

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERRI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D' INFORMATIQUE  
DEPARTEMENT D'AUTOMATIQUE

## Mémoire de Fin d'Etudes de MASTER PROFESSIONNEL

Domaine : Sciences et Technologies

Filière : Automatique

Spécialité : Automatique industrielle

*Présenté par*

**Lina AMEZIANE**

Thème

## Identification des systèmes de Wiener basée sur la représentation de l'espace d'état non linéaire polynomiale.

*Mémoire soutenu publiquement le 30/09/2024 devant le jury composé de :*

**Mr Boualem SALHI**

MCB, UMMTO, Président

**Mme Lamia SERSOUR**

MCB, UMMTO, Encadrant

**Mme Ghania IDIRI**

MCB, UMMTO, Examineur

**M Farida ALMANSBA**

MMA, UMMTO, Examineur

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D' INFORMATIQUE  
DEPARTEMENT D'AUTOMATIQUE

## Mémoire de Fin d'Etudes de MASTER ACADÉMIQUE

Domaine : Sciences et Technologies

Filière : Automatique

Spécialité : Automatique et Systèmes

*Présenté par*

**Rosa DIBOUN**

Thème

# Identification des systèmes de Wiener basée sur la représentation de l'espace d'état non linéaire polynomiale.

*Mémoire soutenu publiquement le 30/09/2024 devant le jury composé de :*

**Mr Boualem SALHI**  
MCB, UMMTO, Président

**Mme Lamia SERSOUR**  
MCB, UMMTO, Encadrant

**Mme Ghania IDIRI**  
MCB, UMMTO, Examineur

**Mme Farida ALMANSBA**  
MMA, UMMTO, Examineur

# Dédicaces

J'ai le grand honneur de dédier ce modeste travail : À mes chers parents, en témoignage de ma profonde gratitude pour votre amour inconditionnel, votre soutien constant et vos sacrifices inestimables. Vous avez toujours cru en moi, même dans les moments de doute, et votre présence a été une source de force et d'inspiration tout au long de ce parcours. Je vous en serai éternellement reconnaissante.

À mes sœurs, Kahina et Djouher.

À mon frère, Mustapha.

À tous mes amis.

À tous ceux qui me sont chers.

*Rosa*

# Remerciements

Au terme de ce mémoire, je tiens à exprimer ma profonde reconnaissance aux personnes qui, de près ou de loin, ont rendu ce travail possible.

Je tiens tout d'abord à exprimer ma gratitude à Madame SERSOUR, pour avoir dirigé cette recherche et contribué à son aboutissement.

Je tiens à remercier tous les membres du jury qui ont accepté de consacrer le temps qu'il faut pour évaluer ce modeste travail. Leur participation à ce jury me fait honneur.

Ma profonde gratitude s'adresse à tous mes enseignants qui m'ont accompagné tout au long de mon parcours académique. Grâce à leur savoir, leur dévouement et leur engagement, j'ai pu acquérir les compétences et connaissances nécessaires pour mener à bien ce travail.

Mes plus vifs remerciements s'adressent au chef de département, Monsieur TOUAT pour son appui constant et son engagement envers notre réussite. Sa bienveillance et son professionnalisme ont été des atouts majeurs tout au long de mon parcours académique.

Enfin, ce travail n'aurait pu aboutir sans le soutien continu de ma famille et mes amis, je les en remercie de tout mon cœur.

*Rosa*

Avant de présenter ce travail, je commence par remercier le bon Dieu de m'avoir donné la santé, le courage et la volonté pour mener à bien ce modeste projet. Je souhaite également exprimer ma profonde gratitude à mon encadrante, Madame L. Sersour, pour ses conseils avisés et son accompagnement tout au long de notre projet de fin d'étude, ainsi qu'aux membres du jury qui me font l'honneur d'examiner notre travail.

Ma reconnaissance la plus profonde va à ma famille, qui a été un pilier essentiel tout au long de ce parcours. Je remercie de tout cœur mes parents, Rachid et Samia, pour leur amour inconditionnel, leurs sacrifices et leur soutien indéfectible qui m'ont permis d'avancer avec confiance et détermination.

Mon cher frère Lyes et ma chère sœur Melissa ont également été d'un soutien précieux tout au long de cette période mais aussi mes chères nièces Emna et Assya.

Un immense merci à mes ami(e)s qui, de près ou de loin, m'ont encouragé(e) et accompagné(e) avec joie et bienveillance durant toute cette période.

*Lina*

# Table des matières

Notations	vi
Abréviations	vii
Liste des figures	viii
Liste des tableaux	ix
Introduction Générale	1
<b>1 Systèmes non linéaires</b>	<b>4</b>
1.1 Introduction . . . . .	4
1.2 Classe des systèmes non linéaires . . . . .	4
1.2.1 Modèle non linéaire à réponse impulsionnelle finie . . . . .	5
1.2.2 Modèle autoRégressif à variable eXogène non linéaire . . . . .	6
1.2.3 Modèle AutoRégressif à moyenne ajustée et variable eXogène Non linéaire . . . . .	6
1.2.4 Modèle Box-Jenkins non linéaire . . . . .	7
1.2.5 Modèle Séries de Volterra . . . . .	8
1.2.6 Modèles blocs orientés . . . . .	9
1.3 Représentation d'espace d'état non linéaire polynomial . . . . .	12
1.3.1 Forme générale . . . . .	13
1.3.2 Description du modèle de Wiener sous forme PNLSS . . . . .	14
1.4 Conclusion . . . . .	19
<b>2 Identification des systèmes non linéaires</b>	<b>20</b>
2.1 Introduction . . . . .	20

2.2	Identification . . . . .	20
2.2.1	Définition de l'identification . . . . .	20
2.2.2	Étapes de l'identification . . . . .	21
2.2.3	Méthodes de l'identification . . . . .	22
2.3	État de l'art sur l'identification des systèmes non linéaires . . . . .	33
2.4	Identification des systèmes de Wiener PNLSS . . . . .	35
2.4.1	Calcul des fonctions de sensibilité . . . . .	35
2.5	Conclusion . . . . .	36
<b>3</b>	<b>Résultats de simulation en utilisant l'algorithme Levenberg-Marquardt</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Identification par l'algorithme LM . . . . .	37
3.2.1	Premier exemple : Ordre linéaire = 2, non linéaire = 2 . . . . .	37
3.2.2	Deuxième exemple : Ordre linéaire = 2, non linéaire = 3 . . . . .	42
3.3	Conclusion . . . . .	46
	<b>Conclusion Générale</b>	<b>47</b>
	<b>Bibliographie</b>	<b>49</b>

# Notations

- $\mathbf{A}(\mathbf{q})$  : Polynôme d'ordre  $n_A$ .
- $\mathbf{B}(\mathbf{q})$  : Polynôme d'ordre  $n_B$ .
- $\mathbf{e}(\mathbf{k})$  : Vecteur de bruit.
- $\mathbf{f}(\cdot)$  : Fonction non linéaire.
- $\mathbf{k}$  : Variable de temps discret.
- $\mathbf{M}(\mathbf{q})$  : Polynôme d'ordre  $n_M$ .
- $\mathbf{N}(\mathbf{q})$  : Polynôme d'ordre  $n_N$ .
- $r$  : Ordre du polynôme non linéaire.
- $\mathbf{u}(\mathbf{k})$  : Entrée en temps discret.
- $\mathbf{v}(\mathbf{k})$  : Vecteur de sortie du bloc linéaire en temps discret.
- $\mathbf{x}(\mathbf{k})$  : Vecteur d'état du modèle d'espace d'état en temps discret.
- $\mathbf{y}(\mathbf{k})$  : Vecteur de sortie en temps discret.
- $\boldsymbol{\theta}$  : Vecteur des paramètres.
- $\lambda$  : Paramètre de contrôle de la direction de recherche de l'optimisation.
- $\boldsymbol{\phi}(\mathbf{k})$  : Vecteur de régression en temps discret.
- $\hat{\boldsymbol{\theta}}$  : Vecteur des paramètres estimés.

# Abbreviations

**ARMAX** AutoRegressive Moving Average model with eXogenous Input.

**BJ** Box-Jenkins.

**EDO** Equation aux Dérivées Ordinaires.

**EDP** Equation aux Dérivées Partielles.

**FIR** Finite Impulse Response.

**IIR** Infinite Impulse Response.

**LM** Levenberg Marquardt.

**LS** Least Squares.

**LTI** Linear Time Invariant.

**NARMAX** Nonlinear AutoRegressive Moving Average model with eXogenous Input.

**NARX** Nonlinear AutoRegressive with eXogenous input.

**NBJ** Nonlinear Box-Jenkins.

**NFIR** Nonlinear Finite Impulse Response.

**PNLSS** Polynomial Nonlinear State Space.

**PSO** Particle Swarm Optimization.

**RLS** Recursive Least Square.

**RVV** Recherche locale et de Voisinage Variable.

**SAVPSO** Self Adaptative Velocity Particle Swarm Optimization.

**SISO** Single Input/Single Output.

# Table des figures

1.1	Modèle de Hammerstein . . . . .	9
1.2	Modèle de Wiener . . . . .	10
1.3	Modèle de Wiener-Hammerstein . . . . .	11
1.4	Modèle de Hammerstein-Wiener . . . . .	11
1.5	Modèle de Wiener PNLSS . . . . .	14
2.1	Étapes de l'identification . . . . .	21
2.2	Méthode de Broïda . . . . .	24
2.3	Structure erreur d'équation . . . . .	25
2.4	Principe de la méthode du modèle . . . . .	27
2.5	Principe de l'algorithme de Levenberg-Marquardt . . . . .	29
2.6	Principe de fonctionnement du PSO . . . . .	32
3.1	Erreur de prédiction (SNR=0 dB) . . . . .	39
3.2	Sortie estimée et sortie mesurée (SNR=0 dB) . . . . .	39
3.3	Erreur de prédiction (SNR=34 dB) . . . . .	40
3.4	Sortie estimée et sortie mesurée (SNR=34 dB) . . . . .	40
3.5	Erreur de prédiction (SNR=26 dB) . . . . .	41
3.6	Sortie estimée et sortie mesurée (SNR=26 dB) . . . . .	41
3.7	Erreur de prédiction (SNR=0 dB) . . . . .	43
3.8	Sortie estimée et sortie mesurée (SNR=0 dB) . . . . .	43
3.9	Erreur de prédiction (SNR=34 dB) . . . . .	44
3.10	Sortie estimée et sortie mesurée (SNR=34 dB) . . . . .	44
3.11	Erreur de prédiction (SNR=26 dB) . . . . .	45
3.12	Sortie estimée et sortie mesurée (SNR=26 dB) . . . . .	45

# Liste des tableaux

2.1	Tableau de Strejc . . . . .	23
3.1	Résultats de la simulation pour différents rapports signal sur bruit. . . .	42
3.2	Résultats de la simulation pour différents rapports signal sur bruit. . . .	46

# Introduction générale

Le problème de modélisation et d'identification des systèmes suscite un grand intérêt et ce depuis plusieurs décennies [1]. L'activité de recherche menée autour de ce problème joue un rôle essentiel et fondamental dans de nombreux domaines, allant de l'automatique à l'ingénierie des systèmes complexes. En effet la capacité de représenter mathématiquement le comportement dynamique de systèmes réels à partir de données expérimentales est essentielle pour comprendre, contrôler et optimiser ces systèmes.

Si l'identification des systèmes linéaires est maintenant arrivée à maturité, celle des systèmes non linéaires reste à ce jour un thème de recherche loin d'être tari.

Cependant le développement de méthodes efficaces pour l'identification des systèmes non linéaires permet non seulement de mieux appréhender les interactions entre les différentes variables d'un système, mais aussi de proposer des solutions fiables et efficaces pour leur gestion. Ainsi, l'identification précise de ces systèmes pose encore des défis importants.

Les méthodes d'identification classiques, principalement conçues pour les systèmes linéaires, ne peuvent pas être appliquées directement aux systèmes non linéaires. Là où, aucune forme générale n'est disponible. Toutefois, les études des classes des systèmes non linéaires ne concernent que des structures de modèles précises, tels que les modèles NARMAX, séries de Volterra ou les modèles blocs orientés [1]... En effet, cette dernière classe de systèmes non linéaires se caractérise par la construction de modèles à partir de blocs simples, afin de trouver des structures suffisamment flexibles pour couvrir de nombreux systèmes non linéaires réels. En effet, ils sont définis par la séparation du comportement statique non linéaire et du comportement linéaire invariant dans le temps (LTI) en différents blocs.

Les modèles blocs se prêtent généralement mieux à l'identification moyennant une adaptation des méthodes conçues à l'origine pour les systèmes linéaires. Parmi les modèles de ce type qui ont le plus suscité l'intérêt, on peut citer le modèle de : Hammerstein, Wiener, Wiener-Hammerstein, ...

Dans ce mémoire de fin d'étude, nous nous baserons sur la description et l'identification des systèmes non linéaires, de type Wiener. Mettant l'accent sur des algorithmes de minimisation non linéaire, en se basant sur l'algorithme de Levenberg-Marquardt (LM). Ce dernier est particulièrement adapté pour résoudre des problèmes où les données d'entrée et de sortie sont bruitées ou imprécises [1].

Pour présenter au mieux notre travail, ce mémoire est structuré en trois chapitres, dont le contenu est brièvement introduit comme suit :

#### Chapitre I : Systèmes non linéaires.

Dans cette partie, nous verrons les différentes classes des systèmes non linéaires, et mettrons en lumière la complexité par rapport aux systèmes linéaires, en soulignant l'importance d'une modélisation précise pour concevoir des stratégies de contrôle efficaces. Ce chapitre explore les caractéristiques des systèmes non linéaires, leurs applications dans divers domaines et l'importance de leur compréhension pour les avancées technologiques et scientifiques. L'objectif est de montrer comment une meilleure compréhension de ces systèmes pourrait conduire à des stratégies de contrôle plus efficaces et à des applications plus performantes et plus précises.

#### Chapitre II : Identification des systèmes non linéaires.

Ce chapitre discutera l'identification des systèmes non linéaires, qui est un domaine crucial en ingénierie et en sciences appliquées. L'identification des systèmes est le processus par lequel un modèle mathématique est dérivé à partir de données expérimentales pour représenter le comportement dynamique d'un système. Ce chapitre commencera par une introduction à l'importance de cette pratique, surtout lorsque les systèmes en question présentent des non linéarités complexes, rendant difficile l'utilisation de modèles exacts basés sur les principes fondamentaux. Les différentes étapes nécessaires pour identifier un système seront ensuite abordées. Plusieurs méthodes d'identification seront décrites, chacune adaptée à des contextes et des types de systèmes spécifiques. Parmi elles, les méthodes paramétriques et non paramétriques qui jouent un rôle central. En effet, les premières nécessitant une structure de modèle préexistante, tandis que les secondes tentent de capturer directement la relation entrée sortie sans présupposer de forme de modèle. Le chapitre se basera également sur l'identification des systèmes de Wiener avec des non-linéarités, en détaillant l'utilisation de techniques convenables comme l'algorithme de LM, pour ajuster les paramètres du modèle et minimiser l'erreur de prédiction.

Chapitre III : Résultats de simulation en utilisant l'algorithme de Levenberg-Marquardt. Nous traiterons, dans cette partie, des résultats de simulations réalisés en utilisant l'algorithme de LM pour l'identification des systèmes de Wiener représentés par les modèles de la représentation d'état non linéaires polynomiale (PNLSS).

