

THESE

Présentée à
L'UNIVERSITE DE POITIERS

Pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITE DE POITIERS

ECOLE SUPERIEURE D'INGENIEURS DE POITIERS
ECOLE DOCTORALE DES SCIENCES POUR L'INGENIEUR

Diplôme National - Arrêté du 30 mars 1992
SPECIALITE : Automatique

Présentée par

Guillaume BIBES

Modélisation de Procédés de Traitement des Eaux et Reconstruction de Grandeurs Physico-Chimiques

Directeur de Thèse : **Patrick COIRAUT**

Co-encadrement : **Régis OUVRARD**

Présentée et soutenue publiquement le 07 Septembre 2004

COMPOSITION DU JURY

<i>Rapporteurs :</i>	Michel MONSION Christophe AUBRUN	Professeur des Universités, Université de Bordeaux I Professeur des Universités, Université de Nancy
<i>Examinateurs :</i>	Jean-Claude TRIGEASSOU Patrick COIRAUT Jean-Philippe STEYER Régis OUVRARD	Professeur des Universités, Université de Poitiers Professeur des Universités, Université de Poitiers Directeur de Recherche, INRA-LBE Maître de Conférence, Université de Poitiers
<i>Invité :</i>	Joseph DeLAAT	Professeur des Universités, Université de Poitiers

Thèse préparée au sein du Laboratoire d'Automatique et d'Informatique Industrielle de Poitiers
dans le cadre d'une bourse de la Région Poitou-Charentes

Avant-propos

L'ensemble de ces travaux de recherche a été réalisé au sein de l'équipe Identification du Laboratoire d'Automatique et d'Informatique Industrielle (LAI) de l'École Supérieure d'Ingénieurs de Poitiers.

Je tiens à exprimer ma reconnaissance à Monsieur Gérard CHAMPENOIS, Professeur à l'Université de Poitiers et directeur du LAII, pour m'avoir accueilli parmi son équipe et pour la confiance qu'il m'a témoigné.

Ce travail de thèse a été effectué sous la direction de Monsieur Patrick COIRAUT, Professeur à l'Université de Poitiers, que je remercie vivement pour toutes les qualités humaines et scientifiques dont il a fait preuve : ses précieux conseils issus d'un savoir semble-t-il inépuisable, sa constante disponibilité et ses encouragements permanents m'ont été d'un grand secours et m'ont permis, je l'espère, de rendre fructueuses ces années passées sous sa tutelle.

L'encadrement de ce travail de recherche a également été assuré par Monsieur Régis OUVRARD, Maître de Conférence à l'Université de Poitiers, à qui j'exprime ma gratitude pour l'attention, le soutien et la disponibilité dont il a fait preuve depuis le commencement de ce travail jusqu'à la relecture finale du mémoire. Durant l'évolution de cette étude, ses compétences scientifiques, son souci de clarté, de rigueur et de précision ainsi que son expérience pratique m'ont été très profitables.

Je suis sensible à l'honneur que me font Monsieur Michel MONSION, Professeur à l'Université de Bordeaux I, et Monsieur Christophe AUBRUN, Professeur à l'Université de Nancy, en acceptant de juger ce travail de thèse et d'en être les rapporteurs.

Je remercie également Monsieur Jean-Claude TRIGEASSOU, Professeur à l'Université de Poitiers, et Monsieur Joseph DeLAAT, Professeur à l'Université de Poitiers, d'avoir accepté de faire partie de mon jury de thèse.

Je tiens particulièrement à remercier Monsieur Jean-Philippe STEYER, Directeur de Recherche au Laboratoire de Biotechnologie de l'Environnement (LBE-INRA) de Narbonne, pour l'intérêt qu'il a porté à mes travaux. Je lui suis reconnaissant d'avoir accepté de juger ce travail et de présider mon jury de thèse.

Je remercie également la Région Poitou-Charentes pour l'aide financière allouée durant ces années de recherches.

Je ne saurais oublier mes "collègues de promo" ainsi que tous les membres du Labo pour ces quelques années passées en leur compagnie. Je les remercie chaleureusement pour leur aide, leur disponibilité, leur convivialité et tous les bons moments passés avec eux.

Enfin, je tiens à exprimer ma profonde reconnaissance et tout mon amour à Delphine,

devenue mon épouse durant cette thèse, pour la compréhension dont elle a fait preuve à mon égard. Toutes mes pensées vont également vers mes parents, sans le soutien desquels je n'aurais jamais pu concrétiser toutes ces années d'études.

*A Delphine,
A mes parents*

Table des matières

Introduction générale	1
Chapitre 1 Modélisation des systèmes non linéaires par les séries de Volterra	7
1.1 Introduction	8
1.2 Représentation des systèmes non linéaires	8
1.2.1 Introduction	8
1.2.2 Modèles de comportement	9
1.2.2.1 Les modèles NARMAX	9
1.2.2.2 Les multi-modèles	11
1.2.2.3 Les réseaux de neurones artificiels	13
1.2.3 Conclusion	15
1.3 Les séries de Volterra	15
1.3.1 Introduction	15
1.3.2 Les séries de fonctionnelles	16
1.3.2.1 Présentation	16
1.3.2.2 Propriétés	18
1.3.2.3 Conclusion	18
1.3.3 Définition des séries de Volterra	19
1.3.3.1 Introduction	19
1.3.3.2 Propriétés	20
1.3.4 Cas particuliers	21
1.3.4.1 Introduction	21
1.3.4.2 Modèle de Hammerstein	21
1.3.4.3 Modèle de Wiener	22
1.3.4.4 Modèle d'Uryson	24
1.3.4.5 Modèle de projection-poursuite	25

Table des matières

1.3.5	Conclusion	26
1.4	Détermination et modélisation des noyaux de Volterra	26
1.4.1	Introduction	26
1.4.2	Séries génératrices non commutatives	27
1.4.2.1	Expression analytique des noyaux de Volterra	27
1.4.2.2	Commentaire	29
1.4.3	Modélisation des noyaux de Volterra sur bases de fonctions orthonormées	29
1.4.3.1	Introduction	29
1.4.3.2	Définition et propriétés	30
1.4.3.3	Développement sur une base monovariable	30
1.4.3.4	Développement sur une base multivariable	31
1.4.3.5	Développement des noyaux de Volterra [Monsion, 1976] . .	32
1.4.3.6	Choix des fonctions orthonormées	35
1.4.3.7	Identification	37
1.4.3.8	Discussion	42
1.4.4	Modélisation des noyaux de Volterra sur fonctions de transfert . .	43
1.4.4.1	Introduction	43
1.4.4.2	Cas particulier : modèle de Wiener	44
1.4.4.3	Généralisation : utilisation des séries de Fliess	46
1.4.4.4	Identification et discussion	53
1.4.5	Modèle simplifié des noyaux de Volterra sur fonctions de transfert .	59
1.4.5.1	Introduction	59
1.4.5.2	Résultat préliminaire	60
1.4.5.3	Modèle de Volterra simplifié sur fonctions de transfert . .	61
1.4.5.4	Identification et discussion	64
1.4.5.5	Conclusion	66
1.5	Développement autour d'une composante continue	67
1.5.1	Introduction	67
1.5.2	Expression du modèle développé autour d'une composante continue	68
1.5.3	Bases de fonctions orthonormées	68
1.5.4	Fonctions de transfert	70
1.5.5	Identification du modèle sur fonctions de transfert et application .	71
1.5.6	Conclusion	75

1.6	Comparaison des 4 modèles	76
1.7	Conclusion du chapitre	79
Chapitre 2 Inversion d'un modèle de Volterra		81
2.1	Introduction	82
2.2	Les problèmes inverses	83
2.2.1	Introduction	83
2.2.2	Cadre général : les problèmes mal posés	83
2.2.2.1	Introduction	83
2.2.2.2	Cas linéaire	84
2.2.2.3	Application des notions précédentes au modèle non linéaire de Volterra	85
2.2.2.4	Quelques exemples de problèmes inverses	86
2.2.2.5	Conclusion	89
2.2.3	Méthodes algébriques déterministes	89
2.2.3.1	Moindres carrés et inversion généralisée	89
2.2.3.2	Régularisation	93
2.2.4	Méthodes probabilistes	97
2.2.4.1	Utilisation des moments	98
2.2.4.2	Critère du maximum de vraisemblance	100
2.2.4.3	Critère du maximum d'entropie	103
2.2.4.4	Approche bayésienne des problèmes inverses	105
2.2.5	Conclusion	110
2.3	Inversion du modèle de Volterra par régularisation de Tikhonov	110
2.3.1	Introduction	110
2.3.2	Principe de la méthode	111
2.3.2.1	Modèle de Volterra	111
2.3.2.2	Inversion du modèle de Volterra	112
2.3.3	Application en simulation	117
2.3.4	Volume de calcul	127
2.3.5	Conclusion	131
2.4	Inversion du modèle de Volterra par retour d'entrée reconstruite [George, 1959]	132
2.4.1	Introduction	132
2.4.2	Première approche	132
2.4.2.1	Principe	132

Table des matières

2.4.2.2 Application à un système non linéaire	135
2.4.2.3 Influence du bruit de mesure sur la reconstruction	138
2.4.3 Deuxième approche	140
2.4.3.1 Principe	140
2.4.3.2 Application à un système non linéaire	142
2.4.3.3 Influence du bruit de mesure sur la reconstruction	144
2.4.4 Comparaison des deux approches	145
2.4.5 Conclusion	147
2.5 Conclusion du chapitre	149
Chapitre 3 Application à des procédés	151
3.1 Introduction	152
3.2 Généralités à propos du traitement des eaux	152
3.3 Etude en simulation : procédé d'oxydation de polluant	153
3.3.1 Description du procédé étudié	154
3.3.2 Modèle cinétique établi expérimentalement	155
3.3.3 Modèle cinétique d'un réacteur chimique continu	161
3.3.3.1 Présentation	161
3.3.3.2 Utilisation du modèle cinétique du réacteur chimique . .	163
3.3.4 Application du modèle de Volterra	167
3.3.5 Inversion de modèle de Volterra par régularisation de Tikhonov .	170
3.3.6 Inversion de modèle de Volterra par retour d'entrée reconstruite .	172
3.3.7 Conclusion	175
3.4 Etude expérimentale : procédé de digestion anaérobiose	176
3.4.1 Description du procédé étudié	176
3.4.2 Le modèle dynamique	181
3.4.3 Application du modèle de Volterra au digesteur anaérobiose . .	185
3.4.3.1 Modélisation	185
3.4.3.2 Reconstruction de grandeurs par inversion du modèle de Volterra	194
3.4.4 Conclusion	204
3.5 Conclusion du chapitre	205
Conclusion générale	207

Annexe	211
1 Preuve du théorème 1.3	211
2 Cas particulier du théorème 1.3	215
Bibliographie	219

Table des matières

Table des figures

1.1	Schéma d'un neurone artificiel	14
1.2	Modèle de Hammerstein	21
1.3	Modèle de Wiener	23
1.4	Modèle d'Uryson	24
1.5	Modèle de projection-poursuite	25
1.6	Noyau d'ordre 2 développé sur 2 fonctions orthonormées	34
1.7	Noyau d'ordre i développé sur une base de fonctions orthonormées	34
1.8	Approximation de la sortie $y(t)$ du système non linéaire	35
1.9	Séquence pseudo-aléatoire bornée en amplitude	37
1.10	Réponse du système non linéaire	38
1.11	Signaux d'entrée et de sortie utilisés pour valider le modèle de Volterra . .	39
1.12	Modèle de Volterra tronqué à 2 noyaux développés sur 4 fonctions	40
1.13	Modèle de Volterra tronqué à 2 noyaux développés sur 5 fonctions	41
1.14	Modèle de Volterra tronqué à 2 noyaux développés sur 4 fonctions avec choix convenable des pôles	42
1.15	Modèle de Wiener	44
1.16	Modèle de Wiener représentant le premier noyau de Volterra	45
1.17	Modèle de Wiener représentant le noyau de Volterra d'ordre i	45
1.18	Noyau de Volterra d'ordre i modélisé par un produit de fonctions de transfert	46
1.19	Modèle du noyau d'ordre 3	51
1.20	Structure d'un schéma-bloc	51
1.21	Modèle du noyau d'ordre n	52
1.22	Structure d'un schéma-bloc	53
1.23	Réponse du système non linéaire	54
1.24	Modélisation du premier noyau	55
1.25	Modélisation du deuxième noyau	55
1.26	Résultats de modélisation et d'identification paramétrique	58
1.27	Représentation du noyau d'ordre n	63
1.28	Modèle global développé sur des fonctions de transfert	64
1.29	Résultats de modélisation et d'identification paramétrique	66
1.30	Partie dynamique des signaux	73
1.31	Sortie du système non linéaire (ligne continue) et du modèle de Volterra (ligne pointillée)	74
1.32	Résultats de modélisation et d'identification paramétrique	74
1.33	Résultats de modélisation sur bases de fonctions généralisée	76

