RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE Ministère de l'Enseignement Supérieur et de la Recherche Scientifique UNIVERSITÉ MOULOUD MAMMERI DE TIZI-OUZOU



FACULTÉ DE GENIE ELECTRIQUE ET D'INFORMATIQUE DÉPARTEMENT D'AUTOMATIQUE

THÈSE DE DOCTORAT

Présentée pour l'obtention du diplôme de

Doctorat 3^{ème} Cycle LMD

Spécialité : AUTOMATIQUE

Par :

Lamia SERSOUR

THÈME

IDENTIFICATION DES SYSTÈMES FRACTIONNAIRES NON LINÉAIRES

Président	Saïd DJENNOUNE	Professeur, Université Mouloud MAMMERI de Tizi-Ouzou
Rapporteur	Tounsia DJAMAH	MCA, Université Mouloud MAMMERI de Tizi-Ouzou
Examinateur	Samir LADACI	Professeur, École Nationale Polytechnique de Constantine
Examinateur	Fazia KHELLAS	Professeur, Université Mouloud MAMMERI de Tizi-Ouzou
Examinateur	Ahmed MAIDI	Professeur, Université Mouloud MAMMERI de Tizi-Ouzou
Invité	Maâmar BETTAYEB	Professeur, Université de Sharjah (EAU)

Année : 2018

Je dédie ce travail de thèse a ma très chère maman, la meilleurs mère du Monde. A la mémoire de mon défunt père.

Remerciements

Ce travail a été effectué au Laboratoire de Conception et Conduite des Systèmes de Production (L2CSP), de l'université Mouloud MAMMERI de Tizi-Ouzou.

Je tiens, en premier lieu, à exprimer ma plus profonde gratitude et mon profond respect à Madame Tounsia DJAMAH, Maître de conférences classe A à l'université Mouloud MAMMERI de Tizi Ouzou, qui a dirigé mes travaux de recherche. Je la remercie pour m'avoir fait profiter de son enthousiasme, de sa rigueur scientifique, de son expérience et les nombreux conseils qu'elle n'a pas cessé de me prodiguer tout au long de ce travail.

Je tiens particulièrement à remercier Monsieur Maâmar BETTAYEB, Professeur à l'université de Sharjah (Emirats Arabes Unis), et vice-recteur chargé de la poste-graduation pour avoir accepté de m'accueillir dans son laboratoire à l'université de Sharjah, pour ses conseils, ses critiques, sa disponibilité et sa simplicité. Je lui suis très reconnaissante de la confiance qu'il a su me témoigner. Je tiens également à remercier toute sa famille pour leur sympathie et leur générosité.

J'exprime mes sincères remerciements à Monsieur Saïd DJENNOUNE, Professeur à l'université Mouloud MAMMERI de Tizi-Ouzou, qui m'a fait l'honneur de présider le jury de cette thèse. Je n'ai aucun doute que ses riches connaissances et précieuses suggestions me permettront d'améliorer ce travail.

Mes remerciements vont également en particulier vers Monsieur Ahmed MAIDI, Professeur à l'université Mouloud MAMMERI de Tizi-Ouzou, pour les nombreux conseils qu'il m'a prodigués et pour avoir accepté de participer à ce jury.

Je remercie également Messieurs Samir LADACI, Professeur à l'Ecole Nationale Polytechnique de Constantine, Madame Fazia KHELLAS, Professeur à l'université Mouloud MAMMERI de Tizi-Ouzou pour l'intérêt qu'ils ont manifesté pour ce travail en acceptant de faire partie du Jury.

Mes remerciements s'adressent aussi à tous les membres du Laboratoire de Conception et Conduite des Systèmes de Production (L2CSP), pour leur sympathie, l'ambiance agréable et le bon temps passé ensemble durant ces années. A tous ceux qui m'ont guidé avec gentillesse et efficacité, en particulier Ouerdia MEGHERBI, Sarah KASSIM, Karima HAMMAR et Nouara HABRACHE.

Je souhaite remercier tous mes ami(e)s, qui m'ont encouragé et m'ont accompagné avec joie durant toute cette période, ainsi que tous ceux qui de près ou de loin ont contribué à la réalisation de cette thèse.

La réalisation de cette thèse ne saurait être possible sans le soutien inconditionnel de mes proches. C'est ainsi, avec grand plaisir et reconnaissance, que j'adresse toute ma reconnaissance et mes plus sincères remerciements à tous les membres de ma famille qui m'ont énormément soutenu et encouragé dans les moments difficiles et avec lesquels j'ai partagé l'immense joie de la réussite, plus particulièrement ma mère, mes chères sœurs Kahina et Nadia ainsi que son mari Rachid et surtout ma petite nièce adorée Nélia.

J'adresse enfin toute ma reconnaissance à ma chère maman, pour ses nombreux sacrifices consentis, sa confiance, son aide et sa compréhension totale durant les périodes difficiles durant mes études. Qu'elle trouve ici l'expression de ma reconnaissance et de mon éternelle gratitude.

Table des matières

Notations			ix	
Ta	able (des fig	ures	xii
Li	ste d	les tab	leaux	xiv
Li	ste d	les tra	vaux scientifiques réalisés	1
In	trod	uction	générale	3
1	Sys	tèmes	non linéaires entiers	7
	1.1	Introd	luction	7
	1.2	Classe	s des modèles non linéaires	8
		1.2.1	Modèle NFIR	8
		1.2.2	Modèle NARX	9
		1.2.3	Modèles NARMAX	9
		1.2.4	Modèle NBJ	10
		1.2.5	Modèles non linéaires des fonctions de base orthonormées	10
		1.2.6	Modèle Séries de Volterra	11
		1.2.7	Modèles blocs orientés	12
	1.3	Repré	sentation d'état non linéaire polynomiale	16
	1.4	Descri	ption du modèle de Wiener sous forme PNLSS	17
	1.5	État d	le l'art sur l'identification des systèmes non linéaires entiers \ldots .	20
	1.6	Conclu	usion	21

2	Syst	tèmes fractionnaires 23		
	2.1	Introduction		
	2.2	Opérateurs d'ordre fractionnaire		
		2.2.1 Différence fractionnaire au sens de Riemann-Liouville		
		2.2.2 Différence fractionnaire au sens de Caputo		
		2.2.3	Différence fractionnaire au sens de Grunwald-Letnikov	25
	2.3	Modèles des systèmes d'ordre fractionnaire		
		2.3.1	Cas continu	27
		2.3.2	Cas discret	29
	2.4	Stabili	ité des systèmes fractionnaires	31
	2.5	Simula	ation des systèmes fractionnaires	32
		2.5.1	Simulation d'un modèle d'état fractionnaire	33
	2.6	6 Description du système de Wiener fractionnaire		
		2.6.1	Modèle de Wiener fractionnaire sous forme PNLSS	35
		2.6.2	Modèle de Wiener sous forme de régression fractionnaire	41
	2.7	Conclu	usion	44
3	Ider	ntificat	ion des systèmes non linéaires entiers et non entiers	45
	3.1	Introduction		45
	3.2	État de l'art sur l'identification des systèmes fractionnaires non linéaires $.46$		46
	3.3	Generation d'un système non linéaire de type de Wiener entier 47		47
		3.3.1	Identification du modèle de Wiener PNLSS par l'algorithme de LM	48
		3.3.2	Identification du modèle de Wiener PNLSS par l'algorithme SAVPSO) 52
	3.4	Identi	fication des systèmes de Wiener fractionnaires	56
		3.4.1	Identification du système de Wiener PNLFSS par l'algorithme SAVP	SO 58
		3.4.2	Identification du système de Wiener PNLFSS par l'algorithme LM .	58
		3.4.3	Identification du système de Wiener sous forme de régression fraction	nnaire 63
	3.5	Conclu	usion	64

4	\mathbf{R} és	Résultats de simulation et discussions		67	
	4.1	Introd	luction	67	
	4.2	Identi	fication du modèle de Wiener PNLSS au cas entier	68	
		4.2.1	Identification par l'algorithme LM	68	
		4.2.2	Identification par l'algorithme SAVPSO	72	
4.3 Identification du système de Wiener au cas fractionnaire		fication du système de Wiener au cas fractionnaire	76		
		4.3.1	Identification du système de Wiener PNLFSS	77	
		4.3.2	Identification du modèle de Wiener : régression fractionnaire	96	
	4.4	Concl	usion	99	
С	onclu	ision g	énérale	102	
Bi	ibliog	graphie	e	105	

Notations

Symboles

A(q):	un polynôme d'ordre n_A
B(q):	un polynôme d'ordre n_B
D_t^{α} :	opérateur de dérivation d'ordre non entière α
e(k):	vecteur de bruit
e - 1:	10-1
f(.):	fonction non linéaire
G(z):	fonction de transfert discrète
g:	un paramètre de valeur réelle
g_{best} :	meilleure position de tout le groupe
H(q):	un polynôme d'ordre n_{H}
h:	période d'échantillonnage
$h_K(au_1,\ldots, au_K)$:	K-ème ordre du noyau de Volterra
I^{lpha}_t :	$(\alpha \in \mathbb{R}),$ l'intégration non entière d'ordre α
k:	variable de temps discret
L:	taille de la mémoire
l_i :	le i eme filtre de Laguerre
M(q):	un polynôme d'ordre n_M
N(q):	un polynôme d'ordre $n_{\cal N}$
n_a :	longueur du vecteur a
n_b :	longueur du vecteur b
p_{best} :	meilleure position de chaque particule

r:	ordre du polynôme non linéaire
t:	variable de temps réel
u(k):	entrée en temps discret
v(k):	vecteur de sortie du bloc linéaire en temps discret
x(k):	vecteur d'état du modèle d'espace d'état en temps discret
y(k):	vecteur de sortie en temps discret
ϵ :	erreur de prédiction
$\binom{lpha}{k}$:	$(\alpha \in \mathbb{R}_+),$ binôme de Newton généralisé à des ordres réels
α :	l'ordre d'intégration/dérivation non entier
λ :	paramètre de contrôle de la direction de recherche de l'optimisation
$\phi(k)$:	le vecteur de régression en temps discret
θ :	vecteur des paramètres
$\hat{ heta}$:	vecteur des paramètres estimés
Δ :	Opérateur de différence
$\Delta^{\alpha}x$:	tous les éléments du vecteur x sont dérivés au même ordre α
$\Delta^{(\alpha)}x$:	chaque élément du vecteur x est dérivé à une composante du vecteur
	lpha

Abréviations

ARMAX :	Autoregressive Moving Average model with exogenous inputs	
ARX :	Autoregressive with exogenous input	
BJ :	Box-Jenkins	
BLA :	Best Linear Approximation	
FIR :	Finite Impulse Response	
GL :	Grünwald_Letnikov	
LM :	Levenberg-Marquardt	
NARMA :	Nonlinear Autoregressive moving average	
NARX :	Nonlinear Autoregressive with exogenous input	
NBJ :	Nonlinear Box-Jenkins	
NFIR :	Nonlinear Finite Impulse Response	
NARMAX :	Nonlinear Autoregressive Moving Average model with exogenous	
	input	
NARX :	Nonlinear Autoregressive with exogenous input	
NOBF :	Nonlinear Orthonormal Basis Function	
OBF :	Orthonormal Basis Function	
PNLFSS :	Polynomial Non-linear Fractional State Space	
PNLSS :	Polynomial Non-linear State Space	
PSO :	Particle Swarm Optimization	
RLS :	Recursive Least Square	
SAVPSO :	Self Adaptative velocity Particle Swarm Optimization	
SISO :	une seule Entrée/Sortie	
SNR :	Signal to Noise Ratio	

Table des figures

1.1	Modèle de Hammerstein	13
1.2	Modèle de Wiener	14
1.3	Modèle de Wiener-Hammerstein	14
1.4	Modèle de Hammerstein-Wiener	15
1.5	Modèle de Wiener PNLSS	17
2.1	Domaine de stabilité des systèmes fractionnaires	32
4.1	Résultats de la simulation avec l'algorithme LM pour un cas sans bruit	70
4.2	Résultats de la simulation avec l'algorithme LM pour $SNR = 34 \ dB$	71
4.3	Résultats de la simulation avec l'algorithme LM pour $SNR = 15 \ dB$	72
4.4	Résultats de la simulation avec l'algorithme SAVPSO pour un cas sans bruit.	75
4.5	Résultats de la simulation avec l'algorithme SAVPSO pour $SNR = 34 \ dB$.	76
4.6	Résultats de la simulation avec l'algorithme SAVPSO pour $SNR = 15 \ dB$.	77
4.7	Résultats de l'exemple 1 fractionnaire, avec SAVPSO, sans bruit	81
4.8	Résultats de l'exemple 1 fractionnaire, avec SAVPSO, $SNR=34\;dB.$	82
4.9	Résultats de l'exemple 1 fractionnaire, avec SAVPSO, $SNR=15\;dB.$	83
4.10	Résultats de l'exemple 2 fractionnaire, avec SAVPSO, sans bruit	85
4.11	Résultats de l'exemple 2 fractionnaire, avec SAVPSO, $SNR=34\;dB.$	86
4.12	Résultats de l'exemple 2 fractionnaire, avec SAVPSO, $SNR=15\;dB.$	87
4.13	Évolution du critère par rapport à la structure, exemple 1	89
4.14	Résultats de l'exemple 1 fractionnaire, avec LM, sans bruit	90
4.15	Sorties estimées et simulées avec LM, exemple 1 fractionnaire, $SNR = 34 \ dB$. 91

4.16 Sorties estimées et simulées avec LM, exemple 1 fractionnaire, $SNR = 15 \ dB$. 91 4.17 Évolution du critère par rapport à la structure, exemple 2. 92 4.18 Résultats de l'exemple 2 fractionnaire, avec LM, sans bruit. 94 4.19 Sorties estimées et simulées avec LM, exemple 2 fractionnaire, $SNR = 34 \ dB$. 95 4.20 Sorties estimées et simulées avec LM, exemple 2 fractionnaire, $SNR = 15 \ dB$. 95 4.21 Résultats de la simulation avec l'algorithme PSO pour un cas sans bruit. 98 4.22 Résultats de la simulation avec l'algorithme PSO pour $SNR = 34 \ dB$ 99 4.23 Résultats de la simulation avec l'algorithme PSO pour $SNR = 15 \ dB$ 100

Liste des tableaux

2.1	Modèle de Wiener avec $n_a = 2$ et $r = 2$
2.2	Vecteurs F et η du modèle de Wiener pour $n_a = 2$ et $r = 2$
2.3	Vecteurs F et η du modèle de Wiener pour $n_a = 2$ et $r = 3$
4.1	Résultats d'identification du modèle de Wiener PNLSS avec LM 69
4.2	Résultats d'identification du modèle de Wiener PNLSS avec SAVPSO $$ 74 $$
4.3	Résultats d'identification du modèle de Wiener PNLFSS avec SAVPSO, exemple 1 80
4.4	Résultats d'identification du modèle de Wiener PNLFSS avec SAVPSO, exemple 2 84
4.5	Évolution du critère par rapport à la structure, exemple 1
4.6	Moyennes des paramètres θ_i du modèle de Wiener PNLFSS avec LM de l'exemple 1 $$ 89
4.7	Évolution du critère par rapport à la structure, exemple 2
4.8	Moyennes des paramètres θ_i du modèle de Wiener PNLFSS avec LM de l'exemple 2. 93
4.9	Résultats de la simulation avec PSO pour différents SNR

Liste des travaux scientifiques réalisés

• Publication internationale

 L. Sersour, T. Djamah and M. Bettayeb, "Nonlinear system identification of fractional Wiener models", *Nonlinear Dynamics*, vol. 92, no.4, pp. 1-13, 2018.

• Conférences internationales

- L. Sersour, T. Djamah and M. Bettayeb, "Wiener System Identification using Polynomial Non Linear State Space Model", *The 3rd International Conference on Control, Engineering and Information Technology : CEIT'2015*, Tlemcen, Algeria, May, 2015.
- L. Sersour, T. Djamah and M. Bettayeb, "Wiener system Identification using Self-Adaptive Velocity Particle Swarm Optimization", *International Conference on Au*tomatic control, Telecommunications and Signals : ICATS'2015, Annaba, Algeria, November, 2015.
- L. Sersour, T. Djamah and M. Bettayeb, "Identification of Wiener Fractional model using Self-Adaptative Velocity Particle Swarm Optimization", *The 7th International Conference on Modelling, Identification and Control : ICMIC'2015*, Sousse, Tunisia, December, 2015.
- L. Sersour, T. Djamah and M. Bettayeb, "Fractional Wiener System Identification using Levenberg Marquardt algorithm", *Graduat Student Recherch Conference*, UAEGSRC 2016, Al Ain, United Arab Emirates, April, 2016.
- L. Sersour, T. Djamah and M. Bettayeb, "Wiener nonlinear system Identification using fractional model", *The International Conference on Fractional Differentiation* and its Applications (ICFDA 2018), Amman, The Hashemite Kingdom of Jordan,

16-18 July, 2018

Introduction générale

L'identification des systèmes est la théorie ou l'art de la construction de modèles mathématiques des systèmes dynamiques à partir de signaux d'entrée-sortie observés. En effet, ce sujet est largement concidéré et a reçu d'énormes contributions. Ainsi, la modélisation et l'identification d'un système dynamique complexe est une tâche importante pour la simulation, la conception de contrôle et le diagnostic de défaut.

Un certain nombre de méthodes ont été développées pour l'identification des modèles linéaires [1].

Cependant, la plupart des systèmes réels montrent un comportement dynamique non linéaire. Dans ce cas, il est très difficile de construire un modèle mathématique sur toute sa plage de fonctionnement et l'utilisation de modèles linéaires est insuffisante pour leur description.

Par conséquent, un intérêt considérable a été relevé pour l'identification des systèmes non linéaires, qui reste un domaine de recherche ouvert, et aucun cadre général n'est disponible pour la théorie des systèmes non linéaires [2–8].

En outre, il est cité dans la littérature que le travail sur la classe des systèmes non linéaires ne concerne que des structures de modèles spécifiques tels que les modèles NAR-MAX, séries de Volterra ou les modèles blocs orientés [5,9,10], ... Ces derniers sont utilisés intensivement et constituent un outil puissant pour modéliser et optimiser les systèmes non linéaires [11]. En effet, cette classe de modèles non linéaires permet de construire des modèles à partir de blocs simples afin de trouver des structures suffisamment flexibles pour couvrir de nombreux systèmes réels non linéaires pertinents. Ainsi, leur identification reste un sujet de recherche très actif depuis plusieurs décennies. Les modèles non linéaires orientés par blocs comprennent des modèles complexes, qui sont définis par la séparation du comportement statique non-linéaire et du comportement linéaire invariant dans le temps (LTI) dans différents blocs.

L'interconnexion de ces parties dans plusieurs structures inclut le modèle de Hammerstein, Wiener, Wiener-Hammerstein, ... [5].

Dans cette étude, le cas du système de Wiener est considéré. Ce type de modèle est constitué de bloc dynamique linéaire en série avec une non-linéarité statique.

Le modèle de Wiener révèle la capacité de décrire une classe importante de systèmes pratiques et industriels dans plusieurs domaines tels que le contrôle [12], l'énergétique [13], l'identification [14–17], le diagnostic de défauts [18,19]. Ils sont appliqués dans : les processus chimiques [20], les échangeurs de chaleur [21] et les micro-ondes non linéaires (nonlinear microwaves) [22], etc ...

Le calcul fractionnaire est la deuxième notion-clé sur laquelle se base cette thèse.

Le calcul d'ordre non entier ou fractionnaire s'est considérablement développé depuis 1960; à ce jour, la théorie de la différentiation fractionnaire a connu un nombre important de contributions et d'applications dans la vie réelle, en raison de sa meilleure représentation du comportement de la mémoire longue et de la structure dimensionnelle infinie [23]. En effet, les systèmes physiques comme les systèmes thermiques [24, 25], chimiques [26], électrochimiques [27] et viscoélastiques [28] sont régis par des équations différentielles à dérivées non entières.

Actuellement, l'intérêt de la dérivation non entière ne cesse de grandir, notamment dans le domaine de l'automatique pour la modélisation, l'identification et la commande des systèmes.

Actuellement, l'intérêt de la dérivation non entière ne cesse de grandir. Notamment un nombre important de systèmes non linéaires présentent un comportement fractionnaire et leur identification reste une tâche importante et un défi de recherche actuel.

L'objectif principal de cette thèse est d'apporter une contribution à l'identification des modèles non linéaires d'ordre fractionnaire, en s'appuyant sur les systèmes régis par les modèles non linéaires de Wiener. Pour décrire au mieux notre contribution, la progression de la thèse est ponctuée par quatre chapitres dont le contenu est présenté ici d'une manière introductive.

Chapitre 1 : Systèmes non linéaires entiers

Une grande variété de systèmes non linéaires est rencontrée dans la réalité, dans ce premier chapitre, nous présenterons les principales classes dans le cas entier.

Notre attention s'est focalisée sur les modèles blocs orientés, en particulier sur le modèle de Wiener. Ce dernier, permet de reproduire fidèlement le comportement de plusieurs systèmes réels.

La description du modèle non linéaire de Wiener sera abordée. Par conséquent, nous introduirons le modèle de Wiener sous forme de représentation d'état non linéaire polynomial (PNLSS).

Chapitre 2 : Systèmes fractionnaires

La notion de différentiation d'ordre non entier ou fractionnaire, qui a intéressé la communauté scientifique et a fait l'objet de plusieurs travaux, sera considérée dans le deuxième chapitre. La représentation des systèmes d'ordre fractionnaire sera ainsi rappelée, dans le cas continu et le cas discret. Nous nous intéresserons aussi à la simulation de ces systèmes, qui se repose essentiellement sur l'approximation de l'opérateur de dérivation (intégration) fractionnaire.

Par la suite, la description du système de Wiener fractionnaire sera considérée, en se basant sur le type du modèle choisi pour les blocs caractérisant le système. En effet, le modèle de Wiener PNLSS fractionnaire (PNLFSS) et le modèle de Wiener décrit sous forme de régression fractionnaire.

Chapitre 3 : Identification des systèmes non linéaires entiers et fractionnaires

La première partie de ce chapitre s'intéressera à l'identification des systèmes non linéaires, dans le cas entier. Pour cela, une méthode d'erreur de sortie basée sur l'algorithme de Levenberg-Marquardt (LM) et une autre méthode heuristique : optimisation par essaim de particules (Particle Swarm Optimization (PSO)) modifiée qui est l'optimisation par essaim de particules à vitesse auto-adaptative (Self Adaptative velocity Particle Swarm Optimization (SAVPSO)) seront adaptées pour l'identification des systèmes de Wiener PNLSS.

La deuxième partie, d'autre part, consiste à développer des algorithmes afin d'identifier des systèmes non linéaires du type de Wiener PNLFSS, en se basant sur une méthode métaheuristique et une méthode d'optimisation non linéaire.

Enfin, pour l'identification du modèle de Wiener décrit sous forme de régression fractionnaire, l'algorithme PSO, basé sur le principe Key term, sera implémenté.

Chapitre 4 : Résultats de simulation et discussions

Le dernier chapitre, objet de notre contribution, se concentrera sur l'application des algorithmes développés dans le troisième chapitre dans différents exemples de simulation. Nous illustrerons dans ce chapitre les résultats de simulation obtenu, permettant de tester et de vérifier l'efficacité des algorithmes présentés précédemment.

Dans ce contexte, différents exemples de simulation seront donnés pour l'identification du modèle de Wiener PNLSS et sous forme de régression, dans le cas entier et dans le cas fractionnaire.

Enfin, une conclusion générale résume les études effectuées et les résultats obtenus et suggère des perspectives futures pour améliorer davantage notre contribution.

