

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 Professionnel

Domaine : Sciences et Technologies

Filière : Automatique

Spécialité : Automatique Industrielle

Présenté par

HADDAD YACINE

HADDAG BELKACEM

Thème

Identification des systèmes blocs orientés

Mémoire soutenu publiquement le 26/ 09/ 2024 devant le jury composé de :

M GHANIA IDIRI UMMTO

Grade, Lieu d'exercice, **Président**

M LAMIA SARSOUR UMMTO

Grade, Lieu d'exercice, Encadrant

M SARAH KASSIM UMMTO

Grade, Lieu d'exercice, Examineur

Remerciements

Nous commençons par remercier le bon Dieu de nous avoir donné la santé, le courage et la volonté pour finir ce modeste travail.

Nous tenons à remercier en premier lieu notre promoteur Madame L. SERSOUR pour son aide, de nous avoir fait profiter de son expérience et de nous avoir encouragés tout au long de ce travail. On la remercie sincèrement pour ses conseils, sa patience et sa disponibilité tout au long de notre projet.

Nous remercions également les membres de jury qui feront l'honneur de juger notre travail, d'apporter leurs réflexions et suggestions scientifiques.

Nous exprimons nos sincères remerciements à nos parents qui nous ont enseigné la patience, la politesse, le sacrifice et qui ont toujours été là pour nous.

Enfin, nos remerciements les plus sincères à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Table des matières :

Introduction générale	1
Chapitre 1 : Les système non linéaire.....	3
1.1 Introduction	4
1.2 Structure des modèles non linéaires	4
1.2.1 Modèle non linéaire à réponse impulsionnelle finie	4
1.2.2 Modèle non linéaire Autorégressif à entrée exogène	5
1.2.3 Modèle non linéaire autorégressif à moyenne ajustée et à entrée exogène.....	6
1.2.4 Modèle non linéaire Box-Jenkins.....	6
1.2.5 Modèle non linéaire des fonctions de base orthonormées.....	7
1.2.6 Modèle des séries de Volterra	8
1.2.7 Modèles blocs orientées	9
1.2.7.1 Modèle de Hammerstein	9
1.2.7.2 Modèle de Wiener	10
1.2.7.3 Modèle Wiener-Hammerstein	11
1.2.7.4 Modèle Hammerstein-Wiener	11
1.3 Représentation de l'espace d'état non linéaire polynomiale	12
1.4 Description des modèles de type Wiener	13
1.5 Conclusion	17
Chapitre 2 : Identification des système bloc orientés	18
2.1 Introduction	19
2.2 Généralité sur l'identification	19
2.2.1 Définition	19
2.2.2 Etape de l'identification	20
2.3 Méthodes d'identifications	22
2.3.1 Méthode Identification en temps continu et en temps discret.....	22
• Identification en Temps Continu.....	22
• Identification en Temps Discret	23
• Comparaison des Approches	23

Table des matière

2.3.2	Méthode de base	24
•	Relation entrée, sortie et système	24
•	Méthode graphique directe	25
•	Méthode de Broïda	27
•	Méthode de Strecj	28
2.3.3	Algorithme à erreur d'équation	29
2.3.3.1	Méthode des moindres carrés	30
•	Moindres Carrés Pondérés	32
•	Moindres Carrés Récursifs	32
2.3.4	Méthode a erreur de sortie.....	33
2.3.4.1	Méthode du gradient.....	35
2.3.4.2	Méthode de gausse newton	35
2.3.4.3	Méthode de Levenberg Marquardt	36
2.3.5	Méthode d'optimisation par essais des particules	37
2.4	Identification des systèmes bloc orientés	38
2.5	Conclusion	40
Chapitre 3 : Résultat de simulation et discussion.....		14
3.1	Introduction	42
3.2	Identification par l'algorithme PSO	42
3.2.1	Exemple 1 :.....	42
3.2.2	Exemple 2 :.....	48
3.3	Conclusion :.....	53
Conclusion générale.....		53

Notations

- **Abbreviation**

NARX: Nonlinear Auto Regressive with External input

NFIR: Nonlinear Finite Impulse-Response

NOE: Nonlinear Output Error

NARMAX: Nonlinear Auto-Regressive Moving Average with Exogeneous Input

NARX: Nonlinear Autoregressive with Exogenous input

AR : modèle autorégressif

FIR : Finite Impulse Response

BJ: Box-Jenkins

OBF: Orthonormal Basis function models

PSO : Particle Swarm Optimisation

• Symboles

$\varphi(k)$: Le vecteur de régression

θ : Le vecteur des paramètres du modèle

$A(q)$: Un polynôme d'ordre n_a

$B(q)$: Un polynôme d'ordre n_b

$x(k)$: Vecteur d'état,

$u(k)$: Vecteur d'entrée

$y(k)$: Vecteur de sortie

$e(k)$: Vecteur de bruit

$f(\cdot)$: Fonction non linéaire

k : Variable de temps discret

n_l : Ordre de la partie linéaire

n_b : Longueur du vecteur b

r : Ordre de la partie non linéaire

$f(\cdot)$: Fonction non linéaire

p_l : Coefficient de la fonction non linéaire

Table des Figures :

Figure 1.1:Modèle de Hammerstein.....	10
Figure 1.2 : Modèle de Wiener.....	10
Figure 1.3:Modèle Wiener-Hammerstein.	11
Figure 1.4: Modèle de Hammerstein-Wiener.....	11
Figure 1.5: Modèle de Wiener PNLSS.....	13
Figure 2.1: Identification à partir du comportement d'entre-sortie.....	20
Figure 2.2: Identification de type paramétrique.	20
Figure 2.3: Procédure d'identification d'un modèle de système.....	21
Figure 2.4: Simulation d'un système 1er ordre soumis à un échelon.	25
Figure 2.5: Simulation d'un système 1er ordre soumis à une rampe.	26
Figure 2.6 : Réponse à un échelon d'un système du second ordre – régime pseudopériodique	27
Figure 2.7: Réponse indicielle du 1er ordre avec retard.	27
Figure 2.8: Réponse indicielle du 1er ordre.	28
Figure 2.9 : Structure erreur d'équation.	29
Figure 2.10:Principe de la méthode a erreur de sortie.....	34
Figure 3.1: Résultats de simulation avec l'algorithme PSO pour SNR=0 dB (sans bruit).	46
Figure 3.2 : Résultats de simulation avec l'algorithme PSO pour SNR=34dB.....	47
Figure 3.3: Résultats de simulation avec l'algorithme PSO pour SNR=26dB.....	47
Figure 3.4 : Résultats de simulation avec l'algorithme PSO pour SNR= 0 dB.....	50
Figure 3.5: Résultats de simulation avec l'algorithme PSO pour SNR=34 dB.....	51
Figure 3.6: Résultats de simulation avec l'algorithme PSO pour SNR=26dB.....	52

Liste des Tableaux :

Tableau 2.1: Paramètres de Strecj pour différents ordre n :.....	28
Tableau 3.1: Résultats d'identification du modèle de Wiener PNLSS avec PSO pour $n_a = 2$ et $r = 2$	43
Tableau 3.2: Résultats d'identification du modèle de Wiener PNLSS avec PSO pour $n_a = 2$ et $r = 3$	48

Introduction générale

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

Pour décrire au mieux notre contribution, la progression du mémoire est ponctuée par trois chapitres dont le contenu est présenté ici d'une manière introductive.

Dans le premier chapitre, les généralités sur les structures non-linéaires sont présentées.

Nous nous sommes intéressés plus particulièrement aux blocs structurés de type 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) [5].

Le deuxième chapitre est consacré à l'identification des systèmes blocs orientés. Après une généralité sur l'identification, on s'est focalisée sur les méthodes d'identification les plus utilisées. A savoir, les méthodes de base, en temps continu et en temps discret, l'algorithme à erreur d'équation et la méthode d'erreur de sortie. Ainsi que la Méthode d'optimisation par essaim de particules (PSO). La fin de ce chapitre est consacrée à l'identification des systèmes bloc orientés.

Le dernier chapitre se concentrera sur l'application de l'algorithme PSO sur un modèle bloc orienté de type Wiener, développés au deuxième chapitre, dans différents exemples de simulation. Nous illustrerons ainsi les résultats de simulation obtenus, permettant de tester et de vérifier l'efficacité de l'algorithme.

Introduction générale

Enfin, une conclusion générale résume l'étude effectuée en discutant les résultats obtenus et suggère des perspectives futures pour améliorer davantage notre contribution.

Chapitre 1

Les Systèmes non linéaire

1.1 Introduction

L'identification des systèmes non linéaires est un défi majeur en ingénierie et en automatique. Cependant, la plupart des systèmes réels présentent des dynamiques complexes qui ne peuvent pas être capturées par des modèles linéaires classiques. Tels que les interactions complexes entre composants. Par conséquent, les modèles non linéaires jouent un rôle essentiel dans la compréhension et la prédiction du comportement des systèmes complexes. Cependant, les modèles non linéaires sont souvent utilisés pour simplifier l'analyse.

On se limitera dans ce chapitre à l'étude des systèmes non linéaires et de présenter les structures des modelés non linéaires.

1.2 Structure des modèles non linéaires

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

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

$f(\cdot)$ 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.

L'équation (1.1) représente la structure la plus générale et selon le contenu du vecteur de régression $\phi(k)$ plusieurs structures peuvent être définies telles que NARMAX, NARX, NFIR, NBJ...

1.2.1 Modèle non linéaire à réponse impulsionnelle finie

Modèle non linéaire à réponse impulsionnelle finie (Nonlinear Finite Impulse Response, (NFIR)) est un concept important pour comprendre et modéliser les systèmes non linéaires, en particulier dans le domaine du traitement du signal et de la communication. Il permet de capturer des comportements complexes qui ne peuvent pas être expliqués par des modèles linéaires simples.

Dans les modèles à réponse impulsionnelle finie (Finite Impulse Response (FIR)) et NFIR le répresseur est seulement en fonction de $u(k - i)$.

Le modèle FIR peut être décrit par [6] :

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

Avec $B(q)$ 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), par conséquent, 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 non linéaire Autorégressif à entrée exogène

C'est un modèle utilisé pour modéliser des séries temporelles complexes. Il combine des valeurs passées de la série temporelle d'intérêt avec des entrées exogènes pour capturer des relations non linéaires et prédire les valeurs futures.

Le modèle autorégressif à entrée exogène (Autoregressive with exogenous input (ARX)) est donné comme suit [6] :

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

$A(q)$ un polynôme d'ordre n_A et $A(q) = 1 + a_1q^{-1} + \dots + a_{n_A}q^{-n_A}$.

Ainsi, la sortie du modèle est représentée comme suit :

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

Le modèle non linéaire autorégressif a entre exogène (Nonlinear Autoregressive with Exogenous input (NARX)) peut aussi être déduit en remplaçant la relation linéaire dans l'équation (1.6) par une fonction non linéaire, comme suit :

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

f est une fonction non linéaire.

1.2.3 Modèle non linéaire autorégressif à moyenne ajustée et à entrée exogène

Dans le cas linéaire, la sortie du modèle non linéaire autorégressif à moyenne ajustée et à entrée exogène (Autoregressive Moving Average Model with eXogenous input (ARMAX)) est représentée par

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

$M(q)$ un polynôme d'ordre n_M .

Le modèle Nonlinear Autoregressive Moving Average Model with eXogenous Inputs (NARMAX), est un modèle utilisé pour prédire et contrôler les systèmes complexes, en particulier dans le domaine des séries temporelles. Il combine des valeurs passées de la série temporelle d'intérêt avec des entrées exogènes pour capturer des relations non linéaires et prévoir le comportement.

Ce type de modèle permet de représenter une grande classe de systèmes non linéaires [2], et peut être défini par l'équation (1.9) : [6]

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

$f(\cdot)$ est une fonction non-linéaire.

1.2.4 Modèle non linéaire Box-Jenkins

Le modèle Box-Jenkins (BJ) fournit une autre forme en multipliant le bruit par $M(p)/N(p)$ [6]. 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.

Avec par exemple :

$$\left\{ \begin{array}{l} B(q) = b_1 q^{-1} + \dots + b_{n_B} q^{-n_B} \\ H(q) = 1 + h_1 q^{-1} + \dots + h_{n_H} q^{-n_H} \\ M(q) = 1 + m_1 q^{-1} + \dots + m_{n_M} q^{-n_M} \\ N(q) = 1 + n_1 q^{-1} + \dots + n_{n_N} q^{-n_N} \end{array} \right. \quad (1.11)$$

Le modèle Nonlinear Box-Jenkins (NBJ) peut être déduit en remplaçant la relation linéaire dans l'équation (1.10) par une fonction non linéaire, comme suit [1] :

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

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 Modèle non linéaire des fonctions de base orthonormées

Les modèles non linéaires des fonctions de base orthonormées (Orthonormal Basis function models (OBF)) sont des approches courantes dans divers domaines, notamment en statistiques et en traitement du signal. Ces modèles permettent de modéliser des relations complexes entre les variables d'entrée et de sortie sans nécessairement spécifier une forme fonctionnelle explicite. 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 Finite Impulse Response (FIR), 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 $l_i(q)$ [6].

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 Nonlinear Orthonormal Basis function models (NOBF) a la forme [3] :

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

Par exemple, le $i^{\text{ème}}$ filtre de Laguerre est donné par :

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

g Un paramètre de valeur réelle et $|g| < 1$.

Alternativement, le $i^{\text{ème}}$ filtre de Laguerre peut être calculé de manière récursive comme suit :

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

1.2.6 Modèle des séries de Volterra

C'est un modèle pour le comportement non linéaire, similaire à la série de Taylor. Cependant, elles diffèrent de la série de Taylor en leur capacité à capturer les effets de mémoire. Alors que la série de Taylor peut être utilisée pour approximer la réponse d'un système non linéaire à une entrée donnée si la sortie du système dépend strictement de l'entrée à ce moment précis, les séries de Volterra permettent à la sortie du système non linéaire de dépendre de l'entrée à tous les autres moments. Cela permet de capturer l'effet de mémoire des dispositifs tels que les condensateurs et les inductances [7].

Afin d'arriver à l'expression de la sortie des séries de Volterra, considérons un système linéaire caractérisé par sa réponse impulsionnelle h_i et un signal d'entrée $u(k)$.

La sortie $y_1(k)$ est exprimée comme le produit de convolution :

$$y_1(k) = \sum_0^{+\infty} h_1(\tau_1) u(k - \tau_1) \quad (1.16)$$

h_i est considéré le 1-ème noyau de Volterra.

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

$$y_2(k) = \sum_0^{+\infty} \sum_0^{+\infty} h_2(\tau_1, \tau_2) u(k - \tau_1) u(k - \tau_2) \quad (1.17)$$

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

$$y(k) = \sum_{k=1}^{+\infty} y_K(k) = y_1(k) + y_2(k) + \dots + y_K(k) + \dots \quad (1.18)$$

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 :

$$y(k)_K = \sum_0^{+\infty} \dots \sum_0^{+\infty} h_K(\tau_1, \dots, \tau_K) \prod_{i=1}^K u(k - \tau_i) \quad (1.19)$$

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 :

$$y(k) = \sum_{K=1}^{+\infty} \sum_0^{+\infty} \dots \sum_0^{+\infty} h_K(\tau_1, \dots, \tau_K) \prod_{i=1}^K u(k - \tau_i) \quad (1.20)$$

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 [8].

1.2.7 Modèles blocs orientées

Les modèles de blocs orientés sont utilisés dans divers domaines, notamment en informatique, en mathématiques et en ingénierie, pour représenter des structures orientées telles que des réseaux et des circuits électriques ...etc. Ces modèles sont des abstractions qui permettent de comprendre et de résoudre des problèmes complexes en les décomposant en éléments plus simples appelés blocs.

Les modèles blocs orientés se décomposent en plusieurs modèles :

1.2.7.1 Modèle de Hammerstein

Le modèle Hammerstein a été introduit en 1930 par le mathématicien allemand A. Hammerstein. Il est un des plus simples et des plus connus des modèles non linéaires Blocs orientés [7].

- **Structure du modèle :**

Ce type de modèle fait partie des modèles à base de blocs structurés. Il est constitué par un

élément non linéaire statique suivi par un élément dynamique linéaire [3].

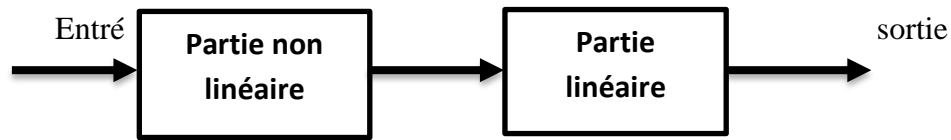


FIGURE1.1:Modèle de Hammerstein.