1.34 Comparaison des résultats de modélisation sur fonctions de transfert	77
2.1 Modèle d'observation	99
2.2 Déconvolution par filtrage de Wiener	100
2.3 Comparaison entre sortie du système non linéaire (ligne continue) et sortie du modèle de Volterra (ligne pointillée)	118
2.4 Entrée du système non linéaire (ligne continue) et entrée reconstruite (ligne pointillée)	119
2.5 Comparaison entre les sorties du système non linéaire et du modèle de Volterra	120
2.6 Reconstruction de l'entrée $u(t)$ pour différentes valeurs de λ	121
2.7 Reconstruction de l'entrée $u(t)$ pour $\lambda = 0.005$	122
2.8 Reconstruction de l'entrée $u(t)$ pour différents niveaux de bruit de la sortie mesurée $y(t)$	123
2.9 Reconstruction de l'entrée $u(t)$ pour différentes valeurs de λ et L matrice des dérivées secondes	126
2.10 Horizon I de minimisation de critère	128
2.11 Horizon $J + I$ de reconstruction	128
2.12 Reconstruction de l'entrée $u(t)$ sur une fenêtre glissante	130
2.13 Reconstruction de l'entrée $u(t)$ sur une fenêtre glissante pour $\lambda = 0.001$	131
2.14 Schéma de reconstruction de l'entrée $u(t)$	133
2.15 Schéma équivalent de reconstruction de l'entrée $u(t)$	134
2.16 Simulation du système non linéaire (2.162)	136
2.17 Sortie du système non linéaire (ligne continue) et du modèle de Volterra (ligne pointillée)	136
2.18 Comparaison de l'entrée du système non linéaire (courbe continue) avec l'entrée reconstruite (ligne pointillée)	137
2.19 Ajout de bruit sur le signal de sortie du système	138
2.20 Sortie bruitée (courbe claire) et sortie du modèle de Volterra (courbe foncée)	139
2.21 Comparaison de l'entrée du système non linéaire (courbe foncée) avec l'entrée reconstruite (courbe claire)	140
2.22 Schéma de reconstruction de l'entrée $u(t)$	141
2.23 Schéma équivalent de reconstruction de l'entrée $u(t)$	141
2.24 Entrée du système non linéaire (ligne continue) et entrée reconstruite (ligne pointillée) pour différentes valeurs de K_1 et K_2	143
2.25 Comparaison de l'entrée du système non linéaire (courbe foncée) avec l'entrée reconstruite (courbe claire)	145
2.26 Schéma de reconstruction de l'entrée $u(t)$ par la deuxième approche avec $K_1 = K_2 = 1$	146
2.27 Schéma de reconstruction de l'entrée $u(t)$ par la première approche	146
3.1 Procédé d'oxydation $H_2O_2/Fe(II)$ pour $[At]_{in} = 8.10^{-7} mol/l$ et $[Fe(II)] = 2.10^{-4} mol/l$	164
3.2 Procédé d'oxydation $H_2O_2/Fe(III)$ pour $[At]_{in} = 8.10^{-7} mol/l$ et $[Fe(III)] = 2.10^{-4} mol/l$	165

3.3	Procédé d'oxydation $H_2O_2/Fe(II)$ pour $[H_2O_2]_{in} = 1.10^{-2} mol/l$ et $[Fe(II)]_{in} = 2.10^{-4} mol/l$	166
3.4	Débit d'atrazine en entrée du réacteur (l/s)	167
3.5	Variations de concentrations (mol/l) donnée par le modèle cinétique du réacteur (courbe continue) et le modèle linéaire (courbe pointillée)	168
3.6	Variations de concentrations (mol/l) donnée par le modèle cinétique du réacteur (courbe continue) et par le modèle de Volterra (courbe pointillée)	169
3.7	Variations données par le modèle cinétique du réacteur (courbe continue) et par le modèle de Volterra (courbe pointillée)	169
3.8	Débit d'atrazine (l/s) (courbe continue) et sa reconstruction (ligne pointillée)	171
3.9	Concentration en H_2O_2 (mol/l) dans le réacteur (courbe claire) et en sortie du modèle de Volterra (courbe foncée)	171
3.10	Comparaison entre le débit d'atrazine (l/s) (courbe foncée) et sa reconstruction (courbe claire) pour différentes valeurs de λ avec utilisation de la matrice des dérivées seconde	172
3.11	Variations de concentrations (mol/l) donnée par le modèle cinétique du réacteur (courbe continue) et le modèle de Volterra (courbe pointillée)	173
3.12	Variations de concentrations (mol/l) données par le modèle cinétique du réacteur (courbe claire) et le modèle de Volterra (courbe foncée)	174
3.13	Débit d'atrazine (l/s) en entrée du réacteur (courbe foncée) et sa reconstruction (courbe claire)	175
3.14	Digesteur anaérobiose en lit fixe à flux ascendant	178
3.15	Mesures effectuées en ligne	179
3.16	Autres mesures effectuées en ligne	180
3.17	Mesures effectuées par spectrométrie infrarouge	181
3.18	Concentrations en biomasse	183
3.19	Comparaison des mesures obtenues en ligne (courbes claires) avec les résultats de simulation (courbes foncées)	184
3.20	Débit d'entrée	186
3.21	Taux de croissance bactérienne	187
3.22	Fichiers de points utilisés pour l'identification de modèles de Volterra	188
3.23	Structure du modèle de Volterra	189
3.24	Comparaison entre sorties des modèles de Volterra (ligne pointillée) et sorties du modèle dynamique (ligne continue)	190
3.25	Validation expérimentale des modèles de Volterra (courbes claires : données expérimentales - courbe foncée : sortie du modèle de Volterra)	192
3.26	Modélisation du lien entre débit d'entrée et DCO en sortie	194
3.27	Comparaison entre $D(t)$ (courbe continue) et sa reconstruction (courbe pointillée)	195
3.28	Comparaison entre DCO bruitée (courbe claire) et DCO en sortie du modèle de Volterra (courbe foncée)	195
3.29	Comparaison entre $D(t)$ (courbe foncée) et sa reconstruction (courbe claire) pour différentes valeurs de λ	196
3.30	Mesures expérimentales du débit d'entrée et de la DCO en sortie	196

Table des figures

3.31 Débit d'entrée expérimental (courbe foncée) et sa reconstruction (courbe claire)	197
3.32 Modélisation du lien entre DCO en entrée et DCO en sortie	198
3.33 Comparaison entre la DCO en entrée (courbe continue) et sa reconstruction (courbe pointillée)	198
3.34 Comparaison entre DCO bruitée (courbe claire) et DCO en sortie du modèle de Volterra (courbe foncée)	199
3.35 Comparaison entre la DCO en entrée (courbe foncée) et sa reconstruction (courbe claire) pour différentes valeurs de λ	199
3.36 Signaux d'entrée et de sortie	200
3.37 Comparaison entre débit d'entrée reconstruit (courbe claire) et mesuré (courbe foncée) pour les 2 méthodes de reconstruction	201
3.38 Mesures expérimentales du débit d'entrée et de la DCO en sortie	201
3.39 Débit d'entrée expérimental (courbe claire) et sa reconstruction (courbe foncée)	202
3.40 Débit d'entrée expérimental filtré (courbe claire) et sa reconstruction (courbe foncée)	202
3.41 Modélisation du lien entre DCO en entrée et DCO en sortie	203
3.42 DCO en entrée (courbe foncée) et DCO reconstruite (courbe claire)	204

Introduction générale

L'étude des systèmes est depuis longtemps une préoccupation majeure du monde de la recherche et ce, quel que soit le domaine scientifique. La compréhension d'un système permet en effet d'agir ensuite sur son comportement naturel et de lui imposer une autre conduite répondant à des performances souhaitées par l'utilisateur. Afin de mieux comprendre un système, une approche naturelle consiste à essayer d'en reproduire le comportement. Une telle démarche implique l'établissement d'un modèle du système étudié. Selon la connaissance des phénomènes physiques mis en jeu dont on dispose, plusieurs possibilités de modélisation peuvent se présenter. Les modèles de connaissance reposent sur une très bonne connaissance des lois physiques qui régissent le procédé étudié. A l'opposé, les modèles de comportement sont purement mathématiques et ne nécessitent aucune connaissance a priori sur la physique du système considéré. Il est évidemment possible d'effectuer un compromis entre ces deux formes de modélisation lorsque la connaissance physique du système est imparfaite.

Quelle que soit leur nature (physique, mathématique ou composition des deux), les modèles ont pour la plus grande partie un point commun : ils sont caractérisés par un certain nombre de variables que l'on nomme communément paramètres. Ces paramètres peuvent avoir une signification physique, en particulier dans le cas d'un modèle de connaissance. De la valeur de ces paramètres dépend en grande partie la qualité du modèle et sa capacité à représenter le comportement du système réel sur un domaine de fonctionnement spécifique.

Une fois définie la forme du modèle la plus apte à représenter correctement le fonctionnement d'un système réel, l'identification des paramètres du modèle consiste à calculer leur valeur optimale. Ces valeurs paramétriques optimales permettent au modèle de représenter fidèlement le système sur le domaine de fonctionnement choisi. En l'absence de connaissance du système, le choix de modélisation se porte naturellement sur un modèle de comportement et l'identification paramétrique ne peut être effectuée qu'à l'aide des seules données disponibles : des signaux d'entrée judicieusement choisis et la réponse du système à ces signaux d'entrée. Une méthode bien connue d'identification paramétrique est la méthode classique des moindres carrés, largement utilisée si le modèle employé peut être exprimé sous forme linéaire par rapport à ses paramètres.

L'essentiel des systèmes réels sont non linéaires par nature. Pourtant, la complexité d'étude et d'implémentation des modèles non linéaires ainsi que l'insuffisance des moyens de simulation disponibles ont d'abord conduit à essayer d'approcher le comportement des

Introduction générale

systèmes réels par des modèles linéaires. Ce type de modèle peut s'avérer efficace si le degré de non-linéarité du système reste faible ou bien si l'on ne s'éloigne pas d'un point de fonctionnement particulier du système. On peut ensuite envisager l'implémentation d'une loi de commande afin d'imposer un comportement au système réel. Cependant, il existe de nombreux cas où les non-linéarités du système étudié ne sont pas négligeables. On doit donc envisager une modélisation non linéaire. Ce domaine de recherche connaît un essor théorique et pratique important car les moyens technologiques actuels permettent la simulation de modèles non linéaires complexes ainsi que la simulation et l'implémentation de lois de commandes évoluées sur des systèmes non linéaires réels [Sjöberg *et al.*, 1995].

Un système non linéaire se démarque d'un système linéaire dans le fait que sa dynamique dépend étroitement du signal d'entrée, qui induit un comportement différent du système non linéaire suivant sa variation d'amplitude et/ou de fréquence. Le comportement du système dépend à son tour des non-linéarités inhérentes à celui-ci. Le choix de la structure du modèle non linéaire doit tenir compte à la fois des non-linéarités et du domaine de fonctionnement sur lequel le modèle doit être valide. Si le modèle est linéaire par rapport aux paramètres, l'identification de ces derniers peut se faire par la méthode des moindres carrés. Dans le cas où le modèle n'est plus linéaire par rapport aux paramètres, un algorithme de programmation non linéaire doit être employé afin d'optimiser la valeur des paramètres. Un algorithme couramment utilisé est celui de Levenberg-Marquardt [Marquardt, 1963], qui consiste en une recherche itérative des paramètres optimaux. Ce type d'algorithme s'appuie lui aussi sur l'utilisation de signaux d'entrée et de sortie du système réel.

L'identification paramétrique repose donc sur l'utilisation de signaux d'entrée et de sortie du système réel [Ljung, 1987]. Dans beaucoup de cas, avec l'amélioration croissante des moyens et des techniques de mesure, le nombre et la précision des capteurs permettent à l'utilisateur de disposer d'un vaste choix de mesures des grandeurs d'entrée et de sortie du système réel. Il peut passer directement à l'étape d'identification paramétrique du modèle. Cependant, dans d'autres cas, la mesure de certaines grandeurs est difficile, voire impossible, du fait de l'absence de capteur adéquat ou du coût prohibitif de ce dernier ou de la mise en place d'un système de mesure. Une solution consiste alors à accéder indirectement à la grandeur non mesurable en résolvant un problème inverse.

Un problème direct consiste à calculer la sortie d'un système lorsque l'on dispose d'un signal d'entrée et d'un modèle suffisamment précis de ce système. Un problème inverse consiste à faire le calcul opposé : étant donné une grandeur de sortie et le modèle du système étudié, on cherche à calculer l'entrée du système. Résoudre un problème direct consiste à déterminer une conséquence à partir d'une cause, résoudre un problème inverse consiste à déterminer une cause connaissant sa conséquence. Dans les deux cas, il est nécessaire de disposer d'un modèle du système réel étudié. Différentes techniques de résolution de problèmes inverses existent et peuvent être appliquées pour obtenir une grandeur non mesurable à partir d'une autre grandeur mesurable [Mohammad-Djafari, 1999b], [Demoment *et al.*]. Etant donnée une grandeur mesurable facilement et un modèle qui la relie à une grandeur non mesurable, la résolution du problème inverse consiste à détermi-

ner la grandeur non mesurable à partir de la grandeur mesurable en inversant le modèle qui les lie. La difficulté de cette méthode réside en l'inversion de modèle, qui peut occasionner une mauvaise reconstruction de la grandeur non mesurable suivant la nature du problème inverse.

La démarche globale entreprise dans ce mémoire peut se décomposer en deux grandes étapes. La première étape consiste en la modélisation de systèmes non linéaires. La seconde étape consiste en l'inversion des modèles non linéaires obtenus afin de déterminer des grandeurs non directement mesurables.

Ce mémoire se compose de trois chapitres.

Le premier chapitre est consacré à la modélisation de systèmes non linéaires par les séries de Volterra [Volterra, 1959], [Schetzen, 1980], [Doyle *et al.*, 2002]. Présentées comme une possibilité parmi d'autres [Chen, 1989], [Boukhris *et al.*, 1999], [Hérault et Jutten, 1994] pour modéliser les systèmes non linéaires, les séries de Volterra peuvent être considérées comme un sous-ensemble des séries de fonctionnelles et un outil particulièrement adapté à la représentation des systèmes non linéaires car elles généralisent la notion de convolution utilisée pour la représentation de systèmes linéaires. Un certain nombre de modèles classiques (Hammerstein, Wiener,...) utilisés depuis longtemps sont présentés comme cas particuliers de modèles de Volterra [Billings et Fakhouri, 1979], [Rugh, 1981], [Ralston et Zoubir, 1995]. Une série de Volterra est entièrement caractérisée par ses noyaux que l'on peut difficilement exprimer de manière analytique et qu'il convient donc de modéliser. La modélisation des noyaux par développement sur une base de fonctions généralisée [Ninness et Gustafsson, 1994], [Akçay et Ninness, 1999] est présentée et l'on met l'accent sur une difficulté majeure de cette approche : une augmentation importante du nombre de paramètres qui rend difficile la procédure d'identification et l'utilisation du modèle à des fins de commande. Une seconde approche de modélisation des noyaux de Volterra est proposée : leur développement sur des fonctions de transfert. Le modèle obtenu s'inspire des séries génératrices non commutatives [Fliess *et al.*, 1983] et permet d'obtenir une parcimonie paramétrique satisfaisante. Afin d'améliorer cette parcimonie, nous proposons également une méthode de développement du modèle de Volterra autour de la composante continue du signal d'entrée. Cette méthode permet de séparer la partie statique de la partie dynamique du modèle. Seule cette dernière est identifiée, réduisant davantage la complexité et le nombre de paramètres du modèle de Volterra global.