Chapitre 1

Systèmes non linéaires entiers

1.1 Introduction

L'étude des systèmes linéaires a connu un grand progrès grâce aux outils d'analyse disponibles (algèbre linéaire, équations différentielles, systèmes différentiels linéaires, etc ...) [29]. Cependant la majorité des systèmes réels ne vérifient pas la propriété de linéarité et les systèmes linéaires ne s'appliquent que dans un domaine de fonctionnement restreint et limité.

Par conséquent, des modèles non linéaires sont nécessaires pour décrire leur riche comportement dynamique.

La notion de système non linéaire est fondée sur le non respect du principe de superposition [30]. Cette définition, ou plutôt cette non-définition, explique la complexité et la diversité des systèmes non linéaires et des méthodes qui leur sont applicables. Il n'y a pas de théorie générale qui s'applique à ces systèmes, par conséquent, il existe plusieurs méthodes adaptées à certaines classes de ces systèmes dits non linéaires.

Historiquement, l'identification des systèmes non linéaires a mis l'accent sur quelques-uns des modèles et s'est focalisée sur quelques classes spécifiques qui peuvent être étroitement définies dans ce qui suit.

Ce chapitre est consacré à définir les systèmes non linéaires et de présenter les principales classes de ces systèmes.

1.2 Classes des modèles non linéaires

Les systèmes non linéaires peuvent être représentés sous différentes classes. En effet, certaines classes ont la forme générale, dans le cas discret, donnée comme suit :

$$y(k) = f(\phi(k), \theta) + e(k)$$
(1.1)

f(.) est une fonction, qui peut être, linéaire dans le cas des systèmes linéaires et non linéaire pour le cas des systèmes non linéaires.

 $\phi(k)$ est le vecteur de régression.

 θ est le vecteur des paramètres du modèle et e(k) le bruit.

Par conséquent, en fonction de la forme du vecteur ϕ , différentes structures de modèles non linéaires peuvent être définies telles que les modèles : NFIR, NARX, NARMAX, ...

1.2.1 Modèle NFIR

Dans les modèles FIR (Finite Impulse Response) et NFIR (Nonlinear Finite Impulse Response) le régresseur est seulement en fonction de u(k - i). Le modèle FIR peut être décrit par [31] :

$$y(k) = B(q)u(k) + e(k)$$
 (1.2)

avec B(q) un polynôme d'ordre n_B , défini par $B(q) = b_1 q^{-1} + \ldots + b_{n_B} q^{-n_B}$. Par conséquent,

$$y(k) = b_1 q^{-1} u(k) + b_2 q^{-2} u(k) + \ldots + b_{n_B} q^{-n_B} u(k) + e(k)$$
(1.3)

Le modèle NFIR peut être défini en utilisant une fonction non linéaire (f) pour l'équation (1.3), par conséquent, la sortie du modèle NFIR est :

$$y(k) = f(u(k-1), \dots, u(k-n_B), \theta) + e(k)$$
(1.4)

1.2.2 Modèle NARX

Le modèle ARX (Autoregressive with exogenous input) est donné comme suit [31] :

$$A(q)y(k) = B(q)u(k) + e(k)$$
 (1.5)

A(q) un polynôme d'ordre n_A et $A(q) = 1 + a_1, \dots, a_{n_A}$.

Ainsi, la sortie du modèle est représentée comme suit :

$$y(k) = b_1 u(k-1) + \ldots + b_{n_B} u(k-n_B) - a_1 y(k-1) - \ldots - a_{n_A} y(k-n_A)$$
(1.6)

Le modèle NARX (Nonlinear Auto-Regressive with Exogenous input) peut aussi être déduit en remplaçant la relation linéaire dans l'équation (1.6) par une fonction non linéaire, comme suit :

$$y(k) = f(u(k-1), \dots, u(k-n_B), y(k-1), \dots, y(k-n_A), \theta) + e(k)$$
(1.7)

f est une fonction non linéaire.

1.2.3 Modèles NARMAX

Dans le cas linéaire, la sortie du modèle ARMAX (Autoregressive Moving Average Model with exogenous inputs) est représentée par :

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{M(q)}{A(q)}e(k)$$
(1.8)

M(q) un polynôme d'ordre n_M .

Le modèle NARMAX (Nonlinear Autoregressive Moving Average Model with exogenous inputs), introduit en 1981 et développé dans [10,32–34], est une extension du modèle ARMAX linéaire. Ce type de modèle permet de représenter une grande classe de systèmes non linéaires [35], et peut être défini par l'équation (1.9) : [31]

$$y(k) = f(y(k-1), y(k-2), \dots, y(k-n_A),$$

$$u(k-1), u(k-2), \dots, u(k-n_B),$$

$$e(k-1), e(k-2), \dots, e(k-n_M), \theta) + e(k)$$
(1.9)

f(.) est une fonction non-linéaire.

De nombreux types de modèles non linéaires, y compris les modèles NFIR (Nonlinear Finite Impulse Response), NARX (Nonlineaire Autoregressive with exogenous input) peuvent facilement être considérés comme des cas particuliers du modèle NARMAX.

1.2.4 Modèle NBJ

Le modèle BJ (Box-Jenkins) fournit une autre forme en multipliant le bruit par M(p)/N(p) [31]. Le modèle BJ est défini comme suit :

$$y(k) = \frac{B(q)}{H(q)}u(k) + \frac{M(q)}{N(q)}e(k)$$
(1.10)

avec B, H, M et N des polynômes d'ordre n_B, n_H, n_M et n_N respectivement. Avec par example :

$$B(q) = b_1 q^{-1} + \ldots + b_{n_B} q^{-n_B}$$

$$H(q) = 1 + h_1 q^{-1} + \ldots + h_{n_H} q^{-n_H}$$

$$M(q) = 1 + m_1 q^{-1} + \ldots + m_{n_M} q^{-n_M}$$

$$N(q) = 1 + n_1 q^{-1} + \ldots + n_{n_N} q^{-n_N}$$
(1.11)

Le modèle NBJ (Nonlinear Box-Jenkins) peut être déduit en remplaçant la relation linéaire dans l'équation (1.10) par une fonction non linéaire, comme suit [31] :

$$y(k) = f(u(k-1), \dots, u(k-n_B-n_N), \dots, u(k-n_B),$$

$$y(k-1), \dots, y(k-n_N), \dots, y(k-n_H), \dots, y(k-n_H-n_N),$$

$$e(k-1), \dots, e(k-n_M), \dots, e(k-n_H), \dots, e(k-n_H-n_M), \theta)$$
(1.12)

De plus, il existe différentes formes utilisées pour la représentation des systèmes non linéaires; conduisant à d'autres modèles non linéaires, tels que les séries de Volterra et les modèles non linéaires de fonction de base orthonormées (NOBF).

1.2.5 Modèles non linéaires des fonctions de base orthonormées

Les modèles non linéaires de fonctions de bases orthonormées (Orthonormal Basis function models (OBF)) peuvent être considérés comme la généralisation du modèle FIR et NFIR.

Dans le domaine temporel, la sortie y(k) peut être représentée par une somme pondérée de ces fonctions de base, à savoir la réponse impulsionnelle des filtres. En effet, pour le modèle FIR (Finite Impulse Response), cette somme pondérée est particulièrement simple car cette fonction de base est l'impulsion de Dirac, mais l'ordre du modèle doit être choisi très élevé. Dans les modèles OBF, les filtres triviaux q^i sont remplacés par des filtres orthonormés plus généraux et plus complexes $l_i(q)$ [31].

Les fonctions de bases orthonormées telles que Laguerre et Kautz sont très répandues, et un traitement spécial est nécessaire pour concevoir les paramètres caractéristiques des fonctions de base. Pour le cas utilisant ces filtres, le vecteur de régression pour le modèle NOBF a la forme [18] :

$$\phi(q) = [l_1(q)u(k) \quad l_2(q)u(k) \quad \dots \quad l_i(q)u(k)]^T$$
(1.13)

Par exemple, le i $^{\grave{e}me}$ filtre de Laguerre est donné par :

$$l_i(q,g) = \frac{\sqrt{1-g^2}}{q-g} \left(\frac{1-gq}{q-g}\right)^{i-1}$$
(1.14)

g un paramètre de valeur réelle et $\left|g\right|<1.$

Alternativement, le i^{eme} filtre de Laguerre peut être calculé de manière récursive comme suit :

$$l_i(q,g) = \frac{1-qg}{q-g} l_{i-1}(q,g)$$
(1.15)

1.2.6 Modèle Séries de Volterra

Volterra, en (1959), fut le premier à introduire la série de Volterra [36]; puis Rugh (1981) [37] pour l'aspect mathématique et Hélie et Roze (2008) [38] pour une application à un système acoustique non linéaire.

Les séries de Volterra sont une extension de l'intégrale de convolution linéaire et représentent les systèmes légèrement non linéaires comme une série de multi-sommations, ou intégrales dans le cas de temps continu, des noyaux de Volterra (h_i) et des entrées [35].

Afin d'arriver à l'expression de la sortie des séries de Volterra, considérons un système linéaire caractérisé par sa réponse impulsionnelle h_1 et un signal d'entrée u(k). La sortie $y_1(k)$ est exprimée comme le produit de convolution :

$$y_1(k) = \Sigma_0^{+\infty} h_1(\tau_1) u(k - \tau_1)$$
(1.16)

 h_1 est considéré le 1-ème noyau de Volterra.

La sortie y_2 , du 2-ème noyau de Volterra, est :

$$y_2(k) = \Sigma_0^{+\infty} \Sigma_0^{+\infty} h_2(\tau_1, \tau_2) u(k - \tau_1) u(k - \tau_2)$$
(1.17)

Pour un système non linéaire, une structure de modèle de la série Volterra est écrite comme la somme infinie :

$$y(k) = \Sigma_{K=1}^{+\infty} y_K(k) = y_1(k) + y_2(k) + \ldots + y_K(k) + \ldots$$
(1.18)

où y_K , la sortie du K-ème noyau de Volterra (h_K) , qui est écrit comme la généralisation du produit de convolution (l'équation (1.16)) au K-ème ordre :

$$y(k)_{K} = \Sigma_{0}^{+\infty} \dots \Sigma_{0}^{+\infty} h_{K}(\tau_{1}, \dots, \tau_{K}) \prod_{i=1}^{K} u(k - \tau_{i})$$
(1.19)

Par conséquent, en combinant les équations (1.18) et (1.19), la sortie y(k) d'un système non linéaire peut être exprimée, dans le cas continu, par :

$$y(k) = \Sigma_{K=1}^{+\infty} \Sigma_0^{+\infty} \dots \Sigma_0^{+\infty} h_K(\tau_1, \dots, \tau_K) \prod_{i=1}^K u(k - \tau_i)$$
(1.20)

avec h_K est le K-ème ordre du noyau de Volterra.

Des études approfondies des modèles des séries de Volterra montrent que le principal inconvénient de cette approche est le grand nombre de paramètres impliqués, donnant lieu à un modèle de grande complexité [39]. Par conséquent, les difficultés pratiques à mesurer et à étendre l'estimation du noyau à des ordres supérieurs au troisième ordre [40]; ce qui limite leurs utilisations.

1.2.7 Modèles blocs orientés

Plus récemment, les recherches menées sur l'identification se sont tournées vers un autre outil puissant pour la modélisation et l'optimisation des systèmes non linéaires, à savoir les modèles blocs orientés [5,41,42]. Ce type de modèles peut être décrit par des connexions de groupes d'éléments ou de blocs dynamiques linéaires et blocs non linéaires statiques. Ainsi, cette classe de systèmes non linéaires est définie par la séparation du comportement statique non linéaire et du comportement linéaire à temps invariant dans différents blocs. En effet, l'interconnexion de ces derniers conduit à plusieurs structures. Parmi les modèles de ce type on peut citer les suivants :

1.2.7.1 Modèle de Hammerstein

Dans le modèle de Hammerstein, une non-linéarité statique est suivie d'une dynamique linéaire, comme montré sur la figure (1.1) [2,7].



FIGURE 1.1: Modèle de Hammerstein

Le modèle de Hammerstein a été utilisé pour modéliser des processus biologiques [43,44], des processus chimiques [45], dans des applications de traitement du signal [46] et pour les problèmes de contrôle [47].

1.2.7.2 Modèle de Wiener

La structure des systèmes Wiener, proposé en 1958 par Wiener, est représentée sur la figure (1.2); Ce modèle a deux blocs de même type que le modèle de Hammerstein, mais dans l'ordre inverse, où le bloc dynamique linéaire est suivi d'un bloc statique non linéaire [7].

Ce modèle peut approximer arbitrairement la plupart des processus dynamiques linéaires avec la sortie obtenue en utilisant un capteur non linéaire comme cité dans [16,17]; ou lorsque le système dynamique linéaire est connecté à une non-linéarité sans mémoire



FIGURE 1.2: Modèle de Wiener

comme une zone morte ou une saturation [48].

En outre, il est utilisé pour décrire un large spectre de procédés industriels tels que les systèmes de contrôle du pH [49,50], échangeurs de chaleur [21,51], ainsi qu'un écoulement de fluide [52], colonne de distillation [53] et exemples biologiques [54], ... etc.

1.2.7.3 Modèle de Wiener-Hammerstein

Les systèmes de Wiener-Hammerstien sont des systèmes dynamiques caractérisés par un raccordement en série de trois blocs : un système dynamique linéaire, une non-linéarité statique et un autre système dynamique linéaire différent, comme montré dans la figure (1.3).



FIGURE 1.3: Modèle de Wiener-Hammerstein

1.2.7.4 Modèle de Hammerstein-Wiener

Un autre modèle bloc est le Modèle de Hammerstein-Wiener. Ce dernier est présenté par la figure (1.4).



FIGURE 1.4: Modèle de Hammerstein-Wiener

L'idée de base de l'approche de la modélisation par blocs structurés consiste à identifier les blocs individuels dans le système en se basant uniquement sur les mesures d'entrée-sortie externes; c'est-à-dire sans accès à des signaux internes de manière à maintenir la relation avec le système sous-jacent et les composants représentés par chaque bloc [35].

La majorité des travaux concernant les systèmes non linéaires de type blocs orientés, ont considéré que la partie dynamique linéaire est paramétrique. Par conséquent, le bloc linéaire est décrit par une fonction de transfert d'ordre connu [55], ou une représentation d'état avec la connaissance du degré de la dynamique [56]. La représentation non paramétrique a également été utilisée; la partie linéaire a donc été choisie comme FIR, IIR et comme la réponse fréquentielle [57].

Les modèles représentants les éléments statiques non linéaires peuvent être sous différentes formes, telles que les formes polynômiales de degré connues [58], les polynômes de Laguerre [59], les splines cubiques [60], une zone morte ou une saturation [48].

En effet, il est cité que l'avantage majeur de l'utilisation des modèles polynomiaux, représentants la partie non linéaire, est l'optimisation efficace des paramètres.

En outre, l'inconvénient des principales approches proposées dans la littérature est la paramétrisation des blocs linéaires et non linéaires en utilisant des modèles d'entrées-sorties, ce qui induit la dépendance de la méthode d'identification par rapport au modèle d'ordre choisi. L'utilisation de la représentation de l'espace d'état, comme partie linéaire, peut être une solution à ce problème. De plus, les modèles d'espace d'état sont souvent préférés aux modèles d'entrées-sorties pour traiter aussi bien les systèmes multivariables que les systèmes SISO.

Dans cette thèse, l'identification des systèmes non linéaires de type blocs orientés est

considérée; en particulier, notre étude se basera sur le modèle de Wiener.

Dans ce but, le modèle d'espace d'état non linéaire polynomial (PNLSS) qui est la généralisation du modèle d'espace d'état au cas non linéaire a été implémenté pour la description du système de Wiener.

1.3 Représentation d'état non linéaire polynomiale

La forme générale du modèle de l'espace d'état non linéaire discret est sous la forme :

$$\begin{cases} x(k+1) = f(x(k), u(k), \theta) \\ y(k) = g(x(k), u(k), \theta) \end{cases}$$
(1.21)

Afin d'appliquer une expansion fonctionnelle des fonctions f(.) et g(.), nous avons opté pour une approche polynomiale. Dans ce cas, la représentation de l'espace d'état non linéaire polynomiale (Polynomial Non-linear State Space (PNLSS)) se compose d'un modèle linéaire classique de l'espace d'état avec des termes non linéaires. Son principal avantage est sa possibilité de décrire une classe très grande des systèmes non linéaires, et son application facile dans le cas multivariable.

Ce modèle est défini comme suit [9, 42, 61]:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases}$$
(1.22)

Les coefficients du sous-ensemble linéaire sont donnés par les matrices $A \in \mathbb{R}^{n_a * n_a}$ et $B \in \mathbb{R}^{n_a * n_u}$ dans l'équation d'état, $C \in \mathbb{R}^{n_y * n_a}$ et $D \in \mathbb{R}^{n_y * n_u}$ dans l'équation de sortie. Les vecteurs $\zeta(t) \in \mathbb{R}^{n_{\zeta}}$ et $\eta(t) \in \mathbb{R}^{n_{\eta}}$ contiennent des monômes non linéaires dans x(t) et u(t) du degré deux jusqu'à un degré choisi r. Les coefficients liés à ces termes non linéaires sont donnés par les matrices $E \in \mathbb{R}^{n_a * n_{\zeta}}$ et $F \in \mathbb{R}^{n_a * n_{\eta}}$.

Notons que les monômes du degré un sont inclus dans la partie linéaire de la structure du modèle PNLSS.

Dans la littérature, il est montré que certains modèles non linéaires de structure blocs orientés peuvent être décrits par le modèle PNLSS.

1.4 Description du modèle de Wiener sous forme PNLSS

Dans cette section, on considère la description du modèle de Wiener. Dans ce but, l'approche PNLSS est utilisée. Cependant on se limitera aux systèmes SISO (une seule E/S).

Dans ce cas, le système de Wiener est composé d'un bloc dynamique linéaire modélisé par un modèle d'espace d'état d'ordre connu n_a (équation 1.23), suivi d'un bloc non linéaire statique (équation 1.24) qui est un polynôme non linéaire d'ordre connu r (équation 1.25) [62,63] comme le montre la figure (1.5).

FIGURE 1.5: Modèle de Wiener PNLSS

avec

$$\begin{cases} x(k+1) = A_0 x(k) + B_0 u(k) \\ v(k) = C_0 x(k) + D_0 u(k) \end{cases}$$
(1.23)

$$f(v(k)) = p_1 f_1(v(k)) + p_2 f_2(v(k)) + \ldots + p_r f_r(v(k))$$
(1.24)

$$y(k) = \sum_{i=1}^{r} p_i v^i(k)$$
 (1.25)

En remplaçant la $2^{\grave{e}me}$ équation de (1.23) dans (1.25), nous avons la sortie du modèle de Wiener est donnée par :

$$\begin{cases} x(k+1) = A_0 x(k) + B_0 u(k) \\ v(k) = C_0 x(k) + D_0 u(k) \\ y(k) = \sum_{i=1}^r p_i \left(C_0 x(k) + D_0 u(k) \right)^i \end{cases}$$
(1.26)

$$\begin{cases} x(k+1) = A_0 x(k) + B_0 u(k) \\ y(k) = p_1 \left(C_0 x(k) + D_0 u(k) \right) + \sum_{i=2}^r p_i \left(C_0 x(k) + D_0 u(k) \right)^i \end{cases}$$
(1.27)

Dans le cas où $n_a = 2$ et r = 2 on a :

$$\begin{cases} x(k+1) = A_0 x(k) + B_0 u(k) \\ y(k) = p_1 C_0 x(k) + p_1 D_0 u(k) + p_2 \left(C_0 x(k) + D_0 u(k) \right)^2 \end{cases}$$
(1.28)

En considérant $C_0 = \begin{bmatrix} c_1 & c_2 \end{bmatrix}$, $D_0 = d$ et $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, la sortie du système de l'équation (1.28) est donnée par : (afin de ne pas surcharger les équations, la variable k est négligée)

$$y = p_1 v^1 + p_2 v^2$$

= $p_1 (C_0 x + D_0 u) + p_2 (C_0 x + D_0 u)^2$ (1.29)

 et

$$y = p_1 (c_1 x_1 + c_2 x_2 + du) + p_2 (c_1 x_1 + c_2 x_2 + du)^2$$
(1.30)

En appliquant l'expansion multinomiale, nous obtenons l'équation suivante :

$$y = p_1 c_1 x_1 + p_1 c_2 x_2 + p_1 du + p_2 \left(c_1^2 x_1^2 + 2c_1 c_2 x_1 x_2 + 2c_1 dx_1 u + c_2^2 x_2^2 + 2c_2 dx_2 u + d^2 u^2 \right)$$
(1.31)

La sortie (équation 1.31) peut être reformulé et donne l'écriture sous forme vectorielle suivante :

$$y = \left[\begin{array}{ccc} p_{1}c_{1} & p_{1}c_{2} \end{array} \right] \left[\begin{array}{c} x_{1} \\ x_{2} \end{array} \right] + \left[p_{1}d \right] u + \left[\begin{array}{ccc} p_{2}c_{1}^{2} & 2p_{2}c_{1}c_{2} & 2p_{2}c_{1}d & p_{2}c_{2}^{2} & 2p_{2}c_{2}d & p_{2}d^{2} \end{array} \right] \left[\begin{array}{c} x_{1}^{2} \\ x_{1}x_{2} \\ x_{2}u \\ x_{2}^{2} \\ x_{2}u \\ u^{2} \end{array} \right]$$

$$(1.32)$$

Par conséquent, les équations PNLSS analogue qui décrivent le comportement du système
de Wiener sont :

$$\begin{cases} x(k+1) = A_0 x(k) + B_0 u(k) \\ y = p_1 \begin{bmatrix} c_1, & c_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + p_1 [d] u + p_2 \begin{bmatrix} c_1^2 & 2c_1c_2 & 2c_1d & c_2^2 & 2c_2d & d^2 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_1u \\ x_2^2 \\ x_2u \\ u^2 \end{bmatrix}$$
(1.33)

Les matrices du système de Wiener sous forme PNLSS (équation 1.22) s'en déduisent comme suit :

$$A = A_0$$

$$B = B_0$$

$$C = p_1 C_0$$

$$D = p_1 D_0$$

$$E = 0$$

$$F = [p_2 [C_0 \quad D_0]^2 \quad \dots \quad p_r [C_0 \quad D_0]^r]$$

$$et$$

$$F = [p_2 [Z]^2 \quad \dots \quad p_r [Z]^r]$$

$$avec$$

$$Z = [C_0 \quad D_0]$$

(1.34)

avec

$$\begin{cases} \zeta(k) = 0 \\ \eta(k) = [\xi^{2}(k) \dots \xi^{r}(k)] \\ \xi(k) = [x(k) u(k)] \end{cases}$$
(1.35)

Le modèle de Wiener, au cas entier général, est décrit par l'équation suivante :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases}$$
(1.36)

Par conséquent, nous pouvons conclure qu'un système de Wiener avec une non-linéarité continue peut-être représenté par l'approche PNLSS.

1.5 État de l'art sur l'identification des systèmes non linéaires entiers

L'identification des systèmes non linéaires est un domaine de recherche en évolution rapide. Ainsi, de nombreuses méthodes et algorithmes d'identification ont été développés, en se concentrant sur les classes principales des modèles non linéaires. En effet, on peut citer l'identification du modèle NARMAX [35,64] et des séries de Volterra [39], ainsi que l'identification de l'espace d'état non linéaire [65].

En outre, le problème d'identification des systèmes blocs structurés a connu beaucoup de contributions; dans ce contexte, la méthode du maximum de vraisemblance [66] et la meilleure approximation linéaire (Best linear approximation) [61] ont été utilisés pour identifier le modèle de Wiener-Hammerstein.