Les simulations seront effectuées, pour différents ordres de linéarité et de non-linéarité, avec une entrée aléatoire de moyenne nulle et de variance unitaire, en présence et en absence de bruit. Les résultats obtenus seront également discutés afin de vérifier l'efficacité de la méthode.

# Chapitre 1

## Systèmes non linéaires

### 1.1 Introduction

En automatique, les systèmes linéaires ont longtemps été privilégiés pour leur comportement prévisible et la robustesse de leurs outils d'analyse, mais la réalité des systèmes physiques est souvent non linéaire, nécessitant des modèles et des stratégies de contrôle plus complexes. Les systèmes non linéaires se caractérisent par des relations de entrée/sortie non proportionnelles, engendrant des comportements dynamiques variés, bien que complexes, ces systèmes sont courants dans des domaines tels que l'aérospatiale et la robotique, où une modélisation précise et un contrôle efficace sont essentiels. Approfondir l'étude des systèmes non linéaires permettrait de concevoir des stratégies de contrôle plus efficaces, conduisant à des applications plus performantes et sûres. Cette recherche est cruciale pour les avancées technologiques et scientifiques, ouvrant ainsi la voie à de nouvelles possibilités et découvertes en automatique.

### 1.2 Classe des systèmes non linéaires

Un système est considéré comme étant non linéaire s'il ne respecte pas le principe de superposition, il ne peut pas être décrit par des équations différentielles linéaires à coefficients constants (au sens physique), ce qui le rend plus complexe, ces systèmes sont souvent rencontrés dans divers domaines tels que l'ingénierie, la physique, la biologie, etc. Il existe de nombreuses classes de systèmes non linéaires, dont certaines ont une forme générale, particulièrement dans le cas discret, comme montré ci-dessous :

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

- $f(\cdot)$  est une fonction qui peut être linéaire pour les systèmes linéaires ou non linéaire pour les systèmes non linéaires.
- $\phi(k)$  est le vecteur de régression.
- $\theta$  est le vecteur des paramètres du modèle.
- $e(k)$  est le bruit.

Ainsi, en fonction de la forme du vecteur  $(\phi)$ , différentes structures de modèles non linéaires peuvent être définies telles que :

### 1.2.1 Modèle non linéaire à réponse impulsionnelle finie

Le modèle de réponse impulsionnelle finie non linéaire (Nonlinear Finite Impulse Response) (NFIR) est un outil puissant pour prédire la réponse des systèmes non linéaires à des signaux d'entrée fini, il est utilisé dans de multiples domaines tels que le traitement du signal, la modélisation de systèmes dynamiques, la modélisation des systèmes biologiques, etc [2] [3]. Le terme «non linéaire» indique que le modèle peut représenter des systèmes avec des comportements non linéaires, et le terme «Réponse impulsionnelle finie» montre comment un système réagit à une entrée impulsionnelle, en générant une réponse de durée limitée, basée sur un nombre fini d'entrées passées. Le modèle (NFIR) est décrit par [4] :

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

avec  $B(q)$  est un polynôme d'ordre  $n_B$ , défini par :

$$B(q) = b_1q^{-1} + \dots + b_{n_B}q^{-n_B}.$$

Par conséquent,

$$y(k) = b_1q^{-1}u(k) + b_2q^{-2}u(k) + \dots + b_{n_B}q^{-n_B}u(k) + e(k) \quad (1.3)$$

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

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

## 1.2.2 Modèle autoRégressif à variable eXogène non linéaire

Les modèle autoRégressif à variable eXogène non linéaire (Nonlinear AutoRegressive with eXogenous input) (NARX) incluent des termes autorégressifs et des termes dépendants d'entrées exogènes, ce qui les rend appropriés pour modéliser des systèmes dynamiques non linéaires, Ce modèle est utilisé pour estimer la réponse d'un système non linéaire à un signal d'entrée fini, et il trouve des applications dans divers domaines tels que la prédiction de séries temporelles et la modélisation de systèmes dynamiques complexes [5].

Ce modèle est donné comme suit :

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

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

Ainsi, la sortie du modèle est représentée comme suit :

$$y(k) = b_1u(k-1) + \dots + b_{n_B}u(k-n_B) - a_1y(k-1) - \dots - a_{n_A}y(k-n_A) + e(k) \quad (1.6)$$

Ce modèle peut également être déduit en remplaçant la relation linéaire dans l'équation (1.6) par une fonction non linéaire, comme suit :

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

$f$  est une fonction non linéaire.

## 1.2.3 Modèle AutoRégressif à moyenne ajustée et variable eXogène Non linéaire

Le modèle autoRégressif à moyenne ajustée et variable eXogène non linéaire (Nonlinear AutoRegressive Moving Average with eXogenous input) (NARMAX) est une structure de modèle qui étend le concept NARX, ce type de modèle est utilisé en identification de systèmes non linéaires, Il est utilisé pour modéliser les relations entre les variables d'entrée et de sortie d'un système en prenant en compte à la fois les retards dans les entrées et les sorties, ainsi que les interactions non linéaires entre ces variables, il est largement utilisé dans les domaines de la commande de processus, de la prévision et de l'identification des systèmes dynamiques non Linéaires [2].

Dans le cas linéaire, la sortie du modèle ARMAX est représentée par :

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

$M(q)$  est un polynôme d'ordre  $n_M$

Ce type de modèle peut être défini par l'équation suivante dans le cas non-linéaire [6] :

$$\begin{aligned} y(k) = f(y(k-1), y(k-2), \dots, y(k-n_A), \\ u(k-1), u(k-2), \dots, u(k-n_B), \\ e(k-1), e(k-2), \dots, e(k-n_M), ) + e(k) \end{aligned} \quad (1.9)$$

$f$  est une fonction non-linéaire.

### 1.2.4 Modèle Box-Jenkins non linéaire

Le modèle Box-Jenkins non linéaire (Nonlinear Box-Jenkins) (NBJ) est une extension du modèle classique de Box-Jenkins (BJ) utilisé en statistiques et en modélisation des séries temporelles, il permet de modéliser des systèmes non linéaires en incluant des composantes non linéaires dans la représentation du système. Cela permet une meilleure approximation des relations entre les entrées et les sorties des systèmes complexes et non linéaires. Le modèle NBJ est utilisé dans divers domaines tels que la commande de processus, la prévision et l'identification des systèmes dynamiques [7].

Le modèle BJ fournit une autre forme en multipliant le bruit par  $\frac{M(p)}{N(p)}$  [4].

Le modèle BJ est défini comme suit :

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

avec  $B$  ,  $H$ ,  $M$  et  $N$  des polynômes d'ordre  $n_B$ ,  $n_H$ ,  $n_M$  et  $n_N$  respectivement.

**Exemple :**

$$\begin{cases} B(q) = b_1q^{-1} + \dots + b_{n_B}q^{-n_B} \\ H(q) = 1 + h_1q^{-1} + \dots + h_{n_H}q^{-n_H} \\ M(q) = 1 + m_1q^{-1} + \dots + m_{n_M}q^{-n_M} \\ N(q) = 1 + n_1q^{-1} + \dots + n_{n_N}q^{-n_N} \end{cases} \quad (1.11)$$

Le modèle NBJ peut être déduit en remplaçant la relation linéaire dans l'équation (1.10) par une fonction non linéaire, comme suit [6] :

L'équation est donnée par :

$$\begin{aligned} y(k) = & f(u(k-1), \dots, u(k-n_B-n_N), \dots, u(k-n_B), \\ & y(k-1), \dots, y(k-n_N), \dots, y(k-n_H), \dots, y(k-n_H-n_N), \\ & e(k-1), \dots, e(k-n_M), \dots, e(k-n_H), \dots, e(k-n_H-n_M), \theta) \end{aligned} \quad (1.12)$$

### 1.2.5 Modèle Séries de Volterra

Les séries de Volterra est une méthode mathématique utilisée pour modéliser des systèmes dynamiques non linéaires, cette approche permet de représenter des systèmes non linéaires en tenant compte des interactions entre différentes composantes du système, contrairement aux séries de Taylor, qui décomposent une fonction en termes de polynômes, la série de Volterra considère les interactions non linéaires de manière plus générale, elle offre une approche puissante pour capturer les non-linéarités dans les systèmes et améliorer la précision des modèles. Les séries de Volterra sont utilisées dans divers domaines tels que la modélisation de hautparleurs électrodynamiques, la commande de classes d'équations aux dérivées partielles (EDP) et d'équations aux dérivées ordinaires (EDO) non linéaires analytiques, et l'analyse de systèmes non linéaires avec plusieurs entrées. Cependant, son application peut devenir complexe à mesure que la dimension du système augmente.

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 :

$$y_1(k) = \sum_0^{+\infty} h_1(\tau_1) u(k - \tau_1) \quad (1.13)$$

$h_1$  est le 1<sup>er</sup> noyau de Volterra.

La sortie  $y_2$ , du 2<sup>ème</sup> noyau de Volterra est :

$$y_2(k) = \sum_0^{+\infty} \sum_0^{+\infty} h_2(\tau_1, \tau_2) u(k - \tau_1) u(k - \tau_2) \quad (1.14)$$

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

$$y(k) = \sum_{K=1}^{+\infty} y_K(k) = y_1(k) + y_2(k) + \dots + y_K(k) + \dots \quad (1.15)$$

$y_K$ , la sortie du  $K$ -ème noyau de Volterra ( $h_K$ ), est écrit comme la généralisation du produit de convolution à  $K$ -ème ordre :

$$y_K(k) = \sum_0^{+\infty} \dots \sum_0^{+\infty} h_K(\tau_1, \dots, \tau_K) \prod_{i=1}^K u(k - \tau_i) \quad (1.16)$$

Par conséquent, en combinant les équations (1.15) et (1.16), la sortie  $y(k)$  d'un système non linéaire peut être exprimée, dans le cas continu, par :

$$y(k) = \sum_{K=1}^{+\infty} \sum_0^{+\infty} \dots \sum_0^{+\infty} h_K(\tau_1, \dots, \tau_K) \prod_{i=1}^K u(k - \tau_i) \quad (1.17)$$

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

### 1.2.6 Modèles blocs orientés

Les modèles de blocs orientés sont souvent utilisés pour représenter des systèmes dynamiques, où chaque bloc peut représenter une partie du système ou un composant spécifique. L'inclusion du terme "orientés" pourrait se référer à une configuration spécifique dans laquelle les blocs sont arrangés d'une manière particulière, les modèles de blocs orientés incluent plusieurs types, notamment le modèle de Hammerstein, Wiener, Wiener-Hammerstein et Hammerstein-Wiener.

#### Modèle de Hammerstein :

Dans un modèle de Hammerstein, le système est représenté par une partie statique non linéaire suivie d'une partie dynamique linéaire comme montré sur la figure 1.1. Cela signifie que l'entrée traverse d'abord une partie non linéaire avant d'être traitée par un système linéaire [3].

**Exemple :** Un amplificateur où la non-linéarité provient d'un élément saturable suivi d'une section linéaire.



FIGURE 1.1 – Modèle de Hammerstein

Les blocs orientés Hammerstein sont extrêmement utiles dans de nombreuses applications d'ingénierie, notamment dans la modélisation et le contrôle des systèmes non linéaires, l'identification des systèmes à partir de données expérimentales et la conception de contrôleurs robustes [8].

### Modèle de Wiener :

Un modèle de Wiener se compose d'une partie linéaire suivie d'une partie statique non linéaire comme montré sur la figure 1.2

Cela signifie que l'entrée passe d'abord à travers un système linéaire, puis à travers une fonction non linéaire.

**Exemple :** Un système où la réponse à un signal linéaire est suivie d'une distorsion non linéaire [3].

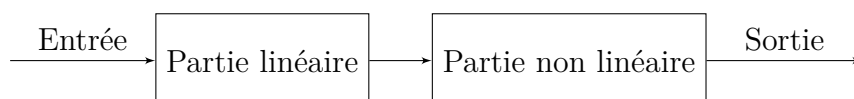


FIGURE 1.2 – Modèle de Wiener

Le bloc orienté Wiener est un outil puissant pour modéliser et contrôler les systèmes non linéaires. Voici quelques exemples concrets du domaine d'utilisation de ce modèle :

1. Traitement du signal audio : Modélisation de filtres audio non linéaires pour améliorer la qualité des signaux audio en éliminant le bruit [3].
2. Électronique : Modélisation de circuits électroniques non linéaires comme les amplificateurs de puissance [3].
3. Réseaux neuronaux : Modélisation de systèmes non linéaires complexes pour comprendre et améliorer les performances des réseaux de neurones [3].
4. Biologie : Caractérisation du comportement dynamique de systèmes biologiques non linéaires pour mieux comprendre les processus biologiques [3].
5. Contrôle des procédés : Modélisation de systèmes de chauffage et autres systèmes industriels non linéaires pour un contrôle optimal de ces systèmes [3].
6. Robotique : Modélisation de robots manipulateurs non linéaires pour améliorer la commande et le contrôle de ces systèmes complexes [3].

### Modèle de Wiener-Hammerstein :

Cette structure combine les modèles Wiener et Hammerstein, comprenant une partie linéaire suivie d'une non-linéarité, puis d'une autre partie linéaire comme montré dans la figure 1.3.

Ce modèle est utilisé pour estimer la réponse d'un système non linéaire à un signal d'entrée fini, et il trouve des applications dans divers domaines tels que la commande de

processus, l'analyse de systèmes non linéaires avec plusieurs entrées, et la modélisation de systèmes dynamiques complexes [3].

**Exemple :** Un système où l'entrée traverse initialement un élément linéaire, puis subit une distorsion non linéaire, suivie d'une autre section linéaire.

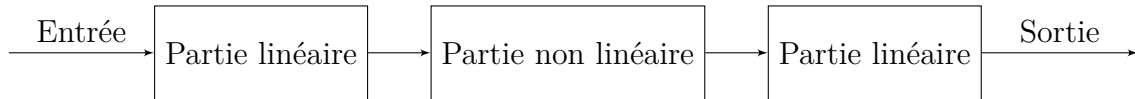


FIGURE 1.3 – Modèle de Wiener-Hammerstein

### Modèle de Hammerstein-Wiener :

Ce modèle est l'inverse du modèle Wiener-Hammerstein, comprenant d'abord une partie non linéaire, puis une section linéaire.

Ce modèle est utilisé pour estimer la réponse d'un système non linéaire à un signal d'entrée fini, et est largement utilisé dans les domaines de l'ingénierie, de la recherche et du contrôle des systèmes dynamiques non linéaires [3].

**Exemple :** Un système où une fonction non linéaire agit initialement sur l'entrée, suivie d'un traitement linéaire.

Ce dernier est présenté par la figure 1.4.