Le deuxième chapitre est consacré à l'inversion du modèle de Volterra présenté au premier chapitre afin de reconstruire une grandeur d'entrée en utilisant la mesure de la grandeur de sortie. La première partie est consacrée à la définition des problèmes inverses et à la présentation succincte de quelques exemples. Un aperçu général non exhaustif des méthodes de résolution de ce type particulier de problème est ensuite exposé, dans lequel on distingue deux grandes catégories : les méthodes algébriques déterministes et les méthodes probabilistes [Mohammad-Djafari, 1999b]. Notre choix se porte d'abord sur la méthode d'inversion de modèle par régularisation de Tikhonov (développée à l'origine dans le cas de modèles linéaires [Tikhonov et Arsénine, 1976]) dont nous élargissons le

Introduction générale

champ d'application au modèle de Volterra. Cette méthode nécessite un algorithme d'optimisation non linéaire afin de reconstruire une grandeur par minimisation de critère. La somme de calcul nécessaire devenant importante, nous proposons d'adopter une démarche de discréétisation du modèle de Volterra afin de simplifier le calcul de dérivation du critère et ainsi diminuer la charge totale de calculs. L'influence du bruit de mesure sur les données utilisées ainsi que celle des paramètres de régularisation de la solution au problème inverse sont étudiées dans le cadre de la simulation. Une approche différente (et plus intuitive) est ensuite proposée pour contourner certains inconvénients de la précédente : l'inversion du modèle de Volterra par retour d'entrée reconstruite. Deux méthodes sont présentées, équivalentes dans le cas linéaire mais sensiblement différentes pour des systèmes non linéaires. Elles utilisent toutes deux l'inverse implicite du modèle de Volterra afin de reconstruire point par point une grandeur d'entrée à partir de la mesure d'une grandeur de sortie et d'un modèle de Volterra suffisamment précis qui les relie. Ces deux méthodes possèdent l'avantage de ne pas nécessiter d'étape d'optimisation lors de la procédure d'inversion, diminuant ainsi de manière significative le temps total de calcul et laissant envisager leur possible implémentation en temps réel.

Le troisième et dernier chapitre est consacré à l'application des méthodes de modélisation et d'inversion proposées dans les deux premiers chapitres à des simulations de procédés et des données expérimentales. Les deux procédés étudiés entrent dans le cadre de la dépollution des eaux. Le premier procédé consiste en la neutralisation de composés organiques considérés comme polluants dans des effluents industriels. L'étude est effectuée dans le cas de l'atrazine, un herbicide systémique. Le procédé de dépollution est le procédé Fenton : la décomposition du peroxyde d'hydrogène par des ions ferreux (ou ferriques) permet la création de radicaux hydroxyle hautement réactifs qui vont oxyder le polluant et le transformer en un produit sans danger pour l'environnement [Barbeni *et al.*, 1987], [Sedlak et Andren, 1991], [DeLaat et Gallard, 1999]. Un modèle cinétique du procédé a déjà été établi expérimentalement [Haber et Weiss, 1934], [Barb *et al.*, 1951], [Walling, 1975], [Gallard et DeLaat, 2000]. Dans un premier temps, nous comparons ce modèle cinétique au modèle de Volterra afin de valider ce dernier dans le cadre de la simulation. Dans un second temps, nous appliquons les méthodes d'inversion proposées précédemment afin de reconstruire une grandeur difficilement accessible (la concentration en atrazine) à partir de la connaissance d'autres grandeurs plus facilement mesurables. Cette étude permet de valider en simulation les méthodes proposées dans les chapitres précédents. Le deuxième cas d'étude est un procédé d'épuration biologique de rejets industriels. L'étude porte sur un procédé dit anaérobiose (absence d'oxygène) de dégradation de composé organique : le carbone organique contenu dans l'effluent à traiter est converti en biogaz par des flores bactériennes. Comme dans l'étude précédente, un modèle dynamique du procédé a déjà été développé [Bernard *et al.*, 2001], [Steyer *et al.*, 2003]. Nous commençons par valider notre modèle de Volterra en le comparant à ce modèle établi expérimentalement. Puis, de la même manière que précédemment, l'inversion du modèle de Volterra permet la reconstruction d'une grandeur (la demande chimique en oxygène) quasiment impossible à mesurer directement vu le coût et la complexité d'une telle procédure. Les résultats des deux approches d'inversion du modèle de Volterra sont comparées et validées à nouveau dans le cadre de la simulation. On utilise ensuite ces méthodes sur

des données expérimentales provenant d'un digesteur anaérobiose traitant des vinasses de distilleries.

Une conclusion générale à ce travail, consacrée au rappel de la démarche globale de recherche et des objectifs, s'attachera en outre à la présentation des différents prolongements et perspectives envisageables tant du point de vue théorique qu'appliqué.

Introduction générale

Chapitre 1

Modélisation des systèmes non linéaires par les séries de Volterra

1.1 Introduction

Cette partie relève du domaine très vaste de la modélisation des systèmes et plus particulièrement de la modélisation de systèmes non linéaires.

Pour plusieurs raisons d'ordre pratique, la théorie de la commande s'est surtout développée dans le cadre des systèmes linéaires. Bien que des travaux théoriques existent déjà depuis longtemps dans le domaine des systèmes non linéaires, de nombreuses limitations techniques, informatiques et financières ont freiné toute tentative d'application pratique en modélisation, identification et commande de systèmes non linéaires. Depuis un certain nombre d'années, la technologie et l'informatique ont comblé ce retard vis-à-vis de la théorie. Le domaine particulier de la commande de systèmes non linéaires prend de plus en plus d'importance, les problèmes posés étant de plus en plus nombreux et toujours ouverts, du fait de la quasi absence de systèmes véritablement linéaires ou linéarisables à l'état naturel.

Une étape importante du processus qui conduit l'utilisateur à la commande de systèmes non linéaires est la modélisation de ces mêmes systèmes. La première section du chapitre est consacrée à la présentation de différents modèles de systèmes non linéaires. La spécification d'objectifs particuliers, tels la simplicité et la parcimonie paramétrique du modèle employé ainsi que la possibilité d'inversion, nous conduira à envisager dans la deuxième section du chapitre une classe particulière de modèles : les séries de Volterra. Les deux dernières sections de ce chapitre sont consacrées à différentes manières de modéliser les noyaux de Volterra en vue d'obtenir un modèle global à la fois simple dans son utilisation et dans la procédure d'identification de ses paramètres.

1.2 Représentation des systèmes non linéaires

1.2.1 Introduction

La plupart des systèmes réels sont non linéaires à des degrés divers. Historiquement et pour des raisons de commodité de mise en oeuvre, l'approche la plus répandue est d'approximer les systèmes non linéaires par des modèles linéaires. Un modèle linéaire suffit le plus souvent à décrire avec une bonne précision le comportement général et la dynamique moyenne d'un système faiblement non linéaire.

Cependant, si les non-linéarités deviennent trop importantes, il n'est plus possible de se contenter d'approximer le comportement du système par un (ou des) modèle(s) linéaire(s). Il faut donc avoir recours à des modèles non linéaires, ce qui entraîne un certain nombre de difficultés, tant au niveau de l'identification paramétrique que de la commande. Si l'on compare les deux types de représentation des systèmes (linéaire et non linéaire), une des principales difficultés de la deuxième forme de représentation est due à l'absence d'une théorie unifiée de représentation des non-linéarités, théorie qui existe dans le cas des systèmes linéaires.

Selon que l'on possède ou non des informations sur la structure du système non linéaire étudié, plusieurs choix possibles de formes de modèles se présentent.

1.2.2 Modèles de comportement

Les modèles de comportement (ou modèles "boîte noire") peuvent être utilisés dans le cas particulier où l'objectif de l'utilisateur n'est pas la connaissance physique du système réel mais plutôt la reproduction de son comportement. De manière générale, on peut dire qu'un modèle est satisfaisant quand il prédit correctement certains phénomènes du monde réel, même s'il n'est qu'une vision partielle de la réalité.

Un modèle de connaissance de type "boîte blanche", construit à partir de lois de représentation de phénomènes physiques, sera plus ou moins satisfaisant en fonction du nombre et de la précision de ces lois physiques.

La notion de modèle "boîte grise" provient du fait que l'utilisateur d'un système réel ne connaît a priori pas toutes les lois physiques qui régissent le comportement de ce système. Certaines relations ou variables demeurent inconnues. D'où l'intérêt d'introduire dans le modèle des relations mathématiques (sans forcément de sens physique) qui viennent compléter les relations physiques connues. La mise au point d'un modèle "boîte grise" nécessite donc une complémentarité entre l'ensemble des lois physiques et l'ensemble des lois mathématiques à la disposition de l'utilisateur.

Les modèles de comportement, quant à eux, sont construits uniquement à partir de lois mathématiques reliant les entrées aux sorties du système. Ils n'ont aucune signification physique particulière et pourront donc donner des résultats satisfaisants dans des conditions beaucoup plus générales d'application. En théorie [Sjöberg *et al.*, 1995], toute structure "boîte noire" convenablement choisie doit pouvoir représenter n'importe quelle forme de non-linéarité.

Parmi les modèles de comportement les plus utilisés, on distinguera entre autres les modèles NARMAX [Chen, 1989], les multi-modèles [Murray-Smith et Johansen, 1997] et les réseaux de neurones [Hérault et Jutten, 1994].

1.2.2.1 Les modèles NARMAX

La modélisation d'un système (linéaire ou non linéaire) nécessite l'identification des paramètres du modèle. Le problème de l'identification est de trouver une ou plusieurs relations entre des données d'entrée-sortie passées et la sortie à calculer. Si l'on dispose d'un nombre fini n_u d'entrées $u(t - i)$ ($1 \leq i \leq n_u$) et n_y de sorties $y(t - j)$ ($1 \leq j \leq n_y$) rassemblées dans un vecteur $\varphi(t)$:

$$\varphi(t) = [-y(t-1) \dots -y(t-n_y) \ u(t-1) \dots u(t-n_u)]^T \quad (1.1)$$

alors le problème est de trouver la relation f entre $\hat{y}(t)$, prédiction de la sortie $y(t)$ et $\varphi(t)$:

$$\hat{y}(t) = f(\varphi(t), \theta) \quad (1.2)$$

la fonction f étant paramétrée par un vecteur θ .

Dans le cas où la fonction f est linéaire, différents modèles existent : modèles ARX, ARMAX, Output Error ou Box-Jenkins. Dans le cas où f n'est plus une fonction linéaire, l'approche polynomiale NARMAX (Nonlinear AutoRegressive Moving Average with eXogenous inputs) a été développée [Chen, 1989]. Le vecteur $\varphi(t)$ comprend, en plus des entrées et des sorties passées, les erreurs de prédiction passées $\varepsilon(t-k)$ ($1 \leq k \leq n_\varepsilon$) :

$$\varphi(t) = [-y(t-1) \dots -y(t-n_y) \ u(t-1) \dots u(t-n_u) \ \varepsilon(t-1) \dots \varepsilon(t-n_\varepsilon)]^T \quad (1.3)$$

L'erreur de prédiction $\varepsilon(t)$ est ajoutée au modèle pour tenir compte du fait que la sortie $\hat{y}(t)$ n'est pas fonction exacte des données d'entrée-sortie passées. L'erreur de prédiction doit être la plus faible possible afin que le modèle fournisse la meilleure prédiction possible de la sortie réelle $y(t)$ connaissant les données passées.

La fonction f , inconnue a priori, est choisie de manière à pouvoir représenter la dynamique du système. Une expression possible (et relativement employée) de f est le développement sur bases de fonctions :

$$f(\varphi(t), \theta) = \sum_{i=1}^M \theta_i f_i(\varphi(t)) \quad (1.4)$$

où f_i est une base de fonctions non linéaires et les termes θ_i sont les paramètres du développement. Le choix des fonctions f_i est très important pour assurer la qualité du modèle. Dans la plupart des structures de modèles non linéaires considérées, les fonctions f_i sont obtenues en paramétrisant une fonction plus générale notée $K(\cdot)$. On écrit de manière générale

$$f_i(\varphi) = K(\varphi, \beta_i, \gamma_i) = K(\beta_i(\varphi - \gamma_i)) \quad (1.5)$$

Les paramètres β_i et γ_i peuvent être de nature différente selon les cas.

Si l'on pose $K(\cdot) = \cos(\cdot)$, on retrouve le développement de $f_i(\varphi)$ en série de Fourier, les paramètres β_i et γ_i étant respectivement la fréquence et la phase.

Dans le cas multidimensionnel, les f_i sont des bases de fonctions multi-variables, construites à partir de fonctions mono-variables du type $K(\cdot)$.

Les bases de fonctions radiales sont couramment utilisées car relativement simples à construire. Les fonctions f_i ont la forme

$$f_i(\varphi) = K(\|\gamma_i - \varphi\|_{\beta_i}) \quad (1.6)$$

où $\|\cdot\|_{\beta_i}$ représente une norme précédemment choisie (par exemple la norme euclidienne). Le terme γ_i est ici appelé centre de la fonction f_i .