De plus, plusieurs travaux ont abordé le problème l'identification des systèmes Hammerstein. En effet, dans [67–69], les auteurs ont utilisé l'algorithme des moindres carrés récursif pour identifier les systèmes non linéaires Hammerstein. Dans [70] les auteurs abordent le principe d'identification hiérarchique, en appliquant l'algorithme des moindres carrés hiérarchique (HLS) pour l'estimation des paramètres du système Hammerstein du type CAR (Controlled Auto Regressive). De plus, l'optimisation par essaim de particules a été proposée dans [71,72] et l'algorithme de Levenberg-Marquardt dans [73], afin d'identifier un système Hammerstein.

Dans la littérature, le problème d'identification des systèmes de Wiener représente un domaine de recherche important où divers travaux ont été réalisés. En effet, plusieurs méthodes ont été proposées : en guise d'exemple, la méthode étudiée dans [74] utilise le principe de sur-paramétrisation basée sur la méthode de programmation semi-définie (SDP), dans [75], les auteurs emploient la méthode des moindres carrés. De leurs coté, les auteurs [76,77] ont fait recours à la méthode de maximum de vraisemblance pour identifier les modèles de Wiener. Dans le même cadre, l'algorithme des moindres carrés [78], la méthode du gradient [79] et la méthode du gradient hiérarchique [80] ont été considérées. En outre, des algorithmes itératifs basés sur l'algorithme des moindres carrés [81] et l'algorithme de Newton [82] ont aussi été utilisés. De plus, l'identification des modèles de Wiener inclut aussi l'utilisation de l'algorithme des moindres carrés récursifs, avec la partie non linéaire supposée être inversible [15, 83, 84].

1.6 Conclusion

Dans ce chapitre, les principales classes des systèmes non linéaires ont été présentées. En effet, nous avons défini les principales classes des modèles des systèmes non linéaires utilisés dans la littérature. Notre attention s'est ainsi focalisée sur la classe des modèles blocs orientés, en particulier le modèle de Wiener, qui sera considéré tout au long de ce travail. Il est décrit dans ce chapitre selon une approches adaptée pour la modélisation et l'identification dans le cas entier. Par la suite, un état de l'art de l'identification des systèmes non linéaires a aussi été donné, pour le cas entier.

Le chapitre suivant, sera consacré aux systèmes d'ordre fractionnaire non linéaires.

Chapitre 2

Systèmes fractionnaires

2.1 Introduction

Le calcul non entier ou fractionnaire a beaucoup intéressé les chercheurs dans plusieurs domaines, particulièrement en mathématiques et en ingénierie.

L'origine du calcul fractionnaire est une question clé de Leibniz en 1695 [85–87]; qui a proposé, dans une lettre a L'Hospital, de généraliser sa formule pour la dérivée $n^{i eme}$, $\frac{d^n y}{dx^n} \equiv D^n y$ pour n > 0 et il posa la question : et si n = 1/2?

Par la suite, on retrouve la contribution de plusieurs mathématiciens connus tels qu'Euler ou Lagrange au XVIII^e siècle, Laplace, Fourier, Liouville (1832, 1837) ou Riemann (1847) au XIX^e siècle, ainsi que Grünwald (1867) et Letnikov (1868) dans la seconde moitié du même siècle.

Le calcul fractionnaire a en effet permis une meilleure représentation et compréhension des phénomènes physiques à mémoire longue et de dimension infinie; tels que l'électrochimie [88], l'électromagnétisme et les machines électriques [89], les systèmes thermiques et la conduction thermique [90,91], la transmission et l'acoustique [92,93], les matériaux viscoélastiques [94] et la robotique [95].

Dans la théorie du contrôle, les premières contributions ont fourni des généralisations de méthodes d'analyse classiques pour les systèmes d'ordre fractionnaire (fonction de transfert, réponse fréquentielle, etc.) [96–98]. Récemment, cet outil a trouvé un regain d'application dans la modélisation des phénomènes physiques et des systèmes réels. Une riche source de références sur le calcul d'ordre fractionnaire peut être fournie par plusieurs ouvrages récents, tel que : [23, 85–87, 97, 99, 100].

L'objectif de ce chapitre est de présenter quelques notions mathématiques du calcul fractionnaire de base, nécessaires pour la suite de notre étude, puis de définir le système de Wiener fractionnaire.

2.2 Opérateurs d'ordre fractionnaire

Le développement du calcul d'ordre fractionnaire est basé sur la généralisation des opérateurs d'intégration et différentiation en un seul opérateur fondamental D_t^{α} , où α est l'ordre de l'opérateur qui est un nombre non entier [101]. L'opérateur integro différentiel continu est défini par :

$$D_t^{\alpha} = \begin{cases} \frac{d^{\alpha}}{dt^{\alpha}} & pour \ \alpha > 0 \\ 1 & pour \ \alpha = 0 \\ I_t^{\alpha} & pour \ \alpha < 0 \end{cases}$$
(2.1)

Les opérateurs D_t^{α} et I_t^{α} représentent respectivement l'opérateur de la dérivation et d'intégration d'ordre non entier.

L'intégration et la dérivation fractionnaires sont la généralisation de l'intégration et de la dérivation classique entière à des ordres quelconques non entiers irrationnels ou complexes.

En effet, en 1947, Riemann a proposé de remplacer la fonction factorielle par la fonction Gamma, à un nombre non entier réel $\alpha \in R_+^*$.

En outre, la généralisation de la dérivation entière à des ordres non entiers a donné le jour à plusieurs définitions, qui ne sont pas toutes équivalentes.

Dans le cas discret, comme dans le cas continu, il existe différentes définitions de l'opérateur de différenciation fractionnaire. Les plus utilisées sont celles de Riemann-Liouville, Caputo et celle de Grunwald-Letnikov [102], qui sont représentées dans ce qui suit.

2.2.1 Différence fractionnaire au sens de Riemann-Liouville

Soit f une fonction à valeur réelle définie sur $(N)_a$, avec $a \in R$. La différence au sens de Riemann-Liouville est donnée, pour $0 < \alpha < 1$ et $f : N_a \longrightarrow R$, par l'équation suivante :

$$(_{a}\Delta_{h}^{\alpha}f)(t) = \Delta_{h}\left(_{a}\Delta_{h}^{-(1-\alpha)}f\right)(t)$$
(2.2)

 $t \in N_{a+(1-\alpha)}$, avec $N_a = \{a, a+1, \ldots\}$.

2.2.2 Différence fractionnaire au sens de Caputo

Caputo, de son coté, a aussi défini les différences d'ordre fractionnaire comme suit; Pour $0 < \alpha < 1$ et $f : N_a \longrightarrow R$.

La différence de type Caputo est définie comme suit :

$$\begin{aligned} \left(_{a}\Delta_{C}^{\alpha}f\right)(t) &= \left(_{a}\Delta^{-(1-\alpha)}\Delta f\right)(t) \\ &= \frac{1}{\Gamma\left(1-\alpha\right)}\Sigma_{s=a}^{t-(1-\alpha)}\left(t-s-1\right)\left(\Delta f\right)(s) \end{aligned}$$

$$(2.3)$$

avec $\Delta f(s) = f(s+1) - f(s)$ la différence classique.

 $t \in N_{a+(1-\alpha)}$, avec $N_a = \{a, a+1, \ldots\}$.

 Γ est la fonction Gamma d'Euler.

2.2.3 Différence fractionnaire au sens de Grunwald-Letnikov

La définition proposée par GL consiste la généralisation de la définition de la dérivée entière d'une fonction temporelle f à des dérivées d'ordre réelle $\alpha \in R_+^*$. Soit le temps d'échantillonnage t = kh pour k = 0, 1, 2, ... et h la période d'échantillonnage qu'on suppose, sans perte de généralité, dans la suite égale à 1 (h = 1). La différence finie d'ordre 1 est donnée comme suit :

$$\Delta^{1} f(k+1) = f(k+1) - f(k)$$
(2.4)

L'équation (2.4) représente aussi l'approximation discret d'Euler de la dérivée d'ordre entier $\frac{df(t)}{t}$; la généralisation à un ordre $\alpha \in R_+^*$ est formulée comme suit [103] :

$$\Delta^{\alpha} f(kh) = \frac{1}{h^{\alpha}} \sum_{j=0}^{k} (-1)^{j} \begin{pmatrix} \alpha \\ j \end{pmatrix} f((k-j)h)$$
(2.5)

 Δ est l'opérateur de dérivation fractionnaire discret avec un temps initial égal à zéro. α est une valeur fractionnaire avec $\alpha \in R_+^*$.

La notation
$$\begin{pmatrix} \alpha \\ j \end{pmatrix}$$
 désigne le binôme de Newton défini par :

$$\begin{pmatrix} \alpha \\ j \end{pmatrix} = \begin{cases} 1, & for \quad j = 0\\ \frac{\alpha(\alpha-1)\dots(\alpha-j+1)}{j!}, & for \quad j > 0 \end{cases}$$
(2.6)

On note que pour la dérivée entière, la dérivée d'ordre 1, mis à part j = 0 et j = 1, les coefficients de pondération $(-1)^j \begin{pmatrix} \alpha \\ j \end{pmatrix}$ sont nuls; impliquant ainsi une caractérisation locale de la fonction.

Par contre, pour des ordres non entiers, ces coefficients de pondération ne s'annulent pas. De plus, la dérivée non entière à chaque instant est une combinaison linéaire de toutes les valeurs de la fonction f((k - j)h), j = 0, 1, ..., k. En effet, la dérivée fractionnaire d'une fonction à un instant t prend en compte les valeurs de la fonction à tous les temps passés, cela montre qu'à l'inverse de la dérivée entière, la dérivée non entière donne une caractérisation globale de la fonction. Par conséquent, les systèmes non entiers sont souvent assimilés à des systèmes à mémoire longue.

Dans cette étude, la définition attribuée à GL, qui est la plus adéquate pour la simulation des systèmes discrets d'ordre fractionnaire, est utilisée pour les calculs numériques.

2.3 Modèles des systèmes d'ordre fractionnaire

La représentation et la description du comportement d'un système d'ordre fractionnaire peut se faire en utilisant trois modèles [101, 104] qui sont :

- Équation différentielle.
- Fonction de transfert .
- Représentation d'état.

Dans ce qui suit, nous développerons, plus en détails, ces représentations dans le cas de systèmes continus et discrets.

2.3.1 Cas continu

Pour cette section, nous nous focaliserons sur la représentation des systèmes d'ordre fractionnaire évoluant en temps continu.

2.3.1.1 Équation différentielle d'ordre fractionnaire

La modélisation du comportement dynamique de plusieurs systèmes peut se faire par des équations différentielles comprenant des dérivées d'ordre fractionnaire.

Un système mono-variable (SISO) peut être identifié par l'équation différentielle d'ordre fractionnaire suivante [104]

$$y(t) + \sum_{i=1}^{n} a_i D^{\alpha_i} y(t) = \sum_{j=1}^{m} b_j D^{\beta_j} u(t) + b_0 u(t)$$
(2.7)

avec $u(t) \in R$ et $y(t) \in R$ sont l'entrée et la sortie du système respectivement. $a_i, b_j \in R. \ \alpha_i, \beta_j \in R^+$ sont les ordres de dérivation, avec $0 < \alpha_1 < \alpha_2 < \ldots < \alpha_n$, $0 < \beta_1 < \beta_2 < \ldots < \beta_m$.

Lorsque les ordres de dérivation α_i, β_j sont tous multiples d'un même nombre réel α tel que $\alpha_i = i\alpha$; $\beta_j = j\alpha$, le système non entier est alors dit commensurable et l'équation (2.7) se réécrit comme suit :

$$y(t) + \sum_{i=1}^{n} a_i D^{i\alpha} y(t) = \sum_{j=1}^{m} b_j D^{j\alpha} u(t) + b_0 u(t)$$
(2.8)

2.3.1.2 Fonction de transfert non entière

L'application de la transformée de Laplace sur l'équation différentielle généralisée (2.7), en considérant les conditions initiales nulles, permet de déduire la fonction de transfert généralisée (2.9) :

$$G(s) = \frac{y(s)}{u(s)} = \frac{b_0 + \sum_{j=1}^m b_j s^{\beta_j}}{1 + \sum_{i=1}^n a_i D^{\alpha_i}}$$
(2.9)

Dans le cas commensurable, la fonction de transfert s'écrit :

$$G(s) = \frac{b_0 + \sum_{j=1}^{m} b_j s^{j\alpha}}{1 + \sum_{i=1}^{n} a_i D^{i\alpha}}$$
(2.10)

Pour le cas multivariable, le système peut être décrit par une matrice de fonctions de transfert non entière, ou un système d'équations différentielles fractionnaires.

Dans le cas entier, la dimension n correspond à l'ordre du polynôme (puissance maximale du polynôme), dans le cas des systèmes fractionnaires; le terme ordre du système est remplacé par le terme dimension du système, et les pôles du système sont les racines du polynôme du dénominateur.

2.3.1.3 Représentation d'état des systèmes non entiers

Le modèle d'état d'un système fractionnaire linéaire, continu invariant multi variable, s'écrit sous la forme :

$$\begin{cases} D^{(\alpha)}x(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$
(2.11)

avec $u \in R, y \in R, \alpha \in R^+$; le vecteur d'ordre fractionnaire est :

$$(\alpha) = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{n_a} \end{bmatrix}^T$$
(2.12)

 et

$$D^{(\alpha)}x = [D^{\alpha_1}x_1 \quad D^{\alpha_2}x_2 \quad \dots \quad D^{\alpha_n}x_{n_a}]^T$$
(2.13)

Dans le cas d'un système d'ordre fractionnaire commensurable, tous les états $x_i(t)$ sont dérivés à un même ordre non entier α :

$$D^{(\alpha)}x = \begin{bmatrix} D^{\alpha}x_1 & D^{\alpha}x_2 & \dots & D^{\alpha}x_{n_a} \end{bmatrix}^T$$
(2.14)

2.3.2 Cas discret

Dans cette partie, les modèles représentant les systèmes d'ordre fractionnaire sont considérés dans le cas discret.

2.3.2.1 Fonction de transfert fractionnaire discrète

Une fonction de transfert fractionnaire au cas discret, notée G(z), est décrite par :

$$G(z) = \frac{B(z)}{A(z)} \tag{2.15}$$

avec

$$\begin{cases} A(z) = 1 + \sum_{i=1}^{n_a} a_i z^{-\alpha_i} \\ = 1 + a_1 z^{-\alpha} + \ldots + a_{n_a} z^{-n_a \alpha_{n_a}} \end{cases}$$
(2.16)

$$\begin{cases}
B(z) = \sum_{j=1}^{n_b} b_j z^{-\alpha_j} \\
= b_1 z^{-\alpha} + \ldots + b_{n_b} z^{-n_b \alpha_{n_b}}
\end{cases}$$
(2.17)

 $a = [a_1 \dots a_{n_a}]$ et $b = [b_1 \dots b_{n_b}]$ sont des vecteurs de coefficients de longueur respectivement n_a et n_b .

 α_i et α_j sont les ordres fractionnaires. Dans le cas commensurable, ces ordres fractionnaires sont des multiples de la même base α ($\alpha_i = i\alpha$ et $\beta_j = j\alpha$ pour $i = 1, ..., n_a$ et $j = 1, ..., n_b$).

La fonction de transfert d'ordre fractionnaire commensurable est donnée comme suit :

$$G(z) = \frac{b_1 z^{-\alpha} + \dots + b_{n_b} z^{-n_b \alpha}}{1 + a_1 z^{-\alpha} + \dots + a_{n_a} z^{-n_a \alpha}}$$
(2.18)

2.3.2.2 Équation aux différences fractionnaire dans le cas discret

A partir de la fonction de transfert (équation 2.15), on a :

$$y(k) = \frac{B(z)}{A(z)}u(k) \tag{2.19}$$

L'équation (2.19) peut être réécrite comme suit :

$$A(z)y(k) = B(z)u(k)$$
(2.20)

 et

$$(1 + \sum_{i=1}^{n_a} a_i z^{-\alpha_i}) y(k) = \sum_{j=1}^{n_b} b_j z^{-\beta_j} u(k)$$
(2.21)

Dans le cas d'ordre commensurable, on a :

$$y(k) = -\sum_{i=1}^{n_a} a_i (z^{-i})^{\alpha} y(k) + \sum_{j=1}^{n_b} b_j (z^{-j})^{\alpha} u(k)$$
(2.22)

On note que z^{-1} est l'opérateur de retard, avec :

 $z^{-1}y(k) = y(k-1)$

En utilisant l'opérateur de différence d'ordre fractionnaire Δ , on obtient la relation suivante :

$$y(k) = -a_1 \Delta^{\alpha} y(k-1) - \dots - a_{n_a} \Delta^{\alpha} y(k-n_a) + b_1 \Delta^{\alpha} u(k-1) + \dots + b_{n_b} \Delta^{\alpha} u(k-n_b)$$
(2.23)

avec

$$\Delta^{\alpha} y(k) = \frac{1}{h^{\alpha}} \sum_{j=0}^{k} (-1)^{j} \begin{pmatrix} \alpha \\ j \end{pmatrix} y((k-j)h)$$
(2.24)

 et

$$\Delta^{\alpha} u(k) = \frac{1}{h^{\alpha}} \sum_{j=0}^{k} (-1)^{j} \begin{pmatrix} \alpha \\ j \end{pmatrix} u((k-j)h)$$
(2.25)

2.3.2.3 Représentation d'état des systèmes d'ordre fractionnaire

Dans la littérature, il est cité qu'il existe plusieurs définitions de la représentation de l'espace d'état fractionnaire [102].

Parmi ces définitions, nous considérons le modèle d'état fractionnaire à temps discret, utilisé dans [105], donné par :

$$\begin{cases} \Delta^{(\alpha)}x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$
(2.26)

 $\Delta^{(\alpha)}x(k)$ est le vecteur d'espace d'état fractionnaire exprimé comme suit :

$$\Delta^{(\alpha)}x(k) = [\Delta^{\alpha_1}x_1(k) \quad \Delta^{\alpha_2}x_2(k) \dots \quad \Delta^{\alpha_n}x_n(k)]^T$$
(2.27)

Dans le cas commensurable, on a le vecteur des variables d'état est :

$$\Delta^{(\alpha)}x(k) = [\Delta^{\alpha}x_1(k) \quad \Delta^{\alpha}x_2(k) \dots \quad \Delta^{\alpha}x_{n_a}(k)]^T$$
(2.28)

2.4 Stabilité des systèmes fractionnaires

Il est noté [106,107] qu'un système fractionnaire d'ordre commensurable est dit BIBO (bounded-input bounded-output) stable lorsque les pôles ρ_i du système vérifient la condition suivante :

$$|arg(\rho_i)| > \alpha \frac{\pi}{2} \quad i = 1, \dots, n$$
 (2.29)

avec $0 < \alpha < 2$, $arg(\rho_i)$ est l'argument de la racine ρ_i . Par conséquent, les différentes régions de stabilité sont illustrées par la figure (2.1).



FIGURE 2.1: Domaine de stabilité des systèmes fractionnaires

Dans le cas où le système est représenté par un modèle d'état au cas discret, il est asymptotiquement stable si et seulement si [105] :

$$\|A\| < 1 \tag{2.30}$$

où $\| \cdot \|$ désigne la norme matricielle définie comme max $\| \gamma_i \|$. γ_i est la i^{ème} valeur propre de la matrice A.

2.5 Simulation des systèmes fractionnaires

La représentation irrationnelle des systèmes fractionnaires complique leurs simulations qui repose essentiellement sur la modélisation et la simulation de l'opérateur de dérivation ou d'intégration non entier. Pour résoudre ce problème, plusieurs méthodes, basées sur l'approximation de l'opérateur de dérivation (intégration) fractionnaire ont été développées. En effet, deux approches différentes peuvent être distinguées; à savoir l'approche continue et l'approche discrète [104, 108].

L'approche continue consiste à approximer le système à simuler par un modèle rationnel continu équivalent. Dans cette approche, plusieurs méthodes d'approximation ont été développées. La technique d'Oustaloup [99], Charef [109] et Trigeassou [110] reposent sur l'utilisation d'une distribution récursive de zéros et pôles d'ordre entier, répartis dans une bande de fréquence limitée. La méthode de Carlson est fondée sur un processus itératif de Newton [111]. Par ailleurs, la méthode de Matsuda [112] utilise le principe du développement en fractions continues (CFE : Continued Fraction Expansions), permettant d'approximer la réponse du dérivateur généralisé sur un intervalle de fréquences espacé.

D'autre part, les réalisations temporelles discrètes sont souvent préférées aux réalisations en temps continu puisqu'elles peuvent être facilement mises en œuvre en pratique et mises à jour en fonction de la modification du but de l'application. Dans le cas discret, l'approximation peut être obtenue selon deux approches différentes : méthodes de discrétisation directes et méthodes de discrétisation indirectes. La méthode directe consiste à approximer le modèle d'ordre fractionnaire par un modèle rationnel discret basé sur la définition de Grünwald-Letnikov [99, 113]. Elle peut être facilement développée dans les cas simples des systèmes monovariables commensurables, elle peut aussi être généralisée au cas des systèmes multivariables non commensurables.

La discrétisation indirecte s'effectue en deux étapes, d'abord réaliser l'approximation continue, puis discrétiser cette dernière au moyen des méthodes numériques usuelles [114].

De plus, d'autres méthodes existent, basées sur la discrétisation classique dans le domaine fréquentiel, tel que les méthodes de : Euler, Tustin, Simpson, El-Alaoui [114,115].

2.5.1 Simulation d'un modèle d'état fractionnaire

La représentation de l'espace d'état fractionnaire est basée sur les définitions de l'opérateur de différence d'ordre fractionnaire [102]. Dans la suite de notre travail, nous utiliserons la définition basée sur l'opérateur de différence de GL donné dans [105]. En effet, la simulation du modèle fractionnaire (équation 2.26) peut être effectuée en utilisant la différence de GL donnée dans l'équation (2.5).

Définissons cette notation :

$$\beta(j) = (-1)^j \begin{pmatrix} \alpha \\ j \end{pmatrix}$$
(2.31)

L'équation (2.5) peut être réécrite sous la forme (2.32) :

$$\Delta^{\alpha} x(k) = \sum_{j=0}^{k} \beta(j) x(k-j)$$
(2.32)

Où une relation de récurrence entre $\beta(j)$ et $\beta(j-1)$ peut être déduite :

$$\beta(j) = \begin{cases} 1 & for \quad j = 0\\ \beta(j-1)\frac{(j-1)(\alpha-1)}{j} & for \quad j > 0 \end{cases}$$
(2.33)

et par conséquent :

$$\Delta^{(\alpha)} x(k+1) = \sum_{j=0}^{L} \beta(j) x(k+1-j)$$
(2.34)

Dans le cas d'ordre fractionnaire général, le terme $\beta(j)$ est donné comme suit :

$$\beta(j) = \begin{bmatrix} \beta_1(j) & 0 & \dots & 0 \\ 0 & \beta_2(j) & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \beta_{n_a}(j) \end{bmatrix}$$
(2.35)

avec par exemple $\beta_1(j)$ donné par :

$$\beta_1(j) = (-1)^j \begin{pmatrix} \alpha_1 \\ j \end{pmatrix}$$
(2.36)

Ainsi, on a :

$$\Delta^{(\alpha)}x(k+1) = x(k+1) + \sum_{j=1}^{L} \beta(j)x(k+1-j)$$
(2.37)

Par conséquent, on obtient :

$$x(k+1) = \Delta^{(\alpha)} x(k+1) - \sum_{j=1}^{L} \beta(j) x(k+1-j)$$
(2.38)

L'équation de récurrence pour la simulation d'espace d'états fractionnaire est la suivante :

$$\begin{cases} \Delta^{(\alpha)} x(k+1) = Ax(k) + Bu(k) \\ x(k+1) = \Delta^{(\alpha)} x(k+1) - \sum_{j=1}^{L} \beta(j) x(k-j+1) \\ y(k) = Cx(k) + Du(k) \end{cases}$$
(2.39)

Afin de faciliter les simulations, nous considérons une mémoire de longueur limitée notée L.