- **Domaines d'utilisation**

Les modèles de Hammerstein sont utilisés dans divers domaines tel que le traitement du signal, la modélisation de systèmes dynamiques [9] et le contrôle des processus [10]. Ils offrent une bonne balance entre la simplicité des modèles linéaires et la capacité à capturer les effets non linéaires des systèmes réels.

1.2.7.2 Modèle de Wiener

Le modèle Wiener a été étudié pour la première fois par N. Wiener en 1958 [11].

La structure du modèle Wiener (L-NL) représenté en figure 1.2 peut être obtenue en permutant les éléments linéaire et non linéaire dans le modèle Hammerstein [12].

- **Structure du modèle :**

Ce type de modèle est également à base de blocs structurés. Il est constitué par un élément dynamique linéaire suivi par un élément non-linéarité statique [13].

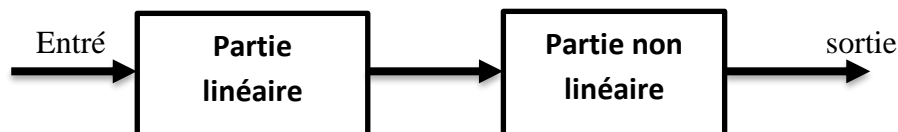


FIGURE1.2 : Modèle de Wiener.

- **Domaines d'utilisation**

Les modèles de Wiener sont utilisés dans divers domaines tels que les systèmes de contrôle du pH [14,15], colonne de distillation [16] et le contrôle des processus [9]. Ils sont particulièrement adaptés aux systèmes où les effets non linéaires dominent mais où une composante linéaire est également présente.

1.2.7.3 Modèle Wiener-Hammerstein

La combinaison en série d'un modèle Wiener et d'un modèle Hammerstein donne un nouveau modèle appelé Wiener- Hammerstein- illustré sur la figure (1.3) [17].

- **Structure du modèle :**

Le modèle de Hammerstein-Wiener est constitué d'un bloc linéaire dynamique en série avec un bloc non linéaire statique suivie d'un block linéaire dynamique [17].



FIGURE 1.3:Modèle Wiener-Hammerstein.

- **Domaines d'utilisation**

Le modèle de Wiener-Hammerstein est utilisé pour modéliser des systèmes biologiques et physiologiques complexes, tels que les systèmes neurologiques, cardiovasculaires, et musculosquelettiques, où les interactions entre les composantes linéaires et non linéaires sont importantes [10].

1.2.7.4 Modèle Hammerstein-Wiener

La combinaison en série d'un modèle Hammerstein et d'un modèle Wiener donne un nouveau modèle appelé Hammerstein-Wiener illustré sur la figure (1.4) [7].

- **Structure du modèle :**

Le modèle de Hammerstein-Wiener est constitué d'un bloc non linéaire statique en série avec un bloc linéaire dynamique suivi d'un block non linéaire dynamique [17].



FIGURE 1.4: Modèle de Hammerstein-Wiener.

- **Domaines d'utilisation**

Le modèle de Hammerstein-Wiener est utilisé pour modéliser des processus industriels complexes présentant à la fois des non-linéarités précédant et suivant des éléments linéaires, tels que les réacteurs chimiques, les turbines, les systèmes de chauffage et de refroidissement, etc. [10].

1.3 Représentation de l'espace d'état non linéaire polynomiale

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

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

Où :

- $x(k)$ est le vecteur d'état,
- $u(k)$ est le vecteur d'entrée,
- $y(k)$ est le vecteur de sortie,
- f et g sont des fonctions non linéaires qui peuvent être développées en séries polynomiales.

La représentation d'état non linéaire polynomiale (PNLSS) est une approche utilisée pour modéliser des systèmes dynamiques non linéaires. Cette méthode étend la représentation d'état classique, qui est généralement linéaire, en intégrant des termes non linéaires dans les équations d'évolution et d'observation.

Ce modèle est défini comme suit [18] :

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

Les coefficients du sous-ensemble linéaire sont donnés par les matrices $A \in R^{n_a \times n_a}$ et matrices $B \in R^{n_a \times n_a}$ dans l'équation d'état, $C \in R^{n_a \times n_a}$ et $D \in R^{n_a \times n_a}$ dans l'équation de sortie.

Chapitre 1 : Les système non linéaire

Les vecteurs $\zeta(t) \in R^{n_\zeta}$ et $\eta(t) \in R^{n_\eta}$ 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 $E \in R^{n_a * n_\zeta}$ et $F \in R^{n_a * n_\eta}$.

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 des modèles de type Wiener

Dans cette partie, nous allons décrire le modèle de Wiener, en considérant que la partie dynamique linéaire du modèle est choisi comme un modèle de l'espace d'état à temps discret d'ordre n_a (équation (1.23)). Le bloc statique non linéaire est défini par un polynôme non linéaire d'ordre connu r (équations (1.24))

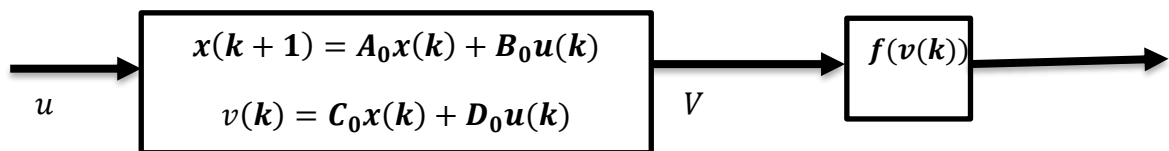


FIGURE 1.5: Modèle de Wiener PNLSS.

Avec :

$$\begin{cases} x(k+1) = A_0x(k) + B_0u(k) \\ v(k) = C_0x(k) + D_0u(k) \end{cases} \quad (1.23)$$

$$f(v(k)) = p_1v^1(k) + p_2v^2(k) + \dots + p_rv^r(k) \quad (1.24)$$

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

En remplaçant la deuxième équation de (1.23) dans (1.25), nous avons la sortie du modèle de Wiener est donnée par :

$$\begin{cases} x(k+1) = A_0x(k) + B_0u(k) \\ v(k) = C_0x(k) + D_0u(k) \\ y(k) = \sum_{i=1}^r p_i (C_0x(k) + D_0u(k))^i \end{cases} \quad (1.26)$$

Chapitre 1 : Les système non linéaire

$$\left\{ \begin{array}{l} x(k+1) = A_0x(k) + B_0u(k) \\ y(k) = p_1(C_0x(k) + D_0u(k)) + \sum_{i=2}^r p_i(C_0x(k) + D_0u(k))^i \end{array} \right. \quad (1.27)$$

- Dans le cas où $n_a = 2$ et $r = 2$

$$y(k) = p_1(C_0x + D_0u) + p_2(C_0x + D_0u)^2 \quad (1.28)$$

En remplaçant 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'équation suivante : (la variable k est omise pour ne pas surcharger les expressions)

$$\begin{aligned} y &= p_1v^1 + p_2v^2 \\ &= p_1(C_0x + D_0u) + p_2(C_0x + D_0u)^2 \quad (1.29) \\ &= p_1(c_1x_1 + c_2x_2 + du) + p_2(c_1x_1 + c_2x_2 + du)^2 \end{aligned}$$

En développant l'équation précédente, nous aboutissons à :

$$\begin{aligned} y &= p_1c_1x_1 + p_1c_2x_2 + p_1du + p_2[c_1^2x_1^2 + 2c_1c_2x_1x_2 + 2c_1dx_1u + c_2^2x_2^2 \\ &\quad + 2c_2dx_2u + d^2u^2] \quad (1.30) \end{aligned}$$

L'expression de la sortie peut être donnée sous forme vectorielle suivante :

$$y = [p_1c_1 \ p_1c_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [p_1d]u + [p_2c_1^2 \ 2p_2c_1c_2 \ 2p_2c_1d \ p_2c_2^2 \ 2p_2c_2d \ p_2d^2] \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_1u \\ x_2^2 \\ x_2u \\ u^2 \end{bmatrix} \quad (1.31)$$

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

Chapitre 1 : Les système non linéaire

$$\left\{ \begin{array}{l} x(k+1) = A_0x(k) + B_0u(k) \\ y = [p_1c_1 \quad p_1c_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [p_1d]u + [p_2c_1^2 \quad 2p_2c_1c_2 \quad 2p_2c_1d \quad p_2c_2^2 \quad 2p_2c_2d \quad p_2d^2] \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_1u \\ x_2^2 \\ x_2u \\ u^2 \end{bmatrix} \end{array} \right. \quad (1.32)$$

- Dans le cas où $n_a = 2$ 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 (1.33).

$$\left\{ \begin{array}{l} x(k+1) = A_0x(k) + B_0u(k) \\ v(k) = C_0x(k) + D_0u(k) \\ y(k) = p_1v^1(k) + p_2v^2(k) + p_3v^3(k) \end{array} \right. \quad (1.33)$$

La sortie y peut être alors formulée comme suit :

$$\begin{aligned} y = p_1(c_1x_1 + c_2x_2 + du) + p_2(c_1x_1 + c_2x_2 + du)^2 \\ + p_3(c_1x_1 + c_2x_2 + du)^3 \end{aligned} \quad (1.34)$$

En développant l'équation (1.34), on obtient :

$$\begin{aligned} y = p_1c_1x_1 + p_1c_2x_2 + p_1du \\ + p_2[c_1^2x_1^2 + 2c_1c_2x_1x_2 + 2c_1dx_1u + c_2^2x_2^2 + 2c_2dx_2u + d^2u^2] \\ + p_3[c_1^3x_1^3 + 3c_1^2c_2x_1^2x_2 + 3c_1^2dx_1^2u + 3c_1c_2^2x_1x_2^2 + 6c_1c_2dx_1x_2u + 3c_1d^2x_1u^2 \\ + c_2^3x_2^3 + 3c_2^2dx_2^2u + 3c_2d^2x_2u^2 + d^3u^3] \end{aligned} \quad (1.35)$$

Chapitre 1 : Les système non linéaire

Aux finales on peut reformuler et donnez l'écriture sous forme vectorielle suivante :

$$\left\{ \begin{array}{l} x(k+1) = A_0 x(k) + B_0 u(k) \\ y = p_1 [C_1 \quad C_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + p_1 [d] u + \begin{bmatrix} p_2 c_1^2 \\ 2p_2 c_1 c_2 \\ 2p_2 c_1 d \\ p_2 c_2^2 \\ 2p_2 c_2 d \\ p_2 d^2 \\ p_3 c_1^3 \\ 3p_3 c_1^2 c_2 \\ 3p_3 c_1^2 d \\ 3p_3 c_1 c_2^2 \\ 6p_3 c_1 c_2 d \\ 3p_3 c_1 d^2 \\ p_3 c_2^3 \\ 3p_3 c_2^2 d \\ 3p_3 c_2 d^2 \\ p_3 d^3 \end{bmatrix}^T \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_1 u \\ x_2^2 \\ x_2 u \\ u^2 \\ x_1^3 \\ x_1^2 x_2 \\ x_1^2 u \\ x_1 x_2^2 \\ x_1 x_2 u \\ x_1 u^2 \\ x_2^3 \\ x_2^2 u \\ x_2 u^2 \\ u^3 \end{bmatrix} \end{array} \right.$$

(1.36)

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

$$\left\{ \begin{array}{l} A = A_0 \\ B = B_0 \\ C = p_1 C_0 \\ D = p_1 D_0 \\ E = 0 \\ F = p_2 [C_0 x + D_0 u]^2 \cdots p_r [C_0 x + D_0 u]^r \\ \text{et} \\ F = [p_2 [Z]^2 \cdots p_r [Z]^r] \\ \text{avec} \\ Z = [C_0 \quad D_0] \end{array} \right. \quad (1.37)$$

Avec

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

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

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

1.5 Conclusion

Dans ce chapitre, on a présenté les principaux modèles des systèmes non linéaires. En effet, on s'est concentré sur les modèles blocs orientés, en particulier le modèle de Wiener qui est étudié tout au long de ce mémoire.

Le chapitre suivant se portera sur l'identification des systèmes bloc orientés.

Chapitre 2

Identification des systèmes blocs orientés

2.1 Introduction

Dans l'étude des systèmes complexes, qu'ils soient mécaniques, électriques ou informatiques, la notion d'ensemble de blocs orientés est utilisée. Ces blocs représentent les éléments de base, qui une fois assemblés, définissent l'architecture globale du système. L'orientation de ces blocs est déterminante car elle influence la direction du flux d'informations ou d'énergie au sein du système.

Ce chapitre se consacre à l'identification, en mettant en lumière les étapes d'identification et les diverses méthodes d'identification. En effet, les approches en temps continu et en temps discret, les méthodes de base telles que la méthode graphique directe, la méthode de Broïda, la méthode de Strecj et la méthode fréquentielle, seront détaillées.

Le chapitre explore également les algorithmes à erreur d'équation, notamment la méthode des moindres carrés, les moindres carrés pondérés et les moindres carrés récursifs. Les méthodes à erreur de sortie, telles que la méthode du gradient, la méthode de Gauss-Newton et la méthode de Levenberg-Marquardt, sont également présentées.

Une section est consacrée à la méthode d'optimisation par essaim de particules (PSO), qui est une technique puissante pour l'identification des systèmes. Enfin, l'identification des systèmes bloc orientés est abordée.

2.2 Généralités sur l'identifications

2.2.1 Définition

L'identification est l'opération de détermination du modèle dynamique d'un procédé (système) à partir des mesures entrées/sorties. La connaissance du modèle dynamique est nécessaire pour la conception et la mise en œuvre d'un système performant de régulation. Pratiquement, l'identification a généralement pour but la détermination de modèle de conduite, utilisables pour simuler, commander ou régler un processus. Ce modèle peut être physique (au sens de simulateur analogique ou numérique et de modèle réduit), ou bien un modèle abstrait (modèle mathématique, le système d'équations algébriques ou différentielle) [19].

Chapitre 2 : identification des système blocs orientés

L'identification consiste à chercher les paramètres des modèles mathématiques d'un système, à partir des données expérimentales et des connaissances disponibles a priori. Ces paramètres peuvent avoir une signification physique, ou ne pas en avoir, comme c'est le cas pour les modèles de comportement.

De nombreuses méthodes d'identification sont précisées dans la suite correspondant le plus souvent à l'un des schémas des figures (2.1) et (2.2) [19].

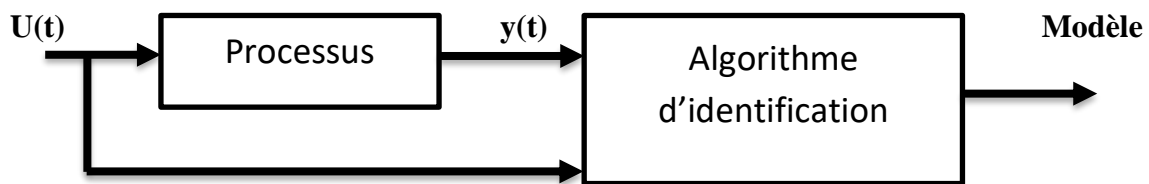


FIGURE 2.1: Identification à partir du comportement d'entre-sortie.

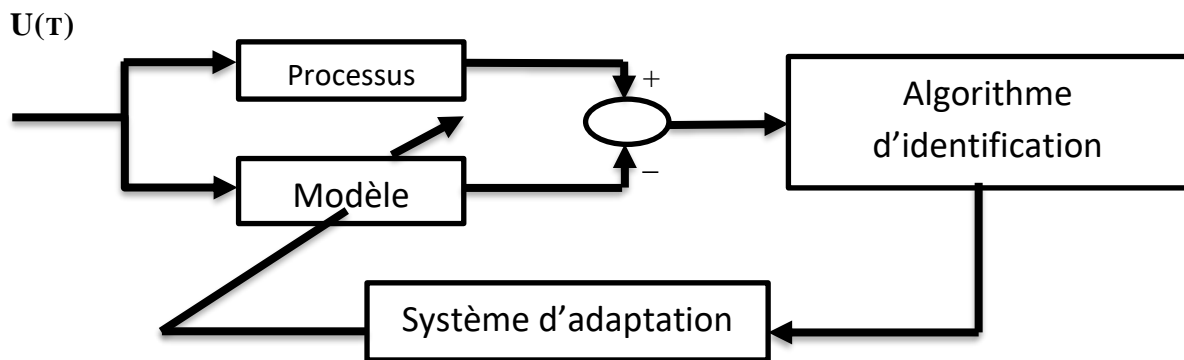


FIGURE 2.2: Identification de type paramétrique.