La décomposition en ondelettes est un autre exemple typique. Dans ce cas précis, la fonction mono-variable utilisée pour la construction de la base d'ondelettes est appelée *ondelette mère*. L'expression est doublement indexée et on pose $\beta_i = 2^i$ et $\gamma_j = j$ pour obtenir :

$$f_{i,j}(\varphi) = 2^{\frac{i}{2}} K(2^i \varphi - j) \quad (1.7)$$

Dans tous les cas considérés, le modèle NARMAX est linéaire par rapport aux paramètres. Les paramètres de ce modèle peuvent donc être estimés de manière relativement simple par la méthode classique des moindres carrés.

Un autre point important, sujet à discussion, est la capacité qu'a le modèle NARMAX à pouvoir représenter toutes les non-linéarités des systèmes considérés. Vu dans le contexte de cette thèse, cet avantage n'en est pas forcément un car les modèles NARMAX utilisés sont bien souvent complexes et surparamétrisés. Bien que l'on puisse envisager de réduire leur complexité par la suite, cela ne répond pas à l'objectif de parcimonie paramétrique que nous nous sommes fixés dès le départ.

Une solution envisageable est de découper l'espace des données entrées/sorties en sous-espaces locaux pour lesquels un modèle plus simple (structure moins complexe et nombre de paramètres moins important) peut être appliqué. Le modèle général consiste alors à combiner les modèles locaux.

1.2.2.2 Les multi-modèles

L'approche multi-modèles [Murray-Smith et Johansen, 1997], [Boukhris, 1998] repose sur l'établissement de plusieurs modèles simples, encore appelés modèles locaux. Chaque modèle est valable autour d'un point de fonctionnement, dont la zone d'influence est définie au moyen d'une fonction poids. Tous les modèles locaux sont ensuite agrégés afin d'obtenir un modèle global qui relie les entrées et les sorties du système.

Plusieurs problèmes se posent alors : la décomposition de l'espace global en différentes zones, le choix du nombre de zones, le choix de la structure des modèles locaux et leur agrégation.

De même que dans la section précédente, on représente la relation entrée-sortie du système de la manière suivante

$$\hat{y}(t) = f(\varphi(t), \theta) \quad (1.8)$$

avec

$$\varphi(t) = [-y(t-1) \dots -y(t-n_y) \ u(t-1) \dots u(t-n_u)]^T \quad (1.9)$$

$\varphi(t)$ représente le vecteur d'information et θ le vecteur paramètre.

Le problème de la modélisation locale consiste à décomposer le domaine global de fonctionnement Φ du système en un ensemble de zones de fonctionnement Φ_i . Dans chaque zone Φ_i de l'espace d'information, la fonction $\hat{y}(t)$ est représentée par la fonction f_i paramétrée par le vecteur θ_i :

$$\hat{y}(t) = f_i(\varphi(t), \theta_i) \quad (1.10)$$

La parcimonie paramétrique du modèle global dépend donc de celle de chaque modèle local. Le choix de la structure des modèles locaux est donc crucial. De plus, le nombre de paramètres à identifier dépend du nombre de modèles locaux employés, et donc de la manière dont sera découpé l'espace global en zones de fonctionnement.

Dans un des cas les plus simples, on considère le système SISO statique représenté par la fonction f :

$$\hat{y}(t) = f(u(t)) \quad (1.11)$$

Le multi-modèle quasi-linéaire permettant de décrire le système représenté par f est obtenu en décomposant le domaine de variation de la variable u en n_u zones de fonctionnement Φ_i où la fonction f est définie par le modèle linéaire :

$$\hat{y}_i(t) = b_i u(t) + c_i \quad (1.12)$$

Dans ce cas simple, le nombre total de paramètres du modèle global dépend du nombre total n_u de zones et de la forme du modèle employé dans chacune des zones. Cependant, décomposer le domaine de fonctionnement en zones où le système peut être représenté par un modèle linéaire peut s'avérer délicat, surtout si le système est fortement non linéaire et si les variations de sa sortie $y(t)$ sont importantes.

Si l'on considère un modèle dynamique du premier ordre

$$\hat{y}(t) = f(u(t-1), y(t-1)) \quad (1.13)$$

alors, dans chaque zone de fonctionnement, le système est décrit par le modèle

$$\hat{y}_i(t) = -a_i y(t-1) + b_i u(t-1) + c_i \quad (1.14)$$

ce qui augmente le nombre total de paramètres si l'on a conservé le même nombre n_u de zones de fonctionnement. Cependant, on s'aperçoit que plus la complexité d'un modèle local augmente, plus il est à même de pouvoir représenter correctement le système étudié

sur un domaine étendu. On peut donc diminuer le nombre n_u de zones de fonctionnement. L'objectif devient alors de trouver un compromis entre la complexité des modèles employés et l'étendue de leur validité.

On peut bien évidemment complexifier davantage la structure des modèles locaux employés afin que le modèle global fournisse une approximation encore meilleure du système réel. Selon les cas, on peut utiliser des fonctions de transfert, des fonctions de Laguerre ou encore la structure d'Hammerstein généralisée. Il va sans dire que ces modèles locaux nécessitent davantage de paramètres que les modèles simples que l'on vient d'évoquer. Il ne répondent donc plus à l'objectif de parcimonie paramétrique. De plus, le problème d'identification paramétrique devient de plus en plus complexe car on aboutit à un modèle global qui n'est plus linéaire par rapport aux paramètres, contraignant l'utilisateur à recourir à des méthodes de programmation non linéaire.

La méthode multi-modèles ne répondra donc à l'objectif de parcimonie que dans le cas de systèmes aux non-linéarités douces que l'on peut décrire localement par un ensemble de modèles linéaires.

1.2.2.3 Les réseaux de neurones artificiels

Les réseaux de neurones sont à la base un outil d'analyse statistique que l'on peut utiliser afin de construire un modèle de comportement de système. Les réseaux artificiels neuronaux permettent de représenter des relations fonctionnelles complexes, difficiles à décrire sous une forme analytique, de systèmes non linéaires ou de systèmes variables en fonction du temps [Hérault et Jutten, 1994].

Le neurone biologique est une cellule vivante spécialisée dans le traitement des signaux électriques. Les neurones sont reliés entre eux par des liaisons particulières appelées axones, qui conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone. Les neurones effectuent une sommation des signaux reçus en entrée et, en fonction du résultat obtenu, vont fournir un signal électrique en sortie.

Par analogie, le neurone artificiel est un processeur élémentaire qui reçoit un certain nombre de variables d'entrée en provenance d'autres neurones appartenant à un niveau situé en amont. A chacune des entrées est associé un poids w représentatif de l'importance de la connexion. Chaque neurone est doté d'une sortie unique qui se ramifie pour alimenter à son tour un certain nombre de neurones appartenant à un niveau situé en aval (cf figure 1.1).

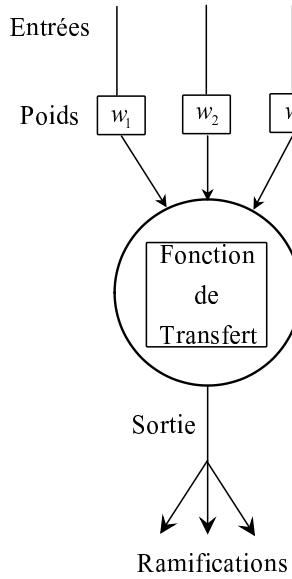


Fig. 1.1 - Schéma d'un neurone artificiel

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle général. Cette topologie fait le plus souvent apparaître une certaine régularité et l'on peut donc distinguer différentes sortes de réseaux :

- Les réseaux multicouches classiques : il n'y a pas de connexion entre les neurones d'une même couche et les connexions ne se font qu'avec les neurones de la couche aval. Tous les neurones de la couche amont sont connectés à tous les neurones de la couche aval. Les couches extérieures du réseau sont appelées respectivement couches d'entrée et de sortie. Les couches intermédiaires sont appelées couches cachées.
- Les réseaux à connexion locale : ce sont également des réseaux multicouches mais tous les neurones d'une couche amont ne sont pas connectés à tous les neurones de la couche aval. Le nombre de connexions est donc moins important que dans le cas d'un réseau multicouches classique.
- Les réseaux à connexions récurrentes : une ou plusieurs sorties de neurones d'une couche aval sont connectées aux entrées des neurones de la couche amont ou bien de la même couche. Ces connexions récurrentes ramènent de l'information en arrière par rapport au sens de propagation défini dans un réseau multicouches classique.
- Les réseaux à connexions complexes : chaque neurone est connecté à tous les neurones du réseau, y compris lui-même. On ne fait plus de distinction entre les différentes couches. Aucun sens général de propagation n'est défini. C'est la structure d'interconnexion la plus générale possible.

Afin de modéliser correctement un système, il est nécessaire que les sorties du réseau de neurones soient aussi proches que possible des sorties réelles. Pour arriver à ce but, il est nécessaire de passer par une phase d'apprentissage, durant laquelle on calcule le nombre et/ou le poids de chacun des neurones. Cela équivaut à une identification paramétrique de modèle.

Les réseaux artificiels neuronaux sont des outils permettant de représenter les relations fonctionnelles complexes nécessitées par les systèmes modernes. La plupart de ces systèmes sont non linéaires. étant donné que les réseaux artificiels de neurones sont très souples dans leur structure, on considère qu'ils constituent des modules universels de représentation du comportement de systèmes réels, linéaires ou pas.

Cependant, un des inconvénients de cette universalité se traduit par une fréquente surparamétrisation des modèles. Outre le fait que cette surparamétrisation ne répond pas à l'objectif de parcimonie fixé au départ, le problème de l'apprentissage (ou identification paramétrique) est généralement mal conditionné. On doit donc recourir à des méthodes de régularisation pour transformer le problème mal posé en un problème bien posé et ainsi obtenir une solution paramétrique optimale unique.

De par sa nature, le réseau de neurones permet donc de représenter un très grand nombre de systèmes non linéaires, mais ceci au détriment de la simplicité et de la parcimonie.

1.2.3 Conclusion

Au travers des trois exemples de modèles évoqués, on se rend compte que l'un des problèmes majeurs entraîné par la représentation précise du comportement de systèmes non linéaires est la surparamétrisation des modèles employés. Par la suite, un trop grand nombre de paramètres rend les modèles obtenus difficiles à manipuler et à identifier de manière globale. Certaines techniques permettent donc de simplifier les structures obtenues en éliminant les paramètres non significatifs des modèles mais cette élimination nécessite un volume et un temps de calcul supplémentaires. L'objectif de parcimonie des modèles de systèmes non linéaires est donc largement justifié.

1.3 Les séries de Volterra

1.3.1 Introduction

La représentation des systèmes non linéaires est actuellement un problème ouvert aussi bien du point de vue théorique qu'appliqué. On a vu que des méthodes nombreuses et variées existent déjà et sont largement appliquées dans le domaine de la recherche. Tou-

tefois, le manque de cohésion entre ces différentes méthodes de modélisation fait que leur application pratique dans le domaine industriel pour la commande de procédés reste relativement marginale. Le principal obstacle est qu'il n'existe aucune approche systématique pour la construction de modèles dynamiques non linéaires dédiés à la commande. A l'inverse, la modélisation des systèmes linéaires est très utilisée dans l'industrie. Et ceci car il est aujourd'hui relativement simple d'obtenir des modèles linéaires à partir de méthodes cohérentes, fiables et largement éprouvées.

Doyle [Doyle *et al.*, 2002] fait remarquer que les systèmes non linéaires sont définis en terme de manque : un système non linéaire ne répond pas à une propriété précise. A l'opposé, un système linéaire est défini à partir d'une propriété mathématique extrêmement spécifique. Il existe de nombreuses formes de non-linéarités. Pour les modéliser correctement, il faudra donc se focaliser sur des classes de modèles particulières, capables de représenter les non-linéarités considérées. Toutes les classes de modèles non linéaires ne sont pas capables de représenter tous les types de non-linéarités. Pour ces raisons, dans le domaine industriel, la commande par modèle non linéaire est encore considérée comme un concept théorique en développement plutôt qu'une stratégie de commande fiable.

Une approche possible pour remédier à cet état de fait pourrait être la modélisation des systèmes non linéaires par les séries de Volterra [Flake, 1963], [Schetzen, 1980], [Rugh, 1981], [Doyle *et al.*, 2002]. D'une part, les séries de Volterra sont un sous-ensemble des séries de fonctionnelles et permettent de représenter un grand nombre de non-linéarités. D'autre part, elles généralisent la notion de convolution utilisée dans l'analyse des systèmes linéaires. Le premier terme de la série de Volterra est une intégrale de convolution qui représente un système linéaire (ou bien la partie linéaire d'un système non linéaire) tandis que les autres termes, qui sont ajoutés, vont modéliser la partie non linéaire du système considéré.

Les séries de Volterra peuvent donc être un moyen pratique et acceptable aux yeux des utilisateurs (car elles généralisent une notion qu'ils connaissent bien : la convolution) de représenter les systèmes non linéaires pour ensuite envisager la commande des procédés modélisés.

1.3.2 Les séries de fonctionnelles

1.3.2.1 Présentation

Avant de parler des séries de Volterra, nous allons d'abord introduire les séries de fonctionnelles, outil mathématique qui permet la représentation de dynamiques non linéaires très variées. Les séries de Volterra trouvent leur origine dans les mathématiques de l'analyse fonctionnelle, née au début du XX^e siècle. L'analyse fonctionnelle est la branche des mathématiques consacrée à l'étude des espaces de fonctions.

Déjà mise en évidence par des précurseurs italiens (Ascoli et Arzela), l'analyse fonc-

tionnelle et la notion même de fonctionnelle furent introduite par Jacques Hadamard en 1910. De nombreux mathématiciens vont ensuite développer cette nouvelle branche des mathématiques : Frédéric Riesz, René Baire, Maurice Fréchet, Stephan Banach et Vito Volterra.