Le modèle d'espace d'états fractionné décrit dans cette section est utilisé comme la partie linéaire du système de Wiener dans ce qui suit.

2.6 Description du système de Wiener fractionnaire

Dans cette partie, on propose la généralisation du système de Wiener décrit précédemment au cas d'ordre fractionnaire.

2.6.1 Modèle de Wiener fractionnaire sous forme PNLSS

Dans cette partie, on propose l'extension du modèle de Wiener avec la forme PNLSS d'ordre entier à l'ordre non entier.

Etant donné le modèle de Wiener représenté sur la figure (1.5), on considère maintenant que la partie dynamique linéaire est représentée par un modèle de l'espace d'état fractionnaire à temps discret d'ordre n_a tel que donné dans l'équation (2.40) [116–118]. Cette dernière est supposée asymptotiquement stable. Le bloc statique non linéaire est défini par une fonction non linéaire f(), qui peut être choisie comme un polynôme d'ordre r associé au vecteur de coefficients p ($p = p_1, p_2, ..., p_r$), comme donné dans l'équation (2.41).

$$\begin{cases} \Delta^{(\alpha)} x(k+1) = A_0 x(k) + B_0 u(k) \\ v(k) = C_0 x(k) + D_0 u(k) \end{cases}$$
(2.40)

$$f(v(k)) = p_1 f_1(v(k)) + p_2 f_2(v(k)) + \ldots + p_r f_r(v(k))$$
(2.41)

$$y(k) = \sum_{i=1}^{r} p_i v^i(k)$$
 (2.42)

Les équations liant ces deux modèles sont développées, donnant la sortie du système de Wiener comme suit :

$$\begin{cases} \Delta^{(\alpha)} x(k+1) = A_0 x(k) + B_0 u(k) \\ y(k) = \sum_{i=1}^r p_i v^i(k) \\ = \sum_{i=1}^r p_i \left(C_0 x(k) + D_0 u(k) \right)^i \end{cases}$$
(2.43)

En appliquant l'expansion multinomiale, les équations du modèle de Wiener PNLFSS peuvent être dérivées.

Considérons, dans ce qui suit, deux cas du système de Wiener avec la partie linéaire du

même ordre d'état $n_a = 2$ et complexité croissante r = 2 et r = 3 pour le bloc non linéaire polynomial.

• Modèle de Wiener avec $n_a = 2$ et r = 2

Dans ce cas, le modèle de Wiener est représenté par les équations suivantes :

$$\begin{cases} \Delta^{(\alpha)} x(k+1) = A_0 x(k) + B_0 u(k) \\ v(k) = C_0 x(k) + D_0 u(k) \\ y(k) = p_1 v^1(k) + p_2 v^2(k) \end{cases}$$
(2.44)

La sortie système de l'équation (2.44) est donnée comme suit : (la variable k est omise pour ne pas surcharger les expressions)

$$y = p_1 v^1 + p_2 v^2$$

= $p_1 (C_0 x + D_0 u) + p_2 (C_0 x + D_0 u)^2$ (2.45)

En remplaçant les vecteurs C_0 , D_0 et x par $C_0 = [c_1 \ c_2]$, $D_0 = d$ et $x = [x_1 \ x_2]^T$ respectivement, nous obtenons l'expression suivante :

$$y = p_1 (c_1 x_1 + c_2 x_2 + du) + p_2 (c_1 x_1 + c_2 x_2 + du)^2$$
(2.46)

En développant l'expression précédente, nous aboutissons à :

$$y = p_1c_1x_1 + p_1c_2x_2 + p_1du + p_2[c_1^2x_1^2 + 2c_1c_2x_1x_2 + 2c_1dx_1u + c_2^2x_2^2 + 2c_2dx_2u + d^2u^2]$$
(2.47)

L'expression de la sortie peut être réécrite comme suit :

$$y = \begin{bmatrix} p_{1}c_{1} & p_{1}c_{2} \end{bmatrix} \begin{bmatrix} x_{1} \\ x_{2} \end{bmatrix} + \begin{bmatrix} p_{1}d \end{bmatrix} u + \begin{bmatrix} p_{2}c_{1}^{2} & 2p_{2}c_{1}c_{2} & 2p_{2}c_{1}d & p_{2}c_{2}^{2} & 2p_{2}c_{2}d & p_{2}d^{2} \end{bmatrix} \begin{bmatrix} x_{1} \\ x_{1}u \\ x_{2}^{2} \\ x_{2}u \\ u^{2} \end{bmatrix}$$

$$(2.48)$$

On déduit alors le modèle PNLFSS suivat :

$$\Delta^{(\alpha)}x(k+1) = A_0x(k) + B_0u(k)$$

$$y = p_1 \left[c_1 \ c_2 \right] \left[\begin{array}{c} x_1 \\ x_2 \end{array} \right] + p_1 \left[d \right] u$$

$$+ p_2 \left[\begin{array}{c} c_1^2 \ 2c_1c_2 \ 2c_1d \ c_2^2 \ 2c_2d \ d^2 \end{array} \right] \left[\begin{array}{c} x_1^2 \\ x_1x_2 \\ x_1u \\ x_2^2 \\ x_2u \\ u^2 \end{array} \right]$$
(2.49)

La séparation de la partie linéaire et de la partie non linéaire du modèle donné par l'équation (2.49) conduit à ses équations de modèle PNLFSS correspondantes. Cela peut être résumé dans le Tableau (2.1), pour les matrices de la partie linéaire (A, B, C et D), la matrice E et le vecteur ξ , et dans le Tableau (2.2) pour les éléments non linéaires η et ses coefficients F.

Dans ce cas, la matrice E = 0, et $\xi = 0$ et η contient l'expansion des monômes de u et xpour r = 2.

PNLSS	Wiener	
A	A_0	
B	B_0	
C	p_1C_0	
D	$p_1 D_0$	
E	0	
ξ	ξ 0	

TABLE 2.1: Modèle de Wiener avec $n_a = 2$ et r = 2

Nous pouvons remarquer que le vecteur F contient les éléments de C, D et p_2 .

F	η
$p_2 c_1^2$	x_{1}^{2}
$2p_2c_1c_2$	$x_1 x_2$
$2p_2c_1d$	x_1u
$p_2 c_2^2$	x_{2}^{2}
$2p_2c_2d$	x_2u
$p_2 d^2$	u^2

TABLE 2.2: Vecteurs F et η du modèle de Wiener pour $n_a = 2$ et r = 2

• Modèle de Wiener avec $n_a = 2$ et r = 3

Dans ce cas, l'expansion multinomiale génère un nombre important d'éléments lorsque r augmente; le modèle de Wiener est décrit par l'équation (2.50).

$$\begin{cases} \Delta^{(\alpha)}x(k+1) = A_0x(k) + B_0u(k) \\ v(k) = C_0x(k) + D_0u(k) \\ y(k) = p_1v^1(k) + p_2v^2(k) + p_3v^3(k) \end{cases}$$
(2.50)

La sortie y peut être alors formulée comme suit :

$$y = p_1 (c_1 x_1 + c_2 x_2 + du) + p_2 (c_1 x_1 + c_2 x_2 + du)^2 + p_3 (c_1 x_1 + c_2 x_2 + du)^3$$
(2.51)

En développant l'équation (2.51), on obtient :

$$y = p_{1}c_{1}x_{1} + p_{1}c_{2}x_{2} + p_{1}du$$

$$+p_{2}[c_{1}^{2}x_{1}^{2} + 2c_{1}c_{2}x_{1}x_{2} + 2c_{1}dx_{1}u + c_{2}^{2}x_{2}^{2} + 2c_{2}dx_{2}u + d^{2}u^{2}]$$

$$+p_{3}[c_{1}^{3}x_{1}^{3} + 3c_{1}^{2}c_{2}x_{1}^{2}x_{2} + 3c_{1}^{2}dx_{1}^{2}u + 3c_{1}c_{2}^{2}x_{1}x_{2}^{2} + 6c_{1}c_{2}dx_{1}x_{2}u$$

$$+3c_{1}d^{2}x_{1}u^{2} + c_{2}^{3}x_{2}^{3} + 3c_{2}^{2}dx_{2}^{2}u + 3c_{2}d^{2}x_{2}u^{2} + d^{3}u^{3}]$$

$$(2.52)$$

et au finale on peut reformuler la représentation du modèle comme suit :

$$\Delta^{(\alpha)}x(k+1) = A_{0}x(k) + B_{0}u(k)$$

$$y = p_{1}\left[c_{1} \ c_{2}\right]\left[x_{1} \ x_{2}\right] + p_{1}\left[d\right]u + \left[x_{2}^{2} \ x_{2}^{2} \ x_{2}^{2}$$

Pour ce second cas $(n_a = 2, r = 3)$, l'équation d'état de la représentation PNLFSS reste inchangée, alors que les éléments correspondants des vecteurs η et F sont donnés dans le Tableau (2.3).

Nous remarquons, tel que rapporté dans la littérature, que le nombre de paramètres des modèles transformés augmente fortement lorsqu'on augmente l'ordre du bloc non linéaire r.

Notez que le nombre des composants du vecteur F est élevé (selon les ordres n_a et r) et que F contient les éléments des matrices C, D (le vecteur Z) et du vecteur p; de plus, nous avons une redondance de ces paramètres.

	F	η	
$p_2 Z^2$	$p_2 c_1^2$	$\eta_1 = \xi^2(k)$	x_{1}^{2}
	$2p_2c_1c_2$		$x_1 x_2$
	$2p_2c_1d$		x_1u
	$p_2 c_2^2$		x_{2}^{2}
	$2p_2c_2d$		x_2u
	$p_2 d^2$		u^2
p_3Z^3	$p_3 c_1^3$	$\eta_2 = \xi^3(k)$	x_{1}^{3}
	$3p_3c_1^2c_2$		$x_1^2 x_2$
	$3p_3c_1^2d$		$x_1^2 u$
	$3p_3c_1c_2^2$		$x_1 x_2^2$
	$6p_3c_1c_2d$		$x_1 x_2 u$
	$3p_3c_1d^2$		$x_1 u^2$
	$p_3 c_2^3$		x_{2}^{3}
	$3p_3c_2^2d$		$x_2^2 u$
	$3p_3c_2d^2$		x_2u^2
	p_3d^3		u^3

TABLE 2.3: Vecteurs F et η du modèle de Wiener pour $n_a=2$ et r=3

Par conséquent, lorsque l'ordre r augmente, nous avons les mêmes paramètres que dans le cas d'ordre précédent en ajoutant les éléments Z^3, \ldots, Z^r .

La représentation PNLFSS de Wiener peut être écrite dans le cas général, comme suit :

$$\begin{cases} \Delta^{(\alpha)}x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases}$$
(2.54)

Le modèle de Wiener PNLFSS est simulé en utilisant l'équation qui suit :

$$\Delta^{(\alpha)} x(k+1) = Ax(k) + Bu(k)$$

$$x(k+1) = Ax(k) + Bu(k) - \sum_{j=1}^{L} \beta(j)x(k-j+1)$$

$$y(k) = Cx(k) + Du(k) + F\eta(k)$$
(2.55)

avec

$$\begin{cases} F = [p_2 Z^2 \dots p_r Z^r] \\ Z = [c_1 \dots c_{n_a} d] \end{cases}$$
(2.56)

 et

$$\begin{cases} \eta(k) = [\xi^{2}(k) \dots \xi^{r}(k)] \\ \xi(k) = [x_{1}(k) \dots x_{na}(k) u(k)] \end{cases}$$
(2.57)

2.6.2 Modèle de Wiener sous forme de régression fractionnaire

Le modèle de Wiener peut aussi être décrit sous forme de régression fractionnaire. Ceci, en supposant que la partie linéaire du modèle bloc est choisie comme une fonction de transfert d'ordre fractionnaire.

Dans ce cas, la fonction de transfert fractionnaire (équation (2.58)) est considérée pour représenter la partie linéaire.

$$G(z) = \frac{B(z)}{A(z)} \tag{2.58}$$

avec

$$\begin{cases}
A(z) = 1 + \sum_{i=1}^{n_a} a_i z^{-\alpha_i} \\
= 1 + a_1 z^{-\alpha} + \ldots + a_{n_a} z^{-n_a \alpha}
\end{cases}$$
(2.59)

$$\begin{cases} B(z) = \sum_{j=1}^{n_b} b_j z^{-\beta_j} \\ = b_1 z^{-\beta} + \ldots + b_{n_b} z^{-n_b\beta} \end{cases}$$
(2.60)

avec : n_a et n_b sont respectivement les longueurs des vecteurs a et b.

 α_i et β_j sont les ordres fractionnaires.

Pour la différentiation d'ordre fractionnaire, la définition de GL est utilisée. Ainsi la relation qui décrit la sortie de la partie linéaire est donnée comme suit :

$$v(z) = \frac{\sum_{j=1}^{n_b} b_j z^{-\beta_j}}{1 + \sum_{i=1}^{n_a} a_i z^{-\alpha_i}} u(k)$$
(2.61)

Dans le cas d'ordre fractionnaire commensurable, on a :

$$v(k) = -\sum_{i=1}^{n_a} a_i(z^{-i})^{\alpha} v(k) + \sum_{j=1}^{n_b} b_j(z^{-j})^{\alpha} u(k)$$
(2.62)

 z^{-1} l'opérateur de retard, et $z^{-1}y(k) = y(k-1)$.

L'utilisation de l'opérateur de différence d'ordre fractionnaire Δ , nous conduit à la relation suivante :

$$v(k) = -\sum_{i=1}^{n_a} a_i \Delta^{\alpha} v(k-i) + \sum_{j=1}^{n_b} b_j \Delta^{\alpha} u(k-j)$$
(2.63)

Le système linéaire dynamique est supposé asymptotiquement stable. La partie non linéaire est un polynôme d'ordre r, donné par :

$$f(v(k)) = \sum_{i=1}^{r} p_i v^i(k)$$

= $p_1 v^1(k) + p_2 v^2(k) + \dots + p_r v^r(k)$ (2.64)

Pour unifier les estimations de paramètres, on fixe le premier coefficient de la fonction non linéaire égal à 1 $(p_1 = 1)$. Par conséquent, on aura :

$$y(k) = v(k) + \sum_{i=2}^{r} p_i v^i(k) + e(k)$$
(2.65)

Pour éviter le produit des paramètres et en substituant l'équation (2.63) dans l'équation (2.65), la sortie du modèle de Wiener peut être écrit sous la forme :

$$y(k) = -\sum_{i=1}^{n_a} a_i \Delta^{\alpha} v(k-i) + \sum_{j=1}^{n_b} b_j \Delta^{\alpha} u(k-j) + \sum_{i=2}^{r} p_i v^i(k) + e(k)$$
(2.66)

$$y(k) = -a_1 \Delta^{\alpha} v(k-1) - \dots - a_{n_a} \Delta^{\alpha} v(k-n_a) + b_1 \Delta^{\alpha} u(k-1) + \dots + b_{n_b} \Delta^{\alpha} u(k-n_b) + p_2 v^2(k) + \dots + p_r v^r(k) + e(k)$$
(2.67)

La sortie du modèle de Wiener peut être décrite comme suit :

$$y(k) = \phi_1^T(k)\tilde{\theta}_1 + \phi_2^T(k)\tilde{\theta}_2 + e(k)$$
(2.68)

La sortie du modèle de Wiener (2.68) peut être aussi être donné par l'équation (3.45).

$$y(k) = \phi^T(k)\tilde{\theta} + e(k)$$
(2.69)

En supposant

$$\begin{cases} \phi(k) = \left[\phi_1^T(k) & \phi_2^T(k) \right]^T \\ \\ \tilde{\theta} = \left[\tilde{\theta}_1^T & \tilde{\theta}_2^T \right]^T \end{cases}$$
(2.70)

Avec

$$\tilde{\theta_1} = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_1 & \dots & b_{n_b} \end{bmatrix}^T$$
(2.71)

$$\phi_{1}(k) = \begin{bmatrix} -\Delta^{\alpha}v(k-1) \\ \vdots \\ -\Delta^{\alpha}v(k-n_{a}) \\ \Delta^{\alpha}u(k-1) \\ \vdots \\ \Delta^{\alpha}u(k-n_{b}) \end{bmatrix}^{T}$$
(2.72)
$$\tilde{\theta_{2}} = \begin{bmatrix} p_{2} & p_{3} & \dots & p_{r} \end{bmatrix}^{T}$$
(2.73)

$$\phi_2(k) = \begin{bmatrix} v^2(k) \\ v^3(k) \\ \vdots \\ v^r(k) \end{bmatrix}$$
(2.74)

Notez que le vecteur d'information ϕ est en fonction de la sortie intermédiaire non mesurable, cela signifie que l'identification du modèle ne peut pas se faire directement. Dans ce cas, on peut utilisé le principe key term et remplacer v(k) par son estimation $\hat{v}(k)$.

2.7 Conclusion

Dans ce chapitre, nous avons rappelé quelques généralités fondamentales des systèmes d'ordre non entier dits fractionnaires. Nous avons introduit plusieurs notions des opérateurs fractionnaires et les différentes représentations ainsi que la propriété de stabilité et les méthodes de simulation des systèmes d'ordre fractionnaire.

Par la suite, un modèle non linéaire au cas fractionnaire est considéré. En effet, nous avons détaillé la description du modèle de Wiener dans le cas fractionnaire. Ce dernier est représenté sous forme PNLFSS et régression fractionnaire.

Dans le chapitre qui suit, nous proposons l'identification des systèmes fractionnaires, en se basant sur les modèles de Wiener représentés dans le présent chapitre.

Chapitre 3

Identification des systèmes non linéaires entiers et non entiers

3.1 Introduction

La plupart des systèmes dans la vie réelle montrent un comportement dynamique non linéaire. Ainsi, un intérêt considérable a été consacré pour l'identification des systèmes non linéaires qui reste une tâche très difficile.

D'autre part, plusieurs systèmes non linéaires présentent une dynamique complexe fractionnaire, caractérisée par une mémoire longue et une structure dimensionnelle infinie. En conséquence, une autre classe basée sur la dérivée fractionnaire appelée modèles fractionnaires a été développée pour leur modélisation.

Toutefois, bien qu'une certaine quantité de connaissances sur le sujet de l'identification du système fractionnaire a été accumulée à travers la littérature, seules quelques études ont été proposées pour l'identification des systèmes non linéaires fractionnaires.

Ce chapitre présente l'identification des systèmes non linéaires de type de Wiener. Dans un premier temps, un état de l'art sur l'identification des systèmes fractionnaires non linéaires sera présenté. Par la suite, l'identification d'un système non linéaire entier du type de Wiener, décrit sous forme PNLSS, sera considérée.

Dans la deuxième partie de ce chapitre, l'identification du modèle de Wiener d'ordre

fractionnaire sous forme PNLFSS est développée, en utilisant l'algorithme de Levenberg-Marquardt (LM) et l'optimisation par essaim de particules à vitesse auto-adaptative (SAVPSO). L'identification du modèle non linéaire de Wiener décrit sous forme de régression fractionnaire est aussi étudiée. Dans ce cas, la sortie du système étant en fonction du signal intermédiaire non mesurable, l'identification peut être effectuée par l'algorithme PSO.

3.2 État de l'art sur l'identification des systèmes fractionnaires non linéaires

L'identification des systèmes fractionnaires a connu un nombre important de contributions et d'applications dans divers champs.

La grande partie de ces études ont porté sur les modèles fractionnaires linéaires. Un aperçu sur l'identification des systèmes linéaires est présenté dans [119].

Bien que plusieurs études ont été faites sur le sujet de l'identification des systèmes fractionnaires dans la littérature, seules quelques études ont été proposées pour l'identification des systèmes non linéaires fractionnaires.

Dans ce contexte, les séries de Volterra fractionnaires ont été proposées pour modéliser et identifier une diffusion thermique avec de grandes variations de température [120]. Une approche à erreur de sortie a été considérée dans [121], où les réseaux de neurones temporels continus (CTNN) ont été utilisés.

Les modèles blocs orientés ont également été appliqués dans le cas d'ordre fractionnaire. Cependant, l'ordre fractionnaire est souvent supposé connu et les majeurs travaux considèrent le cas particulier des modèles d'ordre commensurable, où les ordres fractionnaires sont multiple de la même base α .

Dans ce but, la structure Hammerstein a été considérée; la méthode des moindres carrés a été utilisée [122] pour identifier les paramètres des parties linéaires et non linéaires d'un système Hammerstein fractionnaire commensurable. Dans [123], l'algorithme des moindres carrées récursif (RLS) a été appliquée pour identifier les systèmes Hammerstein d'ordre fractionnaire. De plus, l'identification du modèle Hammerstein d'ordre fractionnaire commensurable a été effectuée sur la base des variables instrumentales (IV) [124]. Dans [125], l'identification du système de Hammerstein neuro-fractionnaire a été rapportée.

La structure de Wiener a également été utilisée dans [126, 127], où le système fractionnaire de Wiener a été modélisé en utilisant les fonctions de base orthonormées (OBF) et l'identification est effectuée sur la base de l'algorithme RLS avec la connaissance préalable de l'ordre fractionnaire.

Dans [128], les auteurs ont présenté une approche fractionnaire pour l'initialisation d'un modèle de Wiener-Hammerstein avec des multiplicités fractionnaires de pôles et des zéros de la meilleure approximation linéaire estimée (BLA).

Cependant, l'identification du système fractionnaire non linéaire reste un domaine de recherche d'actualité très délicat. En effet, le but de notre étude sera l'identification du modèle de Wiener en utilisant un modèle non linéaire fractionnaire.

3.3 Identification d'un système non linéaire de type de Wiener entier

Dans cette partie on considère l'identification du système de Wiener, décrit dans la section (1.4) au cas entier, en utilisant les méthodes de Levenberg-Marquardt (LM) et SAVPSO.

Étant donné le système de Wiener PNLSS suivant :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases}$$
(3.1)

L'objectif de l'identification du modèle de Wiener PNLSS consiste à estimer les matrices PNLSS inconnues : A, B, C, D et F.

Cependant, dans ce cas, l'identification implique un nombre important de paramètres à estimer et de nombreuses structures possibles peuvent être sélectionnées. Afin d'améliorer l'efficacité de l'algorithme et de réduire simultanément le nombre de paramètres, une forme compagne commandable peut être sélectionnée pour la matrice A du bloc linéaire du modèle de Wiener. Par conséquent, la partie linéaire du modèle de Wiener est donnée comme suit :

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -a_1 & -a_2 & \dots & -a_{n_a} \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} c_1 & c_2 & \dots & c_{n_a} \end{bmatrix} x(k) + \begin{bmatrix} d \end{bmatrix} u(k) + \begin{bmatrix} F_1 & \dots & F_{n_F} \end{bmatrix} \eta(k) \\ (3.2)$$

Dans ce cas, les matrices inconnues à estimer sont les matrices A, C et D, qui contiennent les paramètres de la partie linéaire, et les coefficients polynomiaux de la partie non linéaires $p_{2:r}$, qui sont contenus dans le vecteur F.

Pour unifier les estimations des paramètres, on suppose que le premier coefficient de la fonction non linéaire est égal à 1 ($p_1 = 1$).

Basé sur le principe surparamétrisation (Over parameterization), le vecteur des paramètres θ , de longueur $n_{\theta} = 2n_a + 1 + n_F$, à estimer est donc donné par :

$$\theta = [-a_1 \ \dots \ -a_{n_a} \ c_1 \ \dots \ c_{n_a} \ d \ F_1 \ \dots \ F_{n_F}]$$
(3.3)

avec n_F le nombre des éléments du vecteur F.

