. Wang, and W.R. Cluett, "Identification of time-varying pH processes using sinusoidal signals", *Automatica*, vol. 41, pp. 685-691, 2005.
- [24] I.W.Hunter, and M.J.Korenberg, "The identification of nonlinear biological systems :Wiener and Hammerstein cascade models", *Biological Cybernetics*, vol. 55, pp. 135-144, 1986.
- [25] Y.J.Lee, S.W.Sung, S.Park, and S.Park, "Input test signal design and parameter estimation method for the Hammerstein–Wiener processes", *Industrial and Engineering Chemistry Research*, vol. 43, pp. 7521-7530, 2004.
- [26] J.Vörös, "Parameter identification of Wiener systems with discontinuous nonlinearities", *Systems et Control Letters*, vol. 44, pp. 363-372, 2001.
- [27] S.Rangan, G.Wolodkin, and K. Poolla, "New Results for Hammerstein System Identification". *Proceedings 34th CDC News Orleans*, LA - December, 1995.
- [28] G.Mzyk, " Nonparametric identification of Wiener systems - A median based approach", *Proceedings of the 10th IEEE International Conference MMAR*, 2004.
- [29] W.Greblicki, and M.Pawlak , "Nonparametric recovering nonlinearities in block oriented systems with the help of Laguerre polynomials", *Control-Theory and Advanced Technology*, vol. 10, part 1, pp. 771-791, 1991 b.
- [30] W.Greblicki, "Nonparametric identification of Wiener systems". *IEEE Transactions on Information Theory*, vol.38, pp. 1487-1493, 1992.
- [31] E.W.Bai, "Frequency domain identification of Wiener models", *Automatica*, Vol. 39, pp.1521-1530, 2003.
- [32] Y.Zhu, "Estimation of an N-L-N Hammerstein-Wiener model", *Automatica*, Vol. 38, pp. 1607-1614, 2002.

- [33] W.Greblicki, and M.Pawlak, "Nonparametric recovering nonlinearities in block oriented systems with the help of Laguerre polynomials", *Control-Theory and Advanced Technology*, vol. 10,part 1, pp. 771-791, 1994.
- [34] Bai E.W., "Identification of systems with hard input nonlinearities of unknown structure", *Automatica*, vol. 38, pp. 853-860, 2002.
- [35] F.Giri, F.Z.Chaoui, and Y.Rochdi, "System Identification of Hammerstein models including a known structure input nonlinearity - The Hysterisis case", *IFAC Workshop on Adaptation and Learning in Control and Signal Processing*, Yokohama, Japan, Session MM3-A-4, 2004a.
- [36] J.Vörös, "Iterative algorithm for identification of Hammerstein systems with two-segment nonlinearities", *IEEE Transactions on Automatic Control*, vol. 44, pp. 2145-2149, 1999b.
- [37] C.T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.
- [38] D.W. Marquardt, "An algorithm for least squares estimation of nonlinear parameters", *Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, no 2, pp. 431-441, Paris France, 1963.
- [39] J. Kennedy, and R.C. Eberhart, "Particle Swarm Optimization", *Proc. IEEE International Conference on Neural Networks*, Perth, Australia, pp. 1942-1948, 1995.
- [40] J. Kennedy, R.C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, Academic Press, San Fransisco, 2001.
- [41] Y. Liu, and E. W. Bai, "Iterative identification of Hammerstein systems", *Automatica*, Vol.43, no. 2, pp. 346-354, 2007.
- [42] P. Stocia, "On the convergence of an iterative algorithm used for Hammerstein system identification", *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 967-969, 1981.
- [43] J. Vörös, "Recursive identification of Hammerstein systems with discontinuous nonlinearities containing dead-zones", *IEEE Trans. Automatic Control*, vol.48 no. 12, pp. 2203-2206, 2003.

- [44] F. Ding, Y. Shi, and T. Chen, "Gradient-based identification methods for Hammerstein nonlinear ARMAX models", *Nonlinear Dynam*, vol. 45 (1-2), pp. 31-43, 2006.
- [45] F. Ding and T. Chen, "Identification of Hammerstein nonlinear ARMAX systems", *Automatica*, vol. 41, no. 9, pp. 1479-1489, 2005.
- [46] F. Ding, and W. Fan, "Parameter Estimation for Hammerstein Nonlinear Controlled Auto-Regression Models", *Proc. IEEE International Conference on Automation and Logistics*, Jinan, China, 2007.
- [47] E. W. Bai, and D. Li, "Convergence of the iterative Hammerstein system identification algorithm", *IEEE Transactions on Automatic Control*, vol. 49, no. 11, pp. 1929-1940, 2004.
- [48] F. Giri, Y. Rochdi, F.Z. Chaoui, and A. Brouri, "Identification of Hammerstein systems in presence of hysteresis-backlash and hysteresis-relay nonlinearities", *Automatica*, vol. 44, no. 3, pp.767-775, 2008.
- [49] H.Chen, and F.Ding, "Hierarchical Least Squares Identification for Hammerstein Nonlinear Controlled Autoregressive Systems", *Circuits Syst Signal Process DOI 10.1007/s00034-014-9839-9*, 2014.
- [50] F. Ding, Y. Shi, and T. Chen, "Auxiliary model based least-squares identification methods for Hammerstein output-error systems", *Systems Control Lett*, vol. 56, no. 5, pp. 373-380, 2007.
- [51] F.Wang, K.Xing, X.Xu, and H.Liu, "An Identification Approach of Hammerstein Model", *978-1-4244-5182-1/10 26.0.IEEE*, 2010.
- [52] D. F. Wang, Y. Y. Ren, C L. Liu, and P. Han, "Identification of Thermal Process Using Hammerstein Model Based on Particle Swarm Optimization Algorithm", *Lecture Notes in Electrical Engineering 238, DOI 10.1007/978-1-4614-4981-2-214, Springer Science+Business Media, New York 2014.*
- [53] X.G. Liu, and J. Lu, "Least squares based iterative identification for a class of multirate systems", *Automatica*, vol. 46, no. 3, pp. 549-554, 2009.