Une fonctionnelle réelle est une fonction à valeurs réelles définie sur un espace linéaire. Dans ce cas, l'espace linéaire est supposé être un espace de fonctions, à savoir les signaux d'entrée $u(t)$. A n'importe quel temps t , la sortie $y(t)$ d'un système dépend des entrées précédentes $u(t - \sigma)$, $\sigma \geq 0$. On dit que la sortie $y(t)$ à un instant t particulier est une fonctionnelle réelle du signal d'entrée $u(t)$.

Une fonctionnelle régulière et homogène d'ordre i est donnée par l'expression générale

$$H_i[u(t)] = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i u(t - \tau_j) d\tau_j \quad (1.15)$$

On suppose que le noyau $h_i(\tau_1, \dots, \tau_i)$ est symétrique par rapport à ses i variables.

La somme de n fonctionnelles régulières et homogènes H_i , $i = 1, \dots, n$ est une fonctionnelle H d'ordre n régulière et homogène. La fonctionnelle H qui relie l'entrée $u(t)$ et la sortie $y(t)$ du système peut donc être notée sous la forme

$$y(t) = H[u(t)] = \sum_{i=1}^{\infty} \left(\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i u(t - \tau_j) d\tau_j \right) \quad (1.16)$$

La réponse $y(t)$ du système s'exprime plus simplement en fonction de l'entrée $u(t)$

$$y(t) = \sum_{i=1}^{\infty} H_i[u(t)] \quad (1.17)$$

Une fonctionnelle régulière et homogène d'ordre i vérifie la propriété suivante :

$$H_i[\lambda u(t)] = \lambda^i H_i[u(t)] \quad (1.18)$$

λ étant un réel non nul et i un entier naturel.

Si $i > 1$ alors la fonctionnelle $H_i[u(t)]$ est non linéaire. Elle admet la représentation intégrale (1.15) où la fonction $h_i(\tau_1, \dots, \tau_i) : \mathbb{R}^i \rightarrow \mathbb{R}$, $i = 1, 2, \dots$ est localement bornée et continue par morceaux.

On remarque que le premier terme de la somme

$$H_1[u(t)] = \int_{-\infty}^{+\infty} h_1(\tau_1) u(t - \tau_1) d\tau_1 \quad (1.19)$$

n'est autre que l'intégrale de convolution utilisée pour représenter un système linéaire. Dans un cadre plus général, ce premier terme représente le système linéarisé autour de

son point de fonctionnement. $h_1(t)$ est donc la réponse impulsionale de la partie linéaire du système considéré.

Les termes $H_i[u(t)]$ ($i > 1$) généralisent la notion de convolution et permettent d'appliquer cette représentation aux systèmes non linéaires afin de représenter les non-linéarités qui apparaissent au fur et à mesure que le système s'éloigne de son point de fonctionnement. On pourra donc considérer les termes $h_i(t)$ ($i > 1$) comme des réponses impulsionales multidimensionnelles car chaque fonctionnelle $H_i[u(t)]$ est obtenue par convolution avec l'entrée $u(t)$ du système.

1.3.2.2 Propriétés

L'outil mathématique que sont les séries de fonctionnelles doit être adapté à la représentation des systèmes en général et des systèmes non linéaires dans le cas qui nous concerne. Dans cette optique, une série de fonctionnelles possède deux propriétés très intéressantes :

- Tout d'abord, la représentation d'un système non linéaire par une série de fonctionnelles H possède une relation entrée-sortie **explicite**, qui se prête bien à la représentation par schéma-bloc déjà utilisée pour la modélisation des systèmes linéaires.
- Ensuite, cette représentation par schémas-blocs autorise, comme dans le cas linéaire, l'interconnexion en série, en parallèle ou en cascade de plusieurs modèles de systèmes non linéaires. L'association de modèles linéaires et non linéaires est évidemment possible et nous verrons qu'elle était déjà employée depuis un certain temps (modèles de Hammerstein-Wiener).

1.3.2.3 Conclusion

Le formalisme des séries de fonctionnelles, d'abord utilisé par Wiener dans le domaine particulier de l'analyse des circuits électroniques non linéaires [Wiener, 1943], s'est depuis étendu à tous les domaines de la physique. Mais la notion de système peut également s'appliquer à d'autres domaines scientifiques (chimie, biologie,...).

Les méthodes de modélisation de systèmes non linéaires se sont donc récemment "exportées" avec succès vers des domaines auxquels elles n'étaient pas forcément prédestinées, dont la chimie et la biochimie. La prévision du comportement de réacteurs chimiques ou biochimiques, fermés ou continus, intéresse forcément les chercheurs de ces domaines et cette prévision passe par une modélisation précise des phénomènes en jeu. Le lien entre différentes espèces chimiques ou biochimiques peut être établi à l'aide de modèles linéaires ou, dans la plupart des cas, non linéaires. Le formalisme des séries de fonctionnelles, qui comprend une relation explicite entrée-sortie représentable sous forme de schéma-bloc, peut donc être envisagé pour la modélisation de systèmes chimiques ou biochimiques

[Corriou, 2001].

1.3.3 Définition des séries de Volterra

1.3.3.1 Introduction

Les travaux de Volterra (1860-1940) se sont portés, entre autres, sur la classe des équations intégrales qui porte son nom. Il a contribué de manière significative aux premiers développements de ce qui allait plus tard devenir l'analyse fonctionnelle.

Les équations intégrales [Tricomi, 1985], de même que les équations différentielles et les équations aux dérivées partielles, font partie de l'ensemble plus vaste des équations fonctionnelles. Les équations intégrales de Volterra sont donc contenues dans l'ensemble des équations fonctionnelles et les séries de Volterra qui découlent de ces équations intégrales peuvent être considérées comme un sous-ensemble des séries de fonctionnelles.

Les équations intégrales de Volterra proviennent naturellement de méthodes développées pour la résolution d'équations différentielles non linéaires ordinaires du type

$$\frac{dy}{dx} = F(x, y) \quad (1.20)$$

avec la condition initiale $y(x_0) = y_0$. Si l'on intègre cette équation entre x_0 et x , on obtient l'équation suivante

$$y(x) = y_0 + \int_{x_0}^x F[t, y(t)]dt \quad (1.21)$$

Une forme plus générale de cette équation est l'équation non linéaire de Volterra

$$\phi(x) = f(x) + \int_0^x F[x, y, \phi(y)]dy \quad (1.22)$$

où les fonctions f et F sont connues et la fonction ϕ inconnue et déterminée par l'équation (1.22). Ces équations sont résolues par l'approche itérative de substitution suivante (Tricomi 85) : on pose

$$\phi_n(x) = f(x) + \int_0^x F[x, y, \phi_{n-1}(y)]dy \quad (1.23)$$

pour $n \geq 1$. Cette séquence est initialisée en définissant $\phi_0(x) = f(x)$ et, pour $n \rightarrow \infty$, $\phi_n(x)$ converge vers la solution $\phi(x)$.

Si l'on considère maintenant le cas particulier où la fonction F peut être mise sous la forme

$$F[x, y, \phi(y)] = \lambda K(x, y)\phi(y) \quad (1.24)$$

avec $K(x, y)$ fonction réelle ($K(x, y) = 0$ si $y \geq x$), alors la résolution par approche itérative de substitution conduit à l'équation de Volterra de deuxième espèce

$$\phi(x) = f(x) + \sum_{n=1}^{\infty} \lambda^n \int_0^x K_n(x, y) f(y) dy \quad (1.25)$$

qui est une série infinie et où $K_n(x, y)$ est un noyau itéré, défini par $K_1(x, y) = K(x, y)$ pour $n = 1$ et, pour $n > 1$,

$$K_n(x, y) = \int_0^x K(x, z) K_{n-1}(z, y) dz \quad (1.26)$$

Le noyau itéré d'ordre n est donc ici une intégrale d'ordre n (pour $n \geq 2$).

On constate alors que la solution de l'équation intégrale, calculée de manière itérative, prend la forme d'une série dont le terme d'ordre 1 est une intégrale simple

$$\lambda \int_0^x K_1(x, y) f(y) dy \quad (1.27)$$

Le terme d'ordre 2 est une intégrale double

$$\lambda^2 \int_0^x \int_0^x K(x, z) K(z, y) f(y) dz dy \quad (1.28)$$

Le terme général d'ordre n est donc une intégrale d'ordre n . On se rapproche donc du formalisme des séries de fonctionnelles défini dans la section précédente. Pour retrouver le formalisme plus exact des séries de fonctionnelles, on pose d'abord $f(y) = 1$, $\lambda = 1$, puis pour le terme d'ordre 1

$$K_1(x, y) = K(x, y) = h_1(x - y) u(y) \quad (1.29)$$

Pour le terme d'ordre 2

$$K_2(x, y) = \int_0^x K(x, z) K(z, y) dz = \int_0^x h_2(x - y, x - z) u(y) u(z) dz \quad (1.30)$$

et ainsi de suite pour tous les termes d'ordre $n > 2$. On retrouve ainsi le formalisme des séries de fonctionnelles défini précédemment. Cela confirme que les séries de Volterra sont bien une classe particulière de séries de fonctionnelles, répondant à certaines propriétés.

1.3.3.2 Propriétés

De manière générale, on dira qu'une fonctionnelle $H_i[u(t)]$ est réalisable et de type Volterra [Volterra, 1959] si le terme $h_i(\tau_1, \dots, \tau_i)$ (pour $i = 1, \dots, \infty$) répond aux propriétés suivantes :

Causalité :

$$h_i(t - \tau_1, \dots, t - \tau_i) = 0 \text{ pour } \tau_k > t, \quad k = 1, \dots, i \quad (1.31)$$

Mémoire finie :

$$h_i(\tau_1, \dots, \tau_i) = 0 \text{ pour } \tau_k > \alpha, \alpha > 0 \text{ et } k = 1, \dots, i \quad (1.32)$$

Stationnarité :

Le noyau $h_i(\tau_1, \dots, \tau_i)$ est invariant dans le temps et ne dépend que des différences $t - \tau_i$.

Si le noyau $h_i(\tau_1, \dots, \tau_i)$ répond à ces propriétés pour $i = 1, \dots, \infty$, alors l'ensemble des noyaux, qu'on appellera désormais noyaux de Volterra, caractérise complètement le système non linéaire considéré.

1.3.4 Cas particuliers

1.3.4.1 Introduction

Limiter la complexité des modèles de Volterra n'est pas un sujet nouveau. Nous allons voir dans cette section que de nombreuses méthodes de représentation de systèmes non linéaires existent déjà. Ces méthodes peuvent conduire à la synthèse de modèles non linéaires très efficaces et moins complexes que les modèles généraux de Volterra, mais ceci uniquement dans certains cas particuliers. Nous verrons alors que ces classes particulières de modèles ne sont que des sous-ensembles de la classe des modèles de Volterra.

1.3.4.2 Modèle de Hammerstein

Le modèle de Hammerstein est un des plus simples et des plus connus de la famille plus générale des modèles non linéaires dynamiques orientés par blocs [Billings et Fakhouri, 1979], [Ralston et Zoubir, 1995]. La structure du modèle de Hammerstein consiste en une non-linéarité statique $g(\cdot)$ reliée en cascade avec un modèle linéaire dynamique défini par une fonction de transfert notée $A(s)$.

Dans le cas particulier où le modèle linéaire est à réponse impulsionnelle finie et si la fonction $g(\cdot)$ est un polynôme de degré n , on parle de modèle de Hammerstein fini.

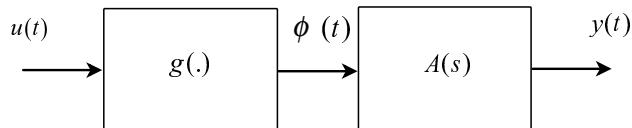


Fig. 1.2 - Modèle de Hammerstein

D'après la figure 1.2, la sortie intermédiaire $\phi(t)$ peut s'écrire sous la forme

$$\phi(t) = g(u(t)) = \sum_{i=1}^n \gamma_i u^i(t) \quad (1.33)$$

De même, la sortie générale $y(t)$ du modèle de Hammerstein s'écrit sous la forme

$$y(t) = \int_0^\infty \alpha(\tau) \phi(t - \tau) d\tau \quad (1.34)$$

où $\alpha(\tau)$ est la réponse impulsionnelle de la fonction de transfert $A(s)$.

Si l'on combine ces deux expressions, on obtient finalement l'expression de $y(t)$ suivante

$$y(t) = \sum_{i=1}^n \int_0^\infty \gamma_i \alpha(\tau) u^i(t - \tau) d\tau \quad (1.35)$$

Si l'on pose ensuite

$$h_i(\tau_1, \dots, \tau_i) = \begin{cases} \gamma_i \alpha(\tau) & \text{si } \tau_1 = \dots = \tau_i \\ 0 & \text{sinon} \end{cases} \quad (1.36)$$

le modèle obtenu peut être vu comme un modèle de Volterra "diagonal" car les termes hors diagonale, qui correspondent au cas où $\tau_i \neq \tau_j$ ($j \neq i$), sont nuls. Les paramètres non nuls du modèle non linéaire ainsi obtenu sont proportionnels aux paramètres du modèle linéaire : $h_i(\tau, \dots, \tau) = \gamma_i h_1(\tau)$.

On peut donc considérer le modèle de Hammerstein comme une approximation du modèle de Volterra. Cette approximation suffit à réduire la complexité du modèle de Volterra. Cette classe de modèle est plus simple à utiliser et plus parcimonieuse dans les paramètres que la classe des modèles de Volterra. Ces avantages sont obtenus au détriment de la flexibilité du modèle : les restrictions inhérentes au modèle de Hammerstein l'empêchent de représenter autant de non-linéarités différentes que le modèle plus général de Volterra.