3.3.1 Identification du modèle de Wiener PNLSS par l'algorithme de LM

La sortie du modèle étant non linéaire par rapport à θ , une méthode d'optimisation non linéaire basée sur l'algorithme de Levenberg-Marquardt (LM) est utilisée [9, 42, 62, 129]. L'estimateur utilise une technique de programmation non linéaire qui interpole l'algorithme de Gauss-Newton et l'algorithme de gradient [1, 130] et permet de mieux ajuster les paramètres en fonction des variations du critère.

Les données échantillonnées sont composées de N observations $\{u_k, y_k^*\}$. $\hat{\theta}$ est l'estimation

des paramètres du vecteur θ .

L'erreur de sortie de prédiction est donnée par

$$\varepsilon_k = y_k - \hat{y}_k \tag{3.4}$$

 y_k et \hat{y}_k correspond respectivement à la sortie mesurée et estimée.

La valeur optimale de $\hat{\theta}$ est obtenue en minimisant le critère quadratique suivant :

$$J(\theta) = \frac{1}{N} \sum_{k=1}^{N} \varepsilon_k^2$$
(3.5)

avec les équations suivantes :

$$\theta^{(i+1)} = \theta^{(i)} - \left\{ \left[J_{\theta\theta}^{"} + \lambda I \right]^{-1} J_{\theta}^{'} \right\}_{\hat{\theta} = \theta^{(i)}}$$
(3.6)

$$\begin{cases} J'_{\theta} = \frac{-2}{N} \left(\sum_{k=1}^{N} \varepsilon_k \, \sigma_{y_{k/\theta}} \right) \text{ est le gradient} \\ J''_{\theta} = \frac{2}{N} \left(\sum_{k=1}^{N} \sigma_{y_{k/\theta}} \, \sigma_{y_{k/\theta}}^T \right) \text{ est le Hessien} \\ \sigma_{y_{k/\theta_j}} = \frac{\partial \hat{y}(k,\theta)}{\partial \theta_j} \text{ est la fonction de sensibilité de la sortie par rapport aux paramètres} \end{cases}$$
(3.7)

avec I la matrice identité et λ le coefficient de Marquardt, qui permet le pilotage de l'algorithme.

Cet algorithme, grâce au réglage du paramètre λ en cours de recherche, permet d'évoluer entre la technique de gradient loin de l'optimum ($\lambda \gg 1$) et la technique de Newton (lorsque $\lambda \to 0$) qui permet d'accélérer la convergence au voisinage de l'optimum. Cette procédure est réalisée suivant les principales étapes résumées comme suit :

- Initialisation des paramètres ($\theta^0 = 1e 6$).
- Calcul du critère $J(\theta^0)$ donné par l'équation (3.5).
- Itération : i = 1.
- Calcul des fonctions de sensibilités $(\sigma_{y_{k/\theta}})$, du Gradient (J'_{θ^i}) et du Hessien (J''_{θ^i}) (l'équation (3.7)), permettant ainsi le calcul de $\hat{\theta}^i$ (équation (3.6)).
- Calcul de $J\left(\hat{\theta}^{i}\right)$.

(a, i)

- Évaluation du critère : $J\left(\hat{\theta}^{i}\right) < J\left(\theta^{i}\right)$? Si oui, $\hat{\theta}^{i} = \theta^{i}$ et $J\left(\hat{\theta}^{i}\right) = J\left(\theta^{i}\right)$; Diminuer λ_{i} ($\lambda_{i}/10$). Sinon, augmenter λ_{i} .
- i = i + 1.
- Répéter le processus jusqu'à ce que le critère d'arrêt soit atteint $(J \le 1e 16)$ et obtenir l'estimation des paramètres.

L'optimisation du critère se fait sur la base des calculs des fonctions de sensibilités [1, 119], leur calcul dépend essentiellement de la structure du modèle choisi et leur rôle est analogue au vecteur regresseur dans le cas linéaire par rapport aux paramètres [1], ce qui rend l'identification très sensible aux calculs de ces fonctions de sensibilité [131].

3.3.1.1 Calcul des fonctions de sensibilité paramétriques

Les fonctions de sensibilité traduisent l'effet d'une variation d'un paramètre sur la sortie du système [130]. Dans notre travail, nous nous somme intéressés à la description du modèle de Wiener par un modèle PNLSS, et en considérant la forme générale du système comme suit :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases}$$
(3.8)

L'application de la dérivation de l'équation (3.8) par rapport à θ donne :

$$\begin{cases} \frac{\partial}{\partial \theta_j} x(k+1) &= \frac{\partial A}{\partial \theta_j} x(k) + A \frac{\partial x(k)}{\partial \theta_j} + \frac{\partial B}{\partial \theta_j} u(k) + B \frac{\partial u(k)}{\partial \theta_j} \\ \frac{\partial y(k)}{\partial \theta_j} &= \frac{\partial C}{\partial \theta_j} x(k) + C \frac{\partial x(k)}{\partial \theta_j} + \frac{\partial D}{\partial \theta_j} u(k) + D \frac{\partial u(k)}{\partial \theta_j} + \frac{\partial F}{\partial \theta_j} \eta(k) + F \frac{\partial \eta(k)}{\partial \theta_j} \end{cases}$$
(3.9)

Avec

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

$$\frac{\partial x(k)}{\partial \theta_{j}} = \sigma_{x/\theta_{j}}$$

$$\frac{\partial u(k)}{\partial \theta_{j}} = 0$$

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

$$\frac{\partial \eta(k)}{\partial \theta_{j}} = \frac{\partial \eta(k)}{\partial x(k)} \frac{\partial x(k)}{\partial \theta_{j}} = \left[\frac{\partial \eta(k)}{\partial x_{1}(k)} \dots \frac{\partial \eta(k)}{\partial x_{n_{a}}(k)}\right] \frac{\partial x(k)}{\partial \theta_{j}} = \eta' \sigma_{x/\theta_{j}}$$
avec
$$\eta' = \left[\frac{\partial \eta(k)}{\partial x_{1}(k)} \dots \frac{\partial \eta(k)}{\partial x_{n_{a}}(k)}\right]$$

En conséquence :

$$\begin{cases} \sigma_{x_{(k+1)}/\theta_j} = \frac{\partial A}{\partial \theta_j} x(k) + A \sigma_{x/\theta_j} + \frac{\partial B}{\partial \theta_j} u(k) \\ \sigma_{y/\theta_j} = \frac{\partial C}{\partial \theta_j} x(k) + C \sigma_{x/\theta_j} + \frac{\partial D}{\partial \theta_j} u(k) + \frac{\partial F}{\partial \theta_j} \eta(k) + F \eta' \sigma_{x/\theta_j} \end{cases}$$
(3.11)

pour $j = 1, \ldots, n_{\theta}$.

La valeur optimale de θ est obtenue en minimisant le critère J par l'application de l'algorithme LM. Ce dernier nécessite le calcul du modèle des fonctions de sensibilité, rappelé comme suit :

$$\sigma_{x_{(k+1)}/\theta_{j}} = A \sigma_{x/\theta_{j}} + \left[\frac{\partial A}{\partial \theta_{j}} \quad \frac{\partial B}{\partial \theta_{j}} \quad 0 \right] \begin{bmatrix} x \\ u \\ \eta \end{bmatrix}$$

$$\sigma_{y/\theta_{j}} = (C + F\eta') \sigma_{x/\theta_{j}} + \left[\frac{\partial C}{\partial \theta_{j}} \quad \frac{\partial D}{\partial \theta_{j}} \quad \frac{\partial F}{\partial \theta_{j}} \right] \begin{bmatrix} x \\ u \\ \eta \end{bmatrix}$$
(3.12)

Considérant la forme compagne commandable, l'équation (3.12) peut être réécrite

comme suit :

$$\sigma_{x_{(k+1)}/\theta_{j}} = A \sigma_{x/\theta_{j}} + \begin{bmatrix} \frac{\partial A}{\partial \theta_{j}} & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ \eta \end{bmatrix}$$

$$\sigma_{y/\theta_{j}} = (C + F\eta')\sigma_{x/\theta_{j}} + \begin{bmatrix} \frac{\partial C}{\partial \theta_{j}} & \frac{\partial D}{\partial \theta_{j}} & \frac{\partial F}{\partial \theta_{j}} \end{bmatrix} \begin{bmatrix} x \\ u \\ \eta \end{bmatrix}$$
(3.13)

Le modèle des fonctions de sensibilité (3.13) est implémenté à chaque itération, et permet le calcul de J' et J'' pour la mise à jour du vecteur de paramètres $\hat{\theta}$.

Les valeurs des coefficients $p_{2:r}$ peuvent être déduits des éléments du vecteur F. Par exemple, pour le cas de Wiener avec $n_a = 2$ et r = 2, p_2 est estimé selon l'équation suivante :

$$p_2 = \frac{1}{n_F} \left(\frac{F(1)}{c_1^2} + \frac{F(2)}{c_2^2} + \frac{F(3)}{2c_1c_2} + \frac{F(4)}{2c_1d} + \frac{F(5)}{2c_2d} + \frac{F(6)}{d^2} \right)$$
(3.14)

avec $n_F = 6$.

Cette méthode sera appliquée pour l'identification du modèle de Wiener PNLSS.

Les méthodes d'identification non linéaires sont largement utilisées, que ce soit vis-à-vis des types de systèmes ou des domaines d'application, et constituent un outil d'identification puissant. D'un autre côté, le problème d'optimisation des fonctions non-linéaires a conduit à beaucoup d'autres algorithmes; où les méthodes méta heuristique ont connu un intérêt particulier pour résoudre ce problème. Ce qui a motivé l'utilisation d'une autre approche d'identification basée sur une méthode heuristique qui sera présentée ci-après.

3.3.2 Identification du modèle de Wiener PNLSS par l'algorithme SAVPSO

L'apparition des méthodes d'identification heuristique ainsi que les algorithmes évolutionnaires a connu d'énormes contributions dans la modélisation des problèmes complexes [132]. Ces méthodes partent des phénomènes physiques ou biologiques. En effet les réseaux de neurones artificiels s'inspirent du fonctionnement du cerveau humain [133], l'algorithme de recuit simulé de la thermodynamique [134], les algorithmes génétiques [135] étant basés sur des phénomènes biologiques (génétique), ...

D'autres méthodes sont basées sur la reproduction d'un comportement social tels que les algorithmes de colonie de fourmis sont issus du comportement des colonies de fourmis [136]. L'optimisation par essaim de particules (Particule Swarm Optimisation (PSO)) vient du comportement de déplacement chez certains groupes d'animaux, comme les vols groupés d'oiseaux et de bancs de poissons [137, 138].

Dans la suite de notre travail, nous nous intéressons à une méthode nommée SAVPSO qui est dérivée de l'algorithme PSO.

3.3.2.1 Méthode PSO

L'algorithme d'optimisation par essaim de particules (PSO) [137,138] est une technique d'optimisation métaheuristique, et de nombreuses études ont récemment été menées pour modifier et améliorer ses performances ([139,140]).

Les principales propriétés de l'algorithme PSO sont résumées comme suit :

- C'est un algorithme de recherche basé sur la population (essaim) où chaque individu est appelé particule et représente une solution candidate.
- Chaque particule déplace à travers l'espace de recherche multidimensionnel avec une vitesse adaptable. Cette dernière est ajustée dynamiquement en fonction de sa propre expérience de déplacement et celle des autres particules.
- A chaque instant, chaque particule a une position et une vitesse.
- Le vecteur de position d'une particule par rapport à l'origine de l'espace de recherche représente une solution du problème de recherche.
- Au début, une population des particules est initialisée à des positions aléatoires marquées par les vecteurs X_i et la vitesse aléatoire V_i .
- Chaque particule a une mémoire et est donc capable de se souvenir de sa meilleure position visitée dans l'espace de recherche.
- La meilleure position de chaque particule est connue sous le nom p_{best} et celle de

tout le groupe est appelée g_{best} .

Dans l'algorithme PSO standard, l'essaim est manipulé selon les équations de mise à jour suivantes :

$$\begin{cases} V_{id}(k+1) = \omega V_{id}(k) + z_1 r_1 (p_{best}(k) - X_{id}(k)) \\ + z_2 r_2 (g_{best}(k) - X_{id}(k)) \\ X_{id}(k+1) = X_{id}(k) + V_{id}(k+1) \end{cases}$$
(3.15)

Où : i = 1, 2, ..., N est l'indice de particule, d = 1, 2, ..., n représente la dimension de l'espace de recherche et indique la composante d^{eme} de la particule.

Le paramètre ω est le poids d'inertie, z_1 et z_2 sont des constantes positives et r_1 et r_2 sont des valeurs aléatoires.

Cependant, l'algorithme PSO standard ne prend pas en compte l'impact des contraintes sur le mécanisme de recherche. Il est donc habituellement difficile de concentrer la particule dans la région approximative du domaine de définition.

3.3.2.2 Méthode SAVPSO

Dans ce travail, nous considérons un algorithme amélioré de PSO, appelé PSO à vitesse auto-adaptative (SAVPSO) [63,140]. Ce dernier a la propriété que chaque particule a la capacité d'ajuster de façon auto-adaptative sa vitesse en fonction de certaines caractéristiques du domaine de définition. Par conséquent, l'algorithme SAVPSO permet le traitement des contraintes pour résoudre les problèmes d'optimisation avec contraintes (COP).

L'algorithme SAVPSO est caractérisé par ce qui suit :

• L'équation de mise à jour de la vitesse est modifiée comme suit :

$$V_{id}(k+1) = \omega |p_{i'd}(k) - p_{best}(k)| sign(V_{id}(k)) + r(p_{best}(k) - X_{id}(k)) + (1-r)(g_{best}(k) - X_{id}(k))$$
(3.16)

Où $z_1 = z_2 = 1$, w et r des valeurs aléatoires. i' un entier aléatoire.
Ainsi, le terme $\omega V_{id}(k)$ (méthode PSO) est remplacé par :

$$\omega |p_{i'd}(k) - p_{best}(k)| sign(V_{id}(k))$$
(3.17)

où $sign(V_{id}(k))$ représente le signe de $V_{id}(k)$, ce qui indique la direction de déplacement de $V_i(k)$ dans la d^{ime} dimension.

• Le terme de la vitesse nouvellement acquis ferait déplacer la particule i vers une position entre g_{best} et p_{best} . Par conséquent, la particule i ne dérivera pas du domaine de définition.

• L'expérience de la vitesse de la particule *i* est seulement limitée à la direction de déplacement, où son amplitude est déterminée par $\omega |p_{i'd}(k) - p_{best}(k)|$ selon l'effet du domaine de définition.

 p_{best} et $p_{i'd}(k)$ sont proches ou dans le domaine de définition, alors $|p_{i'd}(k) - p_{best}(k)|$ traduit la taille du domaine de définition.

Par conséquent, la particule *i* ne s'éloignera pas trop du domaine de définition; et la valeur de $|p_{i'd}(k) - p_{best}(k)|$ peut varier de façon auto-adaptative avec les changements de la portée de recherche de l'essaim.

• $X_{id} = (X_{i1}, X_{i2}, ..., X_{in}) \subset S$ est le vecteur de la solution. L'espace de recherche S est défini comme un espace n dimensionnel limité par des contraintes paramétriques :

$$X_{id}^{l} \le X_{id} \le X_{id}^{u}$$
; $d = 1, \dots n.$ (3.18)

• De plus, si les particules sont hors de l'espace de recherche, le processus de résolution se déroulera comme suit :

$$X_{id}(k) = \begin{cases} \bar{x}_d(k) + r(X_{id}^l(k) - \bar{x}_{id}(k)) & \text{if } X_{id}(k) < X_{id}^l(k) \\ \bar{x}_d(k) + r(X_{id}^u(k) - \bar{x}_{id}(k)) & \text{if } X_{id}(k) > X_{id}^u(k) \end{cases}$$
(3.19)

Où \bar{x}_d est la valeur moyenne des d^{eme} composantes de toutes les particules, donnée par :

$$\bar{x}_d = \frac{\sum_{i=1}^N X_{id}(k)}{N}$$
 (3.20)

L'algorithme SAVPSO peut ainsi être résumé par les principales étapes suivantes :

- Créer et initialiser un essaim à n dimensions, chaque particule a une position initiale
 X_{i0} et une vitesse initiale V_{i0} aléatoires.
- Définir les contraintes paramétriques $(X_{id}^l \text{ et } X_{id}^u)$.
- Calcul de la valeur moyenne de toutes les particules (équation 3.20).
- Calcul du critère $J(\hat{\theta}_j)$.
- Mise a jour de la meilleur position de chaque particule p_{best} et celle de tout le groupe g_{best} (équation (3.16)).
- Évaluation des particules $(X_{id}^l \leq X_{id} \leq X_{id} ; d = 1, ...n)$? Si $X_{id}(k) < X_{id}^l(k)$, alors $X_{id}(k) = \bar{x}_d(k) + r(X_{id}^l(k) - \bar{x}_{id}(k))$. Si $X_{id}(k) > X_{id}^u(k)$, alors $X_{id}(k) = \bar{x}_d(k) + r(X_{id}^u(k) - \bar{x}_{id}(k))$.
- Répéter le processus jusqu'à ce que le critère d'arrêt soit atteint.

3.4 Identification des systèmes de Wiener fractionnaires

L'objectif de cette étude est l'identification du modèle de Wiener fractionnaire, décrit par le modèle PNLFSS dans le chapitre précédent par :

$$\begin{cases} \Delta^{(\alpha)} x(k+1) = Ax(k) + Bu(k) \\ x(k+1) = Ax(k) + Bu(k) - \sum_{j=1}^{L} \beta(j) x_i(k-j+1) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases}$$
(3.21)

Dans la suite, pour unifier les estimations des paramètres, nous supposons que le premier coefficient non nul de la fonction non linéaire est égal à 1, soit $p_1 = 1$.

Par conséquent, nous devons estimer les paramètres du système de Wiener fractionnaire contenu dans les matrices A, B, C, D, F et l'ordre fractionnaire α .

Comme indiqué dans la littérature, la difficulté dans l'identification des systèmes non linéaires PNLSS est de traiter l'inconvénient de la dimensionnalité (grande dimension du vecteur F) aussi bien que la redondance des paramètres qui apparaît dans le vecteur F. Dans ce cas, la présente étude contourne ce problème en estimant seulement les éléments de p_i (i = 2, ..., r) au lieu des éléments du vecteur F; de cette façon, l'inconvénient d'estimer un modèle surparamétré est évité.

Sans perte de généralités, la forme canonique commandable est considérée. Par conséquent, la partie linéaire du modèle de Wiener est représenté comme suit :

$$\begin{cases} \Delta^{(\alpha)}x(k+1) = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -a_1 & -a_2 & \dots & -a_{n_a} \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} u(k) \\ (3.22)$$
$$v(k) = \begin{bmatrix} c_1 & c_2 & \dots & c_{n_a} \end{bmatrix} x(k) + \begin{bmatrix} d \end{bmatrix} u(k)$$

Le vecteur des paramètres θ de longueur $(n_{\theta} = 3n_a + r)$ à estimer est donné par :

$$\theta = \begin{bmatrix} \tilde{\theta} & \alpha_1 & \dots & \alpha_{n_a} \end{bmatrix}$$
(3.23)

avec

$$\tilde{\theta} = \begin{bmatrix} -a_1 & \dots & -a_{n_a} & c_1 & \dots & c_{n_a} & d & p_2 & \dots & p_r \end{bmatrix}$$
(3.24)

Tous les paramètres de ce système de Wiener sont explicitement donnés dans le modèle de l'équation (2.55); l'utilisation du modèle PNLFSS évite la combinaison croisée (cross combination) des coefficients du bloc non linéaire avec ceux du bloc linéaire, et ici $(p_2 \ldots p_r)$ sont contenus dans le F vecteur.

La méthode d'optimisation consiste à minimiser un résidu entre les données de sortie y_k et l'estimation de sortie \hat{y}_k . Par conséquent, l'erreur de prédiction quadratique moyenne de la sortie évalue le critère J, comme indiqué par les équations (3.4) et (3.5)

3.4.1 Identification du système de Wiener PNLFSS par l'algorithme SAVPSO

Rappelons que le modèle de Wiener PNLFSS est donné comme suit :

$$\begin{cases} \Delta^{(\alpha)}x(k+1) = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -a_1 & -a_2 & \dots & -a_{n_a} \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} c_1 & c_2 & \dots & c_{n_a} \end{bmatrix} x(k) + \begin{bmatrix} d \end{bmatrix} u(k) + \begin{bmatrix} F_1 & \dots & F_{n_F} \end{bmatrix} \eta(k) \\ (3.25) \end{cases}$$

L'identification du modèle de Wiener PNLFSS peut être aussi effectuée en utilisant l'algorithme SAVPSO [116], décrit précédemment dans la section (3.3.2.2).

Dans ce cas, tous les paramètres du modèle fractionnaire sont estimés, y compris l'ordre de la dérivation non entière en considérant comme contrainte la connaissance a priori que l'ordre de dérivation non entière est compris entre 0 et 1.

$$0 < \alpha < 1 \tag{3.26}$$

3.4.2 Identification du système de Wiener PNLFSS par l'algorithme LM

Le modèle de Wiener PNLFSS est non linéaire par rapport aux paramètres. Par conséquent, l'utilisation de l'algorithme LM peut être bien adéquate pour son identification [118].

On rappelle que l'algorithme LM [131] est manipulé selon l'équation de récurrence suivante :

$$\theta^{(i+1)} = \theta^{(i)} - \left\{ \left[J_{\theta\theta}^{"} + \lambda I \right]^{-1} J_{\theta}^{'} \right\}_{\hat{\theta} = \theta^{(i)}}$$
(3.27)

où λ est le paramètre de pilotage, J'_{θ} est le Gradient et $J''_{\theta\theta}$ est le Hessien défini par les équations suivantes :

$$\begin{cases}
J'_{\theta} = \frac{-2}{N} \left(\sum_{k=1}^{N} \varepsilon_k \sigma_{y/\theta} \right) \\
J''_{\theta} = \frac{2}{N} \left(\sum_{k=1}^{N} \sigma_{y/\theta} \sigma_{y/\theta}^T \right)
\end{cases}$$
(3.28)

avec $\sigma_{y/\theta}$ la fonction de sensibilité de sortie.

$$\sigma_{y/\theta} = \frac{\partial \hat{y}(k,\theta)}{\partial \theta} \tag{3.29}$$

La partie principale de cet algorithme pour les systèmes non linéaires est le calcul crucial du Gradient et du Hessien basé sur les fonctions de sensibilité à chaque itération; ceux-ci sont développés dans la sous-section suivante.