2.2.2 Etape de l'identification

L'identification est une approche expérimentale pour la détermination du modèle dynamique d'un système. Cette approche peut être décomposée en quatre étapes selon la figure ci-après [19] comme suit :

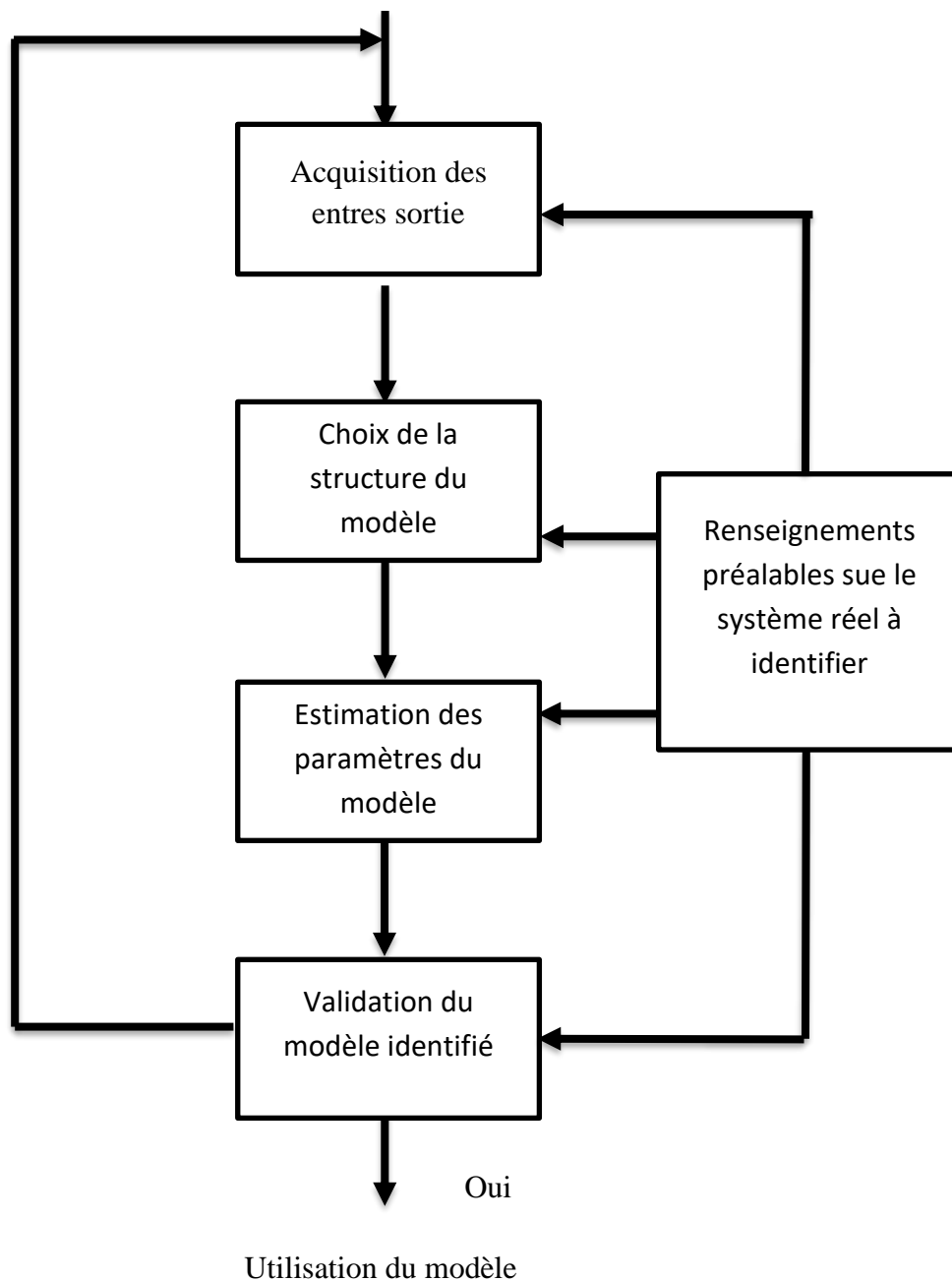


FIGURE 2.3: Procédure d'identification d'un modèle de système.

- **Acquisition des entrées et sorties**

La première étape fournit les données entrées/sorties susceptibles de permettre l'extraction d'un modèle de procédé significatif. Le protocole d'acquisition est un problème-clé car il conditionne en grande partie le succès d'une opération d'identification. Il doit tenir compte d'une part des contraintes pratiques, ce qui requiert en général l'utilisation des signaux d'excitation de faible amplitude et d'autre part de la nécessité d'exciter le système à identifier

dans une bande de fréquence suffisamment large afin d'obtenir un modèle significatif pour le calcul de la commande.

- **Choix de la structure du modèle**

Cette étape implique de choisir le type de modèle, son ordre et son retard pour représenter efficacement le système à identifier.

Le choix d'une structure non convenable conduit souvent à une mauvaise estimation, malgré toutes les précautions prises aux niveaux de la période d'échantillonnage, de la méthode d'identification paramétrique, du signal d'excitation, du gain d'adaptation et du nombre d'observations effectuées.

- **Estimation des paramètres du modèle**

Elle consiste à trouver les valeurs numériques des coefficients qui interviennent dans la structure du modèle. Ces valeurs numériques sont déterminées pour que le comportement du modèle soit le plus proche de celui du procédé. Cette proximité se mesure à l'aide d'un critère qui devra être minimisé.

- **Validation du modèle identifié**

Une fois l'algorithme d'identification appliqué et le modèle obtenu, il reste à examiner celui-ci. Selon la qualité du modèle obtenu on remet en cause l'une ou l'autre des étapes (signal d'excitation, structure du modèle,) et on recommence jusqu'à obtenir un modèle satisfaisant. Dans la pratique, on essaie un certain nombre de modèles avant de choisir le meilleur.

2.3 Méthodes d'identifications

2.3.1 Méthode Identification en temps continu et en temps discret

- **Identification en Temps Continu**

L'identification en temps continu concerne les systèmes dont les modèles sont décrits par des équations différentielles continues. Ces systèmes sont souvent rencontrés dans des applications où le temps est une variable continue et où les signaux sont analogiques.

Dans ce cas en peut citer les méthodes :

- **Identification Paramétrique** : Les techniques de paramétrisation consistent à estimer les paramètres d'un modèle continu à partir de données expérimentales. Cela peut être

réalisé à l'aide des méthodes telles que les moindres carrés, les moindres carrés généralisés, les méthodes de maximum de vraisemblance, etc. [20].

- **Identification non Paramétrique** : Ces méthodes visent à estimer la réponse en fréquence ou la fonction de transfert directement à partir des données sans faire d'hypothèses sur la structure du modèle. Les méthodes de convolution, les méthodes de corrélation croisée et les méthodes basées sur les séries de Fourier sont couramment utilisées [20].
- **Approches basées sur les Réseaux de Neurones** : Les réseaux de neurones peuvent également être utilisés pour identifier des modèles en temps continu. Ces approches peuvent être particulièrement efficaces pour modéliser des systèmes complexes et non linéaires.

- **Identification en Temps Discret**

L'identification en temps discret concerne les systèmes dont les modèles sont exprimés en termes de différences finies ou de récursives discrètes. Ces systèmes sont couramment rencontrés dans le domaine du traitement numérique du signal et du contrôle numérique.

Dans ce cas, on peut citer les méthodes courantes :

- **Identification Paramétrique** : Les techniques de paramétrisation pour les systèmes en temps discret sont similaires à celles utilisées en temps continu, mais les modèles sont exprimés en fonction des échantillons discrets des signaux. Les méthodes des moindres carrés, des moindres carrés récursifs (RLS), et les méthodes de prédiction d'erreur sont souvent utilisées [21].
- **Identification non Paramétrique** : Les méthodes non paramétriques en temps discret peuvent inclure l'utilisation des fonctions de transfert discrètes, d'approximations polynomiales, ou d'estimations de réponses en fréquence à partir des données échantillonnées [21].
- **Approches basées sur les Modèles ARMA** : Les modèles Auto-Regressive Moving Average (ARMA) sont couramment utilisés pour modéliser des systèmes en temps discret. L'identification des modèles ARMA implique souvent l'estimation des coefficients de l'auto régression et de la moyenne mobile à partir des données échantillonnées [21].

- **Comparaison des Approches**

- a. **Temps Continu**

- Avantages : Adapté aux systèmes analogiques, modèles continus fournissent une représentation précise, idéal pour les applications en temps réel.
- Inconvénients : Les systèmes analogiques peuvent nécessiter une conversion analogique-numérique pour l'acquisition des données, certaines méthodes peuvent être complexes à mettre en œuvre.

b. Temps Discret

- Avantages : Adapté aux systèmes numériques et échantillonnés, convient aux systèmes régis par des contrôleurs numériques, implémentation plus simple de certain algorithme.
- Inconvénients : La discrétisation des signaux peut introduire des erreurs, nécessite souvent une étape de conversion de modèle continu en modèle discret.

2.3.2 Méthode de base

Les méthodes de base d'identification des systèmes consiste à déterminer de manière récursive un prédicteur linéaire pour identifier les paramètres d'un système physique à partir de ses entrées et sorties [22]. Elles s'appuient sur l'analyse graphique des courbes expérimentales en quelques points particuliers et ne tiennent pas compte de l'ensemble des mesures. Leur robustesse au bruit de mesure est faible.

• Relation entrée, sortie et système

La sortie d'un système linéaire est le produit de convolution entre le signal d'entrée et la réponse impulsionnelle du système.

$$s(t) = e(t) * h(t) = \int_0^t e(\tau)h(t - \tau)d\tau \quad (2.1)$$

La sortie est caractéristique du système pour des signaux élémentaires tels que l'impulsion de Dirac, l'échelon ou la rampe

Le signal d'excitation le plus courant est l'échelon. L'identification se fait donc à partir de la réponse indicielle.

Remarque : on peut passer de la réponse impulsionnelle à la réponse indicielle par intégration (numérique) des mesures

Si le système présente une intégration dans sa fonction de transfert (pôle nul), il suffit de dériver (numériquement) les mesures pour obtenir la réponse d'un système équivalent sans intégration.

Exemple du circuit du 1er ordre de la forme :

$$L(p) = \frac{1}{1+\tau p} \quad (2.2)$$

La réponse impulsionnelle est :

$$S(p) = 1 \cdot \frac{1}{1+\tau p} = \frac{1/\tau}{p+1/\tau} \quad (2.3)$$

D'où :

$$S(t) = \frac{1}{\tau} e^{-t/\tau} \quad (2.4)$$

L'intégration de la réponse impulsionnelle donne la réponse indicielle :

$$\int_0^t S(u) du = \frac{1}{\tau} [-\tau e^{-t/\tau}]_0^t = 1 - e^{-t/\tau} \quad (2.5)$$

- **Méthode graphique directe**

- ❖ Pour un système du 1^{ER} ordre $H(p) = \frac{K}{(1+\tau p)}$ soumis à un échelon [22].

- $e(t) = e_1 u(t)$

La réponse indicielle est donnée par l'équation (2.6) et son allure est représentée par

la Figure (2.4).

$$s(t) = Ke_1 \left(1 - e^{-\frac{t}{\tau}}\right) + S_0 \quad (2.6)$$

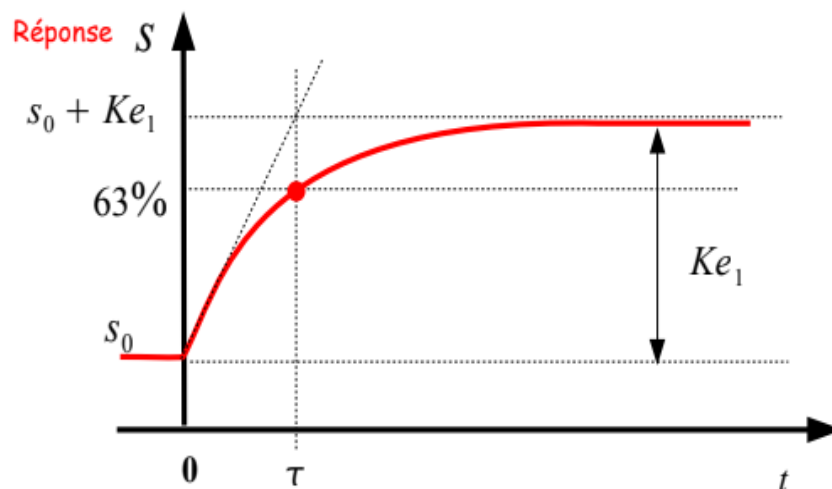


FIGURE 2.4: Simulation d'un système 1er ordre soumis à un échelon.

Le temps de repense a 95% et égale 3τ

Le temps de montrée est :

$$t_m = t_{90} = t_{10} = \tau \cdot \ln(9) \approx 2.2\tau \quad (2.7)$$

Chapitre 2 : identification des système blocs orientés

Dans le cas d'un système du 1^{ER} ordre $H(p) = \frac{K}{(1+\tau p)}$ soumis à une rampe $e(t) = a t u(t)$

La réponse du système est donnée par l'équation (2.8) et représentée par la Figure (2.5).

$$s(t) = aK\tau e^{-\frac{t}{\tau}} + aK(t - \tau) \quad (2.8)$$

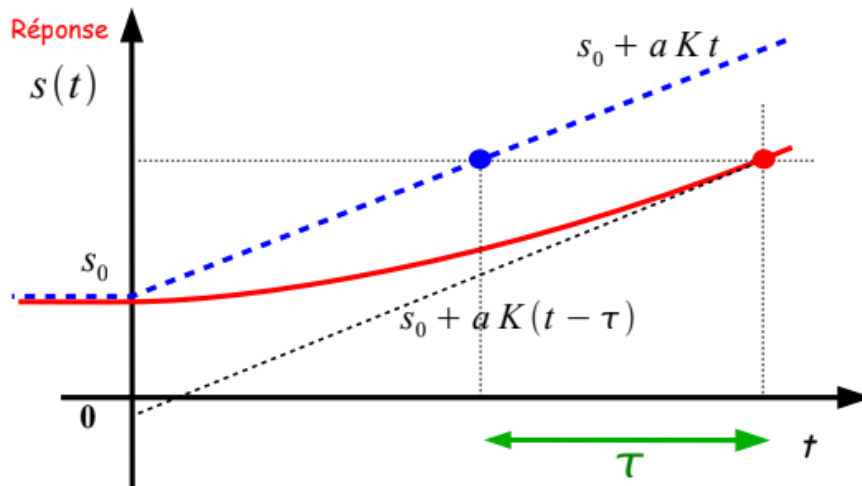


FIGURE 2.5: Simulation d'un système 1er ordre soumis à une rampe.

❖ Pour un système du 2^{eme} ordre pseudopériodique

$$H(p) = \frac{K\omega_n^2}{p^2 + 2Z\omega_n p + \omega_n^2} \quad (2.9)$$

Paramètre :

Calculer le gain du système :

$$K = \frac{\Delta y}{\Delta u} \quad (2.10)$$

Le dépassement du système :

$$D\% = 100 * e^{\frac{-\pi Z}{\sqrt{1-Z^2}}} \quad (2.11)$$

Graphiquement :

$$D\% = 100 * \frac{y(t_1) - y(\infty)}{y(\infty)} \quad (2.12)$$

Fréquence de résonance (ω_d) :

$$\omega_d = \omega_n \sqrt{1 - Z^2} \quad (2.13)$$

Période des oscillations :

$$T_d = \frac{2\pi}{\omega_d} \quad (2.14)$$

Allure de régime pseudopériodique est représentée par la Figure 2.6 :

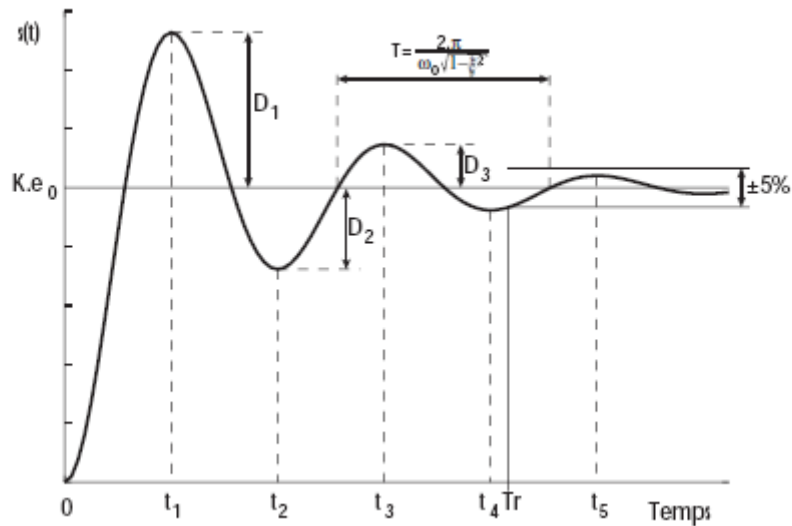


FIGURE 2.6 : Réponse à un échelon d'un système du second ordre – régime pseudopériodique

- **Méthode de Broïda**

Réponse indicielle du 1^{ER} ordre $H(p) = \frac{e^{-Tp}}{(1+\tau p)}$ avec retard [22]