1.3.4.3 Modèle de Wiener

Le modèle de Wiener peut être vu comme le dual du modèle de Hammerstein [Rugh, 1981], [Doyle *et al.*, 2002]. Il comporte les deux mêmes composantes mais reliées dans l'ordre inverse. Le signal d'entrée $u(t)$ passe d'abord par le modèle linéaire dynamique de fonction de transfert $A(s)$ pour donner le signal de sortie intermédiaire $\psi(t)$, lui-même transformé par la non-linéarité statique $g(\cdot)$ pour donner le signal de sortie $y(t)$ du modèle.

De la même manière que pour le modèle de Hammerstein, on peut définir un modèle de Wiener fini si le modèle linéaire est à réponse impulsionnelle finie, et si la fonction $g(\cdot)$ est un polynôme de degré n .

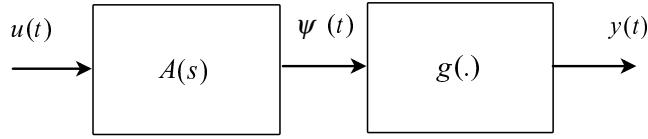


Fig. 1.3 - Modèle de Wiener

D'après la figure 1.3, la sortie intermédiaire $\psi(t)$ peut s'écrire sous la forme

$$\psi(t) = \int_0^\infty \alpha(\tau)u(t-\tau)d\tau \quad (1.37)$$

La sortie générale $y(t)$ du modèle de Wiener s'écrit sous la forme

$$y(t) = \sum_{i=1}^n \gamma_i \psi^i(t) \quad (1.38)$$

En tenant compte des deux expressions précédentes, la sortie $y(t)$ du modèle de Wiener peut s'exprimer ainsi

$$y(t) = \sum_{i=1}^n \int_0^\infty \dots \int_0^\infty \gamma_i \alpha^i(\tau) u^i(t-\tau) (d\tau)^i \quad (1.39)$$

On définit ensuite, pour $i = 1, \dots, \infty$:

$$h_i(\tau_1, \dots, \tau_i) = \gamma_i \alpha(\tau_1) \dots \alpha(\tau_i) \quad (1.40)$$

Si l'on s'en tient aux schémas, la structure du modèle de Wiener est très proche de celle du modèle de Hammerstein. Ils utilisent en effet les mêmes composantes, à savoir un transfert linéaire et une non-linéarité statique.

Cependant, les équations (1.36) et (1.40) nous montrent que ces deux modèles sont très différents car le seul cas où les coefficients $h_i(\tau_1, \dots, \tau_i)$ sont égaux est le cas linéaire ($n = 1$) où la relation entre l'entrée $u(t)$ et la sortie $y(t)$ est un produit de convolution classique. Le comportement des deux types de modèles dans le cas non linéaire et dynamique ne sera donc pas le même.

On remarque également que, par construction, la complexité du modèle de Wiener est du même ordre que celle du modèle de Hammerstein. Le modèle de Wiener peut donc également être considéré comme une approximation du modèle de Volterra, approximation dont la complexité est réduite mais dont la flexibilité (la capacité à représenter une large variété de non-linéarités) est moindre par rapport à celle du modèle de Volterra.

1.3.4.4 Modèle d'Uryson

Le modèle d'Uryson est généralement moins connu que les modèles de Hammerstein et de Wiener mais on peut considérer qu'il représente un sous-ensemble de la classe des modèles de Volterra [Doyle *et al.*, 2002]. Il consiste à mettre en parallèle un certain nombre de modèles de Hammerstein (r modèles dans notre cas).

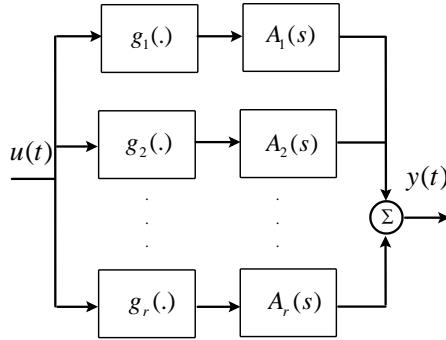


Fig. 1.4 - Modèle d'Uryson

On suppose que les non-linéarités statiques $g_i(\cdot)$ sont des polynômes d'ordre n_i ($i = 1, \dots, r$) et que les modèles linéaires dynamiques peuvent être représentés par des fonctions de transfert $A_i(s)$ ($i = 1, \dots, r$).

La sortie $y_i(t)$ du $i^{\text{ème}}$ modèle de Hammerstein s'écrit :

$$y_i(t) = \sum_{j=1}^{n_i} \int_0^\infty \gamma_j^{(i)} \alpha_i(\tau) u^j(t-\tau) d\tau \quad (1.41)$$

La sortie $y(t)$ du modèle d'Uryson s'écrit donc sous la forme :

$$y(t) = \sum_{i=1}^r y_i(t) = \sum_{i=1}^r \sum_{j=1}^{n_i} \int_0^\infty \gamma_j^{(i)} \alpha_i(\tau) u^j(t-\tau) d\tau \quad (1.42)$$

Les coefficients de ce modèle sont les suivants :

$$h_j(\tau_1, \dots, \tau_j) = \begin{cases} \sum_{i=1}^r \gamma_j^{(i)} \alpha_i(\tau) & \text{si } \tau_1 = \dots = \tau_j \\ 0 & \text{sinon} \end{cases} \quad (1.43)$$

Comme dans le cas du modèle de Hammerstein, on remarque que le modèle d'Uryson peut être vu comme un modèle de Volterra "diagonal". Dans le cas où $r = 1$, on se ramène à un modèle de Hammerstein classique, beaucoup plus flexible que le modèle de Volterra classique mais moins efficace. Le fait de coupler plusieurs modèles de Hammerstein pallie au problème d'efficacité car on peut ainsi représenter davantage de systèmes non linéaires mais c'est au détriment de la simplicité de modélisation. En effet, le modèle d'Uryson

comporte r fois plus de paramètres que celui de Hammerstein.

Le modèle d'Uryson est donc plus flexible que le modèle de Hammerstein, mais il reste plus complexe et ne permet pas de représenter autant de types de non-linéarités que le modèle de Volterra. Dans certains cas, le modèle d'Uryson peut apparaître comme un bon compromis entre le modèle de Hammerstein (trop peu flexible) et celui de Volterra (relativement complexe).

1.3.4.5 Modèle de projection-poursuite

Le modèle de projection-poursuite peut être vu comme le dual du modèle d'Uryson [Doyle *et al.*, 2002]. Le terme de projection-poursuite vient de la littérature consacrée aux statistiques et se réfère à une classe de modèles de régression non linéaire. Le modèle de projection-poursuite peut être obtenu à partir du modèle d'Uryson si l'on remplace chacun des sous-modèles de Hammerstein par des modèles de Wiener (en intervertissant la non-linéarité statique et le modèle linéaire dynamique).

Au final, cela revient à mettre en parallèle un nombre r de modèles de Wiener.

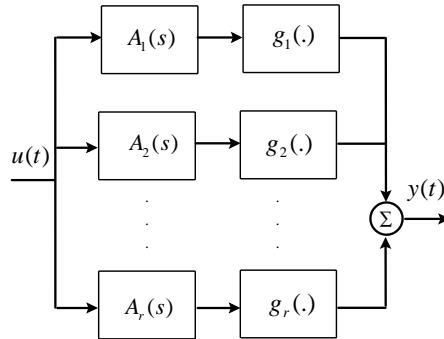


Fig. 1.5 - Modèle de projection-poursuite

On suppose que les non-linéarités statiques $g_i(\cdot)$ sont des polynômes d'ordre n_i ($i = 1, \dots, r$) et que les modèles linéaires dynamiques peuvent être représentés par des fonctions de transfert $A_i(s)$ ($i = 1, \dots, r$).

La sortie $y_j(t)$ du $j^{\text{ème}}$ modèle de Wiener s'écrit :

$$y_j(t) = \sum_{i=1}^{n_j} \int_0^\infty \dots \int_0^\infty \gamma_i^{(j)} \alpha_j^i(\tau) u^i(t-\tau) (d\tau)^i \quad (1.44)$$

La sortie $y(t)$ du modèle de projection-poursuite s'écrit donc sous la forme :

$$y(t) = \sum_{j=1}^r y_j(t) = \sum_{j=1}^r \sum_{i=1}^{n_j} \int_0^\infty \dots \int_0^\infty \gamma_i^{(j)} \alpha_j^i(\tau) u^i(t-\tau) (d\tau)^i \quad (1.45)$$

Les coefficients de ce modèle sont les suivants :

$$h_i(\tau_1, \dots, \tau_i) = \sum_{j=1}^r \gamma_i^{(j)} \alpha_j(\tau_1) \dots \alpha_j(\tau_i) \quad (1.46)$$

Il a été démontré [Doyle *et al.*, 2002] que si r est suffisamment grand, alors la classe des modèles de projection-poursuite est équivalente à la classe des modèles de Volterra.

Comme dans le cas des modèles d'Uryson, l'efficacité des modèles de projection-poursuite dépend du modèle considéré : si l'on prend r faible, le modèle de projection-poursuite est plus efficace en termes de nombre de paramètres qu'un modèle de Volterra. Cependant, le nombre de non-linéarités représentables est plus faible que dans le cas d'un modèle de Volterra.

1.3.5 Conclusion

Les séries de Volterra apparaissent donc comme un moyen acceptable, pratique et souple de modéliser un grand nombre de systèmes non linéaires. Leur formalisme permet une représentation sous forme de schémas-blocs relativement aisée car déjà utilisée dans la représentation de systèmes linéaires. Certaines méthodes de représentation de systèmes non linéaires sont déjà utilisées avec succès depuis un certain nombre d'années, voire de décennies, et apparaissent rétrospectivement comme des cas particuliers de modèles de Volterra, justifiant ainsi le choix de ce type de modèle.

1.4 Détermination et modélisation des noyaux de Volterra

1.4.1 Introduction

Dans tout ce qui suit, on considérera que les conditions initiales sont nulles. Lorsque les conditions initiales sont non nulles, on montre que la connaissance de la réponse impulsionnelle et des noyaux de la réponse d'un système non linéaire en partant de conditions initiales nulles permet d'évaluer les noyaux de la réponse de ce système avec des conditions initiales non nulles [Hassouna, 2001].

On considère le système non linéaire mono entrée-mono sortie dont la représentation d'état est la suivante :

$$\begin{cases} \dot{x}(t) = f(x(t)) + u(t)g(x(t)) \\ y(t) = h(x(t)) \end{cases} \quad (1.47)$$

où

$$\begin{aligned} f(x(t)) &= [f^1(x(t)), \dots, f^n(x(t))]^T \\ g(x(t)) &= [g^1(x(t)), \dots, g^n(x(t))]^T \end{aligned} \quad (1.48)$$

sont des vecteurs de fonctions analytiques et où $x(t) \in \mathbb{R}^n$.

Dans le cas où l'entrée $u(t)$ est bornée en amplitude, la sortie $y(t)$ de ce système non linéaire admet un développement en série de Volterra. Obtenir l'expression analytique des noyaux de Volterra $h_k(t_1, \dots, t_k)$ nécessite la connaissance des vecteurs de fonctions f et g , et de la fonction analytique h . Ces expressions peuvent être relativement difficiles à calculer selon la complexité des fonctions précédentes et le nombre de termes de degré élevé souhaités dans le développement en série de Volterra.

Pour calculer analytiquement les noyaux de Volterra, on peut appliquer la transformée de Laplace multidimensionnelle au système non linéaire global, décomposé sous forme de blocs linéaires et non linéaires. La transformée de Laplace multidimensionnelle inverse permet d'obtenir l'expression temporelle de la réponse impulsionale de chaque noyau de Volterra [George, 1959], [Hassouna, 2001], [Hassouna *et al.*, 2001].

1.4.2 Séries génératrices non commutatives

1.4.2.1 Expression analytique des noyaux de Volterra

La formule fondamentale de Fliess est l'équivalent pour une fonctionnelle de ce qu'est la série de Taylor pour une fonction.

Théorème 1.1 [Lamnabhi-Lagarrigue, 1995]

Soit le système non linéaire mono-entrée mono-sortie (1.47). Dans le cas où l'entrée $u(t)$ du système est suffisamment bornée, on peut associer à la sortie $y(t)$ le développement en série suivant :

$$g = h(x_0) + \sum_{v \geq 0} \sum_{j_0, \dots, j_v=0}^1 L_{j_v} \dots L_{j_1} h(x_0) z_{j_0} \dots z_{j_{v-1}} z_{j_v} \quad (1.49)$$

Les opérateurs L_0 et L_1 sont appelés opérateurs de Lie et définis de la manière suivante :

$$\begin{aligned} L_0 &= \sum_{j=1}^n f^j(x) \frac{\partial}{\partial x_j} \\ L_1 &= \sum_{j=1}^n g^j(x) \frac{\partial}{\partial x_j} \end{aligned} \quad (1.50)$$

Les variables z_0 et z_1 correspondent respectivement à l'opérateur d'intégration par rapport au temps et à l'opérateur d'intégration par rapport au temps après multiplication par l'entrée $u(t)$:

$$\begin{aligned} z_0 &\Leftrightarrow \int_0^t \cdot d\tau \\ z_1 &\Leftrightarrow \int_0^t \cdot u(\tau) d\tau \end{aligned} \quad (1.51)$$

On remarque que les deux variables z_0 et z_1 ne sont pas commutatives. En effet, si on pose $u(t) = t^\alpha$, $\alpha \neq 0$, on obtient

$$z_0 z_1 \Leftrightarrow \int_0^t d\tau_2 \int_0^{\tau_2} u(\tau_1) d\tau_1 = \frac{t^{\alpha+2}}{(\alpha+2)(\alpha+1)} \quad (1.52)$$

qui est différent de

$$z_1 z_0 \Leftrightarrow \int_0^t u(\tau_2) d\tau_2 \int_0^{\tau_2} d\tau_1 = \frac{t^{\alpha+2}}{\alpha+2} \quad (1.53)$$

A partir de ces définitions, le théorème suivant donne une expression explicite de la série de Volterra :

Théorème 1.2 [Fliess et al., 1983]