3.4.2.1 Calcul des fonctions de sensibilité

Considérons le système de Wiener PNLFSS suivant :

$$\begin{cases} \Delta^{(\alpha)} x(k+1) = Ax(k) + Bu(k) \\ x(k+1) = Ax(k) + Bu(k) - \sum_{j=1}^{L} \beta(j)x(k-j+1) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases}$$
(3.30)

Les fonctions de sensibilité sont obtenues de la dérivée du modèle de Wiener PNLFSS par rapport aux éléments de θ_j comme suit :

$$\begin{cases}
\Delta^{(\alpha)} \frac{\partial x(k+1)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} (Ax(k) + Bu(k)) \\
\frac{\partial y(k)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} (Cx(k) + Du(k) + F\eta(k))
\end{cases}$$
(3.31)

Le modèle (3.31) peut être réécrit comme suit :

$$\Delta^{(\alpha)} \frac{\partial x(k+1)}{\partial \theta_{j}} = \frac{\partial A}{\partial \theta_{j}} x(k) + A \frac{\partial x(k)}{\partial \theta_{j}} + \frac{\partial B}{\partial \theta_{j}} u(k) + B \frac{\partial u(k)}{\partial \theta_{j}}$$

$$(3.32)$$

$$\frac{\partial y(k)}{\partial \theta_{j}} = \frac{\partial C}{\partial \theta_{j}} x(k) + C \frac{\partial x(k)}{\partial \theta_{j}} + \frac{\partial D}{\partial \theta_{j}} u(k) + D \frac{\partial u(k)}{\partial \theta_{j}} + \frac{\partial F}{\partial \theta_{j}} \eta(k) + F \frac{\partial \eta(k)}{\partial \theta_{j}}$$

avec :

$$\frac{\partial x(k)}{\partial \theta_{j}} = \sigma_{x/\theta_{j}} \quad \text{la fonction de sensibilité des l'états / } \theta_{j}$$

$$\frac{\partial x(k+1)}{\partial \theta_{j}} = \sigma_{x_{(k+1)}/\theta_{j}}$$

$$\frac{\partial y(k)}{\partial \theta_{j}} = \sigma_{y/\theta_{j}} \quad \text{la fonction de sensibilité de la sortie / } \theta_{j}$$

$$\frac{\partial u(k)}{\partial \theta_{j}} = 0$$

$$\frac{\partial \eta(k)}{\partial \theta_{j}} = \frac{\partial \eta(k)}{\partial x(k)} \frac{\partial x(k)}{\partial \theta_{j}} = \left[\frac{\partial \eta(k)}{\partial x_{1}(k)} \dots \frac{\partial \eta(k)}{\partial x_{n_{a}}(k)}\right] \frac{\partial x(k)}{\partial \theta_{j}} = \eta' \sigma_{x/\theta_{j}}$$
avec
$$\eta' = \left[\frac{\partial \eta(k)}{\partial x_{1}(k)} \dots \frac{\partial \eta(k)}{\partial x_{n_{a}}(k)}\right]$$

Ainsi :

$$\begin{cases}
\Delta^{(\alpha)} \left[\sigma_{x_{(k+1)}/\theta_j} \right] = \frac{\partial A}{\partial \theta_j} x(k) + A \sigma_{x/\theta_j} + \frac{\partial B}{\partial \theta_j} u(k) \\
\sigma_{y/\theta_j} = \frac{\partial C}{\partial \theta_j} x(k) + C \sigma_{x/\theta_j} + \frac{\partial D}{\partial \theta_j} u(k) + \frac{\partial F}{\partial \theta_j} \eta(k) + F \eta' \sigma_{x/\theta_j}
\end{cases}$$
(3.34)

et $j = 1, ..., n_{\theta}$.

Les fonctions de sensibilité peuvent être écrites sous forme d'un modèle d'espace d'état fractionnaire pseudo compact (a fractional compact pseudo state space model) comme suit :

$$\begin{cases}
\Delta^{(\alpha)} \left[\sigma_{x_{(k+1)}/\theta_j} \right] = A \sigma_{x/\theta_j} + \left[\frac{\partial A}{\partial \theta_j} \quad \frac{\partial B}{\partial \theta_j} \quad 0 \right] \begin{bmatrix} x \\ u \\ \eta \end{bmatrix}
\end{cases}$$

$$\sigma_{y/\theta_j} = (C + F\eta') \sigma_{x/\theta_j} + \left[\frac{\partial C}{\partial \theta_j} \quad \frac{\partial D}{\partial \theta_j} \quad \frac{\partial F}{\partial \theta_j} \right] \begin{bmatrix} x \\ u \\ \eta \end{bmatrix}$$

$$(3.35)$$

Dans le cas de la forme compagne commandable, $\frac{\partial B}{\partial \theta_j} = 0$ et la représentation (3.35) se réduit à :

$$\Delta^{(\alpha)} \begin{bmatrix} \sigma_{x_{(k+1)}/\theta_j} \end{bmatrix} = A \sigma_{x/\theta_j} + \begin{bmatrix} \frac{\partial A}{\partial \theta_j} & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ \eta \end{bmatrix}$$

$$\sigma_{y/\theta_j} = (C + F\eta') \sigma_{x/\theta_j} + \begin{bmatrix} \frac{\partial C}{\partial \theta_j} & \frac{\partial D}{\partial \theta_j} & \frac{\partial F}{\partial \theta_j} \end{bmatrix} \begin{bmatrix} x \\ u \\ \eta \end{bmatrix}$$
(3.36)

L'algorithme LM est basé sur le calcul des fonctions de sensibilité par rapport à chaque paramètre de θ , qui s'expriment comme suit :

• Sensibilité par rapport à a_j $(\sigma_{y_j a_j})$:

$$\begin{cases}
\Delta^{(\alpha)} \left[\sigma_{x_{(k+1)}/a_j} \right] = A \sigma_{x/a_j} + I_{n_a j} x \\
\sigma_{y_j a_j} = (C + F \eta') \sigma_{x/a_j}
\end{cases}$$
(3.37)

 $I_{n_a j}$ est une matrice nulle avec un seul élément égal à un à l'entrée (n_a, j) .

• Sensibilité par rapport à c_j $(\sigma_{y_jc_j})$:

$$\Delta^{(\alpha)} \left[\sigma_{x_{(k+1)}/c_j} \right] = A \sigma_{x/c_j}$$

$$\sigma_{y_j c_j} = (C + F \eta') \sigma_{x/c_j} + \left[\frac{\partial C}{\partial c_j} \quad \frac{\partial F}{\partial c_j} \right] \left[\begin{array}{c} x \\ \eta \end{array} \right]$$

$$(3.38)$$

• Sensibilité par rapport à d $(\sigma_{y_{/}d})$:

$$\Delta^{(\alpha)} \begin{bmatrix} \sigma_{x_{(k+1)}/d} \end{bmatrix} = A \sigma_{x/d}$$

$$\sigma_{y/d} = (C + F\eta') \sigma_{x/d} + \begin{bmatrix} \frac{\partial D}{\partial d} & \frac{\partial F}{\partial d} \end{bmatrix} \begin{bmatrix} u \\ \eta \end{bmatrix}$$
(3.39)

Afin de réduire la charge de calcul de l'algorithme, les fonctions de sensibilité par rapport au vecteur p sont calculées à la place de celles du vecteur F. Par exemple, pour le cas $r = 2, p = \begin{bmatrix} 1 & p_2 \end{bmatrix}$, la longueur du vecteur F est 6, tandis que pour $r = 3, p = \begin{bmatrix} 1 & p_2 & p_3 \end{bmatrix}$ et la longueur du vecteur F est 16; de cette manière, le calcul des fonctions de sensibilité par rapport à p au lieu de la fonction de sensibilité par rapport à F, réduit drastiquement la complexité de l'algorithme.

• Sensibilité par rapport à p_j $(\sigma_{y_j p_j})$:

$$\begin{cases}
\Delta^{(\alpha)} \left[\sigma_{x_{(k+1)}/p_j} \right] = A \sigma_{x/p_j} \\
\sigma_{y/p_j} = (C + F \eta') \sigma_{x/p_j} + \left[\frac{\partial F}{\partial p_j} \right] [\eta]
\end{cases}$$
(3.40)

avec

$$\frac{\partial F}{\partial p} = \begin{bmatrix} Z^2 & Z^3 \dots & Z^r \end{bmatrix}$$
(3.41)

 et

$$Z = \begin{bmatrix} c_1 & \dots & c_{n_a} & d \end{bmatrix}$$
(3.42)

• Sensibilité par rapport à α :

La fonction de sensibilité par rapport à chaque ordre fractionnaire α est calculée en utilisant l'approximation numérique de la dérivée :

$$y(k, \alpha + \delta \alpha) - y(k, \alpha) \simeq \delta \alpha \frac{\partial y}{\partial \alpha} = \delta \alpha \sigma_{y/\alpha}$$
 (3.43)

avec

$$\sigma_{y/\alpha} = \frac{y(k, \alpha + \delta\alpha) - y(k, \alpha)}{\delta\alpha}$$
(3.44)

et $\delta \alpha$ étant une petite variation de α .

A chaque itération, les fonctions de sensibilité sont évaluées afin de calculer le gradient J'et le Hessian J'' et ainsi mettre à jour le vecteur de paramètres θ .

Dans la suite, une autre approche d'identification est proposée pour estimer le vecteur de paramètres du modèle de Wiener fractionnaire, écrit sous forme de régression fractionnaire.

3.4.3 Identification du système de Wiener sous forme de régression fractionnaire

Dans cette section l'identification du modèle de Wiener fractionnaire, décrit dans le chapitre précédent (Section 2.6.2), est étudiée en utilisant l'algorithme PSO.

Les équations qui décrivent le système non linéaire de Wiener fractionnaire sont données par :

$$y(k) = \phi^T(k)\theta + e(k) \tag{3.45}$$

Avec

$$\begin{cases} \phi(k) = \left[\phi_1^T(k) & \phi_2^T(k) \right]^T \\ \\ \tilde{\theta} = \left[\tilde{\theta}_1^T & \tilde{\theta}_2^T \right]^T \end{cases}$$
(3.46)

 et

$$\tilde{\theta}_1 = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_1 & \dots & b_{n_b} \end{bmatrix}^T$$
(3.47)

$$\phi_{1}(k) = \begin{bmatrix} -\Delta^{\alpha_{i}}v(k-1) \\ \vdots \\ -\Delta^{\alpha_{i}}v(k-n_{a}) \\ \Delta^{\alpha_{j}}u(k-1) \\ \vdots \\ \Delta^{\alpha_{j}}u(k-n_{b}) \end{bmatrix}^{T}$$
(3.49)
$$\tilde{\theta_{2}} = \begin{bmatrix} p_{2} & p_{3} & \dots & p_{r} \end{bmatrix}^{T}$$

$$\phi_2(k) = \begin{bmatrix} v^2(k) \\ v^3(k) \\ \vdots \\ v^r(k) \end{bmatrix}$$
(3.50)

Nous notons que le vecteur d'information $\phi(k)$ contient le signal intermédiaire inconnu v(k). Par conséquent, l'utilisation de l'algorithme PSO directement ne sera pas efficace. La solution proposée dans ce cas est l'utilisation du principe key term et de remplacer le signal interne non mesurable par son estimé et le vecteur θ par son estimé $\hat{\theta}$ à l'itération (k-1) [141].

Soit $\hat{v}(k)$ l'estimation de v(k) à l'itération k, et $\hat{\phi}(k)$ l'estimation du vecteur d'informations $\phi(k)$ qui est est obtenu en remplaçant $\Delta^{\alpha}v(k-i)$ par $\Delta^{\alpha}\hat{v}(k-i)$ et v(k) avec son estimé $\hat{v}(k)$.

$$\hat{\phi}(k) = \begin{bmatrix} \hat{\phi}_1^T(k) & \hat{\phi}_2^T(k) \end{bmatrix}^T$$
(3.51)

$$\hat{\phi}_1(k) = \begin{bmatrix} -\Delta^{\alpha} \hat{v}(k-1) & \dots & -\Delta^{\alpha} \hat{v}(k-n_a) \\ \Delta^{\alpha} u(k-1) & \dots & \Delta^{\alpha} u(k-n_b) \end{bmatrix}^T$$

$$\hat{\phi}_2(k) = \begin{bmatrix} \hat{v}^2(k) & \hat{v}^3(k) & \dots & \hat{v}^r(k) \end{bmatrix}^T$$
(3.52)

Ainsi, le modèle de Wiener décrit sous forme de régression fractionnaire peut être identifié; en considérant le signal intermédiaire non mesurable (v(k)) comme key term.

3.5 Conclusion

L'objectif de ce chapitre est l'identification des systèmes non linéaires entiers et fractionnaires. L'identification du système de Wiener non linéaire a en particulier été mis en évidence.

La première partie de ce chapitre a été consacrée à l'identification du modèle de Wiener dans le cas entier. Pour cela, une méthode à erreur de sortie et une méthode heuristique ont été adaptées pour l'estimation des paramètres du modèle de Wiener, décrit sous forme PNLSS.

Par la suite, l'identification des systèmes fractionnaires a été développée. Dans le cas du système de Wiener représenté sous forme PNLFSS, les méthodes d'identification utilisées

au cas entier sont également adaptées et considérées dans le cas fractionnaire. En effet, la méthode de LM a été utilisée pour l'identification des paramètres des deux partie linéaire et non linéaire du modèle ainsi que l'ordre fractionnaire. Cet estimateur est basé sur un calcul complexe des fonctions de sensibilité. De plus, l'algorithme SAVPSO a aussi été considéré pour identifier les paramètres et l'ordre fractionnaire du système. Cet algorithme prend aussi compte des contraintes appliquées a la fonction objective à minimiser. Cette contrainte concerne en particulier l'ordre de la dérivation non entière.

De plus, l'identification du modèle non linéaire de Wiener a été étudiée, dans le cas où le système est écrit sous forme de régression fractionnaire. En fait, l'algorithme de PSO a été utilisé afin d'estimer tous les paramètres du système fractionnaire non linéaire écrit sous cette forme.

Le chapitre suivant fera l'objet des résultats de simulations obtenus, illustrant par conséquent l'efficacité des algorithmes d'identification utilisés.

Chapitre 4

Résultats de simulation et discussions

4.1 Introduction

Le large domaine d'application du système de Wiener, ainsi que sa capacité de reproduire le comportement de nombreux phénomènes physiques rendent l'étude de ce modèle intéressante, non seulement pour les chercheurs, mais aussi pour les ingénieurs. Par conséquent, malgré les difficultés théoriques, le problème de l'identification du système de Wiener mérite d'être approfondi.

Dans ce chapitre, des exemples numériques seront présentés afin d'évaluer et d'illustrer les performances des méthodes d'identification développées dans le chapitre précédent. Ces méthodes sont appliquées aux modèles de Wiener dans le cas entier, en considérant deux approches, ainsi que leurs application au cas fractionnaire.

Dans les exemples suivants, l'entrée u(k) est une séquence d'excitation persistante aléatoire de moyenne nulle et de variance unitaire.

La perturbation e(k) est considérée comme une séquence de bruit blanc de moyenne nulle. Les résultats de simulations, réalisées pour le cas sans bruit et pour différents rapport signal sur bruit (SNR), seront donnés.

4.2 Identification du modèle de Wiener PNLSS au cas entier

L'identification du modèle de Wiener PNLSS, décrit dans la section (1.4) au cas entier, est effectuée en utilisant les algorithmes de LM et SAVPSO.

4.2.1 Identification par l'algorithme LM

La méthode d'identification de LM, détaillée dans la section (3.3.1) est appliquée à un modèle de Wiener avec un nombre d'états $n_a = 2$ pour le bloc linéaire, avec les matrices :

$$A = \begin{bmatrix} 0 & 1 \\ -0.3 & -0.4 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$C = \begin{bmatrix} 0.3 & 0.1 \end{bmatrix} \quad d = \begin{bmatrix} 0.2 \end{bmatrix}$$

Le bloc non linéaire est un polynôme de degré r = 2, définie comme suit :

$$f(v(k)) = v(k) + 0.5v^2(k)$$

Alors la forme PNLSS peut se déduire comme suit :

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 1 \\ -0.3 & -0.4 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0.3 & 0.1 \end{bmatrix} x(k) + \begin{bmatrix} 0.2 \end{bmatrix} u(k) \\ + \begin{bmatrix} 0.045 & 0.03 & 0.06 & 0.005 & 0.02 & 0.02 \end{bmatrix} \begin{bmatrix} x_1^2(k) \\ x_1(k)x_2(k) \\ x_1(k)u(k) \\ x_2^2(k) \\ x_2(k)u(k) \\ u^2(k) \end{bmatrix}$$
(4.1)

Le vecteur de paramètres θ à estimer est donc :

$$\theta = \begin{bmatrix} -0.3 & -0.4 & 0.3 & 0.1 & 0.2 & 0.5 \end{bmatrix}$$

Les simulations ont été effectuées d'abord en absence de bruit, puis avec des données corrompues par du bruit blanc pour différents rapports signal/bruit (SNR = 34dB et SNR = 15dB).

Le Tableau 4.1 montre les résultats de la simulation pour différents rapports signal sur bruit. Les résultats obtenus montrent que les erreurs sont très faibles, donc l'efficacité de

	Sansbruit	SNR = 34dB	SNR = 15 dB	$valeurs \ exactes$
a_1	-0.300	-0.294	-0.328	-0.300
a_2	-0.400	-0.397	-0.417	-0.400
c_1	0.300	0.300	0.292	0.300
c_2	0.100	0.102	0.089	0.100
d	0.200	0.018	0.195	0.200
p_2	0.500	0.504	0.548	0.500
J	4e - 16	2e - 3	2e - 2	_

TABLE 4.1: Résultats d'identification du modèle de Wiener PNLSS avec LM

la méthode peut être appréciée.

L'erreur de prédiction de sortie pour le cas sans bruit, représentée sur la figure (4.1(a)), est nulle avec un critère ($J \approx 10^{-16}$); ainsi, la courbe de la réponse temporelle du système estimé chevauche les données sur la figure (4.1(b)).

En présence de bruit $SNR = 34 \, dB$, et $SNR = 15 \, dB$, la réponse de sortie du modèle estimé est comparée aux données respectivement sur la figure (4.2(b)) et sur la figure (4.3(b)), elles montrent une adéquation parfaite. L'erreur de prédiction de sortie pour $SNR = 34 \, dB$ est représentée sur la figure (4.2(a)), et la figure (4.3(a)) représente l'erreur pour $SNR = 15 \, dB$.

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



(b) Sortie estimée et sortie mesurée

FIGURE 4.1: Résultats de la simulation avec l'algorithme LM pour un cas sans bruit.



(b) Sortie estimée et sortie mesurée

FIGURE 4.2: Résultats de la simulation avec l'algorithme LM pour $SNR = 34 \ dB$.



(b) Sortie estimée et sortie mesurée

FIGURE 4.3: Résultats de la simulation avec l'algorithme LM pour $SNR = 15 \ dB$.

4.2.2 Identification par l'algorithme SAVPSO

Nous présentons maintenant l'exemple d'un système non linéaire (Wiener) qui peut être identifié avec la méthode SAVPSO, expliquée dans la section (3.3.2). Les matrices du bloc linéaire sont données comme suit :

$$A = \begin{bmatrix} 0 & 1 \\ -0.3 & -0.2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$C = \begin{bmatrix} 0.5 & 0.1 \end{bmatrix} \quad d = \begin{bmatrix} 0.4 \end{bmatrix}$$

Le bloc non linéaire est un polynôme de degré r = 2 défini comme suit :

$$f(v(k)) = v(k) + 0.2v^2(k)$$

Le système de Wiener est alors décrit par les équations PNLSS suivantes :

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 1 \\ -0.3 & -0.2 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0.5 & 0.1 \end{bmatrix} x(k) + \begin{bmatrix} 0.4 \end{bmatrix} u(k) \\ + \begin{bmatrix} 0.05 & 0.02 & 0.08 & 0.002 & 0.016 & 0.032 \end{bmatrix} \begin{bmatrix} x_1^2(k) \\ x_1(k)x_2(k) \\ x_1(k)u(k) \\ x_2^2(k) \\ x_2(k)u(k) \\ u^2(k) \end{bmatrix}$$
(4.2)

L'idée est de commencer avec une structure de modèle PNLSS connue, de sorte que la fonction de coût converge vers zéro dans le cas sans bruit et dans le cas avec bruit. Le vecteur de paramètres θ à estimer est donné par :

$$\theta = \begin{bmatrix} -0.3 & -0.2 & 0.5 & 0.1 & 0.4 & 0.2 \end{bmatrix}$$

Le Tableau (4.2) montre les résultats de la simulation pour différents rapports signal sur bruit : SNR = 34dB, SNR = 15dB et pour le cas sans bruit. L'erreur de prédiction de la sortie pour le cas sans bruit, représentée sur la figure (4.4(a)), est nulle avec un critère

	Sansbruit	SNR = 34dB	SNR = 15 dB	Valeurs exactes
a_1	-0.300	-0.2899	-0.325	-0.300
a_2	-0.200	-0.2049	-0.161	-0.200
c_1	0.500	0.497	0.493	0.500
c_2	0.100	0.103	0.105	0.100
d	0.400	0.395	0.398	0.400
p_2	0.200	0.202	0.198	0.200
J	6e - 16	9e - 3	0.0769	_

TABLE 4.2: Résultats d'identification du modèle de Wiener PNLSS avec SAVPSO

 $(J \approx 10^{-16})$; ainsi, la courbe de la réponse temporelle du système estimé correspond aux données dans la figure (4.4(b)).

La performance de l'estimateur est également étudiée en présence de bruit. Les résultats d'identification sont indiqués sur la figure (4.5(b)) et la figure (4.6(b)) qui représentent la sortie estimée du système et celle simulée respectivement pour SNR = 34dBet SNR = 15dB; tandis que l'erreur de sortie est observée sur la figure (4.5(a)) pour SNR = 34dB et la figure (4.6(a) pour SNR = 15dB.

Les résultats obtenus démontrent l'efficacité de la méthode proposée.



(b) Sortie estimée et sortie mesurée

FIGURE 4.4: Résultats de la simulation avec l'algorithme SAVPSO pour un cas sans bruit.



(b) Sortie estimée et sortie mesurée

FIGURE 4.5: Résultats de la simulation avec l'algorithme SAVPSO pour $SNR = 34 \ dB$.

4.3 Identification du système de Wiener au cas fractionnaire

Précédemment, l'identification du système de Wiener, décrit sous forme PNLSS au cas entier, était traitée. Dans la suite, le problème d'identification du système de Wiener sera étudié, dans le cas où le système est modélisé sous forme PNLSS fractionnaire (PNLFSS)



(b) Sortie estimée et sortie mesurée

FIGURE 4.6: Résultats de la simulation avec l'algorithme SAVPSO pour $SNR = 15 \ dB$. et sous forme de régression fractionnaire.

4.3.1 Identification du système de Wiener PNLFSS

Dans cette section, l'identification du système de Wiener basée sur l'approche PNLSS fractionnaire, discutée dans la section (2.6.1), est considérée. Dans ce but, les algorithmes de LM et SAVPSO sont utilisés.

4.3.1.1 Identification du système de Wiener PNLFSS en utilisant la méthode SAVPSO

La méthode proposée dans la section (3.4.1) est appliquée pour l'identification du modèle de Wiener PNLSS fractionnaire (PNLFSS).

Les exemples suivants présentent les résultats obtenus dans le cas particulier d'ordre fractionnaire commensurable et le cas général non commensurable.

• Exemple 1

Considérons le modèle de Wiener avec la partie linéaire représentée par un modèle d'état d'ordre fractionnaire commensurable, avec les matrices suivantes :

$$A = \begin{bmatrix} 0 & 1 \\ -0.1 & -0.3 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$C = \begin{bmatrix} 0.2 & 0.5 \end{bmatrix} \qquad D = \begin{bmatrix} 0.1 \end{bmatrix}$$

avec l'ordre fractionnaire $\alpha = 0.5$.

La partie non-linéaire est un polynôme :

$$f(v(k)) = v(k) + 0.3v^2(k);$$

Le système de Wiener PNLFSS au cas commensurable est représenté par les equa-

tions suivantes :

$$\Delta^{\alpha} x(k+1) = \begin{bmatrix} 0 & 1 \\ -0.1 & -0.3 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 0.2 & 0.5 \end{bmatrix} x(k) + \begin{bmatrix} 0.1 \end{bmatrix} u(k)$$

$$+ \begin{bmatrix} 0.012 & 0.060 & 0.012 & 0.075 & 0.030 & 0.003 \end{bmatrix} \begin{bmatrix} x_1^2(k) \\ x_1(k)x_2(k) \\ x_1(k)u(k) \\ x_2^2(k) \\ x_2(k)u(k) \\ u^2(k) \\ u^2(k) \end{bmatrix}$$
(4.3)

Le vecteur des paramètres du modèle à estimer est :

$$\theta = \begin{bmatrix} -0.1 & -0.3 & 0.2 & 0.5 & 0.1 & 0.3 & 0.5 \end{bmatrix}$$
(4.4)

L'utilisation de l'algorithme SAVPSO pour identifier les paramètres du modèle de Wiener PNLSS fractionnaire, donne les résultats suivants :

Pour le cas sans bruit, les résultats obtenus sont représentés par la figure (4.7(a)) qui représente l'erreur de prédiction ($J \approx 10^{-16}$) et la figure (4.7(b)). Cette dernière, illustrant la sortie estimée et la sortie simulée, montre une correspondance parfaite du modèle simulé et estimé.