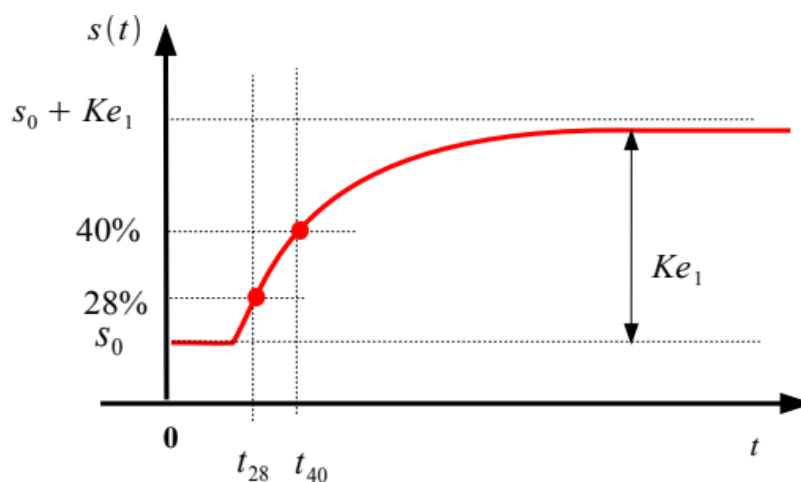


FIGURE 2.7: Réponse indicielle du 1er ordre avec retard.

$$T = 2.8t_{28} - 1.8t_{40} \quad (2.15)$$

$$\tau = 5.5(t_{40} - t_{28}) \quad (2.16)$$

- **Méthode de Strecj**

Réponse indicielle du 1^{ER} ordre $H(p) = \frac{K}{(1+\tau p)^n}$ [22] : (2.17)

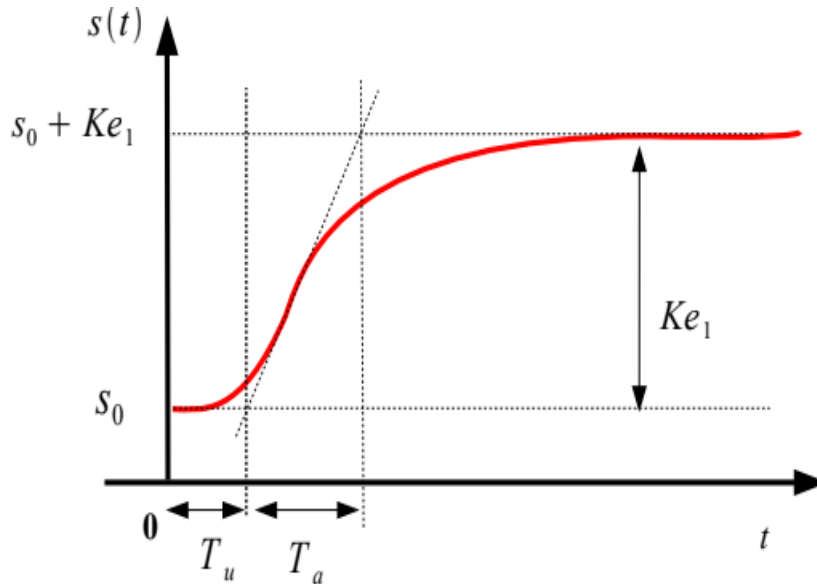


FIGURE 2.8: Réponse indicielle du 1er ordre.

TABLEAU 2.1: Paramètres de Strecj pour différents ordre n :

Ordre n	Amortissement ζ	Constante de temps équivalents T_e	Temps de montée T_m	Temps de réponse T_r
1	1	τ	$1.0 \times \tau$	$4.4 \times \tau$
2	0.707	2τ	$1.6 \times \tau$	$3.9 \times \tau$
3	0.477	3τ	$2.3 \times \tau$	$3.7 \times \tau$
4	0.350	4τ	$2.8 \times \tau$	$3.5 \times \tau$
5	0.280	5τ	$3.2 \times \tau$	$3.4 \times \tau$
6	0.234	6τ	$3.5 \times \tau$	$3.3 \times \tau$

2.3.3 Algorithme à erreur d'équation

Ces algorithmes sont basés sur la méthode classique des moindres carrés [23], [24], [25], la structure de ces méthodes est représentée en Figure (2.9)

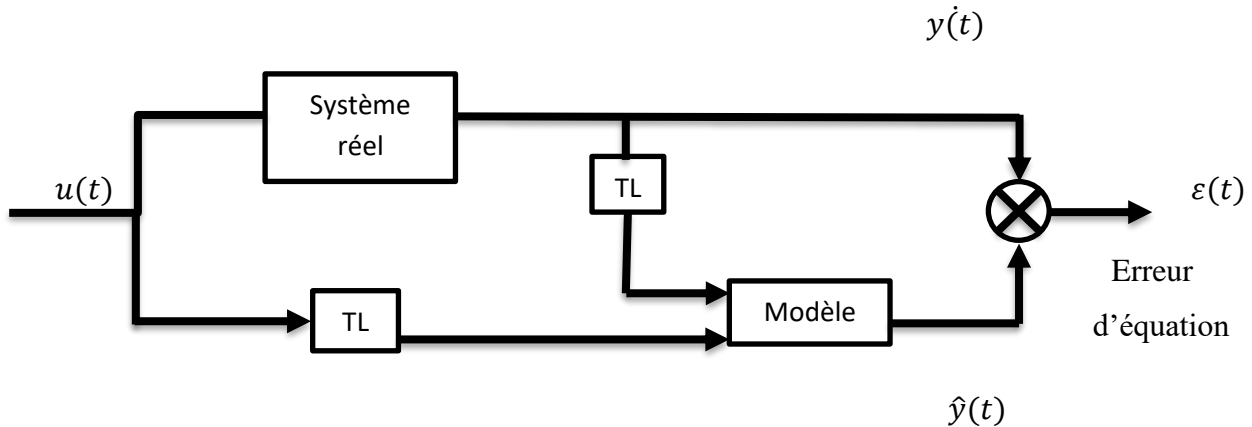


FIGURE2.9 : Structure erreur d'équation.

En continu, le système est décrit par une équation différentielle telle que :

$$y(t) + a_1 \frac{dy(t)}{dt} + \dots + a_{n_a} \frac{d^{n_a} y(t)}{dt^{n_a}} = b_0 u(t) + \dots + b_{n_b} \frac{d^{n_b} y(t)}{dt^{n_b}} + e(t) \quad (2.18)$$

En discrets, Le bloc de transformation linéaire des données (TL) est égal à 1.

Dans les deux cas discret et continu, l'erreur s'écrit sous la forme : $\varepsilon_{EE} = \hat{A}_y - \hat{B}_u$ et le critère à minimiser est un critère quadratique.

L'Equation (2.18) s'écrit sous cette forme :

$$y(t) = \varphi(t)^T \theta + e(t) \quad (2.19)$$

Le vecteur de paramètre est défini par :

$$\theta = [a_1 \dots a_{n_a} b_0 \dots b_{n_b}]^T \quad (2.20)$$

Le régresser s'écrit dans ce cas :

$$\varphi = \left[-\frac{dy(t)}{dt} \dots -\frac{d^{n_a} y(t)}{dt^{n_a}} u(t) \dots \frac{d^{n_b} y(t)}{dt^{n_b}} \right]^T \quad (2.21)$$

Dans ce cas, le régresser est formé des dérivées successives des entrées/sorties, inutilisables en pratique, et une transformation linéaire couplée à la méthode des moindres carrés permet de résoudre le problème de l'estimation des dérivées des signaux.

2.3.3.1 Méthode des moindres carrés

La méthode des moindres carrés est une technique d'estimation très utilisée pour identifier les paramètres des modèles de systèmes dynamiques, en minimisant la somme des carrés des écarts entre les valeurs observées et les valeurs prédites par le modèle. Nous allons explorer cette méthode plus en détail, en commençant par ses principes fondamentaux, puis en examinant comment elle est appliquée aux modèles Autoregressive with exogenous input (ARX), et enfin en considérant les variantes telles que les moindres carrés récurrents.

a. Principes Fondamentaux de la Méthode des Moindres Carrés

L'idée principale des moindres carrés est de trouver les paramètres θ d'un modèle linéaire qui minimisent l'erreur quadratique totale entre les sorties observées $y(t)$ et les sorties prédites.

- Formulation Mathématique

Pour un modèle linéaire simple :

$$y(t) = \phi(t)^T \theta + e(t) \quad (2.22)$$

Où :

$y(t)$ est la sortie observée à l'instant t .

$\phi(t)$ est le vecteur de régression, composé des observations d'entrée et/ou de sortie à des instants antérieurs.

θ est le vecteur des paramètres à estimer.

$e(t)$ est l'erreur de modélisation ou bruit.

- La fonction de coût à minimiser est :

$$J(\theta) = \sum_{t=1}^N e(t)^2 = \sum_{t=1}^N [y(t) - \phi(t)^T \theta]^2 \quad (2.23)$$

- Solution des Moindres Carrés

Pour minimiser $J(\theta)$, nous dérivons cette fonction par rapport à θ et mettons la dérivée à zéro :

$$\frac{\partial J(\theta)}{\partial \theta} = -2 \sum_{t=1}^N \phi(t) [y(t) - \phi(t)^T \theta] = 0 \quad (2.24)$$

Chapitre 2 : identification des système blocs orientés

En résolvant cette équation, nous obtenons :

$$\theta = (\sum_{i=1}^N \phi(t)\phi(t)^T)^{-1} \sum_{i=1}^N \phi(t)y(t) \quad (2.25)$$

Cette formule peut être écrite plus succinctement en utilisant des matrices :

$$\theta = (\phi^T \phi)^{-1} \phi^T Y \quad (2.26)$$

Où :

ϕ est la matrice de régression (contenant les vecteurs $\phi(t)$ pour chaque t).

Y est le vecteur des observations (contenant les valeurs $y(t)$ pour chaque t).

- Application aux Modèles ARX

Les modèles Auto-Regressive with exogenous inputs (ARX) sont largement utilisés pour représenter des systèmes linéaires. Un modèle ARX de type $ARX(n_a, n_b, n_k)$ est défini par :

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + b_2 u(t-2) + \dots + b_{n_b} u(t-n_b) + e(t) \quad (2.27)$$

- Mise en Forme des Équations

En réarrangeant, on obtient :

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) - \dots - a_{n_a} y(t-n_a) + b_1 u(t-1) + b_2 u(t-2) + \dots + b_{n_b} u(t-n_b) + e(t) \quad (2.28)$$

Le vecteur de régression $\phi(t)$ et le vecteur de paramètres θ sont alors :

$$\phi(t) = [-y(t-1) \quad -y(t-2) \quad \dots \quad -y(t-n_a) \quad u(t-1) \quad u(t-2) \quad \dots \quad u(t-n_b)] \quad (2.29)$$

$$\theta = [a_1 \quad a_2 \quad \dots \quad a_{n_a} \quad b_1 \quad b_2 \quad \dots \quad b_{n_b}]^T \quad (2.30)$$

b. Variantes des Moindres Carrés

- **Moindres Carrés Pondérés**

Dans certaines situations, il peut être utile de donner plus d'importance à certaines observations. Cela se fait en introduisant une matrice de poids W dans la formulation :

$$\theta = (\Phi^T W \Phi)^{-1} \Phi^T W Y \quad (2.31)$$

- **Moindres Carrés Récursifs**

Les moindres carrés récursifs (RLS) sont utilisés lorsque les données arrivent de manière séquentielle et que le modèle doit être mis à jour en temps réel. L'algorithme RLS met à jour les estimations des paramètres à chaque nouvelle observation sans avoir à recalculer la solution complète depuis le début.

L'algorithme RLS se base sur les équations suivantes :

1. Calcul de l'erreur de prédiction :

$$e(t) = y(t) - \phi(t)^T \hat{\theta}(t-1) \quad (2.32)$$

2. Calcul du gain :

$$K(t) = P(t-1)\phi(t)[\lambda + \phi(t)^T P(t-1)\phi(t)]^{-1} \quad (2.33)$$

3. Mise à jour des paramètres :

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)e(t) \quad (2.34)$$

4. Mise à jour de la matrice de covariance :

$$P(t) = \frac{1}{\lambda} [P(t-1) - K(t)\phi(t)^T P(t-1)] \quad (2.35)$$

Où λ est un facteur de régularisation (ou facteur d'oubli) qui contrôle la vitesse d'adaptation du modèle.

En continu, le système est décrit par une équation différentielle telle que :

$$y(t) + a_1 \frac{dy(t)}{dt} + \dots + a_{n_a} \frac{d^{n_a} y(t)}{dt^{n_a}} = b_0 u(t) + \dots + b_{n_b} \frac{d^{n_b} u(t)}{dt^{n_b}} + e(t) \quad (2.36)$$

En discrets, Le bloc de transformation linéaire des données (TL) est égal à 1.

Dans les deux cas discret et continu, L'erreur s'écrit sous la forme : $\varepsilon_{EE} + \hat{A}_y - \hat{B}_u$ et le critère à minimiser est un critère quadratique.

L'Equation (2.41) s'écrit sous cette forme :

$$y(t) = \Phi(t)^T \theta + e(t) \quad (2.37)$$

Le vecteur de paramètre est défini par :

$$\theta = [a_1 \cdots a_{na} \ b_0 \cdots b_{nb}]^T \quad (2.38)$$

Le régresser s'écrit dans ce cas :

$$\Phi = \left[-\frac{dy(t)}{dt} \cdots -\frac{d^{na}y(t)}{dt^{na}} \ u(t) \cdots \frac{d^{nb}y(t)}{dt^{nb}} \right]^T \quad (2.39)$$

Dans ce cas, le régresser est formé des dérivées successives des entrées/sorties, inutilisables en pratique, et une transformation linéaire couplée à la méthode des moindres carrés permet de résoudre le problème de l'estimation des dérivées des signaux.

2.3.4 Méthode a erreur de sortie

Cette méthode, connue sous l'appellation de méthode du modèle [26], est une technique d'identification des systèmes où un modèle mathématique est ajusté pour représenter le comportement d'un système réel en utilisant les données d'entrée et de sortie. La sortie simulée est indépendante de la perturbation affectant le système (Figure 2.12). Ces algorithmes à erreur de sortie sont applicables sans restriction aux systèmes linéaires ou non linéaires, ainsi que discrets ou continus, et ne nécessitent pas de transformation linéaire pour ce dernier cas.

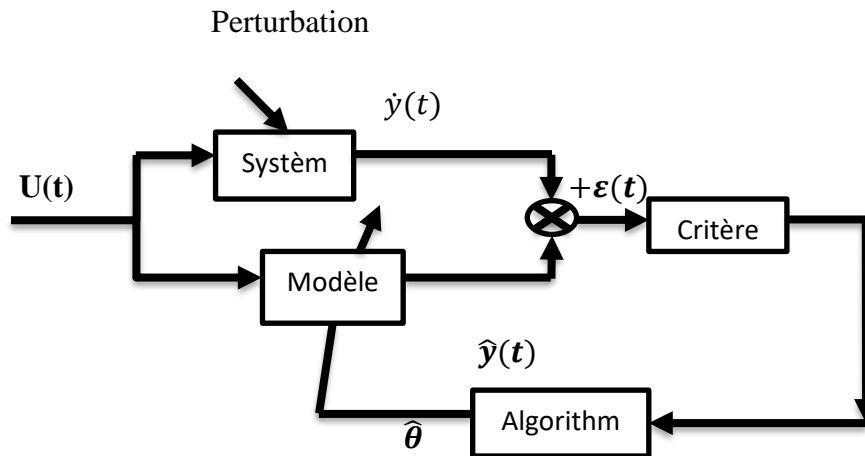


FIGURE2.10:Principe de la méthode a erreur de sortie.

- **Principe de la méthode**

Soit le système monovariante dépendant du vecteur θ de paramètres à estimé

$$y = g(\theta, u) \quad (2.40)$$

Le critère que l'on se propose de minimiser est

$$J = \sum_{k=1}^K \varepsilon_k^2 = \sum_{k=1}^K (y_k - \hat{g}(u, \hat{\theta}))^2 \quad (2.41)$$

Où y_k représente la sortie échantillonnée perturbée par un bruit b_k et $\hat{\theta}$ le vecteur de paramètres estimé.

$y(t)$ n'étant pas linéaire en θ et le critère est minimisé lorsque ses dérivées par rapport aux paramètres sont nulles.

Les algorithmes à erreur de sortie diffèrent par la façon de gérer l'optimisation. Plusieurs méthodes itératives d'optimisation non linéaire peuvent être utilisées [27] :

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

Nous présenterons les deux techniques de base pour l'identification par la méthode à erreur de sortie : la méthode du Gradient, la méthode de Gauss-Newton et une troisième méthode qui concilie les avantages de chacune de ces deux méthodes : l'algorithme de Levenberg-Marquardt.