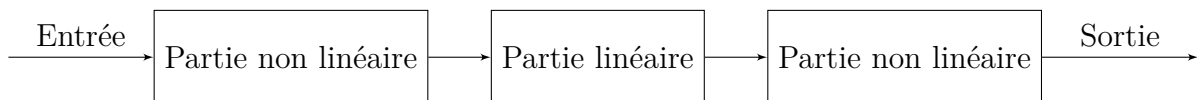


FIGURE 1.4 – Modèle de Hammerstein-Wiener

### Remarque :

La différence dans l'ordre des non-linéarités et de la dynamique linéaire peut entraîner des différences dans la flexibilité de modélisation. Par exemple, le bloc orienté de Wiener-Hammerstein peut être plus approprié pour les systèmes où la non-linéarité à la sortie est plus importante, tandis que le bloc orienté de Hammerstein-Wiener peut être préférable lorsque la non-linéarité à l'entrée est dominante. L'approche de la modélisation par blocs structurés repose sur l'identification des blocs individuels du système uniquement à partir des mesures d'entrée-sortie externes, sans accès aux signaux internes. Cela permet de maintenir la relation avec le système sous-adjacent et les composants représentés par chaque bloc [6].

La plupart des études sur les systèmes non linéaires de type blocs orientés ont généralement considéré que la partie dynamique linéaire est paramétrique. Cela signifie que le bloc linéaire est décrit par une fonction de transfert d'ordre connu ou par une représentation d'état avec une connaissance du degré de la dynamique.

Cependant, une représentation non paramétrique a également été utilisée, où la partie linéaire est choisie comme un filtre à réponse impulsionnelle finie (FIR), un filtre à réponse impulsionnelle infinie (IIR) ou comme la réponse fréquentielle [6].

Les modèles décrivant les éléments statiques non linéaires peuvent adopter différentes formes, comme des polynômes de degré connu, des polynômes de Laguerre, des splines cubiques, une zone morte ou une saturation [6].

En effet, l'avantage principal de l'utilisation de modèles polynomiaux pour représenter la partie non linéaire réside dans l'optimisation efficace des paramètres. De plus, les principales approches proposées dans la littérature présentent un inconvénient lié à la paramétrisation des blocs linéaires et non linéaires à l'aide de modèles d'entrées-sorties. Cela rend la méthode d'identification dépendante du modèle d'ordre choisi. Une solution à ce problème pourrait être l'utilisation de la représentation de l'espace d'état pour la partie linéaire. De plus, les modèles d'espace d'état sont souvent préférés aux modèles d'entrées-sorties pour traiter à la fois les systèmes multivariables et les systèmes SISO [6].

Dans ce mémoire, nous nous intéressons à l'identification des systèmes non linéaires de type blocs orientés, en particulier en nous appuyant sur le modèle de Wiener. Pour ce faire, nous avons mis en oeuvre le modèle d'espace d'état non linéaire polynomial (PNLSS), qui généralise le modèle d'espace d'état au cas non linéaire, pour décrire le système de Wiener [6].

### 1.3 Représentation d'espace d'état non linéaire polynomial

Le modèle d'espace d'état non linéaire polynomial (PNLSS) est une méthode utilisée pour modéliser et identifier les systèmes dynamiques non linéaires. Il repose sur la représentation des systèmes non linéaires par des modèles d'espace d'état non linéaires polynomiaux, ce qui permet de capturer les comportements non linéaires à l'aide de termes polynomiaux. Cette méthode est utilisée pour surmonter les limitations des modèles linéaires en capturant de manière plus précise les comportements non linéaires des systèmes dynamiques. Le modèle PNLSS est largement utilisé dans des domaines tels que la commande de processus, l'analyse de systèmes non linéaires à plusieurs entrées, et la modélisation de systèmes dynamiques complexes.

La méthode PNLSS est basée sur l'identification de modèles d'espace d'état non linéaires polynomiaux à partir de données expérimentales, et elle nécessite des techniques spécifiques pour l'estimation des paramètres du modèle [3][9].

### 1.3.1 Forme générale

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

$$\begin{cases} x(k+1) = f(x(k), u(k), \theta) \\ y(k) = g(x(k), u(k), \theta) \end{cases} \quad (1.18)$$

$x(k)$  est le vecteur d'état.

$u(k)$   $y(k)$  sont respectivement l'entrée et la sortie du système.

$f$  et  $g$  sont des champs de vecteurs.

Nous avons utilisé une approche polynomiale pour étendre les fonctions  $f(\cdot)$  et  $g(\cdot)$  dans le cadre de la modélisation de l'espace d'état non linéaire. Cette approche, appelée Polynomial Non-linear State Space (PNLSS), combine un modèle linéaire classique d'espace d'état avec des termes non linéaires. Le PNLSS offre l'avantage de pouvoir décrire efficacement une grande variété de systèmes non linéaires, ce qui le rend facile à appliquer dans des situations multivariées [10].

Ce modèle est défini comme suit [11][12][13] :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + E\zeta(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases} \quad (1.19)$$

Les paramètres du sous-ensemble linéaire sont représentés par les matrices  $A \in R^{na \times na}$  et  $B \in R^{na \times nu}$  dans l'équation d'état, et par les matrices  $C \in R^{ny \times na}$  et  $D \in R^{ny \times nu}$  dans l'équation de sortie. Les vecteurs  $\zeta(k) \in R^{n\zeta}$  et  $\eta(k) \in R^{n\eta}$  contiennent des termes non linéaires sous forme de monômes dans  $x(k)$  et  $u(k)$ , allant du deuxième degré à un degré choisi  $r$ . Les coefficients associés à ces termes non linéaires sont définis par les matrices  $E \in R^{na \times n\zeta}$  et  $F \in R^{ny \times n\eta}$ .

Les termes non linéaires de degré un sont inclus dans la composante linéaire de la structure du modèle PNLSS.

Le modèle PNLSS est utilisé pour identifier les systèmes non linéaires en représentant leurs comportements complexes à l'aide de polynômes, ce dernier offre une approche pour aborder certains types de systèmes non linéaires, notamment les modèles blocs orientés. Dans la suite, cette approche est utilisée pour le modèle non linéaires de Wiener [6].

### 1.3.2 Description du modèle de Wiener sous forme PNLSS

Dans cette section, on considère la description du modèle de Wiener en se basant sur l'approche PNLSS. Cependant on se limitera aux systèmes Single Input, Single Output (le SISO est un système dans lequel il n'y a qu'une seule entrée et qu'une seule sortie). 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 1.20), suivi d'un bloc non linéaire statique (équation 1.21) qui est un polynôme non linéaire d'ordre connu  $r$  (équation 1.22) [14][15], comme le montre la figure 1.5.

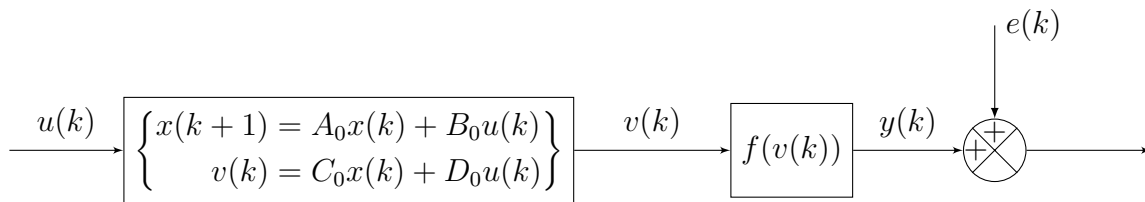


FIGURE 1.5 – Modèle de Wiener PNLSS

Avec :

$$\begin{cases} x(k+1) = A_0x(k) + B_0u(k) \\ v(k) = C_0x(k) + D_0u(k) \end{cases} \quad (1.20)$$

$A_0$ ,  $B_0$ ,  $C_0$  et  $D_0$  sont des matrices de dimensions appropriées.

$$f(v(k)) = p_1f_1(v(k)) + p_2f_2(v(k)) + \dots + p_rf_r(v(k)) \quad (1.21)$$

$$y(k) = \sum_{i=1}^r p_i v^i(k) \quad (1.22)$$

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

$$\begin{cases} x(k+1) = A_0x(k) + B_0u(k) \\ y(k) = \sum_{i=1}^r p_i v^i(k) \\ y(k) = \sum_{i=1}^r p_i (C_0x(k) + D_0u(k))^i \end{cases} \quad (1.23)$$

**Modèle de Wiener avec  $n_a=2$  et  $r=2$  :**

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

$$\begin{cases} x(k+1) = A_0x(k) + B_0u(k) \\ v(k) = C_0x(k) + D_0u(k) \\ y(k) = p_1v^1(k) + p_2v^2(k) \end{cases} \quad (1.24)$$

**Remarque :**

Afin de ne pas surcharger les équations, la variable  $K$  est négligée

La sortie système de l'équation (1.24) après avoir remplacé  $v(k)$  dans  $y(k)$  est donnée comme suit :

$$\begin{aligned} y &= p_1v^1 + p_2v^2 \\ &= p_1(C_0x + D_0u) + p_2(C_0x + D_0u)^2 \end{aligned} \quad (1.25)$$

et

$$y = p_1(c_1x_1 + c_2x_2 + du) + p_2(c_1x_1 + c_2x_2 + du)^2 \quad (1.26)$$

Après avoir développé l'expression précédente en utilisant l'expansion multinomiale, nous obtenons :

$$\begin{aligned} y &= p_1c_1x_1 + p_1c_2x_2 + p_1du \\ &\quad + p_2(c_1^2x_1^2 + 2c_1c_2x_1x_2 + 2c_1dx_1u \\ &\quad + c_2^2x_2^2 + 2c_2dx_2u + d^2u^2) \end{aligned} \quad (1.27)$$

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

$$y = \begin{bmatrix} p_1c_1 & p_1c_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} p_1d \end{bmatrix} u + \begin{bmatrix} p_2c_1^2 & 2p_2c_1c_2 & 2p_2c_1d & p_2c_2^2 & 2p_2c_2d & p_2d^2 \end{bmatrix}^+ \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_1u \\ x_2^2 \\ x_2u \\ u^2 \end{bmatrix} \quad (1.28)$$

Les équations PNLSS analogue qui décrivent le comportement du système de Wiener sont :

$$\left\{ \begin{array}{l} x(k+1) = A_0x(k) + B_0u(k) \\ y = p_1 \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + p_1 \begin{bmatrix} d \end{bmatrix} u + p_2 \begin{bmatrix} c_1^2 & 2c_1c_2 & 2c_1d & c_2^2 & 2c_2d & d^2 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_1u \\ x_2^2 \\ x_2u \\ u^2 \end{bmatrix} \end{array} \right. \quad (1.29)$$

**Modèle de Wiener avec  $n_a=2$  et  $r=3$  :**

Dans ce cas le modèle de Wiener est décrit par l'équation :

$$\left\{ \begin{array}{l} x(k+1) = A_0x(k) + B_0u(k) \\ v(k) = C_0x(k) + D_0u(k) \\ y(k) = p_1v^1(k) + p_2v^2(k) + p_3v^3(k) \end{array} \right. \quad (1.30)$$

La sortie  $y$  peut être alors formulée comme suit :

$$y = p_1(c_1x_1 + c_2x_2 + du) + p_2(c_1x_1 + c_2x_2 + du)^2 + p_3(c_1x_1 + c_2x_2 + du)^3 \quad (1.31)$$

Après avoir développé l'expression (1.31) en utilisant l'expansion multinomiale, nous obtenons :

$$\begin{aligned} y = & p_1c_1x_1 + p_1c_2x_2 + p_1du \\ & + p_2[c_1^2x_1^2 + 2c_1c_2x_1x_2 + 2c_1dx_1u + c_2^2x_2^2 + 2c_2dx_2u + d^2u^2] \\ & + p_3[c_1^3x_1^3 + 3c_1^2c_2x_1^2x_2 + 3c_1^2dx_1^2u + 3c_1c_2^2x_1x_2^2 + 6c_1c_2dx_1x_2u \\ & + 3c_1d^2x_1u^2 + c_2^3x_2^3 + 3c_2^2dx_2^2u + 3c_2d^2x_2u^2 + d^3u^3] \end{aligned} \quad (1.32)$$

Et au final on peut reformuler la représentation du modèle sous forme vectorielle comme suit :



Les matrices du système de Wiener sous forme PNLSS (équation 1.19) s'en déduisent comme suit :

$$\left\{ \begin{array}{l}
 A = A_0 \\
 B = B_0 \\
 C = p_1 C_0 \\
 D = p_1 D_0 \\
 E = 0 \\
 F = [\mathbf{p}_2 [C_0 D_0]^2 \dots \mathbf{p}_r [C_0 D_0]^r] \\
 \text{et} \\
 F = [\mathbf{p}_2 [Z]^2 \dots \mathbf{p}_r [Z]^r] \\
 \text{avec} \\
 Z = [C_0 D_0]
 \end{array} \right. \quad (1.37)$$

Avec

$$\left\{ \begin{array}{l}
 \zeta(k) = 0 \\
 \eta(k) = [\xi^2(k), \dots, \xi^r(k)] \\
 \xi(k) = [x(k) \quad u(k)]
 \end{array} \right. \quad (1.38)$$

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

$$\left\{ \begin{array}{l}
 x(k+1) = Ax(k) + Bu(k) \\
 y(k) = Cx(k) + Du(k) + F\eta(k)
 \end{array} \right. \quad (1.39)$$

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

## **1.4 Conclusion**

Dans ce chapitre, nous avons présenté les systèmes non linéaires et leur comportement complexe, en mettant en avant les différentes classes de systèmes. Cette étude souligne l'importance cruciale de comprendre ces systèmes dans divers domaines, de l'ingénierie à la science. En effet, leur comportement imprévisible nécessite le développement continu de nouvelles théories et approches.

Nous aborderons dans le prochain chapitre un élément essentiel de leur analyse qui est l'identification. Cette étape est essentielle pour modéliser avec précision le comportement de ces systèmes, ouvrant ainsi la voie à des avancées significatives dans leur compréhension et leur contrôle.

# Chapitre 2

## Identification des systèmes non linéaires

### 2.1 Introduction

L'identification des systèmes non linéaires est un domaine de recherche complexe et essentiel dans de nombreux domaines de l'ingénierie, des sciences appliquées et des sciences fondamentales, le but de cette partie est de caractériser mathématiquement le comportement dynamique de ces systèmes à partir de données expérimentales ou de modèles théoriques. Cela permet de comprendre le fonctionnement des systèmes, de les modéliser avec précision et de concevoir des stratégies de commande et de régulation adaptées.

### 2.2 Identification

#### 2.2.1 Définition de l'identification

L'identification de système, est une technique de l'automatique visant à obtenir un modèle mathématique d'un système à partir de mesures de ses signaux d'entrée et de sortie. Cette méthode est essentielle pour caractériser les systèmes dynamiques et construire des modèles mathématiques qui représentent les interactions entre les entrées et les sorties d'un système [16].

En automatique, il est primordial d'obtenir des modèles mathématiques assez précis, car cela constitue la partie la plus importante de l'analyse complète d'un système de commande.