Dans le cas de présence de bruit, l'erreur de prédiction est également donnée par les figures (4.8(a)) et (4.9(a)) pour $SNR = 34 \ dB$ et $SNR = 15 \ dB$ respectivement. De plus, la figure (4.8(b)) et la figure (4.9(b)) représentent la sortie estimée et celle simulée pour $SNR = 34 \ dB$ et $SNR = 15 \ dB$ respectivement.

Ainsi, les résultats d'identification obtenus montrent que l'algorithme estime les paramètres du modèle de Wiener PNLFSS avec une bonne précision.

	Sansbruit	SNR = 34dB	SNR = 15dB	valeurs exactes
a_1	-0.100	-0.093	-0.1048	-0.100
a_2	-0.300	-0.279	-0.311	-0.300
c_1	0.200	0.197	0.211	0.200
c_2	0.500	0.496	0.518	0.500
d	0.100	0.097	0.091	0.100
p_2	0.300	0.302	0.321	0.300
α	0.500	0.369	0.530	0.500
J	7e - 16	5e - 3	4e - 2	_

 TABLE 4.3: Résultats d'identification du modèle de Wiener PNLFSS avec SAVPSO,

 exemple 1



(b) Sortie estimée et sortie mesurée

FIGURE 4.7: Résultats de l'exemple 1 fractionnaire, avec SAVPSO, sans bruit.

• Exemple 2

L'identification du modèle du système de Wiener fractionnaire est évaluée. Dans ce cas, la partie linéaire du modèle est un modèle d'état d'ordre fractionnaire non commensurable, avec les matrices suivantes :



(b) Sortie estimée et sortie mesurée

FIGURE 4.8: Résultats de l'exemple 1 fractionnaire, avec SAVPSO, $SNR = 34 \ dB$.

$$A = \begin{bmatrix} 0 & 1 \\ -0.3 & -0.1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$C = \begin{bmatrix} 0.2 & 0.1 \end{bmatrix} \quad D = \begin{bmatrix} 0.5 \end{bmatrix}$$

L'ordre fractionnaire $\alpha = \begin{bmatrix} 0.3 & 0.4 \end{bmatrix}$.



(b) Sortie estimée et sortie mesurée



Le bloc non linéaire est décrit par le polynôme suivant :

$$f(v(k)) = v(k) + 0.3v^2(k)$$

Par conséquent, le modèle de Wiener PNLFSS est :

$$\begin{split} \left(\begin{array}{ccc} \Delta^{\alpha} x(k+1) &= \left[\begin{array}{ccc} 0 & 1 \\ -0.3 & -0.1 \end{array} \right] x(k) + \left[\begin{array}{c} 0 \\ 1 \end{array} \right] u(k) \\ y(k) &= \left[\begin{array}{ccc} 0.2 & 0.1 \end{array} \right] x(k) + \left[\begin{array}{c} 0.5 \end{array} \right] u(k) \\ &+ \left[\begin{array}{cccc} 0.012 & 0.012 & 0.060 & 0.003 & 0.030 & 0.075 \end{array} \right] \left[\begin{array}{c} x_1^2(k) \\ x_1(k)x_2(k) \\ x_1(k)u(k) \\ x_2^2(k) \\ x_2(k)u(k) \\ u^2(k) \\ u^2(k) \end{array} \right] \right]$$

Ainsi, le vecteur de paramètres à estimer

$$\theta = \begin{bmatrix} -0.3 & -0.1 & 0.2 & 0.1 & 0.5 & 0.3 & 0.3 & 0.4 \end{bmatrix}$$
 (4.6)

TABLE 4.4: Résultats d'identification du modèle de Wiener PNLFSS avec SAVPSO,exemple 2

	Sansbruit	SNR = 34dB	SNR = 15 dB	Valeurs exactes
a_1	-0.300	-0.287	-0.298	-0.300
a_2	-0.100	-0.117	-0.053	-0.100
c_1	0.200	0.198	0.183	0.200
c_2	0.100	0.101	0.089	0.100
d	0.500	0.495	0500	0.500
p_2	0.300	0.304	0.304	0.300
α_1	0.300	0.385	0.512	0.300
α_2	0.400	0.472	0.512	0.400
J	7e - 16	9.3e - 4	2.2e - 3	_

L'erreur de prédiction de sortie pour le cas sans bruit, représentée sur la figure (4.10(a)), est nulle avec un critère ($J \approx 10^{-16}$); ainsi, la courbe de la réponse temporelle du système estimé chevauche les données dans la figure (4.10(b)).



(a) Erreur de prédiction



(b) Sortie estimée et sortie mesurée

FIGURE 4.10: Résultats de l'exemple 2 fractionnaire, avec SAVPSO, sans bruit

En présence de bruit $SNR = 34 \ dB$, et $SNR = 15 \ dB$, la réponse de la sortie du modèle estimé est comparée avec les données respectivement dans les figures (4.11(a)) et (4.12(a)), qui montrent une adéquation parfaite.

L'erreur de prédiction de la sortie pour $SNR = 34 \ dB$, représentée dans la figure (4.8(b)), est très faible ($\approx 10^{-3}$) et la figure (4.9(b)) représente l'erreur pour $SNR = 15 \ dB$.

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



(b) Sortie estimée et sortie mesurée





(b) Sortie estimée et sortie mesurée



4.3.1.2 Identification du système de Wiener PNLFSS en utilisant l'algorithme de LM

Dans cette section, des exemples numériques sont présentés afin d'évaluer et d'illustrer la performance de la méthode proposée dans la section (3.4.2). En effet, le modèle de Wiener fractionnaire est considéré pour le cas commensurable et le cas général non commensurable.

• Exemple 1

Dans cet exemple, le modèle de Wiener fractionnaire dans le cas commensurable est testé avec la partie linéaire d'ordre $n_a = 3$ et l'ordre de la partie non linéaire r = 3. Par conséquent, il est décrit par les équations ci-dessous :

$$\Delta^{0.3}x(k+1) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.2 & -0.4 & -0.3 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$v(k) = \begin{bmatrix} 0.3 & 0.2 & 0.1 \end{bmatrix} x(k) + \begin{bmatrix} 0.1 \end{bmatrix} u(k),$$

$$y(k) = v(k) + 0.1v^{2}(k) + 0.5v^{3}(k)$$

$$(4.7)$$

Le modèle PNLFSS est déduit et le vecteur de paramètres à estimer est :

$$\theta = \begin{bmatrix} -0.2 & -0.4 & -0.3 & 0.3 & 0.2 & 0.1 & 0.1 & 0.1 & 0.5 & 0.3 \end{bmatrix}.$$
(4.8)

Cependant, la procédure d'identification nécessite l'identification paramétrique ainsi que le choix de la structure du modèle. Dans ce but, la structure est sélectionnée sur la base de plusieurs tests sur différents choix de la structure et l'évolution du critère observé.

La Tableau (4.5) donne les résultats de l'évolution du critère par rapport à la structure, et la figure (4.13) confirme que l'ordre de la partie linéaire et non linéaire est respectivement $n_a = 3$ et r = 3 (J = e - 8). Basé sur le résultat du meilleur critère, l'utilisation de $n_a = 3$ et r = 3 est confirmée.

Structuro	$n_a = 2$	$n_a = 2$	$n_a = 3$	$n_a = 3$
Structure	r = 2	r = 3	r = 2	r = 3
J	0.095	0.038	0.061	2e - 8

TABLE 4.5:Évolution du critère par rapport à la structure, exemple 1.



FIGURE 4.13: Évolution du critère par rapport à la structure, exemple 1.

La valeur moyenne des paramètres, utilisant la simulation de Monte Carlo, est donnée dans le Tableau 4.6 pour différents SNR. À partir du Tableau, nous pouvons TABLE 4.6: Moyennes des paramètres θ_i du modèle de Wiener PNLFSS avec LM de l'exemple 1

	Sansbruit	SNR = 34dB	SNR = 15 dB	Valeurs exactes
Moyenne de a_1	-0.200	-0.200	-0.191	-0.200
Moyenne de a_2	-0.400	-0.400	-0.377	-0.400
Moyenne de a_3	-0.300	-0.298	-0.271	-0.300
Moyenne de c_1	0.300	0.299	0.291	0.300
Moyenne de c_2	0.200	0.199	0.196	0.200
Moyenne de c_3	0.100	0.099	0.099	0.100
$Moyenne\ de\ d$	0.100	0.099	0.100	0.100
Moyenne de p_2	0.100	0.102	0.087	0.100
$Moyenne\ de\ p_3$	0.500	0.501	0.448	0.500
Moyenne de α	0.300	0.329	0.250	0.300
$Moyenne\ de\ J$	1e - 15	2e - 3	2e - 2	_

voir qu'une bonne estimation des paramètres est obtenue. En effet, dans le cas sans bruit, l'erreur de prédiction de sortie est nulle, comme le montre la figure (4.14(a)), avec le critère $J \simeq e - 15$. Par conséquent, une adéquation parfaite entre la réponse en sortie du modèle estimé et les données est montrée sur la figure (4.14(b)). Tandis que les figures (4.15) et (4.16) montrent la sortie du modèle simulé et celle du modèle estimé en présence de bruit avec $SNR = 34 \ dB$ et $SNR = 15 \ dB$ respectivement. Ces figures montrent une bonne adéquation, ce qui démontre l'efficacité de l'estimateur.



(b) Sortie estimée et sortie mesurée

FIGURE 4.14: Résultats de l'exemple 1 fractionnaire, avec LM, sans bruit.


FIGURE 4.15: Sorties estimées et simulées avec LM, exemple 1 fractionnaire, $SNR = 34 \ dB$.



FIGURE 4.16: Sorties estimées et simulées avec LM, exemple 1 fractionnaire, $SNR = 15 \ dB$.

• Exemple 2

Ici, on considère un modèle de Wiener fractionnaire dans le cas non commensurable, dont la partie linéaire est d'ordre $n_a = 2$ et la partie non linéaire d'ordre r = 3. Le modèle est décrit par :

$$\begin{cases} \Delta^{(\alpha)}x(k+1) = \begin{bmatrix} 0 & 1 \\ -0.1 & -0.4 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ v(k) = \begin{bmatrix} 0.3 & 0.2 \end{bmatrix} x(k) + \begin{bmatrix} 0.1 \end{bmatrix} u(k) \\ y(k) = v(k) + 0.3v^2(k) + 0.4v^3(k) \end{cases}$$
(4.9)

avec $(\alpha) = [0.2 \quad 0.5].$

Le vecteur des paramètres à estimer est :

$$\theta = \begin{bmatrix} -0.1 & -0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.4 & 0.2 & 0.5 \end{bmatrix}$$
(4.10)

La validation de la structure est déterminée sur la base des valeurs du critère pour différents ordres de n_a et r, comme indiqué dans le Tableau (4.7); Figure (4.17) montre son évolution. Par conséquent, il est confirmé que l'ordre des parties linéaire et non linéaire est respectivement $n_a = 2$ et r = 3.



FIGURE 4.17: Évolution du critère par rapport à la structure, exemple 2.

En utilisant le résultat ci-dessus, la Tableau (4.8) fournit les résultats de la simulation effectuée en l'absence de bruit, et les valeurs moyennes des paramètres estimés avec des données affectées par une séquence de bruit blanc pour différents SNR.

 $n_a = 2$ $n_a = 2$ $n_a = 3$ $n_a = 3$

 Structure
 r = 2 r = 3 r = 2 r = 3

 J
 0.010
 3e - 15 0.016
 0.015

TABLE 4.7: Évolution du critère par rapport à la structure, exemple 2.

TABLE 4.8: Moyennes des paramètres θ_i du modèle de Wiener PNLFSS avec LM de l'exemple 2.

	Sansbruit	SNR = 34dB	SNR = 15 dB	Valeurs exactes
Moyenne de a_1	-0.100	-0.100	-0.105	-0.100
Moyenne de a_2	-0.400	-0.402	-0.389	-0.400
Moyenne de c_1	0.300	0.300	0.297	0.300
Moyenne de c_2	0.200	0.200	0.199	0.200
$Moyenne\ de\ d$	0.100	0.099	0.099	0.100
Moyenne de p_2	0.300	0.300	0.302	0.300
$Moyenne\ de\ p_3$	0.400	0.401	0.488	0.400
Moyenne de α_1	0.200	0.180	0.268	0.200
Moyenne de α_2	0.500	0.484	0.385	0.500
Moyenne de J	4e - 16	2e - 3	2e - 2	_

Les résultats du Tableau (4.8) montrent que l'approche proposée restitue avec précision les paramètres. Ceci peut être observé dans le graphe esquissé dans la figure (4.18(a)), représentant l'erreur de prédiction qui est pratiquement nulle avec le critère J = e - 16. La figure (4.18(b)) montre une correspondance parfaite de la réponse en sortie du modèle estimé et des données pour le cas sans bruit.

Figures (4.19) et (4.20), représentant la sortie du système estimé et celle simulée en présence du bruit, pour $SNR = 34 \ dB$ et $SNR = 15 \ dB$.



(b) Sortie estimée et sortie mesurée

FIGURE 4.18: Résultats de l'exemple 2 fractionnaire, avec LM, sans bruit.



FIGURE 4.19: Sorties estimées et simulées avec LM, exemple 2 fractionnaire, $SNR = 34 \ dB$.



FIGURE 4.20: Sorties estimées et simulées avec LM, exemple 2 fractionnaire, $SNR = 15 \ dB$.

Par conséquent, l'efficacité de la méthode proposée a été validée avec ces exemples de simulation et la bonne performance de l'estimateur est confirmée à la fois pour le cas fractionnaire commensurable et pour l'ordre non commensurable général, même en présence de bruit.

4.3.2 Identification du modèle de Wiener : régression fractionnaire

La sortie du modèle de Wiener, décrit sous forme de régression fractionnaire (Section (2.6.2)) en fonction du signal intermédiaire des deux parties du modèle de Wiener, peut être identifiée en utilisant l'algorithme PSO, basée sur le principe key term.

Ainsi, la méthode est appliquée au système de Wiener fractionnaire, décrit par les équations suivantes :

$$v(k) = 0.2\Delta^{0.3}v(k-1) + 0.3\Delta^{0.3}v(k-2) + 0.8\Delta^{0.3}y(k-1) + 0.1\Delta^{0.3}y(k-2)$$
(4.11)
$$y(k) = v(k) + 0.5v^{2}(k) + e(k).$$

Le vecteur de paramètres à estimer est :

$$\theta = [-0.2 - 0.3 - 0.8 - 0.1 - 0.5 - 0.3]$$
 (4.12)

L'exemple de simulation a été testé pour le cas sans bruit et pour différents rapports signal sur bruit, les résultats sont donnés dans le Tableau (4.9).

La figure (4.21(b)) représente la courbe de la réponse temporelle du système estimé qui correspond aux données, en absence de bruit.

En présence de bruit SNR = 34dB et SNR = 15dB, le critère est très bas (figures (4.22(a)) et (4.23(a)). Les figures (4.22(b)) et (4.23(b)) représentent respectivement la réponse de sortie du modèle estimé par rapport aux données pour SNR = 34dB et SNR = 15dB. Les résultats obtenus, vérifient l'efficacité de l'approche proposée.

	Sansbruit	SNR = 34dB	SNR = 15dB	Valeurs exactes
a_1	-0.200	-0.198	-0.192	-0.200
a_2	-0.300	-0.295	-0.305	-0.300
b_1	0.800	0.798	0.809	0.800
b_2	0.100	0.110	0.055	0.100
p_2	0.500	0.514	0.435	0.500
α	0.300	0.306	0.298	0.300
J	7e - 16	2e - 2	1e - 1	_

TABLE 4.9: Résultats de la simulation avec PSO pour différents SNR



(b) Sortie estimée et sortie mesurée

FIGURE 4.21: Résultats de la simulation avec l'algorithme PSO pour un cas sans bruit.



(b) Sortie estimée et sortie mesurée

FIGURE 4.22: Résultats de la simulation avec l'algorithme PSO pour $SNR = 34 \, dB$.

4.4 Conclusion

Ce chapitre est consacré à présenter les résultats de simulations, obtenus, en utilisant les algorithmes d'identification cités dans le chapitre précédent.

En effet, le chapitre est structuré en deux parties principales. Dans un premier temps, des exemples de simulation pour tester les algorithmes choisis pour d'identification du système de Wiener, au cas entier ont été développés. Cette étude permet de comprendre,



(b) Sortie estimée et sortie mesurée

FIGURE 4.23: Résultats de la simulation avec l'algorithme PSO pour $SNR = 15 \ dB$.

clarifier et justifier l'utilisation de ces algorithmes pour l'identification du système non linéaire de Wiener au cas fractionnaire.

Dans une seconde étape, l'identification du système non linéaire au cas fractionnaire est réalisée. Par conséquent, une méthode d'optimisation non linéaire et une méthode heuristique sont adaptées au cas du système de Wiener fractionnaire.

Les résultats obtenus montrent que la qualité des algorithmes d'identification est satisfai-

sante même pour un niveau de bruit élevé.

Conclusion générale

Notre travail de thèse porte essentiellement sur la description et l'identification des systèmes non linéaires d'ordre fractionnaire du type Wiener. En effet des algorithmes d'identification complexes sont adaptés et appliqués dans ce sens tout en montrant leurs avantages.

La démarche suivie dans cette étude peut se décomposer en quatre chapitres.

Dans le premier chapitre nous avons présenté les principales classes des systèmes non linéaires entiers. L'accent a été mis par la suite sur la description du modèle de Wiener, sur lequel s'appuie notre étude. Cette description peut se faire en imposant que le modèle non linéaire est constitué d'une succession d'un modèle d'état linéaire par une non-linéarité polynomiale. Ceci a permis de décrire le système de Wiener sous forme de l'approche PNLSS.

Le deuxième chapitre a été consacré à donner quelques rappels et notions principales sur le calcul fractionnaire considérés dans le cadre de cette thèse. En effet, les différents modèles permettant la représentation et la description d'un système d'ordre fractionnaire ont été présentés. Par la suite, les méthodes de simulation des systèmes fractionnaires ont été aussi exposées. La dernière partie du deuxième chapitre a été dédiée à l'extension de la description du système non linéaire Wiener au cas fractionnaire, à savoir la représentation sous forme PNLSS fractionnaire et régression fractionnaire.

Dans le troisième chapitre, nous avons introduit et développé en détails les méthodes proposées pour l'identification des systèmes non linéaires de Wiener. Ce dernier décrit sous forme PNLSS et sous forme de régression, a été étudié aussi bien dans le cas d'ordre entier que dans le cas d'ordre fractionnaire. Dans le but de valider et de tester les algorithmes proposés, nous avons illustré dans le dernier chapitre les résultats de simulation. La discussion des principaux résultats obtenus montre l'efficacité des méthodes d'identification utilisées, même en présence d'un pourcentage de bruit important.

En somme, on peut résumer les principales contributions de notre travail comme suit :

- Identification du système non linéaire Wiener PNLSS en utilisant l'algorithme LM.
- Identification du système non linéaire Wiener PNLSS en utilisant l'algorithme SAVPSO.

• Description du système non linéaire Wiener avec l'approche PNLSS fractionnaire et sous forme de régression fractionnaire.

• Identification des systèmes non linéaires des modèles fractionnaires de type Wiener.

• Extension et adaptation de l'algorithme de LM et l'algorithme SAVPSO au cas fractionnaire, permettant l'identification et l'estimation de tous les paramètres du modèle de Wiener PNLSS fractionnaire ainsi que l'ordre de dérivation fractionnaire dans le cas général et dans le cas particulier d'ordre commensurable.

• Identification du système Wiener décrit sous forme de régression fractionnaire en utilisant et l'algorithme PSO standard, basé sur le principe key term.

Notre travail ouvre par ailleurs de nombreuses perspectives qui peuvent améliorer et compléter son contenu. Les perspectives proposées sont :

• Extension des méthodes proposées pour l'identification des modèles de Wiener fractionnaire pour le cas de systèmes multivariables MIMO (Multiple-Input Multiple-Output).

- Identification du système de Wiener fractionnaire sous autres formes.
- Validation des résultats obtenus sur des systèmes réels.

• Utilisation des approches proposées pour l'identification d'autres modèles blocs orientés fractionnaires.

Bibliographie

- [1] L. Ljung, System identification : theory for the user, Prentice Hall, London, 1987.
- [2] R. Haber and L. Keviczky, "Nonlinear System Identification : Nonlinear System Parameter Identification", *Dordrecht, Netherlands : Kluwer Academic Publishers*, vol. 1, 1999.
- [3] T. B. Schön, A. Wills and B. Ninness, "System identification of non-linear state space models", *Automatica*, vol. 47, no. 1, pp. 39-49, 2011.
- [4] T.A. Johansen and B.A. Foss, "Identification of non-linear system structure and parameters using regime decomposition", *Automatica*, vol. 31, no. 2, pp. 321-326, 1995.
- [5] F. Giri and E.W. Bai, Block-oriented non-linear system identification, 1st ed. Springer, 2010.
- [6] K. Li, J. X. Peng and E. W. Bai, "A two-stage algorithm for identification of nonlinear dynamic systems", *Automatica*, vol. 42, no. 7, pp. 1189-1197, 2006.
- [7] R. Haber and H. Unbehauen, "Structure identification of nonlinear dynamic systems-A survey on Input/Ouput Approaches", Automatica, vol. 26, no. 4, pp. 651-677, 1990.
- [8] E. W. Bai, "Identification of linear systems with hard input non-linearities of known structure", Automatica, vol. 38, no. 5, pp. 853-860, 2002.
- [9] 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.

- [10] S.A. Billings and W.S.F. Voon, "Structure detection and model validity tests in the identification of non-linear systems", *IEE Proceedings*, Pt. D, vol. 130, no. 4, pp. 193-199, July 1983.
- [11] J. Zhao, X. Ma, S. Zhao and J. Fei, "Hammerstein identification of supercharged boiler superheated steam pressure using Laguerre-Fuzzy model", *International Journal* of Heat and Mass Transfer, vol. 70, pp. 33-39, 2014.
- [12] M. Lawrynczuk, "Computationally efficient model predictive control algorithms : A Neural Network Approach", Studies in Systems, Decision and Control Cham, Switzerland : Springer, vol. 3, 2014.
- [13] X. Wang, K. Wu, J. H. Lu, and W. G. Xiang, "Nonlinear identification of alstom gasifier based on Wiener model", In International Conference on Sustainable Power Generat Supply (SUPERGEN), Nanjing, China, pp. 1-7, Apr, 2009.
- [14] Y.B. Hu, B.L. Liu, Q. Zhou and C. Yang, "Recursive extended least squares parameter estimation for Wiener nonlinear systems with moving average noises", *Circuits, Systems, and Signal Processing*, vol. 33, no. 2, pp. 655-664, 2014.
- [15] F. Ding, X.M. Liu and M.M. Liu, "The recursive least squares identification algorithm for a class of Wiener nonlinear systems", *Journal of the Franklin Institute*, vol. 353, no. 7, pp. 1518-1526, 2016.
- [16] A. Wills and L. Ljung, Wiener system identification using the maximum likelihood method, In Block-Oriented Nonlinear System Identification, Lecture Notes in Control and Information Science, pp. 89-110. Springer, 2010.
- [17] P.F. Cao and X.L. Luo, "Soft sensor model derived from Wiener model structure : modeling and identification", *Chinese Journal of Chemical Engineering*, vol. 22, no. 5, pp. 538-548, 2014.
- [18] A. Janczak, Identification of nonlinear systems using neural networks and polynomial models : a block-oriented approach, In A Block-Oriented Approach (Lecture Notes in Control and Information Sciences), Springer Science & Business Media, vol. 310, 2004.