- [54] J.Wang, Q.Zhang, and L.Ljung, "Revisiting Hammerstein system identification through the two-stage algorithm for bilinear parameter estimation", *Automatica*, vol. 45, no. 11, pp. 2627-2633, 2009
- [55] F.Ding, X.J.Liu, and J.Chu, "Gradient-based and least-squares-based iterative algorithms for Hammerstein systems using the hierarchical identification principle", *IET.Control Theory and Applications*, vol. 7, no. 2, pp. 176-184, 2013.
- [56] F.Ding, X.,Liu, H.,Chen, and G.Yao, "Hierarchical gradient based and hierarchical least squares based iterative parameter identification for CARARMA", *Signal processing*. vol. 97, pp. 31-39, 2014.
- [57] L.L. Xiang, L.B. Xie, and R.F. Ding, "Hierarchical least squares algorithms for single-input multiple-output systems based on the auxiliary model", *Math. Comput. Modelling*, vol. 52, (5-6), pp. 918-924, 2010.
- [58] Z.N. Zhang, F.Ding, and X.G.Liu, "Hierarchical gradient based iterative parameter estimation algorithm for multivariable output error moving average systems", *Comput.Math.Appl*, vol. 61, no. 3, pp. 672-682, 2011.
- [59] J.Paduart, "Identification of nonlinear systems using Polynomial Nonlinear State Space models. Ph.D thesis. Vrije Universiteit Brussel, 2008.
- [60] P.Brunet, "Nonlinear System Modeling and Identification of Loudspeakers", Ph.D. Northeastern University Boston, Massachusetts, 2014.
- [61] F. Ding, P.X. Liu, and G. Liu, "Identification methods for Hammerstein non-linear systems", *Digital Signal Process*, vol. 21, no. 2, pp. 215-238, 2011.
- [62] J. Ding, Y. Shi, H.G. Wang, and F. Ding, "A modified stochastic gradient based parameter estimation algorithm for dual-rate sampled-data systems", *Digital Signal Processing*, vol. 20, no. 4, pp. 1238-1249, 2010.
- [63] J. Paduart, L. Lauwers, R. Pintelon, and J. Schoukens, "Identification of a Wiener-Hammerstein system using polynomial non-linear state space approach", *Control Engineering practice*, vol. 20, pp. 1133-1139, 2012.

- [64] S. Lu, F.J. Molz, and G.J. Fix, "Possible problems of scale dependency in applications of the three-dimensional fractional advection-dispersion equation to natural porous media", *Water Resources Research*, Vol. 38, no 9, pp. 1-9, 2002.
- [65] K.B.Oldham ,and C.G.Zoski, "Analogue instrumentation for processing polarographic data", *J Electroanal Chem Interfacial Electrochem*, vol.157, no. 1, pp. 27-51, 1983.
- [66] M.D.Ortigueira, "A coherent approach to non-integer order derivatives", *Signal Process*", vol. 86, no. 10, pp. 2505-2515, 2006.
- [67] A.Carpinteri , and F.Mainardi, *Fractal and fractional calculus in continuum mechanics*, Springer, New York, 2006
- [68] I.Petráš, "Chaos in fractional-order population model", *Int J Bifurcat Chaos Appl Sci Eng* 22(4) :28, 2012.
- [69] Djamah T., R. Mansouri, S. Djennoune, and M. Bettayeb, " Heat transfer modeling and identification using fractional order state space models", *J. Europeen des Systemes Automatises, JESA*, Vol. 42, pp. 939-951, 2008.
- [70] T.Djamah , M. Bettayeb, and S. Djennoune, " Identification of multivariable fractional order systems", *Asian Journal of Control*, Vol. 15, No. 2, pp. 1-10, 2013.
- [71] A. Dzieliński, and D.Sierociuk, "Stability of discrete fractional-order state-space systems", *Proc. of the 2nd IFAC Workshop on Fractional Differentiation and its Applications (FDA '06)*, Porto, Portugal, 2006.
- [72] A.Dzieliński , and D.Sierociuk, "Adaptive Feedback Control of Fractional Order Discrete State-Space Systems", *Proc. of the 2005 Intern. Conf. on Computational Intelligence for Modelling, Control and Automation, and Inter. Conf. on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)*, Vienna, Austria, pp. 524-529, 2005.