2.3.4.1 Méthode du gradient

Cette technique est à la base des méthodes d'optimisation non linéaire. Dans ce cas, l'équation de récurrence est de la forme suivante : [28], [29]

$$\theta^{(i+1)} = \theta^{(i)} - \lambda \left. \frac{\partial J}{\partial \theta} \right|_{\theta=\theta^{(i)}} = \theta^{(i)} - \lambda (J'_{\theta})_{\theta=\theta^{(i)}} \quad (2.42)$$

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

λ est le paramètre de contrôle.

L'algorithme consiste à recalculer le gradient J'_{θ} à chaque itération. De ce fait, la direction de recherche de l'optimisation varie à chaque instant et la trajectoire obtenue est la ligne de plus grande pente (steepest descent). Le paramètre de contrôle λ est un coefficient de pondération dont la valeur conditionne la convergence. Cette méthode est stable mais oscille au voisinage de l'optimum, et le temps de convergence peut devenir très long.

2.3.4.2 Méthode de gausse newton

Cette méthode permet d'améliorer la recherche de l'optimisation en utilisant la matrice des dérivés secondes du critère appelée le Hessien ($J''_{\theta\theta}$). L'équation de récurrence s'écrit dans ce cas [27], [29] :

$$\theta^{(i+1)} = \theta^{(i)} - [(J''_{\theta\theta})^{-1} \quad J'_{\theta}]_{\theta=\theta^{(i)}} \quad (2.43)$$

Cette méthode est beaucoup plus rapide que celle du gradient au voisinage de l'optimum mais comporte un risque important de divergence au voisinage d'un point d'inflexion. Pour y remédier partiellement, on utilise une approximation du Hessien : le pseudo Hessien à l'aide des fonctions de sensibilité calculées à partir de l'équation du critère J .

$$J = \sum_{k=1}^K (\mathcal{Y}_k^* - \hat{y}_k)^2 = \sum_{k=1}^K \varepsilon_k^2$$

$$J'_{\theta} = \frac{\partial J}{\partial \theta_n} = -2 \sum_{k=1}^K (\mathcal{Y}_k^* - \hat{y}_k) \frac{\partial \hat{y}_k}{\partial \theta_n} \quad (2.44)$$

où $\sigma_{\theta_n, k} = \frac{\partial \hat{y}_k}{\partial \theta_n}$ est la fonction de sensibilité de la sortie \hat{y}_k par rapport au paramètre θ_n

$$J'_{\theta} = \frac{\partial J}{\partial \theta_n} = -2 \sum_{k=1}^K \varepsilon_k \sigma_{\theta_n} \quad (2.45)$$

La dérivée seconde du critère peut être approximée par le pseudo Hessien :

$$J''_{\theta\theta} = \frac{\partial^2 J}{\partial \theta_n \partial \theta_n} \simeq 2 \sum_{k=1}^K \sigma_{\theta n} \sigma_{\theta n} \quad (2.46)$$

L'algorithme de Gauss Newton présente l'avantage d'une rapide convergence vers l'optimum, mais s'avère instable lorsque l'initialisation n'est pas au voisinage de l'optimum.

2.3.4.3 Méthode de Levenberg Marquardt

C'est un algorithme d'optimisation non linéaire qui fonctionne très bien dans la pratique [30], [31]. Il combine avantageusement les propriétés des deux méthodes décrites précédemment : stabilité du gradient et rapidité de convergence de Gauss-Newton.

Cette méthode est basée sur la recherche itérative d'un vecteur de paramètres θ minimisant Le critère quadratique J_θ :

$$J_\theta = \frac{1}{K} \sum_{k=1}^K \varepsilon^2(k) \quad (2.47)$$

Où

- $\hat{y}(k)$ C'est la sortie estimée
- $y(k)$ C'est la sortie mesurée
- $\varepsilon(k)$ est l'erreur de prédiction entre la sortie mesurée et la sortie estimée
 - Tel que $\varepsilon(k) = y(k) - \hat{y}(k)$
- K est l'horizon d'observation $(u(k), y(k))$.

La relation de récurrence de la méthode est donnée par :

$$\theta^{(i+1)} = \theta^{(i)} - [(J''_{\theta\theta} + \lambda I)^{-1} J'_\theta]_{\theta=\theta^{(i)}} \quad (2.48)$$

λ est le paramètre de contrôle et I est une matrice identité.

La valeur du paramètre de contrôle λ , appelé aussi coefficient de Marquardt, s'adapte en fonction de l'évolution du critère :

Lorsque J augmente, λ est augmenté pour obtenir une évolution de l'algorithme type gradient et retrouver une bonne direction de recherche.

En revanche, si J diminue, λ diminue et l'algorithme se comporte de manière similaire à la méthode de Gauss-Newton.

Cet algorithme nécessite le calcul du gradient J' , effectué de la manière suivante :

$$J' = \left(\frac{\partial J_\theta}{\partial \theta} \right) = \frac{-2}{K} \sum_{k=1}^K (y(k) - \hat{y}(k)) \frac{\partial \hat{y}(k)}{\partial \theta} \quad (2.49)$$

$$= \frac{-2}{K} \sum_{k=1}^K \left(\varepsilon(k) (\sigma_{y/\theta}) \right) \quad (2.50)$$

Ou $(\sigma_{y/\theta}) = \frac{\partial \hat{y}(k)}{\partial \theta}$ est la fonction de sensibilité de la sortie $\hat{y}(k)$ par rapport θ .

La dérivée seconde du critère J_θ peut être approximée par le pseudo Hessian :

$$J'' = \frac{\partial^2 J_\theta}{(\partial \theta)(\partial \theta)} = \frac{2}{K} (\sigma_{y/\theta})^T (\sigma_{y/\theta}) \quad (2.51)$$

La difficulté dans la procédure de minimisation du critère J' réside dans le calcul des fonctions de sensibilités [32], qui dépend essentiellement de la structure du modèle.

Par conséquent, l'identification devient très sensible au calcul des fonctions de sensibilité.

2.3.5 Méthode d'optimisation par essais des particules

L'optimisation par essais de particulaires (OEP ou PSO en anglais) est une métaheuristique d'optimisation, inventée par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995 [33]. Cette méthode s'inspire du comportement de déplacement d'un essaim d'oiseaux ou de poissons pour la recherche de nourriture [34].

L'algorithme PSO génère un essaim de particules dont chaque membre représente une solution potentielle du problème d'optimisation. Cet essaim "vole" dans l'espace de recherche et chaque particule est attirée par sa meilleure solution ainsi que celle de ses voisins. Chaque particule a une mémoire contenant les données relatives à son vol (position, vitesse, meilleure solution) et la capacité de communiquer avec son entourage [34].

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

- PSO utilise une population de solutions candidates, appelées particules, pour explorer l'espace de recherche. Chaque particule représente une solution possible.
- Les particules se déplacent dans l'espace de recherche en mettant à jour leurs positions et vitesses en fonction de leurs propres expériences (meilleur point rencontré) et des expériences des autres particules (meilleur point global rencontré par le groupe).
- A chaque instant, chaque particule a une position et une vitesse et une mémoire et est donc capable de se souvenir de sa meilleure position visitée dans l'espace de recherche.
- Le PSO équilibre l'exploration (recherche de nouvelles régions de l'espace de recherche) et l'exploitation (affinement des solutions dans les régions prometteuses) à travers le contrôle des paramètres ω, C_1, C_2 .

- Au début, une population des particules est initialisée à des positions aléatoires marquées par les vecteurs X_i et la vitesse aléatoire V_i .
- La meilleure position de chaque particule est connue sous le nom p_{best} et celle de tout le groupe est appelée g_{best} .
- L'algorithme PSO est simple à comprendre et à implémenter par rapport à d'autres algorithmes d'optimisation stochastiques.

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

$$\begin{cases} V_{id}(k+1) = \omega V_{id}(k) + z_1 r_1 (p_{best}(k) - X_{id}(k)) + z_2 r_2 (g_{best}(k) - X_{id}(k)) \\ X_{id}(k+1) = X_{id}(k) + V_{id}(k+1) \end{cases} \quad (2.52)$$

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

Le paramètre $\omega = 0.9$ est le poids d'inertie.

$z_1 = z_2 = 0.4$: sont des constantes positives.

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.

2.4 Identification des systèmes bloc orientés

Un système bloc orienté est un modèle conceptuel utilisé dans divers domaines de l'ingénierie et des sciences appliquées pour représenter le comportement dynamique d'un système complexe.

L'identification des modèles blocs orientés est un problème qui remonte au milieu des années 60 [35] et plusieurs méthodes d'identification ont été développées et adaptées pour les modèles de Wiener et Hammerstein.

L'identification des systèmes Hammerstein a été initiée par Narendra et Gallman (1966) [35] ou une méthode itérative a été utilisée. Depuis, de nombreux travaux ont été développés et plusieurs approches et algorithmes ont été proposés dans la littérature.

En 2007, Ding et al. [36] ont abordé le problème d'identification d'un modèle Hammerstein à erreur de sortie basée sur l'algorithme des moindres carrés. L'analyse de la convergence de l'algorithme et les résultats des simulations montrent l'efficacité d'algorithme proposé. L'algorithme des moindres carrés récursifs a été proposé par Giri et al. (2008) [37] pour l'identification des systèmes Hammerstein en présence des non-linéarités de type hysteresis-relais. En 2013, Wang et al. [38] ont présenté une méthode des moindres carrés récursifs basée sur le principe de séparation des termes clés. Ce principe a été associé à l'algorithme de Levenberg Marquardt par Zhou et al. (2015) [4] et à l'algorithme du gradient par Ma et al. (2016) [39] pour l'identification des systèmes Hammerstein.

L'algorithme de Newton basé sur le principe hiérarchique a été proposé par Ding et al. (2014) [40].

Pour les systèmes non linéaires Wiener, plusieurs approches ont été proposées. Par exemple, Hagenblad et al. (2008) [41] ont présenté la méthode du maximum de vraisemblance permettant d'identifier les modèles de Wiener. Chen et al. (2014) [1], ont utilisé l'algorithme des moindres carrés récursif basé sur la théorie du filtrage pour les systèmes non linéaires Wiener en utilisant le principe d'identification hiérarchique. Hu et al.

(2014) [42] ont abordé le problème d'identification d'un système Wiener de type Moving Average Noises. Ils ont utilisé la méthode des moindres carrés récursifs étendue basée sur le principe de sur-paramétrisation. En 2015, Vörös [43] a utilisé l'algorithme des moindres carrés itérative basé sur le principe de séparation des termes clés pour l'estimation d'un système Wiener ou la partie non linéaire est de type hystérésis à temps invariant. De plus, la méthode du gradient a été considérée dans [44].

Le problème d'identification des systèmes Hammerstein-Wiener a été abordé par Bai (2002) [45], l'auteur a proposé une approche pour identifier un modèle Hammerstein-Wiener dont la partie non linéaire a été supposée inversible. L'algorithme de Levenberg Marquardt basé sur l'approche PNLSS a été utilisé par Paduardt et al. (2012) et Mulder et al. (2010) [46]. Wang et al. (2008) [47] ont traité le problème d'identification des modèles Hammerstein-Wiener ARMAX utilisant l'algorithme du gradient stochastique étendu et en 2016, Wang et Ding [48] ont utilisé l'algorithme du gradient et l'algorithme des moindres carrés récursifs basés sur la technique du filtrage des données pour identifier un système Hammerstein-Wiener. De plus, la méthode des sous-espaces a été proposée dans [49].

Le problème d'identification des systèmes Wiener-Hammerstein a été abordé dans Chen et Fang (2012) [50], les auteurs ont estimé récursivement à l'aide de l'algorithme d'approximation stochastique les sous-systèmes linéaires et ils ont utilisé une méthode non

paramétrique pour la partie non linéaire. Paduardt et al. (2012) [50] ont utilisé la méthode d'optimisation non linéaire pour estimer les paramètres d'un modèle Wiener-Hammerstein utilisant l'approche du modèle d'état non-linéaire polynomiale (PNLSS). En 2015, Ase et Katayama [51], ont développé la méthode des sous-espaces pour identifier un modèle Wiener-Hammerstein. De plus, la méthode d'optimisation évolutionnaire a été utilisée par Naitali et Giri (2016) [52] et l'algorithme de Levenberg Maruardt a été proposé par Schoukens et al. (2014) [53].

2.5 Conclusion

Ce chapitre a fourni une vue d'ensemble complète des méthodes d'identification des systèmes blocs orientés, en couvrant à la fois les aspects théoriques et pratiques. Nous avons commencé par introduire les concepts fondamentaux de l'identification des systèmes, en soulignant les étapes clés de ce processus, allant de la définition du modèle à l'estimation des paramètres.

Nous avons ensuite exploré les différentes méthodes d'identification, en distinguant les approches en temps continu et en temps discret, et en comparant leurs avantages et inconvénients respectifs. Des méthodes classiques, telles que les moindres carrés et les algorithmes basés sur l'erreur d'équation et l'erreur de sortie, ont été détaillées, mettant en lumière leurs applications et limitations.

L'accent a été mis sur les méthodes d'optimisation, notamment l'algorithme à essaims de particules (PSO), qui est de plus en plus utilisé pour résoudre des problèmes d'identification complexes grâce à sa capacité à explorer efficacement l'espace de solution.

Enfin, l'identification spécifique des systèmes blocs orientés a été discutée, soulignant l'importance de choisir des méthodes appropriées en fonction de la nature du système à identifier.

En résumé, ce chapitre fournit les bases nécessaires pour aborder l'identification des systèmes blocs orientés avec une compréhension claire des différentes méthodes disponibles et de leurs applications spécifiques.

Chapitre 3

Résultats de
simulation et
discussion

3.1 Introduction

Le modèle de Wiener présente un vaste champ d'applications et la capacité de reproduire le comportement des nombreux phénomènes physiques. Cela en fait un sujet d'étude intéressant, malgré les défis théoriques posés. C'est pourquoi le problème de l'identification des systèmes de Wiener mérite d'être approfondi, malgré sa complexité.

Ce chapitre aura pour objectif de présenter les résultats de simulation de l'estimation des paramètres du modèle de Wiener, par la méthode PSO. Par conséquent pour objectif aussi d'évaluer et d'illustrer l'efficacité de cette méthode d'identification.

Les simulations seront menées avec un signal d'entrée $u(k)$ constitué d'une séquence d'excitation aléatoire persistante, de moyenne nulle et de variance unitaire. La perturbation $e(k)$ est une séquence de bruit blanc de moyenne nulle.

Les résultats de simulations sont réalisés pour le cas sans bruit et pour différents rapport signal sur bruit (SNR).

3.2 Identification par l'algorithme PSO

3.2.1 Exemple 1 :

Nous présentons dans cette section l'exemple d'un système non linéaire (Wiener), qui peut être identifié avec la méthode PSO, expliquée dans la section (2.3.5)

Dans cet exemple, le modèle de Wiener est testé avec la partie linéaire d'ordre $n_a = 2$ et l'ordre de la partie non linéaire $r = 2$

Les matrices du bloc linéaire sont données comme suit :

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

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

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

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

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 1 \\ -0.3 & -0.2 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) = [0,5 \quad 0,1] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0,4]u + [0,05 \quad 0,02 \quad 0,08 \quad 0,002 \quad 0,016 \quad 0,032] \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_1u \\ x_2^2 \\ x_2u \\ u^2 \end{bmatrix} \end{cases} \quad (3.2)$$

On a commencé 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 paramètre θ à estimer est donné par :

$$\theta = [-0.3 \quad -0.2 \quad 0.5 \quad 0.1 \quad 0.4 \quad 2] \quad (3.3)$$

Les simulations ont été effectuées d'abord en absence de bruit, puis avec des données avec bruit pour différents rapports signal/bruit (SNR = 34dB et SNR = 26dB).