## 2.2.2 Étapes de l'identification

Les étapes de l'identification avec minimisation d'un critère sont essentielles pour obtenir un modèle mathématique précis d'un système, afin de faciliter son analyse et sa compréhension pour des applications de simulation, de commande ou de régulation. Les étapes générales de ce processus sont, tel que résumés sur la figure 2.1 :

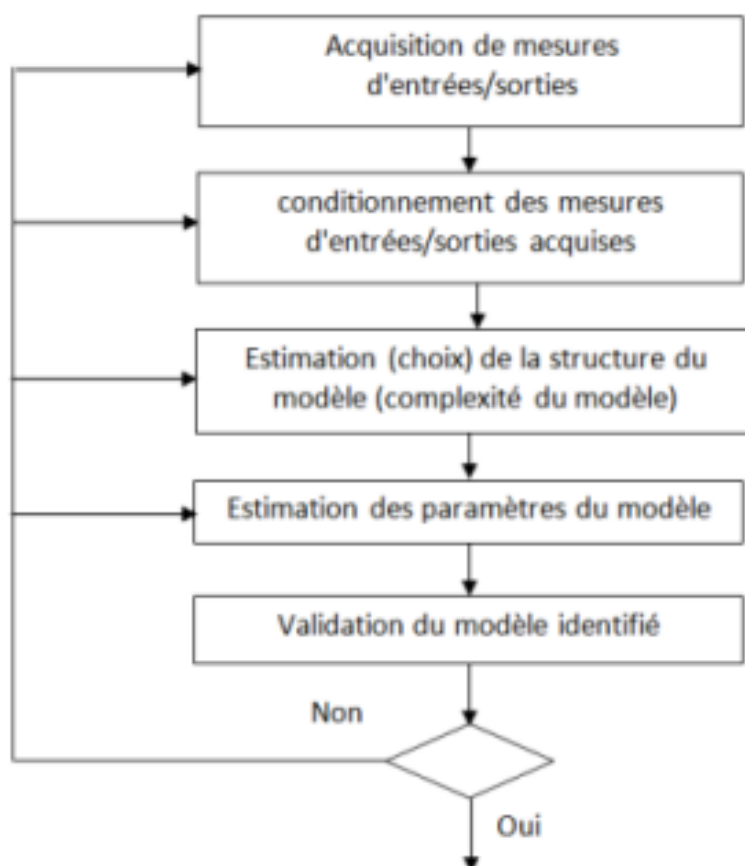


FIGURE 2.1 – Étapes de l'identification

1. Étape 1 : Acquisition de mesures d'entrées/sorties
  - Application d'un signal d'excitation particulier (plusieurs types de signaux d'excitation sont utilisés parmi ces signaux on trouve : l'échelon, la rampe, le signal sinusoïdale, la SBPA,...)
  - Recueillir des données expérimentales ou de simulation sur le système (entrée et sortie), ces données sont nécessaires pour déterminer les paramètres du modèle qui les représente.

2. Étape 2 : Conditionnement des mesures acquises  
Avant d'utiliser les mesures brutes issues de l'acquisition des données d'entrée et de sortie pour l'identification, il est souvent nécessaire de réaliser plusieurs opérations préliminaires, notamment :
  - Élimination des mesures inutiles.
  - Élimination des points aberrants.
  - Filtrage des bruits de mesure.
  
3. Étape 3 : Choix de la structure du modèle  
Dans cette étape, nous devons choisir la structure mathématique du modèle à identifier (le type, l'ordre, le retard ... ).
  
4. Étape 4 : Détermination des paramètres du modèle
  - Choisir une méthode d'identification à partir du type du modèle choisi et des exigences du cahier de charge (précision, rapidité, ..)
  - Appliquer la méthode pour déterminer les valeurs numériques des paramètres du modèle.
  
5. Étape 5 : Validation du modèle
  - Cette étape consiste à vérifier la qualité du modèle identifié si celui-ci peut reproduire les sorties réelles avec une précision acceptable.
  - Cette étape est essentielle pour s'assurer que le modèle est fiable et peut être utilisé pour des applications pratiques.

### 2.2.3 Méthodes de l'identification

L'identification des systèmes est une étape fondamentale en ingénierie et en sciences appliquées. Elle vise à caractériser mathématiquement le comportement de systèmes complexes, tels que les circuits électriques, les processus industriels ou les systèmes biologiques.

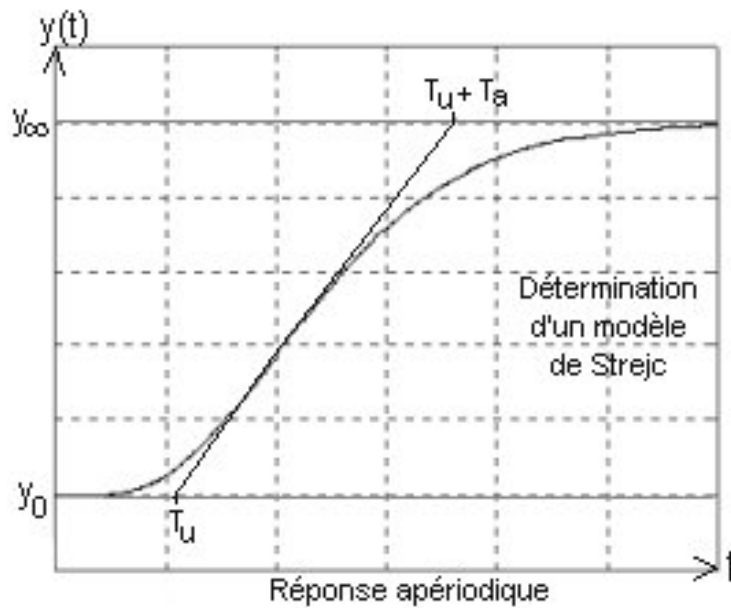
À travers différentes méthodes, cette approche permet de déterminer les paramètres inconnus d'un modèle, offrant ainsi une compréhension approfondie de leur dynamique et facilitant leur analyse et leur contrôle.

Différentes techniques sont utilisées :

a) Méthodes graphiques :

Ces méthodes graphiques permettent d'identifier les systèmes à partir de leur réponse indicielle [16] [17]. Elles incluent :

- **Méthode de Strejc** : Cette méthode s'applique aux systèmes apériodiques, elle consiste à tracer la tangente au point d'inflexion de  $y(t)$  et d'en déduire la valeur des temps  $T_a$  et  $T_u$ . On compare alors le rapport  $T_u/T_a$  expérimental avec les rapports théoriques  $T'_u/T_a$  de la réponse indicielle d'un système ayant  $n$  constantes de temps identiques.



On prend dans le tableau ci-dessous la valeur de  $n$  qui correspond au rapport  $T'_u/T_a$  immédiatement inférieur au rapport  $T_u/T_a$  expérimental.

$T'_u/T_a$	0.104	0.218	0.319	0.410	0.493	0.570	0.642
$n$	2	3	4	5	6	7	8
$T_a/\tau$	2.718	3.695	4.463	5.119	5.699	6.226	6.711

TABLE 2.1 – Tableau de Strejc

La troisième ligne du tableau permet alors de calculer la valeur des  $n$  constantes de temps identiques (pour  $n$  donné,  $\tau$  est proportionnelle au temps  $T_a$ ).

Il ne reste plus qu'à compenser la différence pour une même valeur de  $T_a$  entre  $T_u/T_a$  et  $T'_u/T_a$  avec un retard  $T$  :

$$\frac{T_u}{T_a} = \frac{T'_u + T}{T_a} \Rightarrow T = T_a \left( \frac{T_u}{T_a} - \frac{T'_u}{T_a} \right)$$

La fonction de transfert obtenue est alors :

$$T(p) = \frac{A}{(1 + \tau p)^n} e^{-T_r p}$$

**Remarque :** la mesure du gain statique  $A$  est classique : c'est le rapport de la variation totale de  $y(t)$  sur l'amplitude de l'échelon de commande. Le modèle de Strejc est très utilisé pour l'identification des processus en boucle ouverte car 9 fois sur 10 la réponse indicielle en boucle ouverte est apériodique. L'expression de cette fonction de transfert est simple et permet une synthèse correcte et aisée de l'asservissement.

- **Méthode de Broïda :** cette méthode consiste à assimiler le système à un 1er ordre avec retard pur dont la fonction de transfert est de la forme suivante :

$$H(p) = K \frac{e^{-(T_r p)}}{1 + \tau p} \quad (2.1)$$

Les paramètres à identifier sont :

- $K$  est le gain statique.
- $T_r$  est le retard.
- $\tau$  la constante de temps.

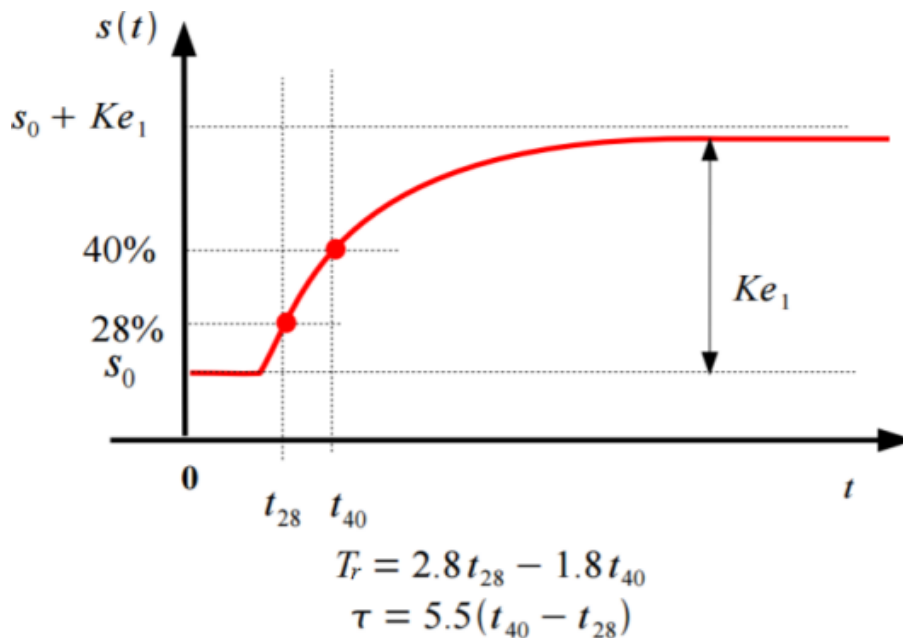


FIGURE 2.2 – Méthode de Broïda

La méthode de Broïda n'est validée que si le rapport  $T_r/\tau$  est inférieur à 0.25.

Cette méthode offre un moyen très avantageux à l'asservissement qui consiste à aider à déterminer le type de correcteur qu'il faut utiliser dans une commande en boucle fermée afin de supprimer l'erreur statique entre le signal à commander et sa consigne, et aussi le dimensionnement de ses coefficients. Cela dépend de la valeur du rapport  $T_r/\tau$  obtenu graphiquement.

Pour les systèmes oscillants, elle est utilisée pour estimer les paramètres d'un modèle à partir de données expérimentales [16][17].

Ces deux méthodes permettent d'identifier les paramètres du modèle sans avoir besoin de connaître en détail les équations physiques du système.

### b) Méthode à erreur d'équation

La méthode à erreur d'équation, utilisée pour identifier les paramètres des modèles dynamiques en minimisant l'écart entre les observations et les prédictions, est souvent mise en œuvre en utilisant la méthode des moindres carrés, qui vise à trouver les paramètres optimaux en minimisant la somme des carrés des erreurs.

La structure de cette méthode est représentée dans la figure suivante :

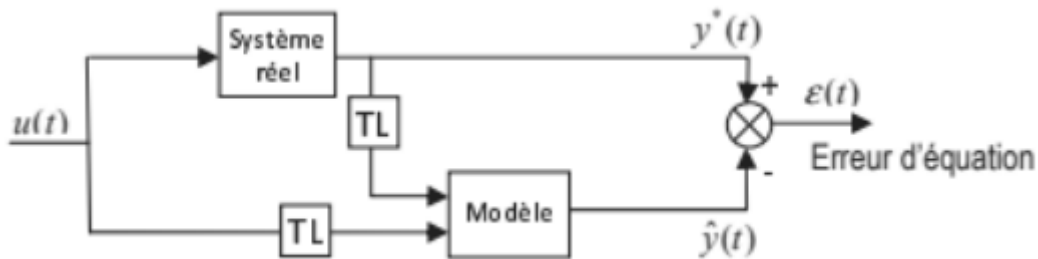


FIGURE 2.3 – Structure erreur d'équation

- $u(t)$  : Signal d'entrée appliqué au système réel et au modèle.
- Système réel : Le système physique dont on veut identifier les paramètres.
- Modèle : Le modèle mathématique utilisé pour représenter le système réel.
- $y^*(t)$  : Sortie réelle du système.
- $\hat{y}(t)$  : Sortie estimée du modèle.
- Erreur d'équation  $\varepsilon(t)$  : Différence entre la sortie réelle et la sortie estimée.
- TL (Transformation Linéaire) : Transformations éventuelles appliquées aux signaux avant la comparaison.

### c) Méthode des moindres carrés

La méthode des moindres carrés (Least square : LS) peut être appliquée de diverses manières, en fonction des caractéristiques des données et des exigences spécifiques du problème.

Voici un aperçu des différentes variantes de la méthode des moindres carrés :

#### — Moindres carrés simples

Utilisé pour ajuster une ligne de régression linéaire à un ensemble de données. Il s'agit de la forme la plus basique et la plus courante de la méthode des moindres carrés.

La minimisation de l'erreur de prédiction est définie par :

$$e(k) = y(k) - \hat{y}(k) \quad (2.2)$$

Avec :

—  $y(k)$  est la sortie échantillonnée du système à identifier.

—  $\hat{y}(k)$  est la sortie échantillonnée du modèle prédictif.

Les valeurs des paramètres du modèle sont obtenues par la minimisation du critère quadratique :

$$J(\theta) = \frac{1}{N} \sum_{k=1}^N e^2(k) \quad (2.3)$$

$$= \frac{1}{N} \sum_{k=1}^N (y(k) - \hat{y}(k, \theta))^2 \quad (2.4)$$

La solution de la méthode des moindres carrés simples est donnée par :

$$\hat{\theta} = (\phi^T \phi)^{-1} \phi^T Y \quad (2.5)$$

Avec :

$$Y = \begin{bmatrix} y(n+1) \\ \vdots \\ y(N) \end{bmatrix}, \phi = \begin{bmatrix} -y(n) & \cdots -y(1) & u(n) & \cdots u(n-m+1) \\ \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & \cdots -y(N-n) & u(N-1) & \cdots u(N-m) \end{bmatrix}$$

La matrice  $(\phi^T \phi)^{-1} \phi^T$  est appelée la pseudo-inverse de  $\phi$ .

— **Moindres carrés récurrents**

Les moindres carrés récurrents (RLS) est une méthode permettant d'estimer les paramètres d'un modèle en temps réel. Comparé à la méthode des moindres carrés ordinaire, l'algorithme récurrent a l'avantage de ne pas nécessiter le stockage de toutes les données passées, ce qui le rend plus efficace en termes de calcul et de mémoire [18].

— **Moindres carrés à erreurs corrigées**

Cette méthode est une variante de la méthode RLS, elle intègre des corrections des erreurs d'observation dans le modèle. elle est utilisée lorsque les données d'entrée et de sortie sont bruitées, permettant des estimations plus robustes des paramètres.

— **Moindres carrés pondérés**

Dans certains cas, les observations montrent des niveaux de précision différents. Cette méthode donne des poids à chaque observation en fonction de leur précision relative.

Chacune des méthodes citées ci-dessus a ses propres applications et avantages. Le choix de la méthode dépend de la nature des données et des objectifs particuliers de l'analyse.

**d) Méthode du modèle**

Ces algorithmes sont utilisés pour minimiser la différence entre la sortie mesurée et la sortie prédite par le modèle, ils sont applicables sans restriction aux systèmes linéaires ou non linéaires, et aussi discrets ou continus. Ils sont basés sur des techniques d'optimisation non linéaire pour minimiser le critère quadratique [17].