La sortie $y(t)$ du système non linéaire défini par (1.47) peut être développée en une série de Volterra :

$$y(t) = h_0(t) + \sum_{i=1}^{\infty} \int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_i} h_i(t, \tau_i, \dots, \tau_1) \prod_{j=1}^i u(\tau_j) d\tau_j \quad (1.54)$$

où les noyaux de Volterra sont des fonctions analytiques. Le terme $h_0(t)$ correspond à la réponse libre (c'est à dire en l'absence d'entrée $u(t)$) du système (1.47). Il a la forme :

$$h_0(t) = \sum_{k=0}^{\infty} L_0^k h(x_0) \frac{t^k}{k!} = e^{tL_0} h(x_0) \quad (1.55)$$

Le noyau $h_1(t, \tau_1)$ d'ordre 1 a la forme :

$$\begin{aligned} h_1(t, \tau_1) &= \sum_{k_0=0}^{\infty} \sum_{k_1=0}^{\infty} L_0^{k_0} L_1 L_0^{k_1} h(x_0) \frac{(t-\tau_1)^{k_1} \tau_1^{k_0}}{k_1! k_0!} \\ &= e^{\tau_1 L_0} L_1 e^{(t-\tau_1)L_0} h(x_0) \end{aligned} \quad (1.56)$$

Le noyau $h_n(t, \tau_1, \dots, \tau_n)$ d'ordre n a la forme :

$$\begin{aligned} h_n(t, \tau_1, \dots, \tau_n) &= \sum_{k_0=0}^{\infty} \dots \sum_{k_n=0}^{\infty} L_0^{k_0} L_1 L_0^{k_1} \dots L_1 L_0^{k_n} h(x_0) \frac{(t-\tau_n)^{k_n} \dots \tau_1^{k_0}}{k_n! \dots k_0!} \\ &= e^{\tau_1 L_0} L_1 e^{(\tau_2 - \tau_1)L_0} \dots L_1 e^{(t - \tau_n)L_0} h(x_0) \end{aligned} \quad (1.57)$$

1.4.2.2 Commentaire

Ces expressions analytiques des noyaux de Volterra ne sont pas utilisables telles quelles car les opérateurs qu'elles comportent sont difficilement calculables. Le calcul effectif des noyaux de Volterra reste donc un problème ouvert.

1.4.3 Modélisation des noyaux de Volterra sur bases de fonctions orthonormées

Dans la majeure partie des cas considérés, on ne peut pas disposer d'une expression analytique (ou d'un modèle de connaissance) des noyaux de Volterra car soit celle-ci ne peut pas être calculée (structure des noyaux inconnue), soit cette expression est trop complexe pour être ensuite utilisée (structure des noyaux trop complexe). Une manière de traiter ce problème est de trouver un modèle paramétrique valide pour les noyaux de Volterra sur une certaine plage de fonctionnement du système non linéaire. On utilisera ensuite les données d'entrée et de sortie du système non linéaire considéré pour identifier les noyaux de Volterra, même si la forme analytique de ces derniers reste inconnue.

Un moyen relativement courant de modéliser les noyaux de Volterra consiste en leur développement sur des bases de fonctions orthonormées multidimensionnelles, elles-mêmes obtenues à partir d'une ou de plusieurs bases de fonctions monodimensionnelles [Wahlberg, 1991], [Lindskog et Wahlberg, 1993], [Ninness et Gustafsson, 1994], [Akçay et Ninness, 1999].

Nous verrons ensuite qu'un autre moyen consiste en l'utilisation de la représentation temporelle de la série génératrice à variables non commutatives (introduite par Fliess) associée à la sortie $y(t)$ du système non linéaire. Cette méthode conduit à un modèle relativement parcimonieux pour chaque noyau de Volterra mais ceci au détriment d'une complexité croissante du modèle au fur et à mesure que le nombre de noyaux augmente.

Un autre modèle, inspiré du précédent, sera ensuite présenté. Ce modèle consiste à développer les noyaux de Volterra sur des produits de fonctions de transfert. Bien que la parcimonie de ce nouveau modèle soit légèrement moindre, nous verrons que sa complexité globale est considérablement diminuée, permettant ainsi une identification paramétrique beaucoup plus aisée.

1.4.3.1 Introduction

Les bases de fonctions orthonormées sont couramment utilisées en matière d'approximation et d'analyse de fonctions complexes. Une des premières études sur le sujet est l'oeuvre du mathématicien Yuk-Wing Lee, un des élèves de Wiener [Lee, 1932]. On considère aujourd'hui qu'il fut le premier à utiliser le terme de "synthèse" pour décrire le

rassemblement de divers éléments dans le but de répondre à un ensemble de performances.

Dans des domaines tels que la théorie de la commande, le traitement du signal et l'identification de systèmes, on utilise depuis longtemps les fonctions de Laguerre [Wahlberg, 1991], développées par Laguerre à la fin du 19^e siècle, étendues ensuite par Kautz à l'ordre 2 afin de modéliser des dynamiques non amorties [Wahlberg, 1994]. Les bases de fonctions orthonormées généralisées, introduites par Ninness et Akçay [Ninness et Gustafsson, 1994], [Akçay et Ninness, 1999], permettent l'introduction de pôles réels ou complexes conjugués afin de modéliser le plus grand nombre possible de dynamiques différentes.

1.4.3.2 Définition et propriétés

Une base de fonctions orthonormées est constituée de fonctions orthonormées entre elles et normales à elles-mêmes. Si l'on considère une base de fonctions orthonormées Φ définie par

$$\{\Phi_i(s)\} \quad i \in \mathbb{N} \quad (1.58)$$

et

$$\Phi_i(s) \in H_2(C_+) \quad (1.59)$$

où $H_2(C_+)$ est l'espace de Hardy, espace des fonctions analytiques dans le demi-plan complexe de Laplace C_+ tel que $Re(s) \geq 0$, et de carré sommable sur l'axe des imaginaires.

Ces fonctions $\Phi_i(s)$ sont orthonormées entre elles si leur produit interne à $H_2(C_+)$ a les propriétés suivantes :

$$\langle \Phi_i, \Phi_j \rangle = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \Phi_i(j\omega) \Phi_j^*(j\omega) d\omega \begin{cases} 1 \text{ si } i = j \\ 0 \text{ sinon} \end{cases} \quad (1.60)$$

où l'indice * désigne le conjugué d'un nombre complexe.

Dans la suite du mémoire, on adopte les notations suivantes : $\Phi(s)$ représente une fonction orthonormée dans le domaine de Laplace tandis que ϕ représente la réponse de la fonction Φ à une excitation $u(t)$.

1.4.3.3 Développement sur une base monovariable

On considère une fonction $f(s)$ de l'espace de Hardy $H_2(C_+)$. Cette fonction admet un développement unique sur la base de fonctions Φ

$$f(s) = \sum_{j=0}^{\infty} \theta_{ej} \Phi_j(s) \quad (1.61)$$

Les coefficients θ_{ej} sont les paramètres du développement.

Suivant le choix des fonctions $\Phi_j(s)$, les coefficients θ_{ej} tendent plus ou moins rapidement vers 0 lorsque j tend vers l'infini. Le développement (1.61) peut donc être tronqué à ses M premiers termes

$$\widehat{f}(s) = \sum_{j=0}^M \theta_{ej} \Phi_j(s) \quad (1.62)$$

1.4.3.4 Développement sur une base multivariable

On considère maintenant l'espace $L^2(T^n)$, $T = [0, \infty[$, des fonctions mesurables et de carré sommable. Toute fonction $f(t_1, \dots, t_n)$ de $L^2(T^n)$, admet un développement unique sur la base orthonormée $\{\phi_{m_1 \dots m_n}\}$ de la forme

$$f(t_1, \dots, t_n) = \sum_{m_1=0}^{\infty} \dots \sum_{m_n=0}^{\infty} f_{m_1 \dots m_n} \phi_{m_1 \dots m_n}(t_1, \dots, t_n) \quad (1.63)$$

où les termes $f_{m_1 \dots m_n}$ sont les paramètres du développement.

Proposition 1.1 : Une base orthonormée $\{\phi_{m_1 \dots m_n}\}$ de $L^2(T^n)$ peut être obtenue à partir de la base orthonormée $\{\phi_m\}$ de $L^2(T)$ en formant les produits directs

$$\phi_{m_1} \otimes \dots \otimes \phi_{m_n} = \prod_{i=1}^n \phi_{m_i} \text{ avec } m_1 \geq \dots \geq m_n \quad (1.64)$$

Preuve :

On considère un multi-index (m_1, \dots, m_n) tel que $m_1 \geq m_2 \geq \dots \geq m_n$ et une base orthonormée $\{\phi_m\}$ monodimensionnelle de l'espace $L^2(T)$. On définit l'ensemble des fonctions $\{\phi_{m_1 \dots m_n}\}$ de $L^2(T^n)$ de la manière suivante :

$$\phi_{m_1 \dots m_n} = \prod_{i=1}^n \phi_{m_i} \quad (1.65)$$

Calculons le produit scalaire $\langle \phi_{m_1 \dots m_n}, \phi_{p_1 \dots p_n} \rangle$:

$$\begin{aligned} \langle \phi_{m_1 \dots m_n}, \phi_{p_1 \dots p_n} \rangle &= \int_{-j\infty}^{j\infty} \dots \int_{-j\infty}^{j\infty} \Phi_{m_1 \dots m_n}(f_1, \dots, f_n) \Phi_{p_1 \dots p_n}^*(f_1, \dots, f_n) df_1 \dots df_n \\ &= \int_{-j\infty}^{j\infty} \dots \int_{-j\infty}^{j\infty} \prod_{i=1}^n \Phi_{m_i}(f_i) \prod_{j=1}^n \Phi_{p_j}^*(f_j) df_1 \dots df_n \\ &= \prod_{i=1}^n \int_{-j\infty}^{j\infty} \dots \int_{-j\infty}^{j\infty} \Phi_{m_i}(f_i) \Phi_{p_i}^*(f_i) df_i \\ &= \prod_{i=1}^n \langle \Phi_{m_i}, \Phi_{p_i} \rangle \\ &= \prod_{i=1}^n \delta_{m_i, p_i} \\ &= \delta_{m_1 \dots m_n, p_1 \dots p_n} \end{aligned} \quad (1.66)$$

où $\delta_{m_1 \dots m_n, p_1 \dots p_n}$ est le symbole de Kronecker défini par

$$\delta_{m_1 \dots m_n, p_1 \dots p_n} = \begin{cases} 1 & \text{si } m_1 = \dots = m_n = p_1 = \dots = p_n \\ 0 & \text{sinon} \end{cases} \quad (1.67)$$

Les fonctions $\phi_{m_1 \dots m_n}$ sont donc orthogonales entre elles et normales à elles-mêmes.

Par conséquent, $\{\phi_{m_1 \dots m_n}\}$ constitue une base orthonormée de $L^2(T^n)$. ■

Suivant le choix de la base de fonctions multivariables, les coefficients du développement peuvent tendre plus ou moins rapidement vers 0 lorsque le nombre de termes tend vers l'infini. On peut donc, comme dans le cas monovariable, approximer le développement (1.63) :

$$\hat{f}(t_1, \dots, t_n) = \sum_{m_1=0}^M \dots \sum_{m_n=0}^M f_{m_1 \dots m_n} \Phi_{m_1 \dots m_n}(t_1, \dots, t_n) \quad (1.68)$$

1.4.3.5 Développement des noyaux de Volterra [Mansion, 1976]

Nous allons utiliser le résultat précédent pour approximer les noyaux de Volterra. On rappelle que la sortie d'un système non linéaire mono entrée-mono sortie peut être développée en une série de Volterra

$$y(t) = \sum_{i=1}^{\infty} \left(\int_0^t \dots \int_0^t h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i u(t - \tau_j) d\tau_j \right) \quad (1.69)$$

Dans le cas général, le noyau de Volterra $h_i(\tau_1, \dots, \tau_i)$ est une fonction multivariable et admet donc un développement unique sur une base de fonctions $\{\phi_{m_1 \dots m_i}\}$ de $L^2(T^i)$:

$$h_i(\tau_1, \dots, \tau_i) = \sum_{m_1=0}^{\infty} \dots \sum_{m_i=0}^{\infty} c_{m_1 \dots m_i} \phi_{m_1 \dots m_i}(\tau_1, \dots, \tau_i) \quad (1.70)$$

Proposition 1.2 : en considérant les propriétés d'orthogonalité de la base de fonctions $\{\phi_{m_1 \dots m_i}\}$, la réponse $H_i[u(t)]$ du noyau (1.70) à l'entrée $u(t)$ peut être approximée par

$$\widehat{H}_i[u(t)] = \sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{M_i} b_{m_1 \dots m_i} I_{m_1}(t) \dots I_{m_i}(t) \quad (1.71)$$

où le terme $I_{m_j}(t)$ est la réponse de la fonction orthonormée ϕ_{m_j} à l'entrée $u(t)$.