Le Tableau 3.1 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 la méthode peut être appréciée

TABLEAU 0.2: Résultats d'identification du modèle de Wiener PNLSS avec PSO

pour $n_a = r = 2$

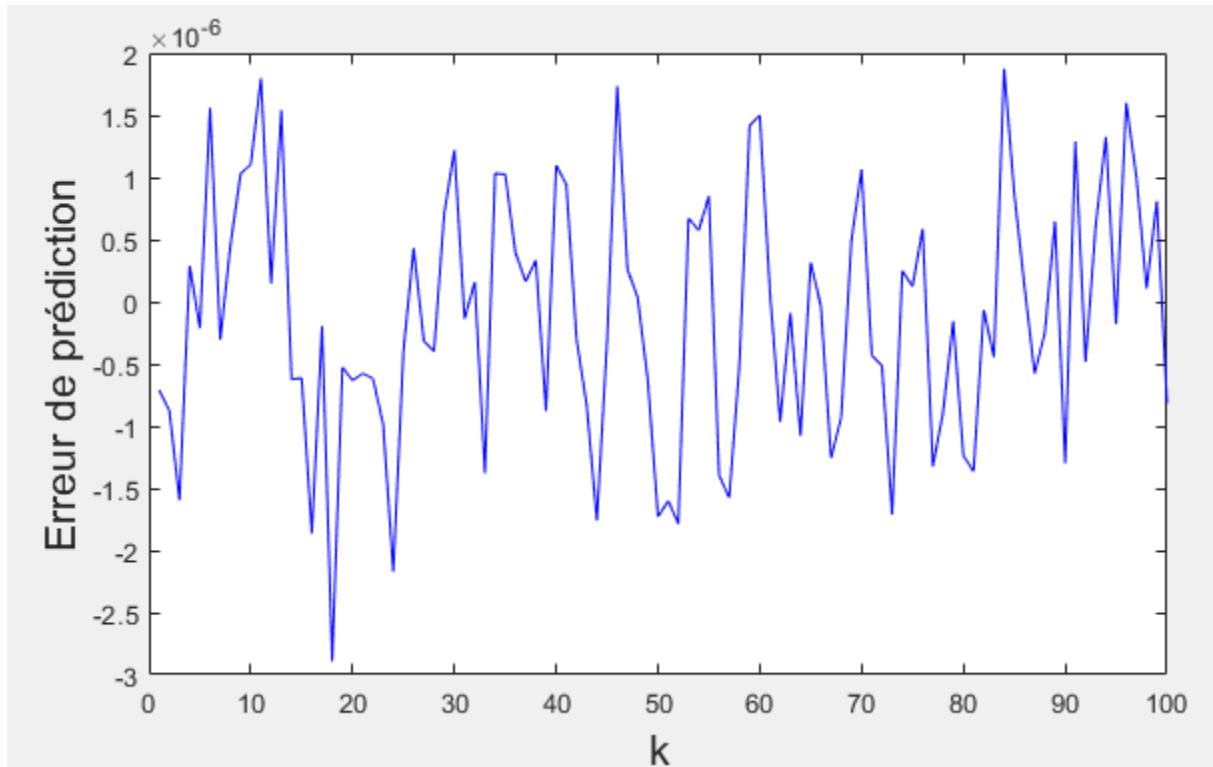
Paramètres	Valeurs exactes	SNR=0 dB	SNR=34 dB	SNR=26 dB
a_1	-0.300	-0.300	-0.319	-0.291
a_2	-0.200	-0.200	-0.183	-0.199
c_1	0.500	0.500	0.491	0.544
c_2	0.100	0.100	0.081	0.104
d	0.400	0.400	0.378	0.409
p_2	2.000	2.000	2.115	1.788
J	-	1.025e-012	0.049	0.085

L'erreur de prédiction de sortie pour le cas sans bruit, représentée sur la figure (3.1(a)), est extrêmement faible (nulle), avec un critère ($J= 10-12$), ainsi la courbe de la réponse temporelle du système estimé chevauche les données sur la figure (3.1(b)).

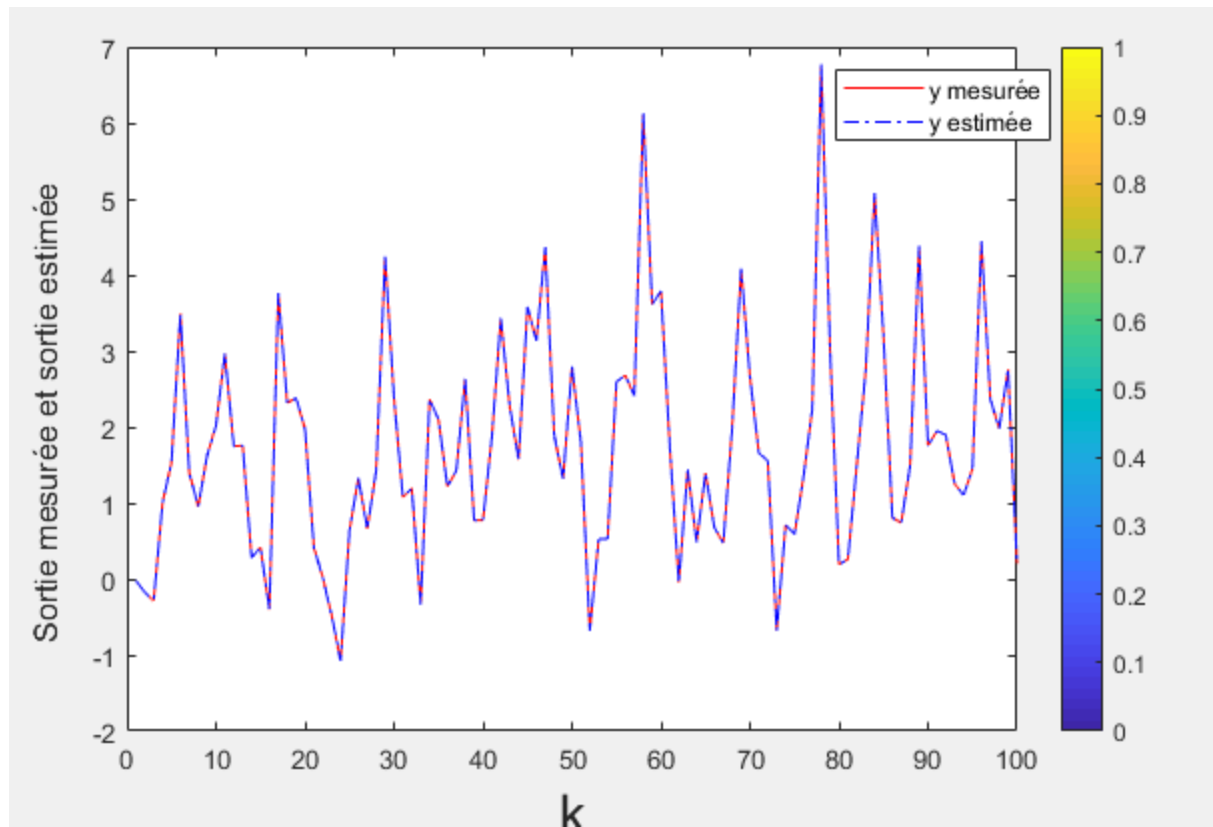
En présence de bruit SNR = 34 dB et SNR = 26 dB, la réponse de sortie du modèle estimé est comparée aux données respectivement sur la figure (3.2(b)) et sur la figure (3.3(b)), elles montrent un petit dépassement.

L'erreur de prédiction de sortie pour SNR = 34 dB est représentée sur la figure (3. 2(a)), et la figure (3.3(a)) représente l'erreur pour SNR =26 dB. On remarque une augmentation de manière significative lorsque le SNR passe de 0 dB à 34 dB et 26 dB.

Une augmentation du SNR entraîne des changements significatifs dans ces valeurs. Cela suggère que la méthode PSO est efficace, mais son efficacité peut être affectée par des niveaux de bruit élevés.

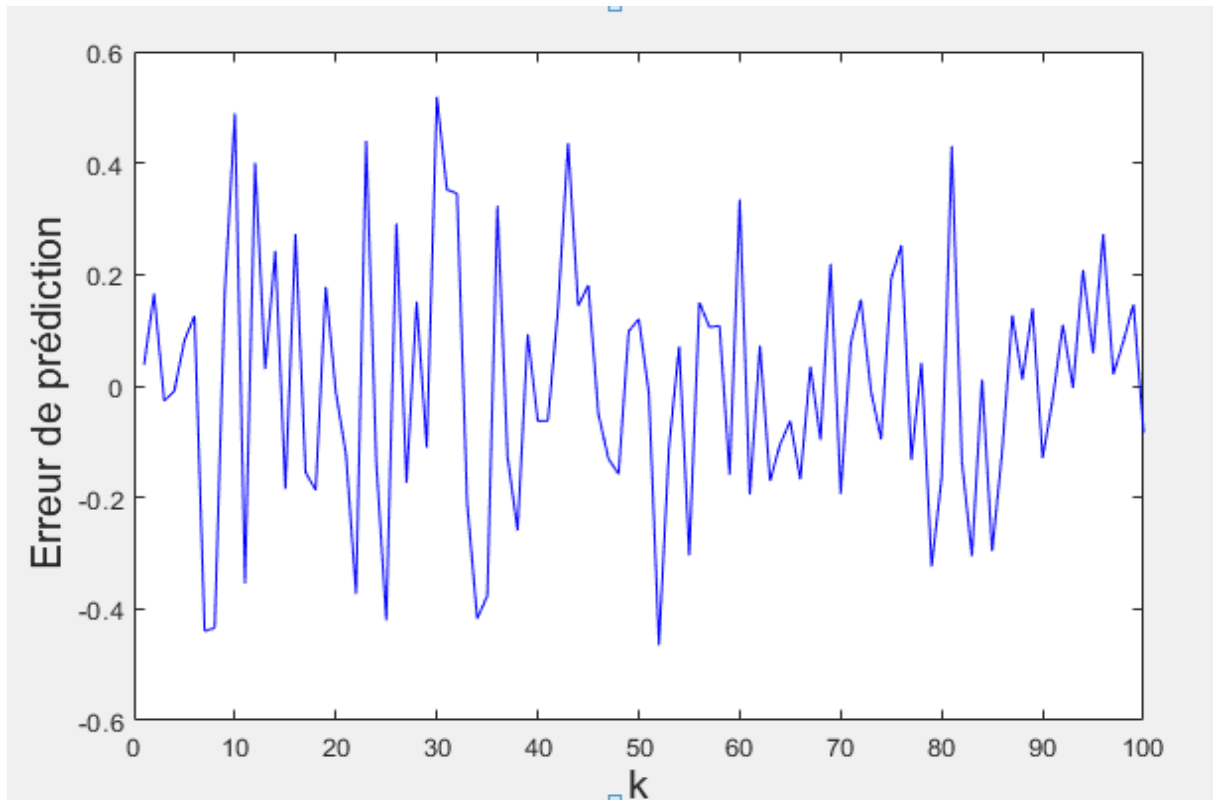


a. Erreur de prédiction

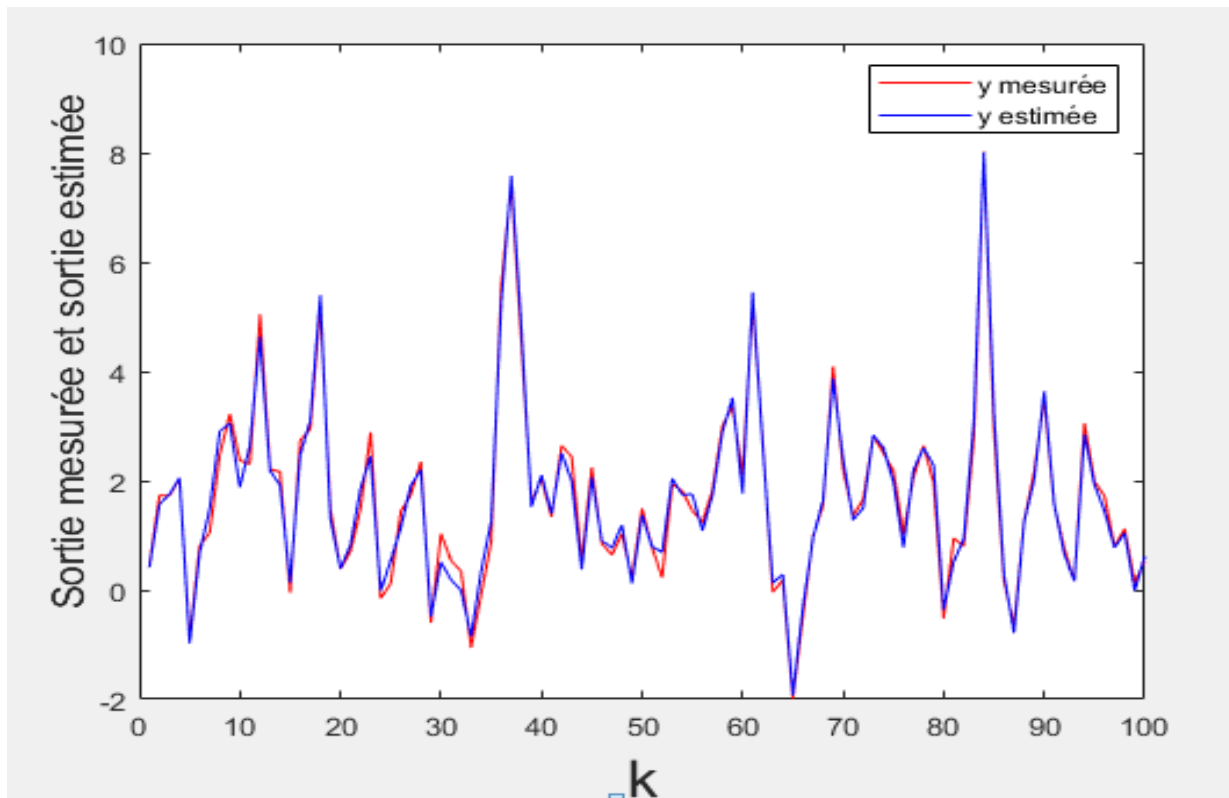


b. Sortie estimée et sortie mesurée

FIGURE 0.11: Résultats de simulation avec l'algorithme PSO pour SNR=0 dB (sans bruit).