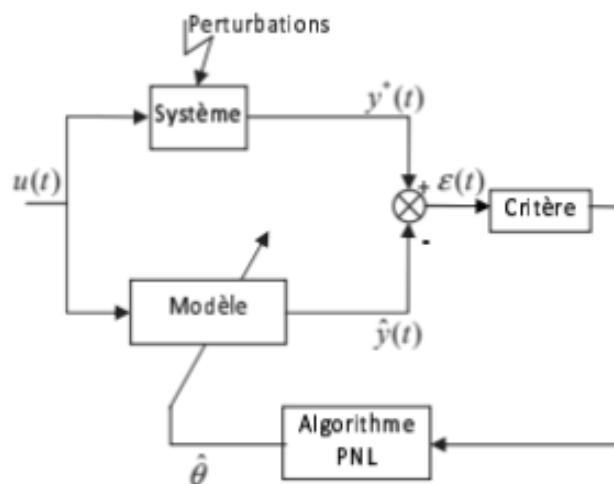


FIGURE 2.4 – Principe de la méthode du modèle

Les algorithmes à erreur de sortie diffèrent dans la manière dont ils gèrent l'optimisation. Plusieurs méthodes d'optimisation non linéaire itératives peuvent être utilisées :

- Méthode du gradient,
- Méthode de Gauss-Newton,
- Méthode de Levenberg-Marquardt. . .

Par la suite, nous présenterons deux techniques de base pour l'identification via la méthode d'erreur de sortie. Ensuite, nous aborderons la méthode de Levenberg-Marquardt, qui combine les avantages de ces deux techniques.

— **Gradient**

Technique d'optimisation utilisée pour trouver le minimum (ou maximum) d'une fonction, en ajustant itérative-ment les paramètres du modèle dans la direction opposée au gradient de la fonction d'erreur, à chaque itération, les paramètres sont mis à jour selon la règle de mise à jour du gradient jusqu'à ce qu'une condition d'arrêt soit atteinte[16].

L'équation de récurrence est de la forme suivante :

$$\theta^{(i+1)} = \theta^{(i)} - \lambda \frac{\partial J}{\partial \theta} \Big|_{\theta=\theta^{(i)}} = \theta^{(i)} - \lambda (J'_{\theta})_{(\theta=\theta^{(i)})} \quad (2.6)$$

$\theta^{(i)}$  est le vecteur de paramètres estimé à l'itération  $i$ .

Le coefficient  $\lambda$  est un paramètre de contrôle.

— **Gauss-Newton (GN)**

Cet algorithme utilise la matrice des dérivés secondes du critère appelée le Hessian ( $J''_{\theta\theta}$ ). Il est plus stable et plus rapide que la méthode du gradient mais sa convergence n'est pas garantie et dépend aussi des paramètres initiaux[19].

L'équation de récurrence est de la forme suivante :

$$\theta^{(i+1)} = \theta^{(i)} - \left[ (J''_{\theta\theta})^{-1} J'_{\theta} \right]_{(\theta=\theta^{(i)})} \quad (2.7)$$

Cependant, cette méthode présente un grand risque de divergence près d'un point d'inflexion.

— **Méthode avec l'algorithme de Levenberg-Marquardt**

L'algorithme de LM, permet d'obtenir une solution numérique au problème de minimisation d'une fonction, souvent non linéaire et dépendant de plusieurs variables[20][21]. Il s'agit d'une méthode d'optimisation qui combine les approches de l'algorithme

de Gauss-Newton et de la descente de gradient[21], cet algorithme est plus stable que celui de Gauss-Newton et peut trouver une solution même s'il est démarré très loin d'un minimum. Cependant, pour certaines fonctions très régulières, il peut converger légèrement moins vite.

En pratique, l'algorithme LM est largement utilisé dans de nombreux domaines, notamment pour les réseaux de neurones, car il nécessite beaucoup moins d'itérations qu'une simple descente de gradient, il est utilisé aussi pour l'ajustement de données expérimentales en utilisant le critère des moindres carrés non linéaires, la résolution de systèmes d'équations non linéaires, la recherche des zéros de fonctions non linéaires etc.

Le principe de la méthode de Levenberg-Marquardt peut être résumé dans la figure 2.5 ci-dessous.

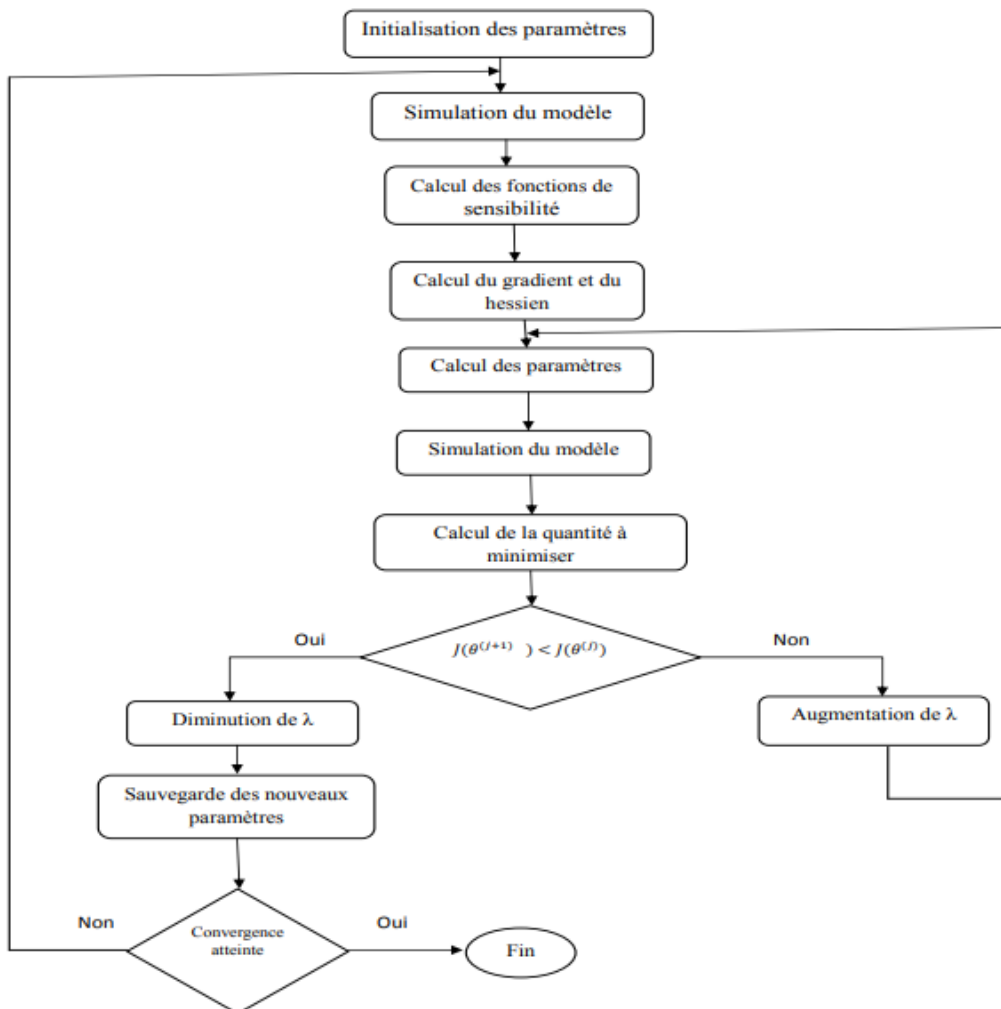


FIGURE 2.5 – Principe de l'algorithme de Levenberg-Marquardt

Le but principal de cet algorithme est de minimiser une fonction de coût qui mesure la différence entre les valeurs enregistrées et les valeurs prédites par le modèle. Cela se fait en mettant à jour de manière itérative les paramètres du modèle pour combler cet écart. La relation de récurrence de la méthode est donnée par :

$$\theta^{(i+1)} = \theta^{(i)} - \left[ (J''_{\theta\theta} + \lambda I)^{-1} J'_{\theta} \right]_{\theta=\theta^{(i)}} \quad (2.8)$$

$\lambda$  : La valeur du paramètre de contrôle  $\lambda$  est appelé coefficient de Marquardt. Lorsque  $J$  augmente,  $\lambda$  est augmenté afin d'obtenir une évolution de l'algorithme type gradient et retrouver une meilleure direction de recherche. Par contre, si  $J$  diminue,  $\lambda$  diminue et l'algorithme se comporte de manière similaire à la méthode de Gauss-Newton.

L'objectif est de trouver un vecteur de paramètres  $\theta$  qui va minimiser le critère quadratique  $J_{\theta}$  défini comme suit :

$$J_{\theta} = \frac{1}{K} \sum_{k=1}^K \epsilon^2(k) \quad (2.9)$$

Avec :

$\epsilon^2(k)$  : l'erreur de prédiction entre la sortie mesurée  $y(k)$  et de la sortie estimée  $\hat{y}(k)$

Tel que  $\epsilon(k) = y(k) - \hat{y}(k)$

$K$  est l'horizon d'observation ( $u(k), y(k)$ ).

La mise à jour du vecteur de paramètres  $\theta$  se fait à partir de l'équation de récurrence :

$$\theta^{(i+1)} = \theta^{(i)} - \left\{ \left[ J^{\theta} + \lambda I \right]^{-1} J' \right\}_{\hat{\theta}=\theta^{(i)}} \quad (2.10)$$

Avec :

$\hat{\theta}$  : l'estimation du vecteur de paramètres  $\theta$  et  $\theta^{(l+1)}$  est son estimation à l'itération  $(l + 1)$ .

$\lambda$  est un paramètre de contrôle (le coefficient de Marquardt).

$I$  est une matrice d'identité.

$$\left\{ \begin{array}{ll} J' = -\frac{2}{N}(\sum_{k=1}^N \epsilon_k \sigma_{yk/\theta}) & \text{est le Gradient} \\ J'' = -\frac{2}{N}(\sum_{k=1}^N \sigma_{yk/\theta} \sigma_{yk/\theta}^T) & \text{est le Hessien} \\ \sigma_{yk/\theta_j} = \frac{\sigma \hat{y}(k, \theta)}{\sigma_{\theta_j}} & \text{est la fonction de sensibilité de la sortie} \\ & \text{par rapport aux paramètres} \end{array} \right. \quad (2.11)$$

La méthode de minimisation du critère repose sur le calcul des fonctions de sensibilités des paramètres, crucial pour l'identification. Ces sensibilités agissent localement comme les vecteurs de régression dans les modèles linéaires, leur calcul précis est essentiel pour une identification réussie. Dans ce travail l'utilisation de la méthode Levenberg-Marquardt est considérée.

### e) Méthodes heuristiques

Ces approches innovantes exploitent des techniques d'optimisation inspirées du comportement naturel pour résoudre des problèmes complexes de manière efficace.

#### — L'optimisation par essaim de particule

Le PSO (Particle Swarm Optimization) est un algorithme d'optimisation inspiré du comportement des essaims [16][17], Il estime les paramètres du modèle en simulant le déplacement des particules dans l'espace de recherche [16].

Cette approche permet de trouver des solutions optimales à des problèmes d'optimisation complexes, son fonctionnement peut être décrit comme suit :

#### — Etape 1 : Initialisation

Un ensemble de particules est créé aléatoirement dans l'espace de recherche, chacune de ces particules représente une solution potentielle au problème d'optimisation, et chacune à une position et une vitesse initiale [22].

#### — Etape 2 : Déplacement des particules

À chaque itération, la vitesse de chaque particule est mise à jour en fonction de trois composantes[23] :

- Sa vitesse actuelle, pondérée par un coefficient d'inertie  $w$ .
- La distance entre sa position actuelle et sa meilleure position personnelle ( $pbest$ ), pondérée par un facteur cognitif  $c_1$ .
- La distance entre sa position actuelle et la meilleure position globale ( $gbest$ ), pondérée par un facteur social  $c_2$ .



**Remarque :** Le PSO et le SAVPSO sont deux types d'algorithmes d'optimisation par essaim de particules. Le PSO est l'algorithme original, où les particules ajustent leur position dans l'espace de recherche en fonction de leur expérience et de celle de leurs voisins. Le SAVPSO est une amélioration qui ajuste automatiquement la vitesse des particules en fonction de leur performance et de la progression de l'algorithme, ce qui peut accélérer la convergence vers une solution optimale. En résumé, la différence principale réside dans la manière dont la vitesse des particules est gérée pendant l'optimisation.

#### f) Méthode aléatoire

Les méthodes aléatoires pour l'identification paramétrique des systèmes sont basées sur l'utilisation de signaux d'entrée aléatoires ou pseudo-aléatoires.

Voici quelques points clés sur leur utilisation :

- Pour la simulation, on utilise systématiquement comme signal d'entrée, la séquence binaire pseudo aléatoire (SBPA) qui est riche en fréquence et en même temps persistante[26]. Ce type de signal permet d'exciter le système sur une large bande de fréquences.
- L'utilisation d'un signal d'entrée aléatoire permet d'éviter les biais d'identification lorsque le système a plusieurs entrées/sorties. Exciter les entrées de façon découplée est important pour avoir un modèle représentatif du fonctionnement normal du système.
- En pratique, pour obtenir un modèle consistant, il est important d'exciter le système avec un signal riche en fréquences, couvrant toute la plage de fonctionnement. Le ratio signal/bruit doit être suffisant et le nombre de points de mesure significatif

## 2.3 État de l'art sur l'identification des systèmes non linéaires

L'identification des systèmes non linéaires est un domaine de recherche important et complexe, il est en évolution rapide. Contrairement aux systèmes linéaires, il n'existe pas de méthode d'identification universelle s'appliquant à tous les types, la recherche dans ce domaine a conduit à une diversité de techniques pour certaines classes de systèmes non linéaires allant des modèles boîte noire aux modèles à structure connue, en passant par des techniques d'optimisation et d'apprentissage automatique. Le choix de la méthode dépend de la complexité du système et des informations disponibles [28].

Dans le cas de modèles à structure connue, plusieurs approches sont utilisées. Tout d'abord, on retrouve les méthodes basées sur des structures de modèles paramétriques spécifiques, telles que les modèles de Wiener, Hammerstein ou NARMAX[3][28][29], ces modèles sont conçus pour représenter et estimer les systèmes non linéaires de manière précise[3][30]. Il existe aussi des méthodes qui font usage des séries de Volterra ou de Wiener pour approximer les non-linéarités des systèmes[28][29], ces séries sont des outils essentiels dans l'identification des systèmes non linéaires, offrant une représentation efficace des comportements complexes[28][29][30].

De plus, les approches par multi-modèles blocs orientés sont utilisées pour simplifier l'analyse et la modélisation des systèmes non linéaires, en contournant leur complexité grâce à une décomposition en sous-modèles plus simples [31]. Ces méthodes offrent une perspective structurée pour aborder la diversité des comportements non linéaires.

L'identification et l'estimation des paramètres des systèmes de Wiener PNLSS, incluant des approches spécialisées incluant :

- Une méthode à erreur de sortie basée sur l'algorithme de Levenberg-Marquardt a été développée pour identifier les systèmes de Wiener PNLSS [6].
- Des méthodes heuristiques comme la méthode à essaim de particules ont également été adaptées pour estimer les paramètres des modèles de Wiener fractionnaires PNLSS [6].
- L'approche PNLSS a aussi été utilisée pour initialiser les modèles d'état non linéaires appliqués aux systèmes de Wiener-Hammerstein.