- [19] A. Janczak, "Fault detection and isolation in Wiener systems with inverse model of static nonlinear element", In : Control Conference (ECC), 1999 European. IEEE, Karlsruhe, Germany, pp. 4249-4254. Aug. 1999.
- [20] A. D. Kalafatis, L. Wang, W. R. Cluett, "Identification of time-varying pH processes using sinusoidal signals", *Automatica*, vol. 41, no. 4, pp. 685-691, 2005.
- [21] A. Srinivasan and P. Lakshmi, "Wiener Model Based Real-Time Identification and Control of Heat Exchanger Process", J. Automation and Systems Engineering, 2008.
- [22] W. Liu, W. Na, L. Zhu, J. Ma and Q. J. Zhang, "A Wiener-Type Dynamic Neural Network Approach to the Modeling of Nonlinear Microwave Devices", *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 6, pp. 2043-2062, 2017.
- [23] A. A. Kilbas, H.M. Srivasta and J. J.Trujillo, Theory and Applications of Fractional Differential Equations, In : vanMill, J. (ed.) North HollandMathematics Studies, Elsevier, Amsterdam, 2006.
- [24] O. Cois, Systèmes linéaires non entiers et identification par modèle non entier : Application en thermique, Thèse de Doctorat, Université de Bordeaux, France, 2002.
- [25] J. Battaglia, L. Le Lay, J.C. Batsale, A. Oustaloup, and O. Cois, "Heat flux stimation through inverted non integer identification models", *International Journal of Thermal Science*, vol. 39, pp. 374-389, 2000.
- [26] J. Sabatier, M. Aoun, A. Oustaloup, G. Gregoire, F. Ragot and P. Roy, "Estimation of lead acid battery state of charge with a novel fractional model", *Inproc IFAC work*shop on fractional differentiation and its applications, pp. 362-67, Porto, Portugal, 2006.
- [27] R. Darling and J. Newman, "On the short behaviour of porous interaction electrodes", Journal of the Electrochemical Society, vol. 144, pp. 3057-3063, 1997.
- [28] C.R. Serment, Synthèse d'un isolateur d'ordre non entier fondé sur une architecture arborescente d'éléments viscoélastiques quasi-identiques, Thèse de Doctorat, Université Bordeaux 1, France, 2001.

- [29] C. Jutten, Systèmes asservis non linéaires, Université Joseph Fourier-Polytech, Grenoble, Août, 2006.
- [30] P. Müllhaupt, "Introduction à l'analyse et à la commande des systèmes non linéaires", PPUR Presses polytechniques, 2009.
- [31] O. Nelles, Nonlinear system identification. From classical approaches to neural networks and fuzzy models, Springer, New York, Berlin, Heidelberg, 2001.
- [32] S.A. Billings and I.J. Leontaritis, "Identification of nonlinear systems using parametric estimation techniques", *Proceedings of the IEE Conference on Control and its Application, Warwick, UK*, pp. 183-187, 1981.
- [33] S.A. Billings and S.Y. Fakhouri, "Identification of systems containing linear dynamic and static nonlinear elements", *Automatica (Journal of IFAC)*, vol. 18, no. 1, pp. 15-26, 1982.
- [34] I.J. Leontaritis and S.A. Billings, "Input-output parametric models for non-linear systems-Part I : Deterministic non-linear systems", *International Journal of Control*, vol. 41, no. 2, pp. 303-328, 1985.
- [35] S.A. Billings, Nonlinear System Identification : NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, Wiley, 2013.
- [36] V. Volterra, Theory of functionals and of integral and integro-differential equations, Dover : New York, 1959.
- [37] W. J. Rugh, Non linear theory : The Volterra/Wiener approach, Baltimore, MD, USA : Johns Hopkins University Press, 1981.
- [38] T. Hélie and D. Roze, "Sound synthesis of a nonlinear string using Volterra series", Journal of Sound and Vibration, vol. 314, no. 1-2, pp. 275-306, 2008.
- [39] C. A. Schmidt, S. I. Biagiola, J. E. Cousseau and J. L. Figueroa, "Volterra-type models for nonlinear systems identification", *Applied Mathematical Modelling*, vol. 38, no 9-10, pp. 2414-2421, 2014.
- [40] V. Z. Marmarelis and X. Zhao, "Volterra models and three-layer perceptrons", *IEEE Trans. Neural Networks*, vol. 8, pp. 1421-1432, 1997.

- [41] Y. Rochdi, Identification de systèmes non linéaires blocs, Thèse de Doctorat, Université de Caen, France, 2006.
- [42] 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, SYSID 2009, pp. 1080-1085, 2009.
- [43] M.J. Korenberg, "Recent advances in the identification of nonlinear systems : Minimum-variance approximation by Hammerstein models", Proceedings of the Annual International Conference of the IEEE, vol.13. pp. 2258-2259, 1991.
- [44] D. Westwick and R. Kearney, "Identification of a Hammerstein model of the stretch reflex EMG using separable least squares", In Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE, Chicago, IL. vol. 3. p. 1901-1904, 2000.
- [45] E. Eskinat, S.H. Johnson and W.L. Luyben, "Use of Hammerstein models in identification of nonlinear systems", AIChE Journal, vol. 37, no. 2, p. 255-268, 1991.
- [46] J.C. Stapleton and S.C. Bass, "Adaptive noise cancellation for a class of nonlinear dynamic reference channels", *IEEE Transactions on Circuits and Systems*, vol. 32, no. 2, pp. 143-150, 1985.
- [47] Z.H. Lang, "Controller design oriented model identification method for Hammerstein systems", Automatica, vol. 29, pp. 767-771, 1993.
- [48] A.G. Wills, T.B. Schon, L. Ljung and B. Ninness, "Blind Identification of Wiener Models", *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 5597-5602, 2011.
- [49] A. Kalafatis, N. Arifin, L. Wang, and W. R. Cluett, "A new approach to the identification of pH processes based on the Wiener model", *Chemical Engineering Science*, vol. 50, no. 23, pp. 3693-3701, 1995.
- [50] G. A. Pajunen, "Adaptive control of Wiener type nonlinear systems", Automatica, vol. 28, no. 4, pp. 781-785, 1992.

- [51] A. Srinivasan and P. Lakshmi, "Identification and control of Wiener type process applied to real-time heat exchanger", Asia-Pacific Journal of Chemical Engineering, vol. 3, no.6, pp. 622-629, 2008.
- [52] Y. Fang and T. W. Chow, "Orthogonal wavelet neural networks applying to identification of Wiener model", *IEEE Transactions on Circuits and Systems I : Fundamental Theory and Applications*, vol. 47, no. 4, pp. 591-593, 2000.
- [53] Y. Zhu, "Distillation column identification for control using Wiener model", American Control Conference, Hyatt Regency San Diego, California, USA, vol. 5, pp. 3462-3466, 1999.
- [54] 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.
- [55] J. Vörös, "Parameter identification of Wiener systems with discontinuous nonlinearities", Systems & Control Letters, vol. 44, no. 5, pp. 363-372, 2001.
- [56] S. Rangan, G. Wolodkin and K. Poolla, "New results for Hammerstein system identification", In Decision and Control, 1995, Proceedings of the 34th IEEE Conference on, vol. 1, pp. 697-702, IEEE, December 1995.
- [57] E.W. Bai, "Frequency domain identification of Wiener models", *Automatica*, vol. 39, no. 9, pp. 1521-1530, 2003.
- [58] V. Cerone and D. Regruto, "Parameter bounds evaluation of Wiener models with noninvertible polynomial nonlinearities", *Automatica*, vol. 42, no. 10, pp. 1775-1781, Oct. 2006.
- [59] W. Greblicki and M. Pawlak, "Nonparametric recovering nonlinearities in block oriented systems with the help of Laguerre polynomials", *Control-Theory and Advanced Technology 10. part 1*, pp. 771-791, 1994.
- [60] M.C. Hughes and D.T. Westwick, "Identification of IIR Wiener systems with spline nonlinearities that have variable knots", *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1617-1622, Oct. 2005.

- [61] J. Paduart, L. Lauwers, L. Pintelon and R. Schoukens, "Identification of a Wiener Hammerstein system using the polynomial nonlinear state space approach", *Control* of engineering practice, vol. 20, pp. 1133-1139, 2012.
- [62] L. Sersour, T. Djamah and M. Bettayeb, "Wiener System Identification using Polynomial Non Linear State Space Model", The 3rd International Conference on Control, Engineering and Information Technology : CEIT'2015, Tlemcen, Algeria, May, 2015.
- [63] L. Sersour, T. Djamah and M. Bettayeb, "Wiener system Identification using Self-Adaptive Velocity Particle Swarm Optimization", *International Conference on Au*tomatic control, Telecommunications and Signals : ICATS'2015, Annaba, Algeria, November, 2015.
- [64] S. Chen and S. A. Billings, "Recursive prediction error parameter estimator for nonlinear models", *International Journal of Control*, vol. 49, no. 2, pp. 569 - 594, 1989.
- [65] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens and R. Pintelon, "Identification of nonlinear systems using polynomial nonlinear state space models", *Automatica*, vol. 46, no. 4, pp. 647-656, 2010.
- [66] A. Haryanto and K. S. Hong, "Maximum likelihood identification of Wiener-Hammerstein models. Mechanical Systems and Signal Processing", *Mechanical Sys*tems and Signal Processing, vol. 41, no. 1-2, pp. 54-70, 2013.
- [67] F. Ding and T. Chen, "Identification of Hammerstein nonlinear ARMAX systems", Automatica, vol. 41, no. 9, pp. 1479-1489, 2005.
- [68] 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.
- [69] F. Ding, Y. Shi and T. Chen, "Auxiliary model-based least-squares identification methods for Hammerstein output-error systems", Systems & Control Letters, vol. 56, no 5, pp. 373-380, 2007.

- [70] H. Chen and F. Ding, "Hierarchical least squares identification for Hammerstein nonlinear controlled autoregressive systems", *Circuits, Systems, and Signal Processing*, vol. 34, no 1, pp. 61-75, 2015.
- [71] F. Wang, K. Xing, X. Xu and H. Liu, "An identification approach of Hammerstein model", In : Control and Decision Conference (CCDC), Chinese. IEEE, pp. 607-612, 2010.
- [72] 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", In : Unifying Electrical Engineering and Electronics Engineering. Springer, New York, NY, pp. 1961-1968, 2014.
- [73] L. Zhou, X. Li, H. Xu and P. Zhu, "Levenberg-Marquardt iterative algorithm for Hammerstein nonlinear systems", In Control, Automation and Information Sciences (ICCAIS), 2015 International Conference on, IEEE, pp. 280-284, 2015.
- [74] Y. Han and R.A. Callafon, "Identification of a Wiener system via semidefinite programming", In Proceedings of the 16th IFAC Symposium on System Identification, Brussel, Belgium, pp. 1109-1113, 2012.
- [75] I. Aljamaan, A.B. bshait, and D. Westwick, "Separable least squares identification of wiener box-jenkins models", In Proceedings of the 18th IFAC World Congress, pp. 4434-4439, 2011.
- [76] L. Vanbeylen, R. Pintelon and J. Schoukens, "Blind maximum-likelihood identification of Wiener systems", *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3017-3029, 2009.
- [77] A. Hagenblad, L. Ljung and A. Wills, "Maximum likelihood identification of Wiener models", *Automatica*, vol. 44, no. 11, pp. 2697-2705, 2008.
- [78] D. Wang and F. Ding, "Least squares based and gradient based iterative identification for wiener nonlinear systems", *Signal Processing*, vol. 91, no. 5, pp. 1182-1189, 2011.
- [79] S. Bedoui, M. Ltaief and K. Abderrahim, "Gradient based identification for discrete time delay systems", In 9th Asian Control Conference, 2013.

- [80] S. Bedoui, M. Ltaief and K. Abderrahim, "Hierarchical gradient based identification of discrete-time delay systems", In 52th IEEE Conference on Decision and Control, Italy, 2013.
- [81] L. Zhou, X. Li, and F. Pan, "Least squares based iterative identification algorithm for Wiener nonlinear systems", *Journal of Applied Mathematics*, doi:10.1155/2013/565841, 2013.
- [82] M. Liu, Y. Xiao and R. Ding, "Iterative identification algorithm for Wiener nonlinear systems using the Newton method", *Applied Mathematical Modelling*, vol. 37, no. 9, pp. 6584-6591, 2013.
- [83] M.S. Revathy and N.N. Singh, "An efficient way of solving inverse problem using nonlinear Wiener filter and its application to pattern recognition", *Procedia engineering*, vol. 38, pp. 708-717, 2012.
- [84] I.A. Krishtal and K.A. Okoudjou, "Invertibility of the Gabor frame operator on the Wiener amalgamspace", *Journal of Approximation Theory*, vol. 153, no. 2, pp. 212-224, 2008.
- [85] K.B. Oldham and J. Spanier, "The fractional calculus theory and applications of differentiation and integration to arbitrary order", *Elsevier*, vol. 111, 1974.
- [86] S.G. Samko, A.A. Kilbas and O.I. Marichev, "Fractional integrals and derivatives : theory and applications", *Gordon and Breach*, 1993.
- [87] K.S. Miller and B. Ross, "An introduction to the fractional calculus and fractional differential equations", Wiley, New York, 1993.
- [88] M. Ichise, Y. Nagayanagi and T. Kojima, "An analog simulation of non integer order transfer functions for analysis of electrode processes", *Journal of Electroanalytical Chemistry*, vol. 33, no. 2, pp. 253-265, 1971.
- [89] J. lin, T. Poinot, J. C. Trigeassou, H. Kabbaj and J. Faucher, "Modélisation et identification d'ordre non entier d'une machine asynchrone", In : Conférence Internationale Francophone d'Automatique, 2000.

- [90] J. L. Battaglia, O. Cois, L. Puigsegur and A. Oustaloup, "Solving an inverse heat conduction problem using a noninteger identified model", *International Journal of Heat and Mass Transfer*, vol. 44, no. 14, pp. 2671-2680, 2001.
- [91] O. Cois, A. Oustaloup, E. Battaglia and J.L. Battaglia, "Non integer model from modal decomposition for time domain identification", 41st IEEE CDC'2002 Tutorial Workshop 2, Las Vegas, USA, 2002.
- [92] D. Matignon, Représentation en variables d'état de modèles de guides d'ondes avec dérivation fractionnaire, Thèse de Doctorat, Universitè Paris XI, France, 1994.
- [93] D. Matignon, B. d'Andrèa Novel, P. Depalle and A. Oustaloup, "Viscothermal Losses in Wind Instruments : A NonInteger Model", *Academic Verlag*, Berlin, 1994.
- [94] A. Hanyga, "Internal variable models of viscoelasticity with fractional relaxation laws", Proceedings of Design Engineering Technical Conference, Mechanical Vibration and Noise, 48395, American Society of Mechanical Engineers, Chicago, USA, 2003.
- [95] D. Valerio and J. Sa da Costa, "Non-integer order control of a flexible robot", Proceedings of the IFAC Workshop on Fractional Differentiation and its Applications, FDA'04, Bordeaux, France, 2004.
- [96] S. Manabe, "The non-integer integral and its application to control systems", Japanese Institute of Electrical Engineers Journal, vol. 80, no. 86, pp. 589-597, 1960.
- [97] A., Oustaloup, Systèmes asservis linéaires d'ordre fractionnaire, Masson, Paris, 1983.
- [98] M. Axtell and E. M. Bise, "Fractional calculus applications in control systems", Proceedings of the IEEE 1990 National Aerospace and Electronics Conference, New York, USA, pp. 536-566, 1990.
- [99] A. Oustaloup, La Dérivation non entière : Théorie, synthèse et applications, Hermès, Paris, 1995.
- [100] I. Podlubny, Fractional Differential Equations, Academic Press, San Diego, 1999.
- [101] S. Hammouche, Identification d'un modèle fractionnaire à l'aide des réseaux de neurones, Mémoire de Magistère. Unniversity Mouloud Mammerie Tizi ouzou, 2012.

- [102] D. Mozyrska, E. Girejko and M. Wyrwas, "Comparison of h-Difference Fractional Operators", In W. Mitkowski, J. Kacprzyk, and J. Baranowski (Eds.) Lecture Notes in Electrical Engineering, Springer, Cham, Heidelberg, New York, Dordrecht, London. pp. 251-256, 2013.
- [103] A. Dzielinski and D. Sierociuk, "Adaptive feedback control of fractional order discrete state-space systems", In Proceedings of the 2005 International Conference on Computational Intelligence for Modelling Control and Automation, Vienna, Austria, November 28-30, vol. 1, pp. 804-809, 2005.
- [104] T. Djamah, Identification de systèmes par des modèles d'ordre fractionnaire, Thèse de Doctorat, Université Mouloud Mammerie Tizi ouzou, 2010.
- [105] A. Dzielinski and D. Sierociuk, "Stability of discrete fractional order state-space systems", Journal of Vibration and Control, vol. 14, no. 9-10, pp. 1543-1556, 2008.
- [106] D. Matignon, "Stability results for fractional differential equations with applications to control processing", In Computational engineering in systems applications IMACS, IEEE-Syst. Man Cyber., Lille, France, 1996.
- [107] J. Sabatier, M. Moze and C. Farges, "On stability of fractional order systems", In : Third IFAC workshop on fractional differentiation and its applications FDA'08, 2008.
- [108] A. Djouambi, Contribution à la Commande CRONE, Thèse de Doctorat, Université de Constantine. 2008.
- [109] A. Charef and H.H. Sun, "Time domain analysis of fractal system", In Engeneering in Medicine and Biology society, Proceedings of the 12th Annual international Conference of the IEEE, no. 18, pp. 597-621, 1990.
- [110] J. C. Trigeassou and al. "Modeling and identification of a non integer order system", Control Conference (ECC), 1999 European. IEEE, 1999.
- [111] G. E. Carlson and C. Halijak, "Approximation of Fractional Capacitors by a Regular Newton Process", *IEEE Transactions on Circuits and Systems*, vol. 11, no. 2, pp. 210-213, 1964.

- [112] K. Matsuda and H. Fujii, "Optimised Wave Absorbing Control : Analytical and Experimental Results", Journal of Guidance Control and Dynamics, vol. 16, no. 6, pp. 1146-1153, 1993.
- [113] C., Lubich, "Discretized fractional calculus", SIAM Journal of Mathematical Analysis, vol. 17, no. 3, pp. 704-719, 1986.
- [114] Y.Q. Chen and K.L. Moore, "Discretization Schemes for Fractional-Order Differentiators and Integrators", *IEEE Transactions on Circuits and Systems-I : Fundamental Theory and Applications*, vol. 49, no. 3, pp. 363-367, 2002.
- [115] B.M. Vinagre, I. Podlubny, A. Hernandez and V. Feliu, "Some approximations of fractional order operators used in control theory and applications", *Fractional Cal*culus & applied Analysis, vol. 3, no. 3, pp. 231-248, 2000.
- [116] L. Sersour, T. Djamah and M. Bettayeb, "Identification of Wiener Fractional model using Self-Adaptative Velocity Particle Swarm Optimization", *The 7th International Conference on Modelling, Identification and Control : ICMIC'2015*, Sousse, Tunisia, December, 2015.
- [117] L. Sersour, T. Djamah and M. Bettayeb, "Fractional Wiener System Identification using Levenberg Marquard algorithm", *Graduat Student Recherch Conference*, UAEGSRC 2016, Al Ain, United Arab Emirates, April, 2016.
- [118] L. Sersour, T. Djamah and M. Bettayeb, "Nonlinear system identification of fractional Wiener models", *Nonlinear Dynamics*, vol. 92, no. 4, pp. 1-13, 2018.
- [119] 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.
- [120] A. Maachou , R. Malti, P. Melchior, J-L. Battaglia, A. Oustaloup and B. Hay, "Nonlinear thermal system identification using fractional Volterra series", *Control Engineering Practice*, vol. 29, pp. 50-60, 2014.
- [121] F. Benoit-Marand, L. Signac, T. Poinot and J. C. Trigeassou, "Identification of non linear fractional systems using continuous time neural networks", *IFAC Proc*, vol. 39, no. 11, pp. 402-407, 2006.

- [122] Y. Zhao and Y. Chen, "Complete parametric identification of fractional order Hammerstein systems", International Conference on Fractional Differentiation and Its Applications (ICFDA), pp. 1-6, 2014.
- [123] D. V. Ivanov, "Identification discrete fractional order Hammerstein systems", Control and Communications (SIBCON), 2015 International Siberian Conference on. IEEE, 2015.
- [124] Z. Liao, Z. Zhu, S. Liang, C. Peng and Y. Wang, "Subspace identification for fractional order Hammerstein systems based on instrumental variables", *International Journal of Control, Automation and Systems*, vol. 10, no. 5, pp. 947-953, 2012.
- [125] M.R. Rahmani and M. Farrokhi, "Identification of neuro-fractional Hammerstein systems : a hybrid frequency-/time-domain approach", *Soft Computing*, pp. 1-10, 2017.
- [126] R. Stanislawski, KJ. Latawiec, M. Galek and M. Lukaniszyn, "Modeling and identification of a fractional-order discrete-time SISO Laguerre-Wiener system", Proceedings of the 19th Int Conf on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, pp. 165-168, 2014.
- [127] N. Kianpour and M. Asad, "A novel identification method for fractional-order Wiener systems with PRBS input", 4th Int Conf on Control, Instrumentation, and Automation (ICCIA), Qazvin, Iran, 2016.
- [128] L. Vanbeylen, "A fractional approch to identify Wiener-Hammerstein systems", Automatica, vol. 50, no. 3, pp. 903-909, 2014.
- [129] K. Levenberg, "A method for the solution of certain problems in least squares", Quart. Appl. Math, vol. 2, pp. 164-168, 1944.
- [130] 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, 1963.
- [131] T. Djamah, S. Djennoune and M. Bettayeb, "Generalized Levenberg- Marquardt algorithm for real dynamic system identification", *Colloque International sur l'Op-*

timisation et les Systèmes d'Information, COSI'08, Tizi-ouzou Algerie, 08-10 Juin 2008.

- [132] J. Dréo, A. Pétrowski, P. Siarry and E. Taillard, "Methaheuristics for hard optimisation", Spring-verlag, Berlin Heidelberg, 2006.
- [133] J. Dreo, A. Petrowski, P. Siarry and E. Taillard, Metaheuristiques for difficult optimization, Eyrolles, Paris, 2003.
- [134] S. Kirkpatrick, D.C. Gelatt and M.P. Vechhil, "Optimization by simulated annealing", *Science*, 220, pp. 671-80, May 1983.
- [135] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1996.
- [136] A. Colorni, M. Dorigo and V. Maniezzo, "Distributed optimization by ant colonies", Proceeding of the first Europeean Conference on Artificial Life, vol. 142, pp. 134-142, 1992.
- [137] R.C. Eberhart and J. Kennedy, "A new Optimizer using particle swarm theory", In : Proceedings of 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43, 1995.
- [138] J. Kennedy, "Dynamic probabilistic particle swarms", GECCO'05, June 2005, Washington, DC, USA, pp. 201-207, 2005.
- [139] Y. Shi and R.C. Eberhart, "A modified particle swarm optimizer", In : Proceeding of IEEE Conference on Evolutionary Computation, Anchorage, AK, pp. 69-73, 1998.
- [140] H. Lu andW. Chen, "Self-adaptive velocity particle swarm optimization for solving constrained optimization problems", *Journal of Global Optimization*, vol. 41, no. 3, pp 427-445, 2008.
- [141] L. Zhou, X. Li, H. Xu and P. Zhu, "Gradient-Based Iterative Identification for Wiener Nonlinear Dynamic Systems with Moving Average Noises", *Algorithms*, vol. 8, pp. 712-722, 2015.