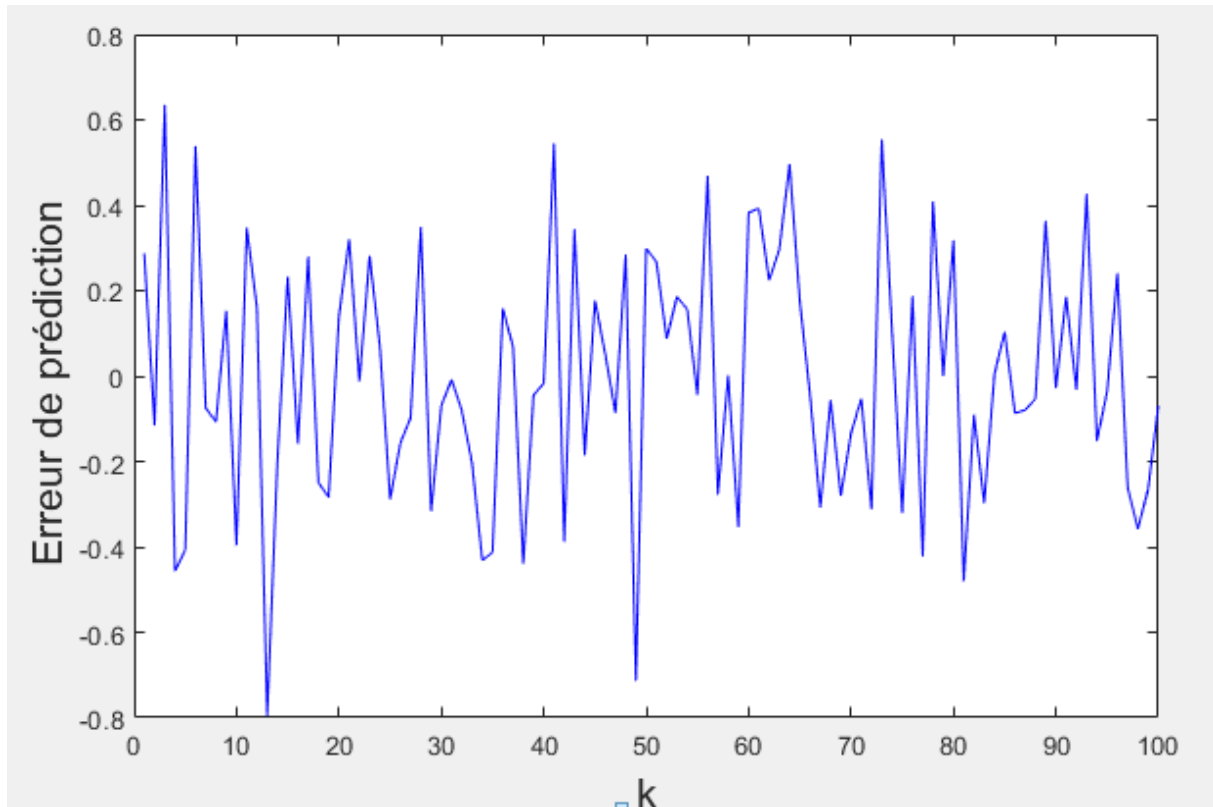


a. Erreur de prédiction

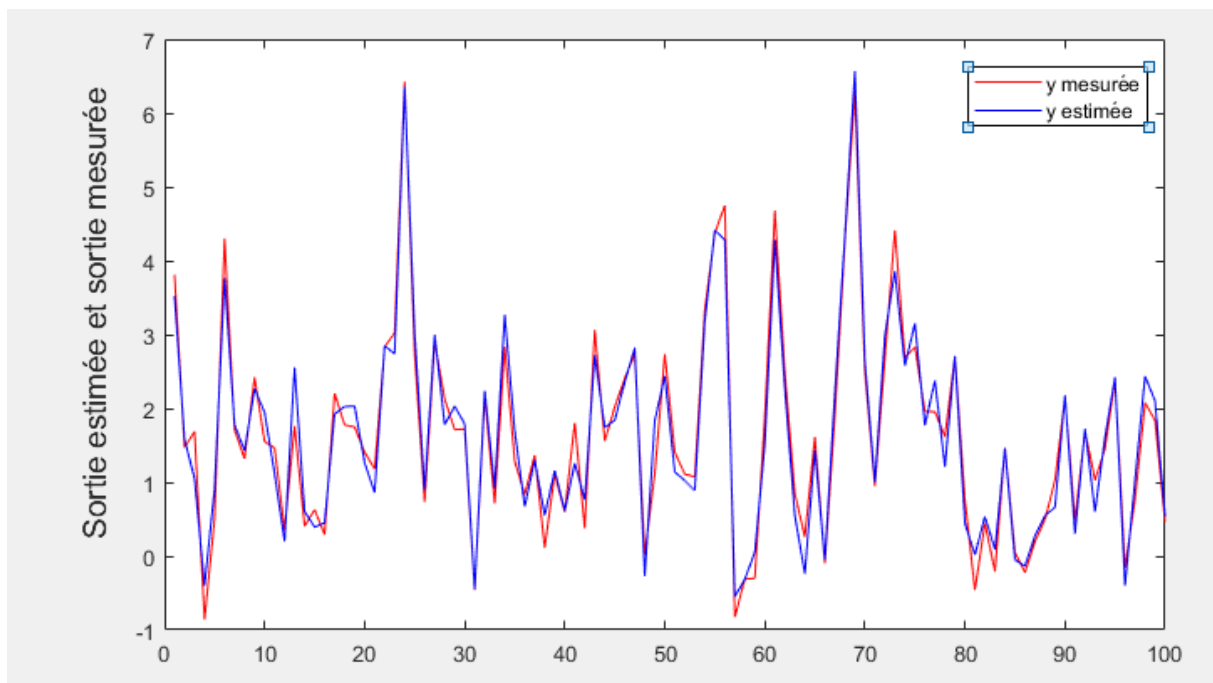


a. Sortie estimée et sortie mesurée

FIGURE 0.12 : Résultats de simulation avec l'algorithme PSO pour SNR=34dB



a. Erreur de prédiction



b. Sortie estimée et sortie mesurée

FIGURE 0.13: Résultats de simulation avec l'algorithme PSO pour SNR=26dB

3.2.2 Exemple 2 :

Dans cet exemple, le modèle de Wiener est testé avec la partie linéaire d'ordre $n_a = 2$ et l'ordre de la partie non linéaire $r = 3$.

Les matrices du bloc linéaire sont données comme suit :

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ -0.7 & -0.2 \end{bmatrix} B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = [0.5 \quad 0.3] \quad D = [0.1]$$

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

$$f(v(k)) = v(k) + 2v^2(k) + 0.1 v^3(k) \quad (3.4)$$

Le vecteur de paramètre θ à estimer est donné par :

$$\theta = [-0.7 \quad -0.2 \quad 0.5 \quad 0.3 \quad 0.1 \quad 2.0 \quad 0.1] \quad (3.5)$$

Le Tableau 3.2 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 la méthode peut être appréciée.

Tableau 0.3: Résultats d'identification du modèle de Wiener PNLSS avec PSO pour $n_a = 2$ et $r = 3$

Paramètres	Valeurs exactes	SNR=0 dB	SNR=34 dB	SNR=26dB
a_1	-0.700	-0.700	-0.697	-0.683
a_2	-0.200	-0.200	-0.203	-0.194
c_1	0.500	0.500	0.494	0.515
c_2	0.300	0.300	0.290	0.323
d	0.100	0.100	0.101	0.113
p_2	2.000	2.000	2.030	1.847
p_3	0.100	0.100	0.112	0.119
J	-	8.248e-012	0.061	0.137

Chapitre 3 : Résultats de Simulation et Discussion

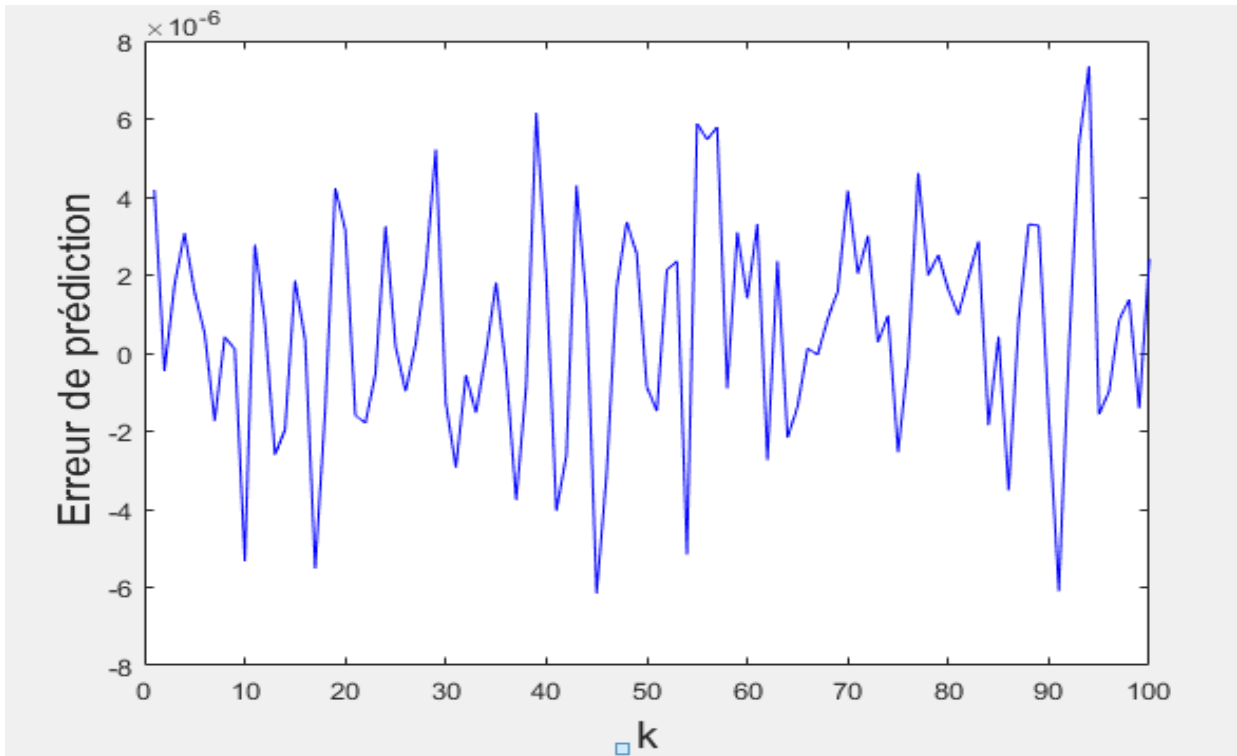
En utilisant le résultat ci-dessus, la Tableau (3.2) fournit les résultats de la simulation des paramètres estimés, effectuée en absence et en présence de bruit pour différents SNR.

L'erreur de prédiction de la sortie pour le cas sans bruit, représentée sur la figure (3.4(a)), est extrêmement faible et s'annule avec un critère ($8.2488e-12$). Indiquant ainsi par la figure (3.4(b)), une très bonne correspondance de la réponse de la sortie du modèle estimé et celle du modèle en absence de bruit,

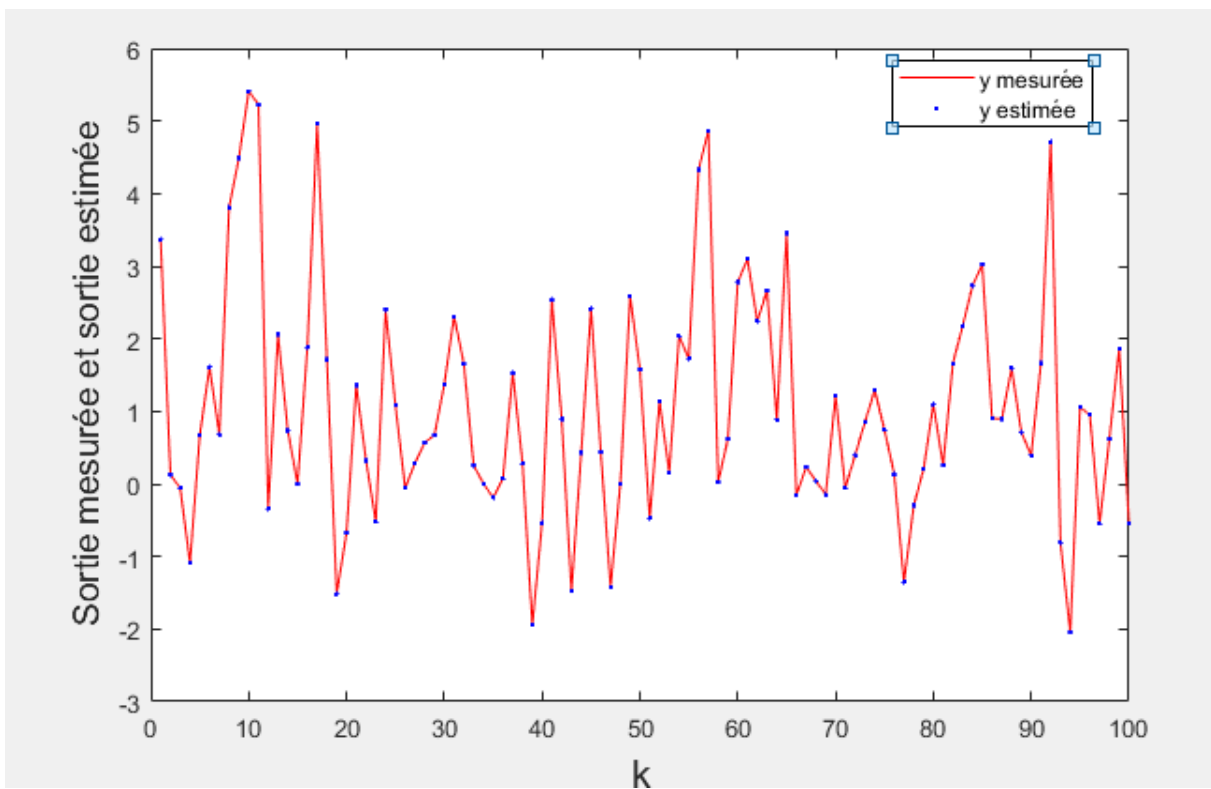
Les Figures (3.5) et (3.6), représentant la sortie du système estimé et celle simulée en présence du bruit, pour $SNR = 34$ dB et $SNR = 26$ dB, montrent un petit dépassement.

Pour l'erreur de prédiction de la sortie pour le cas $SNR=34$ dB et $SNR = 26$ dB représentée sur la figure (3.5(a)) et la figure (3.6(a)), on remarque que les valeurs augmentent respectivement. L'augmentation de J avec le SNR montre que la qualité de l'identification du modèle se dégrade lorsque le bruit augmente.

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.