Cependant il n'y a pas de méthode générale applicable à tous les systèmes non linéaires en raison de leur complexité. Les techniques classiques d'identification ne sont pas toujours applicables, d'où la nécessité de développer de nouvelles approches, comme l'identification neuronale.

## 2.4 Identification des systèmes de Wiener PNLSS

L'approche PNLSS permet de capturer efficacement le comportement non linéaire du système[3], cette approche est utilisée pour modéliser les systèmes non linéaires avec une structure de type Wiener. Ce type de structure se compose d'un bloc linéaire dynamique suivi d'un bloc non linéaire statique[3].

Cette méthode représente un modèle flexible et efficace pour représenter une large classe de systèmes non linéaires et permet de capturer le comportement dynamique et statique non linéaire complexe des systèmes[3].

La description du modèle de Wiener par le modèle PNLSS est :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) + F\eta(k) \end{cases} \quad (2.12)$$

### Remarque :

Dans notre travail, l'identification des systèmes de Wiener PNLSS vas se faire en utilisant l'algorithme de LM, qui nécessite le calcul des fonctions de sensibilité.

### 2.4.1 Calcul des fonctions de sensibilité

Pour ajuster les paramètres du modèle, il est nécessaire de comprendre comment les modifications des paramètres affectent les résultats du modèle. Cela implique le calcul de fonction de sensibilité, qui est la dérivée de la sortie par rapport aux paramètres.

La fonction de sensibilité reflète l'impact des modifications des paramètres sur la sortie du système.

Dans notre cas nous nous intéressons à la description du modèle de Wiener par le modèle PNLSS.

Après dérivation de l'équation (2.12) par rapport à  $\theta$  on aura :

$$\begin{cases} \frac{\partial}{\partial \theta_j} x(k+1) = \frac{\partial A}{\partial \theta_j} x(k) + A \frac{\partial x(k)}{\partial \theta_j} + \frac{\partial B}{\partial \theta_j} u(k) + B \frac{\partial u(k)}{\partial \theta_j} \\ \frac{\partial y(k)}{\partial \theta_j} = \frac{\partial C}{\partial \theta_j} x(k) + C \frac{\partial x(k)}{\partial \theta_j} + \frac{\partial D}{\partial \theta_j} u(k) + D \frac{\partial u(k)}{\partial \theta_j} + \frac{\partial F}{\partial \theta_j} \eta(k) + F \frac{\partial \eta(k)}{\partial \theta_j} \end{cases} \quad (2.13)$$

Avec

$$\begin{cases} \frac{\partial x(k+1)}{\partial \theta_j} = \sigma x(k+1) / \theta_j \\ \frac{\partial x(k)}{\partial \theta_j} = \sigma x(k) / \theta_j \\ \frac{\partial u(k)}{\partial \theta_j} = 0 \\ \frac{\partial y(k)}{\partial \theta_j} = \sigma y(k) / \theta_j \\ \frac{\partial \eta(k)}{\partial \theta_j} = \frac{\partial \eta(k)}{\partial x(k)} \frac{\partial x(k)}{\partial \theta_j} = \left[ \frac{\partial \eta(k)}{\partial x_1} \dots \frac{\partial \eta(k)}{\partial x_{n_a}} \right] \frac{\partial x(k)}{\partial \theta_j} = \eta' \sigma x / \theta_j \end{cases} \quad (2.14)$$

Donc

$$\begin{cases} \sigma x_{(k+1)}/\theta_j = \frac{\partial A}{\partial \theta_j} x(k) + A\sigma_{x/\theta_j} + \frac{\partial B}{\partial \theta_j} u(k) \\ \sigma y/\theta_j = \frac{\partial C}{\partial \theta_j} x(k) + C\sigma_{x/\theta_j} + \frac{\partial D}{\partial \theta_j} u(k) + \frac{\partial F}{\partial \theta_j} \eta(k) + F\eta'\sigma_{x/\theta_j} \end{cases} \quad (2.15)$$

Pour  $j = 1, \dots, n_\theta$ . La valeur optimale de  $\theta$  est déterminée en minimisant le critère  $J$  grâce à l'application de l'algorithme LM. Ce dernier exige le calcul des fonctions de sensibilité, rappelées comme suit :

$$\begin{cases} \sigma x_{(k+1)}/\theta_j = A\sigma_{x/\theta_j} + \begin{bmatrix} \frac{\partial A}{\partial \theta_j} & \frac{\partial B}{\partial \theta_j} & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ \eta \end{bmatrix} \\ \sigma y/\theta_j = (C + F\eta')\sigma_{x/\theta_j} + \begin{bmatrix} \frac{\partial C}{\partial \theta_j} & \frac{\partial D}{\partial \theta_j} & \frac{\partial F}{\partial \theta_j} \end{bmatrix} \begin{bmatrix} x \\ u \\ \eta \end{bmatrix} \end{cases} \quad (2.16)$$

En tenant compte de la forme compagne commandable, on peut reformuler l'équation (2.16) de la manière suivante :

$$\begin{cases} \sigma x_{(k+1)}/\theta_j = A\sigma_{x/\theta_j} + \begin{bmatrix} \frac{\partial A}{\partial \theta_j} & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ \eta \end{bmatrix} \\ \sigma y/\theta_j = (C + F\eta')\sigma_{x/\theta_j} + \begin{bmatrix} \frac{\partial C}{\partial \theta_j} & \frac{\partial D}{\partial \theta_j} & \frac{\partial F}{\partial \theta_j} \end{bmatrix} \begin{bmatrix} x \\ u \\ \eta \end{bmatrix} \end{cases} \quad (2.17)$$

Le calcul du modèle des fonctions de sensibilité (équation 2.17) est effectué à chaque itération. Cela permet de calculer  $J'$  et  $J''$  pour mettre à jour le vecteur de paramètres  $\hat{\theta}$ .

## 2.5 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'identification des systèmes non linéaires, soulignant sa complexité et son importance dans divers domaines. Différentes méthodes d'identification ont été rappelées et discutées. De plus, l'identification des systèmes de Wiener PNLSS utilisant l'algorithme de LM a été considérée.

Le prochain chapitre présentera les résultats de simulations obtenus.

# Chapitre 3

## Résultats de simulation en utilisant l'algorithme Levenberg-Marquardt

### 3.1 Introduction

Le système de Wiener est couramment utilisé pour modéliser des phénomènes physiques complexes, ce qui suscite l'intérêt des chercheurs et ingénieurs. Cependant, identifier précisément ses paramètres reste un défi théorique.

Ce chapitre présente des exemples numériques utilisant l'algorithme LM pour identifier le modèle de Wiener PNLSS, dans des cas entiers. Les simulations incluent  $u(k)$  une entrée aléatoire de moyenne nulle et de variance unitaire et des perturbations  $e(k)$  de bruit blanc de moyenne nulle.

Les résultats des simulations, réalisées pour le cas sans bruit et pour différents niveaux de rapport signal sur bruit (SNR), seront présentés.

### 3.2 Identification par l'algorithme LM

#### 3.2.1 Premier exemple : Ordre linéaire = 2, non linéaire = 2

Un modèle de Wiener décrit sous PNLSS est étudié, en considérant que le bloc linéaire est un modèle d'espace d'état d'ordre  $n_a = 2$ , avec les matrices :

$$A = \begin{bmatrix} 0 & 1 \\ -0.6 & -0.5 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.3 & 0.1 \end{bmatrix}$$

$$D = [0.4]$$

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

$$y(k) = v(k) + 0.7v^2(k) \quad (3.1)$$

Avec  $p_1 = 1$

Alors la forme PNLSS peut se déduire comme suit :

$$\left\{ \begin{array}{l} x(k+1) = \begin{bmatrix} 0 & 1 \\ -0.6 & -0.5 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ v(k) = \begin{bmatrix} 0.3 & 0.1 \end{bmatrix} x(k) + 0.4u(k) + \begin{bmatrix} 0.063 & 0.042 & 0.168 & 0.007 & 0.056 & 0.112 \end{bmatrix} \begin{bmatrix} x_1^2(k) \\ x_1(k)x_2(k) \\ x_1(k)u(k) \\ x_2^2(k) \\ x_2(k)u(k) \\ u^2(k) \end{bmatrix} \end{array} \right.$$

Le vecteur de paramètres  $\theta$  est :

$$\theta = \begin{bmatrix} a_1 & a_2 & c_1 & c_2 & d & p_2 \end{bmatrix}$$

Donc :

$$\theta = \begin{bmatrix} -0.6 & -0.5 & 0.3 & 0.1 & 0.4 & 0.7 \end{bmatrix}$$

Trois valeurs de SNR sont considérées dans cet exemple de simulation :

- SNR = 0 dB : Absence de bruit.
- SNR = 34 dB : Niveau de bruit modéré.
- SNR = 26 dB : Niveau de bruit le plus élevé.

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 en présence de bruit.

On considère les données suivantes :

Nombre d'itérations : 300

Critère d'arrêt :  $\epsilon = 10^{-15}$

$\lambda = 0.01$

Les résultats de la simulation, avec l'algorithme LM pour un cas sans bruit (SNR=0 dB) sont les suivants :

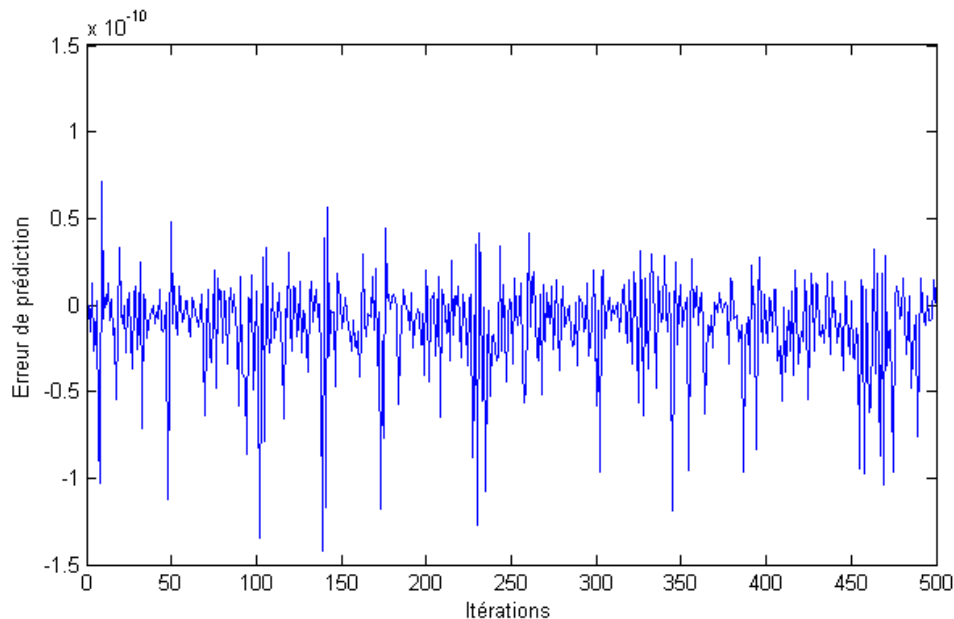


FIGURE 3.1 – Erreur de prédiction (SNR=0 dB)

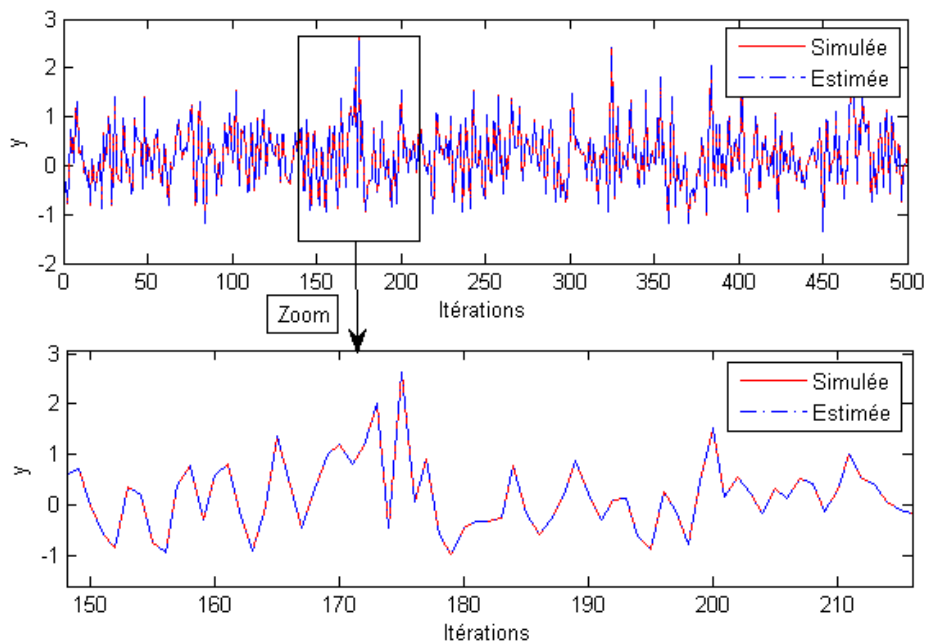


FIGURE 3.2 – Sortie estimée et sortie mesurée (SNR=0 dB)

La figure 3.1, représente l'erreur de prédiction de sortie qui est nulle. En figure 3.2, la sortie estimée recouvre la sortie mesurée leurs tracés montrent une parfaite adéquation.

En présence de bruit, la figure 3.3 et la figure 3.4, montrent respectivement le tracé de l'erreur de prédiction et la sortie estimé par rapport à celle mesurée pour SNR = 34 dB.

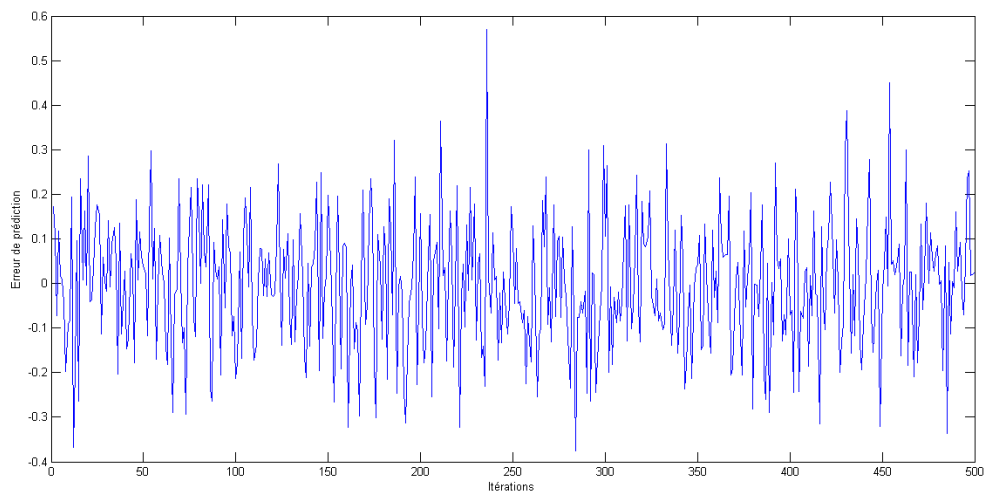


FIGURE 3.3 – Erreur de prédiction (SNR=34 dB)

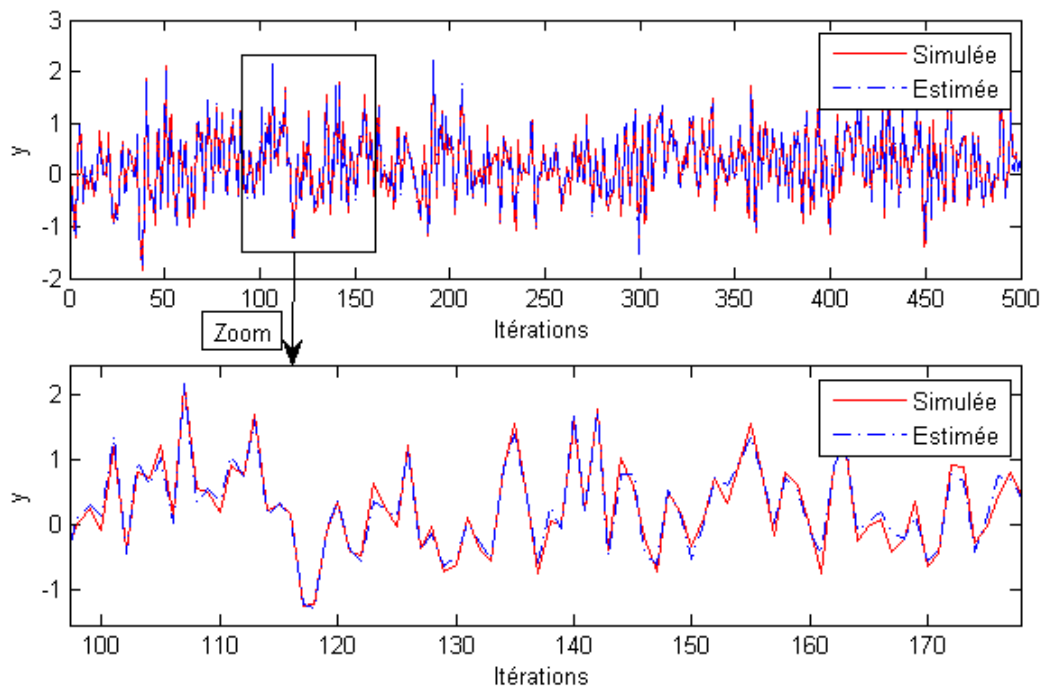


FIGURE 3.4 – Sortie estimée et sortie mesurée (SNR=34 dB)

Pour un cas de bruit plus élevé (SNR=26 dB), l'erreur de prédiction et la sortie estimée par rapport à celle mesurée sont respectivement illustrées par les figures 3.5 et 3.6.

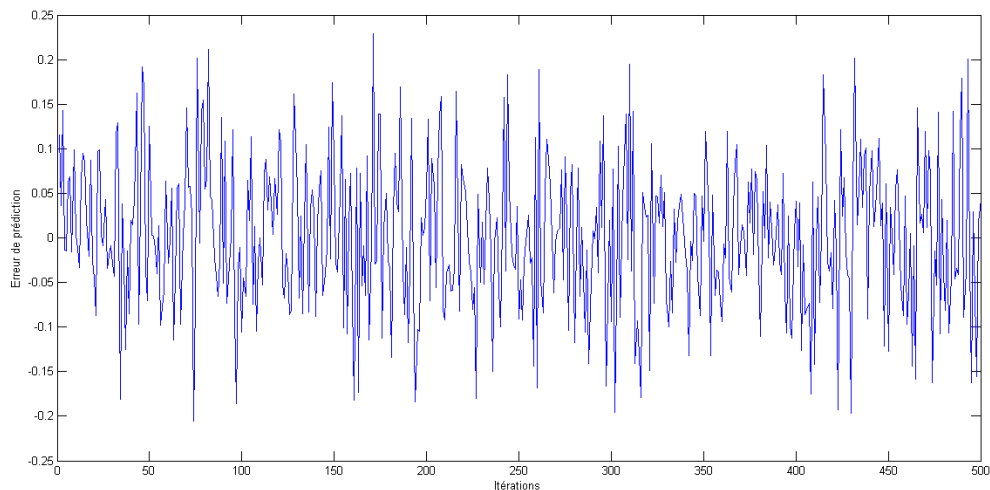


FIGURE 3.5 – Erreur de prédiction (SNR=26 dB)

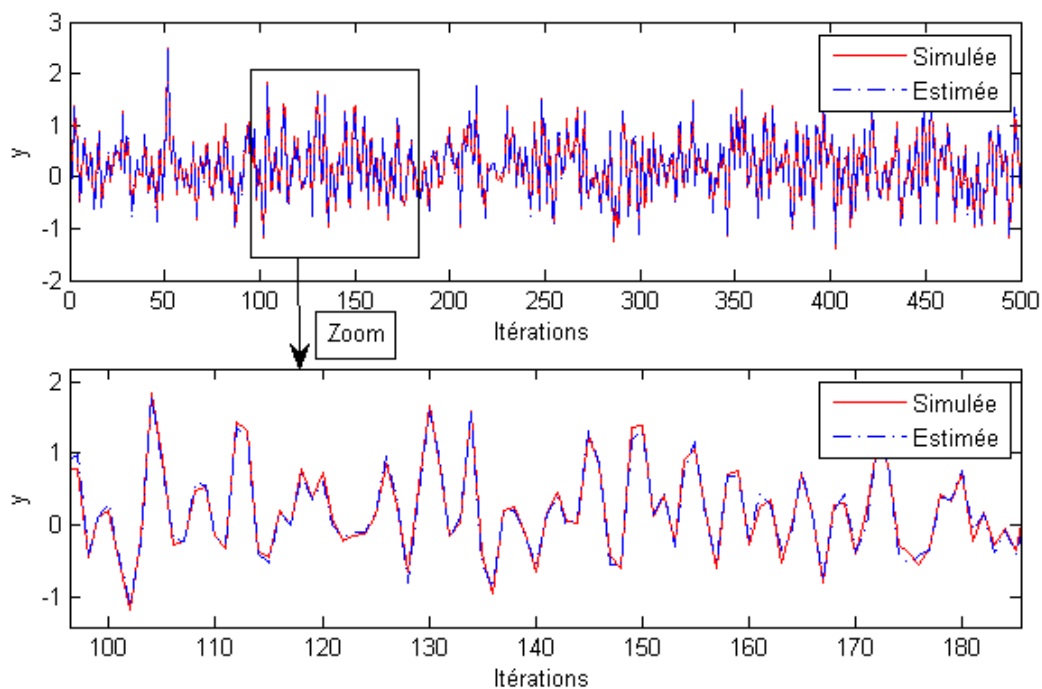


FIGURE 3.6 – Sortie estimée et sortie mesurée (SNR=26 dB)

Le tableau 3.1 montre les résultats de la simulation pour différents SNR. Les résultats trouvés montrent l'efficacité de l'algorithme utilisé.

	Sans bruit	34 dB	26 dB	Valeurs exactes
a <sub>1</sub>	-0.600	-0.605	-0.607	-0.600
a <sub>2</sub>	-0.500	-0.499	-0.470	-0.500
c <sub>1</sub>	0.300	0.097	0.085	0.100
c <sub>2</sub>	0.100	0.294	0.295	0.300
d	0.400	0.398	0.401	0.400
p <sub>2</sub>	0.700	0.726	0.738	0.700
J	9.763 e-22	6.912 e-3	2.048 e-2	-

TABLE 3.1 – Résultats de la simulation pour différents rapports signal sur bruit.

### 3.2.2 Deuxième exemple : Ordre linéaire = 2, non linéaire = 3

Dans cette exemple, on considère un modèle de Wiener l'ordre linéaire  $n_a = 2$  avec les matrices :

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ -0.3 & -0.4 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = [ 0.3 \quad 0.1 ]$$

$$D = [0.2]$$

Le bloc non linéaire est un polynôme de degré  $r = 3$ , défini comme suit :

$$y(k) = v(k) + 0.5v^2(k) + 0.15v^3(k) \quad (3.2)$$

Le modèle Wiener PNLSS peut se déduire sous la forme suivante :

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

Le vecteur de paramètres  $\theta$  est :

$$\theta = [ a_1 \quad a_2 \quad c_1 \quad c_2 \quad d \quad p_2 \quad p_3 ]$$

Donc :

$$\theta = \begin{bmatrix} -0.3 & -0.4 & 0.3 & 0.1 & 0.2 & 0.5 & 0.15 \end{bmatrix}$$

L'exemple de simulation a été testé pour le cas sans bruit et pour différents rapport signal sur bruit.

L'erreur de prédiction de sortie pour le cas sans bruit, illustrée dans la figure 3.7 est nulle ( $J \approx 10^{-22}$ ).

Ainsi, la courbe de la réponse temporelle du système estimé se chevauche avec les données.

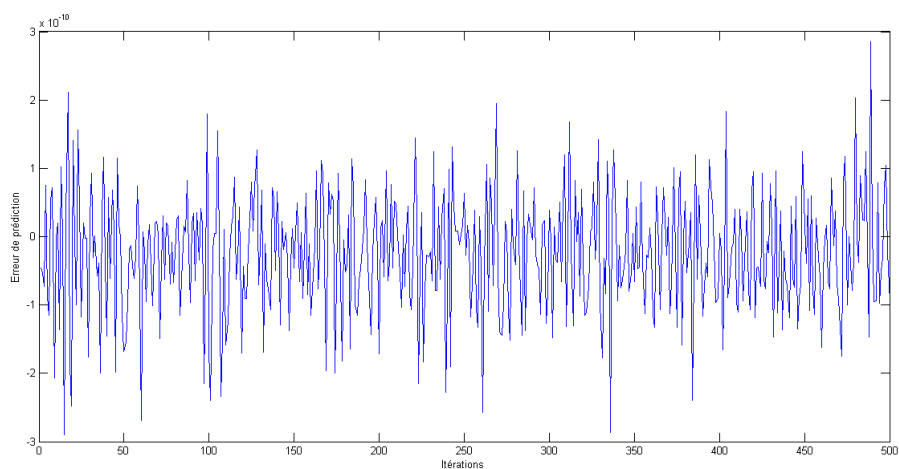


FIGURE 3.7 – Erreur de prédiction (SNR=0 dB)

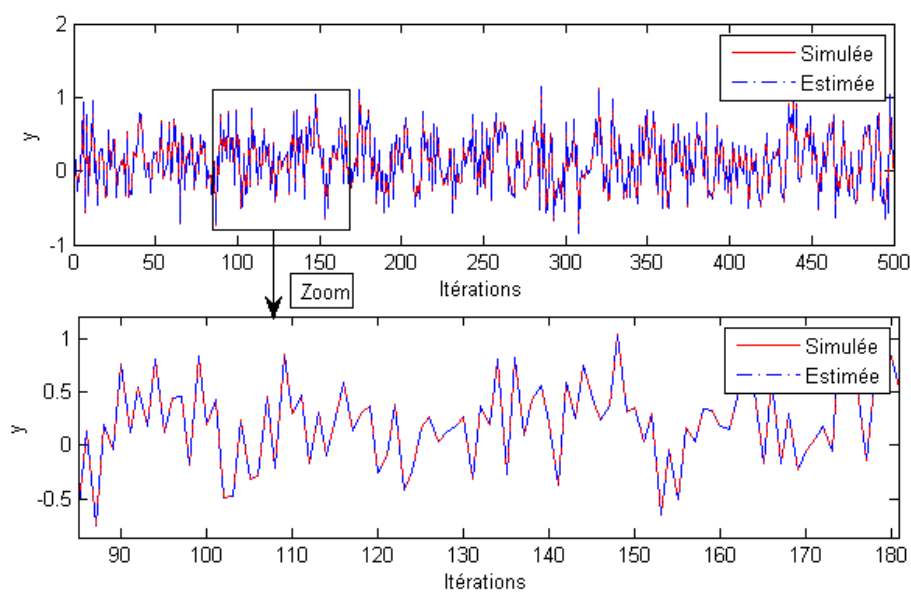


FIGURE 3.8 – Sortie estimée et sortie mesurée (SNR=0 dB)

La figure 3.7 et la figure 3.8 représentent l'erreur de prédiction et la sortie estimée et sortie mesurée au cas sans bruit.

Les résultats de simulation en présence de bruit, à savoir pour un SNR=34 dB et SNR = 26dB sont représentés respectivement dans les figures : 3.9, 3.10, 3.11 et 3.12. Les sorties estimées se chevauchent avec celles simulées et les erreurs sont négligeables.

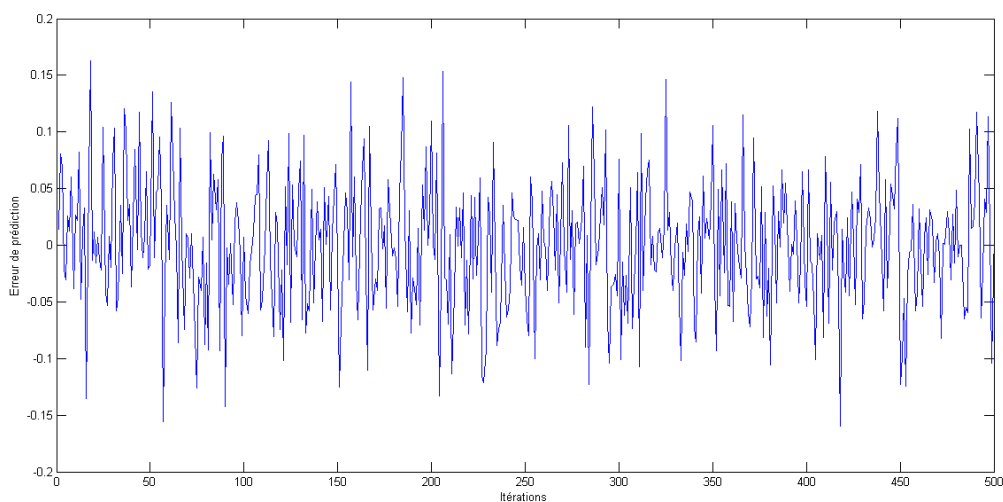


FIGURE 3.9 – Erreur de prédiction (SNR=34 dB)