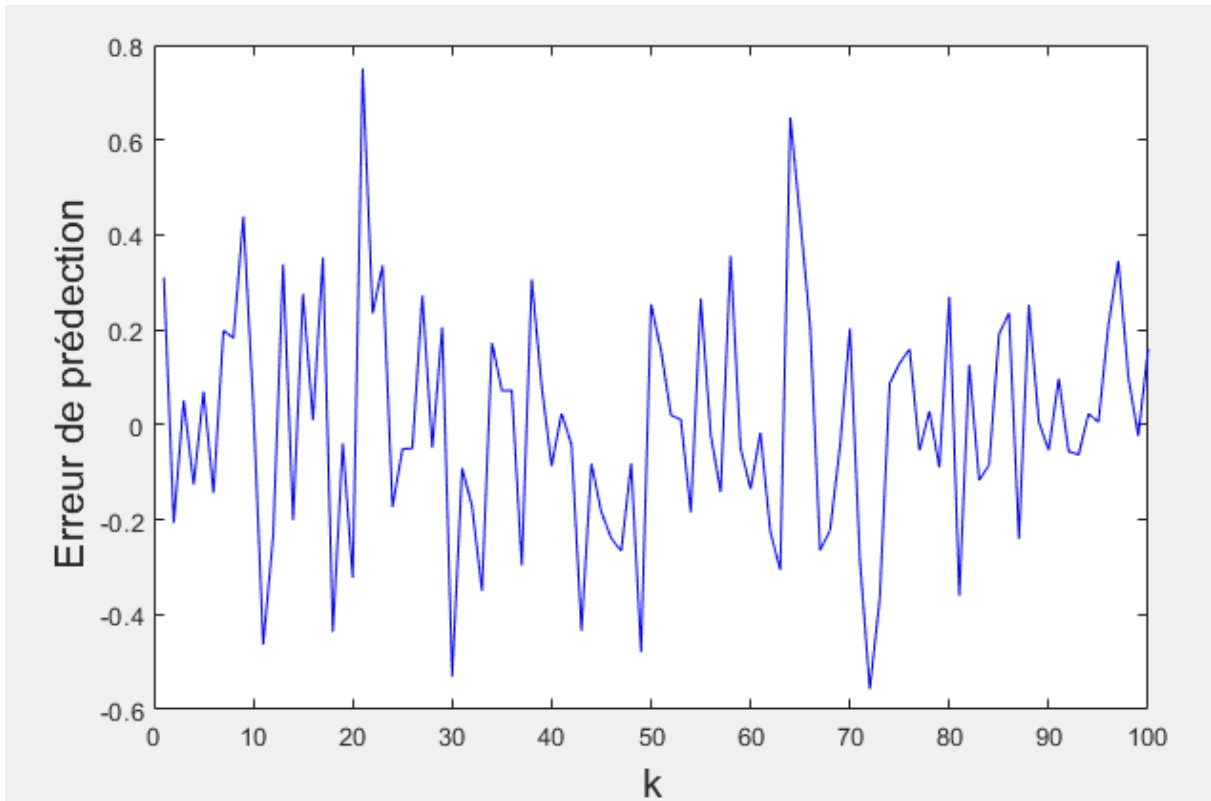


a. Erreur de prédiction

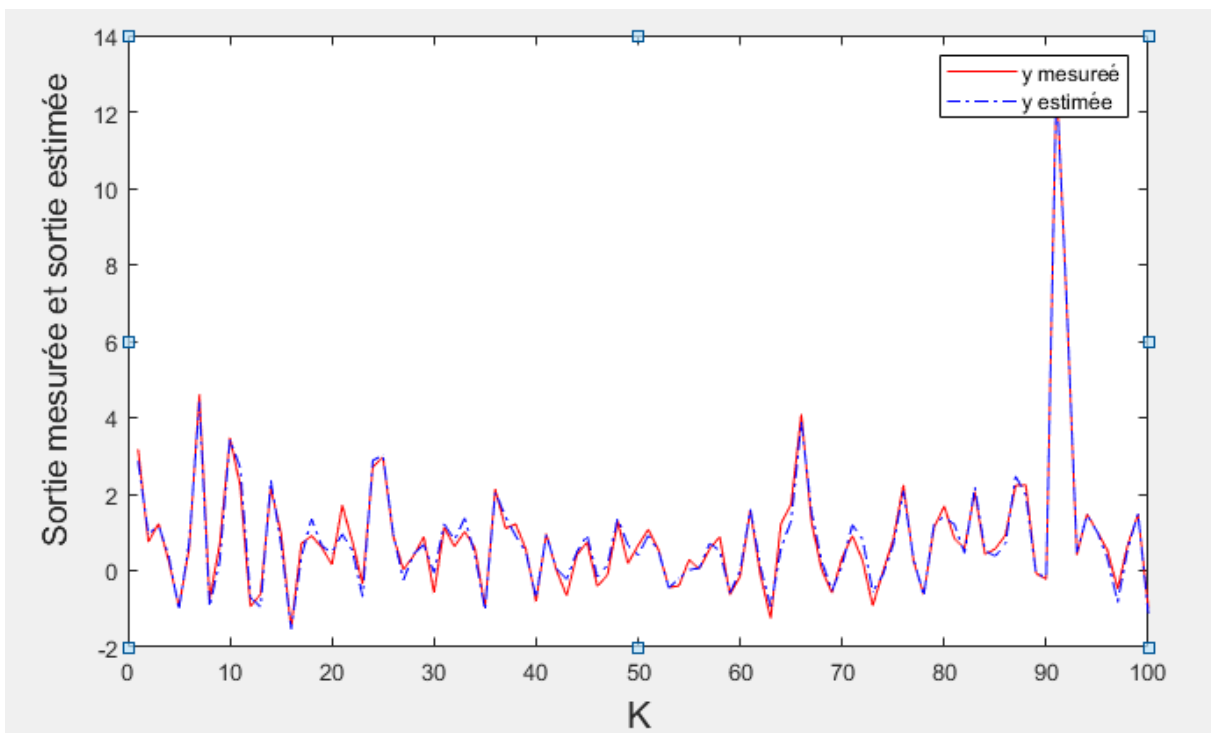


c. Sortie estimée et sortie mesurée

FIGURE 0.14 : Résultats de simulation avec l'algorithme PSO pour SNR= 0 dB

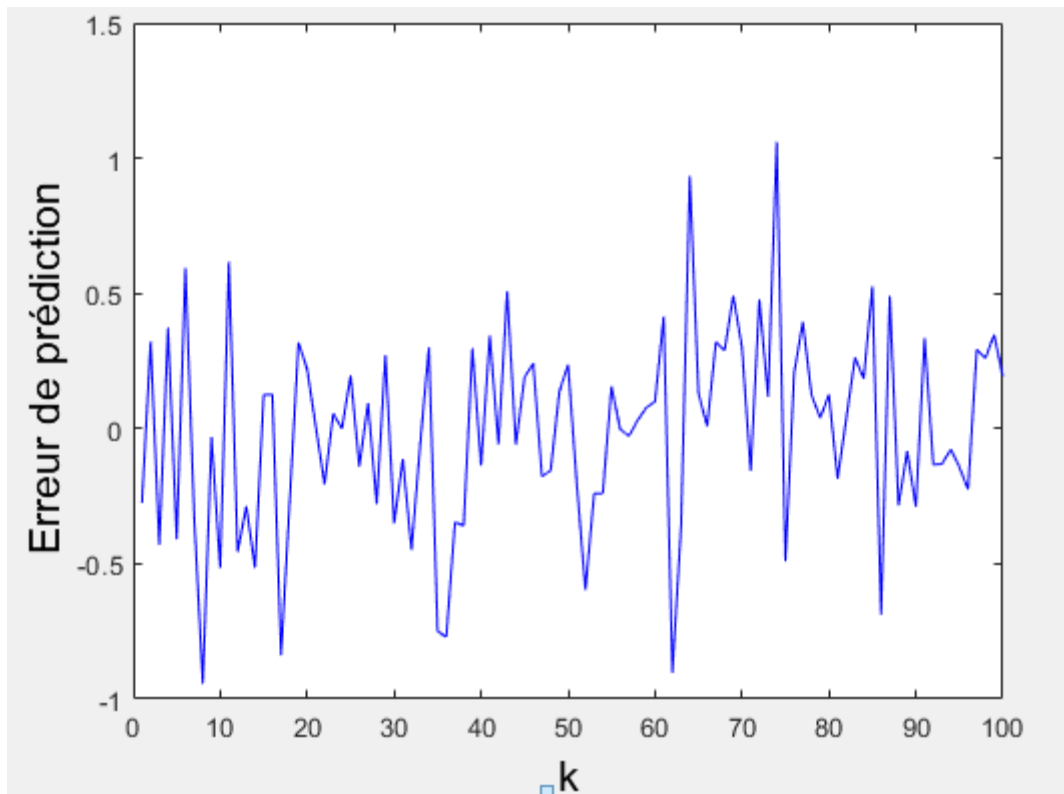


a. Erreur de prédéction

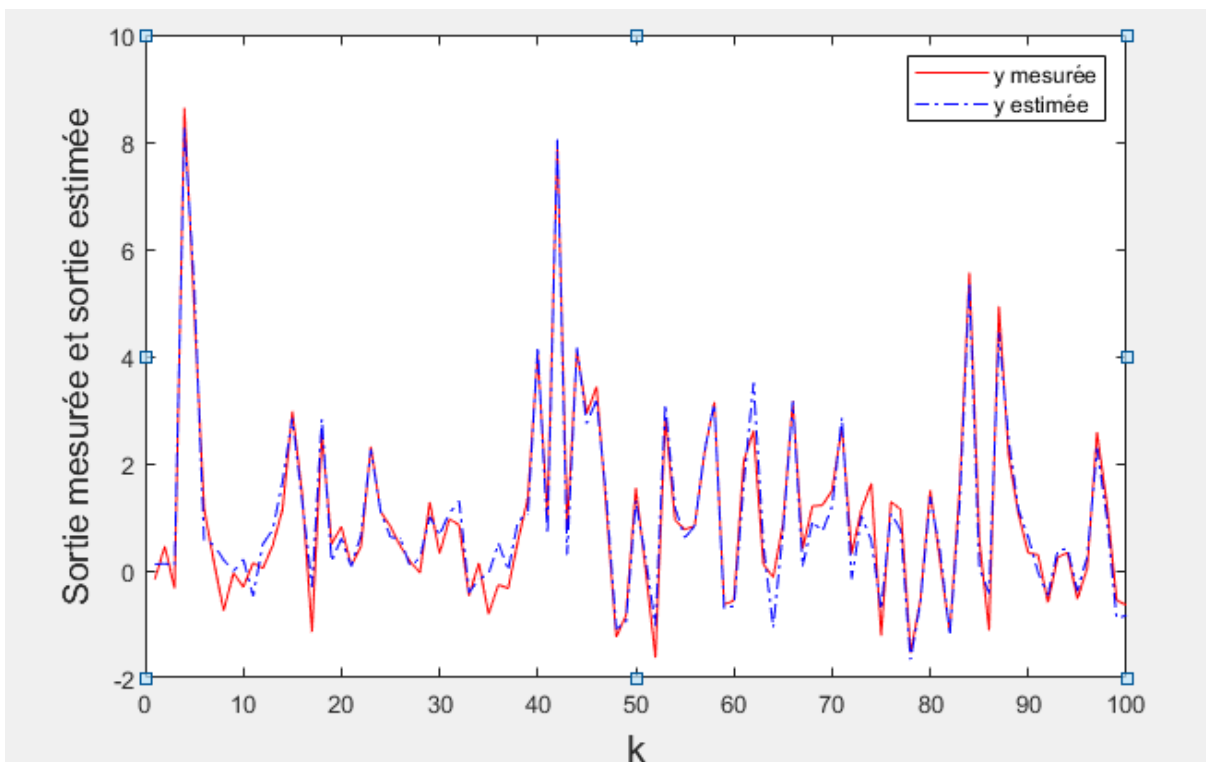


a. Sortie estimée et sortie mesurée

FIGURE 0.15: Résultats de simulation avec l'algorithme PSO pour SNR=34 dB



b. Erreur de prédiction



a. Sortie estimée et sortie mesurée

FIGURE 0.16: Résultats de simulation avec l'algorithme PSO pour SNR=26dB

3.3 Conclusion :

Ce chapitre est consacré à présenter les résultats de simulations, obtenus, en utilisant l'algorithme de PSO. Le modèle PNLSS est utilisé pour décrire la structure du système Wiener. Deux exemples de simulation ont été étudiés et observer pour tester l'algorithme choisis pour l'identification du système de Wiener.

Les résultats obtenus montrent l'efficacité de la méthode PSO pour l'identification du modèle de Wiener PNLSS pour différents niveaux bruit SNR. En effet la qualité de l'identification diminue avec l'augmentation du bruit.

Conclusion générale

Notre travail de mémoire se porte essentiellement sur la description et l'identification des systèmes non linéaires, notamment les systèmes blocs orientés de type Wiener.

Dans le premier chapitre, les généralités sur les structures non-linéaires sont présentées, nous nous sommes intéressés plus particulièrement aux systèmes blocs structurés de type Wiener. Ce dernier, permet de reproduire fidèlement le comportement de la plupart des systèmes réels. La description du modèle de Wiener a été abordée, sous forme de la représentation PNLSS.

Le deuxième chapitre est consacré à l'identification des systèmes blocs orientés. Après des généralités sur l'identification, notre attention s'est focalisée sur les méthodes d'identification les plus utilisées. Tels que les méthodes d'identification en temps continu et en temps discret, les méthodes de base ainsi que l'algorithme à erreur d'équation et la méthode à erreur de sortie, de plus la méthode PSO.

Dans le troisième chapitre, nous avons présenté des exemples de simulation pour l'identification du système Wiener avec la méthode PSO. L'interprétation des principaux résultats obtenus montre l'efficacité de la méthode d'identification utilisée, même en présence d'un pourcentage de bruit important.

Les résultats présentés dans ce mémoire ouvrent également de multiples perspectives :

- Identification d'autres modèles blocs orientés.
- Généralisation aux cas multi variables.
- Extension aux cas fractionnaire
- Validation des résultats obtenus sur des systèmes réels.

Bibliographies

- [1] L. Ljung, System identification: Theory for the User, 1987.
- [2] S.A. Billings, Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, Wiley, 2013.
- [3] 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.
- [4] L. Zhou, X. Li, H. Xu, P. Zhu, Levenberg-Marquardt Iterative Algorithm for Hammerstein Nonlinear Systems, 2015 International Conference on Control. Automation and Information Sciences (ICCAIS) 29-31, Changshu, China, October (2015).
- [5] J. Paduart, L. Lauwers, R. Pintelon, J. Schoukens, Identification of a Wiener-Hammerstein system using polynomial non-linear state space approach, Control Engineering practice, 20, 1133-1139 (2012).
- [6] O. Nelles, Nonlinear system identification. From classical approaches to neural networks and fuzzy models, Springer, New York, Berlin, Heidelberg, 2001.
- [7] K. HAMMAR, Identification des systèmes bloc-structurés Fractionnaires non-linéaires, TIZI- Thèse de Doctorat, UNIVERSITÉ MOULOUD MAMMERI DE OUZOU, 22 Mars 2021.
- [8] 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.
- [9] R. Pintelon et J. Schoukens "Nonlinear System Identification: A User-Oriented Road Map" (IEEE Transactions on Instrumentation and Measurement, 2001).

- [10] Oliver Nelles. Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models.
- [11] N. Wiener: Nonlinear problems in random theory, Wiley, New York, (1958).
- [12] W.J. Rugh, Nonlinear System Theory, The Volterra/Wiener Approach. The Johns Hopkins University Press (1981).
- [13] A. Maria Nagy, Analyse et synthèse de multimodales pour le diagnostic. Application à une station d'épuration Automatique / Robotique. Institut National Polytechnique de Lorraine - INPL, 2010.
- [14] 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.
- [15] G. A. Pajunen, "Adaptive control of Wiener type nonlinear systems", Automatica, vol. 28, no. 4, pp. 781-785, 1992.
- [16] 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.
- [17] S. Riccardo. Risuleo, Nonparametric models for Hammerstein-Wiener and Wiener-Hammerstein system identification, ACCESS Linnaeus Centre, School of Electrical Engineering, KTH – Royal Institute of Technology, Stockholm, Sweden, 2020.
- [18] 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.
- [19] A. Herizi, Généralité sur l'identification.
- [20] H. GARNIER, M. Gilson, T. Bastogne, identification de modèle paramétrique à temps continu Méthode, outil logiciel et avantages, Journées Identification et Modélisation Expérimentale JIME'2006 – 16 et 17 novembre – Poitiers.

- [21] G. MERCÈRE, R. OUVRARD, M. GILSON, H. GARNIER, Identification de systèmes multivariables à temps continu par approche des sous-espaces, Laboratoire d'Automatique et d'Informatique Industrielle, EA 1219 École Supérieure d'Ingénieurs de Poitiers, 40, avenue du Recteur Pineau 86022 Poitiers Cedex, France.
- [22] P. BONNET " Modélisation Identification des Processus", Université Lille 1,2010-2011.
- [23]. C. Young, Parameter estimation for continuous time models, A survey. Automatica. Vol. 17, n° 1, pp 23-39, 1981.
- [24] J.C Trigeassou, (1996), Estimation paramétrique des systèmes continus. Ecole d'été, surveillance des systèmes continus Technique et Documentation. Edition Lavoisier, Paris 1996.
- [25] G.P. Rao, H. Garnier, Identification of continuous time systems, direct or indirect Systems Science. Vol. 30, n° 3, pp25-50, 2004.
- [26] J. Richalet, A. Rault, R. Pouliquen, Identification des processus par la méthode du modèle, Gordon and Breach, 1971.
- [27] C.T. Kelley, Iterative Methods for Optimization, SIAM, Philadelphia, 1999.
- [28] T.A. Johansen, M.S. Waterman, Identification of Non-Linear Systems using Empirical Data and Prior Knowledge-an Optimization Approach, Automatica. Vol. 32, pp 337-356, 1996.
- [29] J. Nocedal, S. Wright, Numerical Optimization. Springer, 2000.
- [30] K. Levenberg. A, method for the solution of certain problems in least squares. Quart. Appl. Math. 2. pp 164-168, 1944.
- [31] D.W. Marquardt, An algorithm for least squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics. Vol. 11, n° 2, pp 431-441, Paris France, 1963.
- [32] T. Djamah, M. Bettayeb, S. Djennoune, Identification of multivariable fractional order systems. Asian J. Control 15(3), 741-750 (2013).
- [33] J. Kennedy, and R.C. Eberhart, Particle Swarm Optimization". Proc. IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948, 1995.

- [34] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, Academic Press, San Fransisco, 2001.
- [35] K. S. Narendra, P. G. Gallman, An iterative method for the identification of nonlinear systems using a Hammerstein model, *IEEE Trans. Automat. Contr.*, 11, 546-550 (1966).
- [36] F. Ding, Y. Shi, T. Chen, , Auxiliary model based least-squares identification methods for Hammerstein output-error systems. *Systems & Control Letters*, 56(5), 373-380 (2007).
- [37] F. Giri, Y. Rochdi, F. Z. Chaoui, A., Brouri, Identification of Hammerstein systems in presence of hysteresis-backlash and hysteresis-relay nonlinearities, *Automatica*, 44(3), 767-775 (2008).
- [38] D.Q. Wang, F. Ding, Y.Y. Chu, Data filtering based recursive least squares algorithm for Hammerstein systems using the key-term separation principle. *Inf. Sci.* 222(10), 203-212 (2013).
- [39] J. Ma, W. Xiong, F. Ding, A. Alsaedi, T. Hayat, Data filtering-based forgetting factor stochastic gradient algorithm for Hammerstein systems with saturation and preload nonlinearities. *Journal of the Franklin Institute*, 353(16), 4280-4299 (2016).
- [40] F. Ding, K. Deng, X. Liu, Decomposition based Newton iterative identification method for a Hammerstein nonlinear FIR system with ARMA noise, *Circuits, Systems, and Signal Processing*, 33(9), 2881-2893 (2014).
- [41] L. Ljung, A. Hagenblad, A. Wills, Maximum likelihood identification of Wiener models, *Automatica*, 44(11), 2697-2705 (2008).
- [42] Y. Hu, B. Liu, Q. Zhou, C. Yang, Recursive extended least squares parameter estimation for Wiener nonlinear systems with moving average noises. *Circuits, Systems, and Signal Processing*, 33(2), 655-664 (2014).

- [43] J. Vörös, Identification of nonlinear cascade systems with output hysteresis based on the key term separation principle. *Applied Mathematical Modelling*, 39(18), 5531-5539 (2015).
- [44] S. Bedoui, M. Ltaief, K. Abderrahim, Gradient based identification for discrete time delay systems, In 9th Asian Control Conference (2013).
- [45] E. W. Bai, A blind approach to the Hammerstein-Wiener model identification. *Automatica*, 38(6), 967-979 (2002).
- [46] A. Van Mulders, J. Schoukens, M. Volckaert, M. Diehl, Two nonlinear optimization methods for black box identification compared, *Automatica*, 46(10), 1675-1681(2010).
- [47] D. Wang, F. Ding, Extended stochastic gradient identification algorithms for Hammerstein-Wiener ARMAX systems, *Journal of Computers and Mathematics with Applications* 56, 3157-3164 (2008).
- [48] Y. Wang, F. Ding, Recursive least squares algorithm and gradient algorithm for Hammerstein-Wiener systems using the data filtering. *Nonlinear Dynamics*, 84(2), 1045-1053 (2016).
- [49] X. S. Luo, Y. D. Song, Data-driven predictive control of Hammerstein-Wiener systems based on subspace identification. *Information Sciences*, 422, 447-46 (2018).
- [50] X. Chen, H. T. Fang, Recursive identification for Wiener-Hammerstein systems using instrumental variable. In *Proceedings of the 10th World Congress on Intelligent Control and Automation* (3043-3048). IEEE. (July 2012).
- [51] H. Ase, T. Katayama, A subspace-based identification of Wiener-Hammerstein benchmark model. *Control Engineering Practice*, 44, 126-137 (2015).
- [52] A. Naitali, F. Giri, Wiener-Hammerstein system identification-an evolutionary approach. *International Journal of Systems Science*, 47(1), 45-61 (2016).

- [53] M. Schoukens, R. Pintelon, Y. Rolain, Identification of Wiener-Hammerstein systems by a nonparametric separation of the best linear approximation. *Automatica*, 50(2), 628-634 (2014).