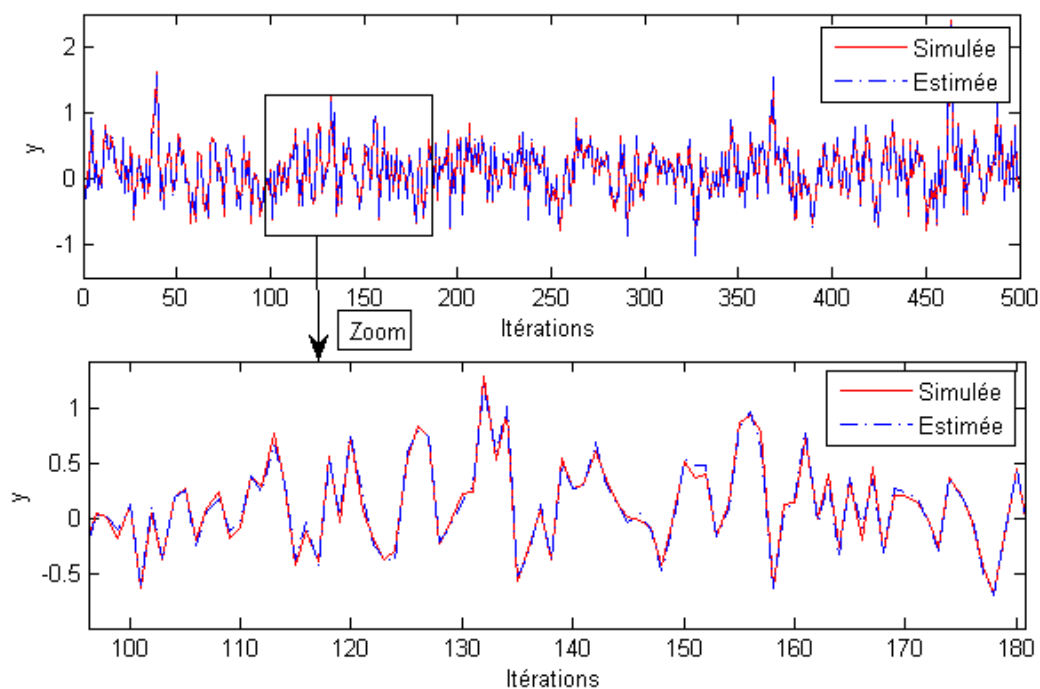


FIGURE 3.10 – Sortie estimée et sortie mesurée (SNR=34 dB)

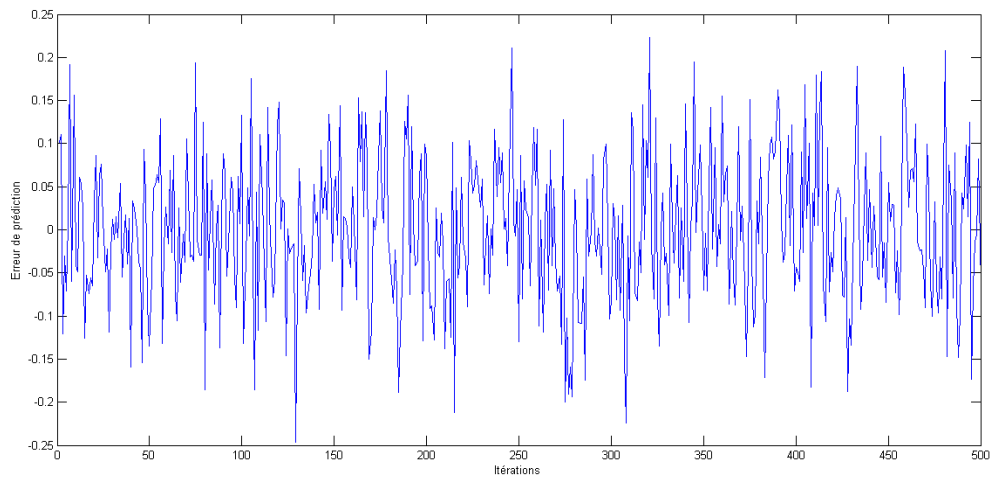


FIGURE 3.11 – Erreur de prédiction (SNR=26 dB)

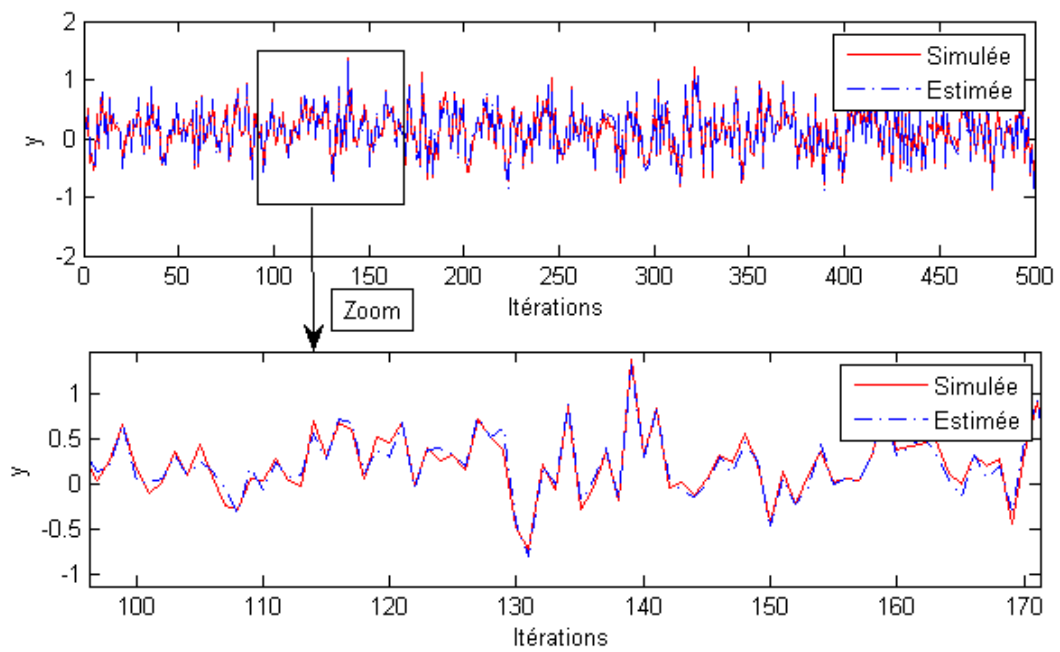


FIGURE 3.12 – Sortie estimée et sortie mesurée (SNR=26 dB)

Le tableau 3.2 montre les résultats de la simulation pour différents rapports signal sur bruit.

	Sans bruit	34 dB	26 dB	Valeurs exactes
$a_1$	-0.300	-0.291	-0.31126	-0.300
$a_2$	-0.400	-0.378	-0.411	-0.400
$c_1$	0.300	0.294	0.300	0.300
$c_2$	0.100	0.103	0.100	0.100
$d$	0.200	0.20125	0.196	0.200
$p_2$	0.500	0.515	0.481	0.500
$p_3$	0.150	0.164	0.155	0.150
J	2.565 e-22	3.225 e-3	7.1761 e-3	-

TABLE 3.2 – Résultats de la simulation pour différents rapports signal sur bruit.

Ces résultats valident la bonne performance de la méthode proposée.

### 3.3 Conclusion

Dans ce chapitre, nous avons exploré l'identification des systèmes de Wiener basée sur la représentation d'état non linéaire polynomiale à l'aide de l'algorithme de LM. Les résultats de simulations des exemples montrent que l'algorithme LM peut estimer avec précision les paramètres du système, même en présence de bruit modéré.

En conclusion, l'algorithme LM s'est révélé être une méthode fiable et efficace pour l'identification des systèmes de Wiener, malgré les défis posés par le bruit.

# Conclusion générale

Dans ce mémoire, notre travail se porte sur une étude des systèmes non linéaires, en se basant particulièrement sur les systèmes de type Wiener. Ce type de système est largement utilisé pour modéliser des phénomènes complexes dans divers domaines scientifiques et industriels.

Notre travail ouvre la porte pour explorer les défis associés à la description et l'identification de ces systèmes, ainsi que la méthode spécifique développée à utiliser. Cependant, ce mémoire est composé de trois chapitres.

Dans le chapitre I, nous avons introduit les bases théoriques des systèmes non linéaires, soulignant leur importance croissante dans la modélisation de processus complexes. Après avoir présenté les principales catégories de ces systèmes, nous nous sommes basés sur le modèle de Wiener, qui constitue le pilier de notre étude. Ce modèle, décrit par une combinaison d'un modèle d'état linéaire et d'une non-linéarité polynomiale, permettant de le représenter selon l'approche PNLSS, qui servira de base aux développements ultérieurs.

Le chapitre II, s'intéresse aux techniques d'identification des systèmes non linéaires. Ce chapitre a fait l'objet d'un rappel théorique de base sur l'identification, permettant d'obtenir un modèle mathématique précis à partir des données d'entrée et de sortie. Ainsi que les méthodes pour minimiser l'impact des bruits et des perturbations.

Le chapitre III a présenté les résultats des simulations utilisant l'algorithme LM pour l'identification des systèmes de Wiener. Les simulations ont démontré que l'algorithme LM est efficace pour estimer les paramètres des systèmes non linéaires, même en présence d'un pourcentage de bruit important. Ces résultats confirment la fiabilité de l'approche proposée et son utilité pour l'identification de systèmes complexes.

En résumé, ce mémoire a permis de faire progresser la compréhension et la description des systèmes non linéaire de Wiener et de proposer une méthode fiable pour leur identification, malgré les défis posés par la non-linéarité et le bruit.

Les travaux réalisés constituent une contribution significative à la recherche dans ce domaine et ouvrent la voie à de futures perspectives, notamment :

- Généralisation aux cas fractionnaire
- Identification des systèmes de Wiener sous autres formes
- Extension aux cas multi variables

# Bibliographie

- [1] LJUNG, L. System Identification : Theory for the User. 1987.
- [2] ORJUELA, R. Contribution à l'estimation d'état et au diagnostic des systèmes représentés par des multimodèles. 2009.
- [3] HAMMAR, K. Identification des systèmes non-linéaires à structure Wiener, Hammerstein, application au cas fractionnaire. 2015.
- [4] NELLES, O. Nonlinear System Identification : From Classical Approaches to Neural Networks and Fuzzy Models. Springer, New York, Berlin, Heidelberg. 2001.
- [5] LEMMOU, A ; KHAOUKHA, A ; LEDJEDEL, A. Identification et commande des systèmes non linéaires. Mémoire d'ingénieur en électronique, Université de M'sila, Algérie. 2011.
- [6] SERSOUR, L. Identification des systèmes fractionnaires non linéaires. Thèse de doctorat. 2018.
- [7] GUERMAH, S. Commande CRONE, application à un procédé pilote de laboratoire (CRONE control application to a pilot laboratory process). Université Mouloud Mammerie Tizi Ouzou. 2011.
- [8] NAGY, A.M. Analyse et synthèse de multimodèles pour le diagnostic. Application à une station d'épuration. Ecole doctorale IAEM Lorraine, Département de Formation Doctorale en Automatique. 2010.
- [9] BALAJI, N. ; SHUQING, L ; SCHEEL, M ; BRAKE, M.R.W ; TISO, P ; NOËL, J-P ; KRACK, M. Numerical Assessment of Polynomial Nonlinear State-Space and Nonlinear-Mode Models for Near-Resonant Vibrations. 2020.

- [10] PADUART, J. Identification of nonlinear systems using Polynomial Nonlinear State Space models. Ph.D. thesis, Vrije Universiteit Brussel. 2008.
- [11] VAN MULDER, A ; VOLCKAERT, M ; DIEHL, M ; SCHOUKNES, J. Two nonlinear optimization methods for black box identification compared. In Proc. 15th IFAC Symposium on System Identification, SYSID09, pp. 1086-1091. 2009.
- [12] PADUART, J ; LAUWERS, L ; PINTELON, L ; SCHOUKENS, R. Identification of a Wiener Hammerstein system using the polynomial nonlinear state space approach. In Proc. 15th IFAC Symposium on System Identification, SYSID 2009, pp. 1080-1085.
- [13] PADUART, J ; LAUWERS, L ; PINTELON, L ; SCHOUKENS, R. Identification of a Wiener Hammerstein system using the polynomial nonlinear state space approach. Control Engineering Practice, 20, 1133-1139. 2012.
- [14] SERSOUR, L ; DJAMAH, T ; BETTAYEB, M. Wiener System Identification using Polynomial Non Linear State Space Model. In The 3rd International Conference on Control, Engineering and Information Technology : CEIT'2015, Tlemcen, Algeria, May 2015.
- [15] SERSOUR, L ; DJAMAH, T ; BETTAYEB, M. Wiener system Identification using Self-Adaptive Velocity Particle Swarm Optimization. In International Conference on Automatic Control, Telecommunications and Signals : ICATS'2015 , Annaba, Algeria, November 2015.
- [16] CABODEVILA, G. Identification des systèmes, École Nationale Supérieure de Mécanique et des Microtechniques, 26 chemin de l'Épitaphe, 25030 Besançon Cedex, France.
- [17] DJEGHLOUD, H. Module : MODÉLISATION ET IDENTIFICATION. Université des frères Mentouri Constantine Faculté des Sciences de la Technologie Département ELT DES SYSTÈMES ÉLECTRIQUES. 2020.
- [18] TRIKI, A, Cours : Modélisation des processus, Université de Mouloud Mammeri de Tizi Ouzou Département d'Automatique.

- [19] RAGUE, M ; DIAKITE, A ; COMMENGES, D. Package ‘marqLevAlg’ - Algorithme de Marquardt-Levenberg en R : Une alternative à ‘optimx’ pour des problèmes de minimisation. ISPED, Université Bordeaux Segalen, INSERM U897 Épidémiologie et Biostatistique. Rencontres R Bordeaux - 2/3 Juillet 2012.
- [20] KENOUCHE, S. Algorithmes d’optimisation : Application à des fonctions-objectifs non linéaires. Département des Sciences de la Matière, UMKB.
- [21] RONDEPIERRE, A. Méthodes numériques pour l’optimisation non linéaire déterministe. 2017-2018.
- [22] NOUAOURIA, N. Une Approche d’Optimisation par Essaim de Particules pour la Recherche en Mémoire de Cas. Thèse de doctorat en Informatique Cognitive. Université du Québec à Montréal. 2013.
- [23] GHARIB, A . BENHRA, J. Tuning manuel de l’algorithme d’optimisation par essais de particules appliqué au problème de voyageur de commerce. Xème Conférence Internationale : Conception et Production Intégrées, 2015.
- [24] ACHOURI, S. MÉMOIRE DE MASTER sur le thème de ”Etude de l’Identification d’un Système d’Ordre Fractionnaire” Université Mohamed Khider de Biskra, Faculté des Sciences et de la Technologie, Département de Génie Électrique. 2020.
- [25] BELLOUMI, R. Mémoire sur le thème de « Optimisation par Essaim de Particules Application à un système complexe » Université Batna 2 Faculté de Technologie Département d’Électronique.
- [26] BOUGUERRA, I; KHALFAOUI, I. Mémoire sur le thème de «Identification paramétrique basée sur la prédiction et application au signal de parole» Université SAAD DAHLAB de BLIDA, Faculté de Technologie, Département d’Électronique. 2016-2017.
- [27] BOUFATAH, S. Mémoire de fin d’étude sur le thème «Méthodes d’identification paramétrique des structures mécaniques par mesures vibratoires» UNIVERSITÉ MOULOUD MAMMERI DE TIZI-OUZOU FACULTÉ DU GÉNIE DE LA CONSTRUCTION DÉPARTEMENT DE GÉNIE MÉCANIQUE. 2013.

- [28] MELAKHESSOU, La. Contrôle et identification des systèmes non linéaires par les techniques neuronales. Thèse de doctorat, Université El-Hadj Lakhdar - Batna.
  
- [29] THOUVEREZ, F. Identification temporelle des structures non linéaires. Thèse de doctorat, École Centrale de Lyon, 16 décembre 1992.
  
- [30] BUTTINO, J. Identification de systèmes dynamiques non linéaires à l'aide de modèles de Volterra. Année académique 2017-2018.
  
- [31] ORJUELA, R ; MAQUIN, D ; RAGOT, J. Identification des systèmes non linéaires par une approche multi-modèle à états découplés. Centre de Recherche en Automatique de Nancy UMR 7039, Nancy-Université, CNRS. 2006.