Preuve : comme on l'a vu précédemment, le développement (1.70) peut être tronqué :

$$\widehat{h}_i(\tau_1, \dots, \tau_i) = \sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{M_i} c_{m_1 \dots m_i} \phi_{m_1 \dots m_i}(\tau_1, \dots, \tau_i) \quad (1.72)$$

La base de fonctions multidimensionnelles orthonormées $\{\phi_{m_1 \dots m_i}\}$ de $L^2(T^i)$ étant obtenue à l'aide de la base de fonctions monodimensionnelles orthonormées $\{\phi_m\}$ de $L^2(T)$, on peut réécrire l'approximation précédente du développement de la manière suivante :

$$\begin{aligned} \hat{h}_i(\tau_1, \dots, \tau_i) &= \sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{m_{i-1}-1} c_{m_1 \dots m_i} (\phi_{m_1}(\tau_1) \dots \phi_{m_i}(\tau_i) + \dots + \phi_{m_1}(\tau_i) \dots \phi_{m_i}(\tau_1)) \\ &\quad + \sum_{m_1=0}^{M_i} c_{m_1 \dots m_1} \phi_{m_1}(\tau_1) \dots \phi_{m_1}(\tau_i) \end{aligned} \quad (1.73)$$

La réponse $H_i[u(t)]$ du noyau de Volterra d'ordre i à l'entrée $u(t)$ du système s'écrit de la manière suivante :

$$H_i[u(t)] = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i u(t - \tau_j) d\tau_j \quad (1.74)$$

On en déduit donc l'expression approchée $\hat{H}_i[u(t)]$:

$$\begin{aligned} \hat{H}_i[u(t)] &= \sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{m_{i-1}} c_{m_1 \dots m_i} \\ &\quad \int_0^t (\phi_{m_1}(\tau_1) \dots \phi_{m_i}(\tau_i) + \dots + \phi_{m_1}(\tau_i) \dots \phi_{m_i}(\tau_1)) \prod_{j=1}^i u(t - \tau_j) d\tau_j \\ &\quad + \sum_{m_1=0}^{M_i} c_{m_1 \dots m_1} \int_0^t \dots \int_0^t \phi_{m_1}(\tau_1) \dots \phi_{m_1}(\tau_i) \prod_{j=1}^i u(t - \tau_j) d\tau_j \end{aligned} \quad (1.75)$$

Afin de simplifier les notations, on note $I_{m_j}(t)$ la réponse de la fonction orthonormée ϕ_{m_j} à l'entrée $u(t)$:

$$I_{m_j}(t) = \int_0^t \phi_{m_j}(\tau) u(t - \tau) d\tau \quad (1.76)$$

L'expression approchée $\hat{H}_i[u(t)]$ de la réponse $H_i[u(t)]$ peut donc s'écrire :

$$\begin{aligned} \hat{H}_i[u(t)] &= \sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{m_{i-1}} c_{m_1 \dots m_i} (i!) I_{m_1}(t) \dots I_{m_i}(t) \\ &\quad + \sum_{m_1=0}^{M_i} c_{m_1 \dots m_1} I_{m_1}(t) \dots I_{m_1}(t) \end{aligned} \quad (1.77)$$

On peut uniformiser la notation des coefficients du développement de la manière suivante :

$$b_{m_1 \dots m_i} = \begin{cases} c_{m_1 \dots m_i} & \text{si } m_1 = \dots = m_i \\ c_{m_1 \dots m_i} (i!) & \text{sinon} \end{cases} \quad (1.78)$$

avec $m_1 \geq \dots \geq m_i$.

On obtient enfin l'expression approchée générale de la sortie $\hat{H}_i[u(t)]$ du noyau d'ordre i :

$$\hat{H}_i[u(t)] = \sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{m_{i-1}} b_{m_1 \dots m_i} I_{m_1}(t) \dots I_{m_i}(t) \quad (1.79)$$

■

Considérons l'exemple du noyau de Volterra d'ordre 2 développé sur 2 fonctions orthonormées (troncature à $M_2 = 1$). Sa représentation schématique est donnée sur la figure 1.6.

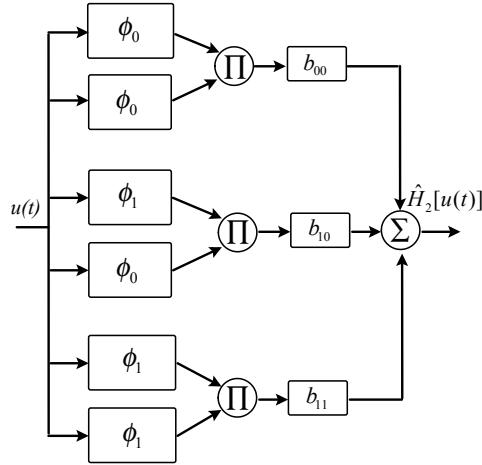


Fig. 1.6 - Noyau d'ordre 2 développé sur 2 fonctions orthonormées

La représentation schématique générale du modèle $\hat{H}_i[u(t)]$ de la réponse du noyau d'ordre i (1.79) est donnée sur la figure 1.7.

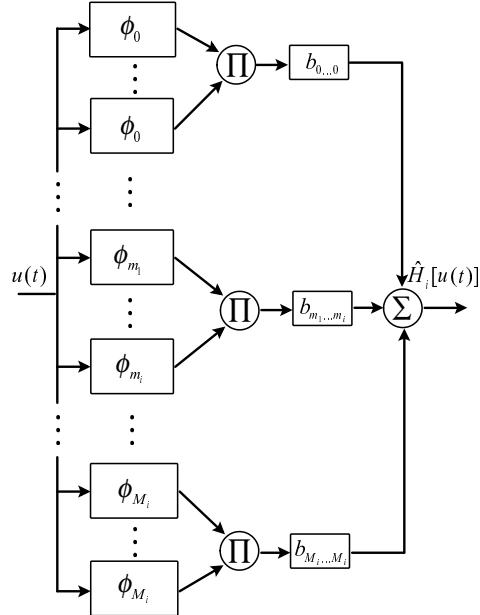


Fig. 1.7 - Noyau d'ordre i développé sur une base de fonctions orthonormées

Enfin, l'approximation $\hat{y}(t)$ (troncature de l'expression (1.69) à l'ordre n) de la sortie $y(t)$ du système non linéaire est donnée par

$$\hat{y}(t) = \sum_{i=1}^n \left(\sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{m_{i-1}} b_{m_1 \dots m_i} I_{m_1}(t) \dots I_{m_i}(t) \right) \quad (1.80)$$

et représentée sur la figure 1.8.

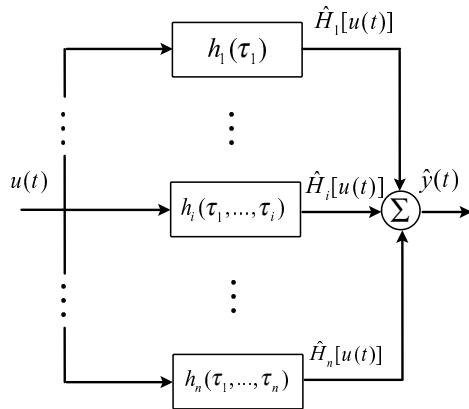


Fig. 1.8 - Approximation de la sortie $y(t)$ du système non linéaire

Une fois le modèle du système non linéaire établi, l'étape suivante consiste à choisir la forme des fonctions de la base orthonormée.

1.4.3.6 Choix des fonctions orthonormées

Le choix de la base orthonormée monodimensionnelle initiale Φ_n est primordial pour la qualité du modèle. Les dynamiques des fonctions génératrices doivent être aussi proches que possible de celles des noyaux de Volterra. Ces derniers dépendant du système non linéaire considéré, le choix de la base de fonctions orthonormées devra donc se faire au cas par cas parmi l'éventail des possibilités.

a) Fonctions de Laguerre

Les fonctions de Laguerre ont été les premières à être utilisées afin de modéliser les systèmes (linéaires ou non linéaires). On les utilise plus particulièrement dans le cas de systèmes apériodiques [Wahlberg, 1991]. Elles sont définies dans l'espace de Laplace de la manière suivante :

$$L_n(s) = \frac{\sqrt{2\alpha}}{s + \alpha} \left(\frac{s - \alpha}{s + \alpha} \right)^{n-1} \quad (1.81)$$

Ces fonctions sont composées d'un filtre passe-bas du premier ordre suivi d'une fonction passe-tout. Le modèle de Volterra ainsi défini ne comporte dans ce cas qu'un seul pôle réel multiple $-\alpha$.

b) Fonctions de Kautz

Les fonctions de Kautz sont une extension des fonctions de Laguerre au deuxième ordre afin de modéliser les systèmes oscillants [Wahlberg, 1994]. Elles sont définies dans l'espace de Laplace de la manière suivante :

$$\begin{cases} \Psi_{2k-1}(s) = \frac{\sqrt{2bs}}{s^2+bs+c} \left[\frac{s^2-bs+c}{s^2+bs+c} \right]^{k-1} \\ \Psi_{2k}(s) = \frac{\sqrt{2bc}}{s^2+bs+c} \left[\frac{s^2-bs+c}{s^2+bs+c} \right]^{k-1} \end{cases} \quad \text{avec } k \geq 0, b > 0, c > 0 \quad (1.82)$$

Les deux pôles conjugués sont calculés par résolution de l'équation $s^2 + bs + c = 0$.

On remarque que dans le cas où $b = \alpha$ et $c = 0$, on retrouve la fonction de Laguerre définie au paragraphe précédent.

c) Fonctions orthonormées généralisées

Un des principaux inconvénients des deux types précédents de fonctions orthonormées est que l'on ne peut introduire qu'un et deux pôles dans chaque fonction de Laguerre et de Kautz respectivement. Dans le cas d'un système comportant plusieurs non-linéarités différentes ou plus simplement des modes éloignées, on ne pourra pas introduire plusieurs dynamiques. Il faut alors considérer un nombre suffisamment grand de fonctions orthonormées, ce qui entraînera un problème de surdimensionnement de modèle et de surparamétrisation.

Pour remédier à ce problème, on dispose des fonctions orthonormées généralisées, introduites par Ninness et Akçay [Ninness et Gustafsson, 1994], [Akçay et Ninness, 1999], définies dans l'espace de Laplace de la manière suivante :

$$\Phi_n(s) = \frac{\sqrt{2Re\{p_n\}}}{s + p_n} \prod_{k=1}^{n-1} \frac{s - \bar{p}_k}{s + p_k} \quad (1.83)$$

où le terme \bar{p}_k désigne le conjugué du nombre complexe p_k .

La construction de telles fonctions préserve l'orthogonalité et permet l'incorporation d'une grande variété de pôles p_k convenablement répartis afin de modéliser les différentes dynamiques ou les non-linéarités du système considéré. Ces fonctions permettent de correctement décrire la dynamique de chaque noyau et d'éviter ainsi d'obtenir un modèle de

dimension trop importante.

Un autre avantage de l'utilisation des fonctions orthonormées généralisées est la possibilité d'introduire des pôles complexes afin de modéliser des dynamiques oscillantes. Ces pôles complexes sont introduits par paires.

On introduit les fonctions $\Phi'_n(s)$ et $\Phi''_n(s)$ définies de la manière suivante :

$$\begin{pmatrix} \Phi'_n(s) \\ \Phi''_n(s) \end{pmatrix} = \begin{pmatrix} C_0 & C_1 \\ C'_0 & C'_1 \end{pmatrix} \begin{pmatrix} \Phi_n(s) \\ \Phi_{n+1}(s) \end{pmatrix} \quad (1.84)$$

où C_0 , C_1 , C'_0 et C'_1 sont des nombres complexes, la fonction $\Phi_n(s)$ est donnée par (1.83) et la fonction $\Phi_{n+1}(s)$ est définie de la manière suivante :

$$\Phi_{n+1}(s) = \frac{\sqrt{2p_n}}{s + \bar{p}_n} \prod_{k=1}^{n-1} \frac{(s - \bar{p}_k)}{(s + p_k)} \quad (1.85)$$

1.4.3.7 Identification

On considère le système non linéaire quadratique dont la représentation d'état est la suivante :

$$\begin{cases} \dot{x}(t) = -x(t) - x^2(t) + u(t) \\ y(t) = x(t) \end{cases} \quad (1.86)$$

L'utilisation d'une série de Volterra pour modéliser un système non linéaire suppose que l'on puisse identifier les termes de la série à partir d'un signal d'entrée $u(t)$ quelconque mais suffisamment riche pour pouvoir sensibiliser l'ensemble des paramètres. Dans le cas du système non linéaire (1.86), l'utilisation d'une séquence pseudo aléatoire bornée en amplitude (figure 1.11(a)) permet d'effectuer une identification globale des paramètres de tous les termes du modèle de Volterra.

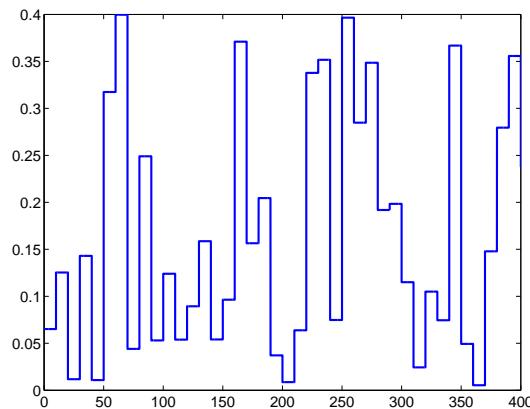


Fig. 1.9 - Séquence pseudo-aléatoire bornée en amplitude

Ce type d'excitation comporte un nombre suffisamment important de variations d'amplitude pour sensibiliser tous les termes de la série de Volterra.

La réponse du système non linéaire (1.86) au signal d'entrée défini par la figure 1.11(a) est donnée par la figure 1.11(b).

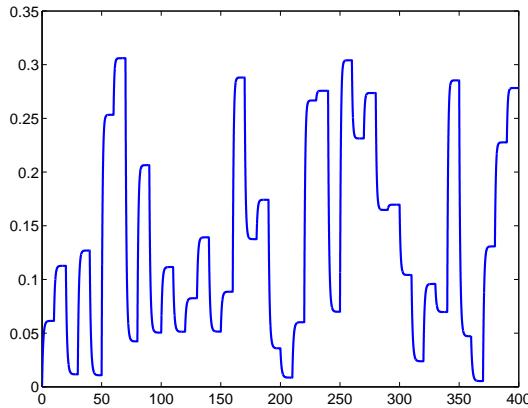


Fig. 1.10 - Réponse du système non linéaire

On utilisera ce signal d'entrée et cette réponse du système non linéaire afin d'identifier les paramètres des modèles de Volterra dans toute la suite de ce chapitre.

La structure du modèle de Volterra est d'abord fixée en effectuant une troncature à ses 2 premiers termes. Le premier terme permet de modéliser la partie linéaire du système considéré. Le second terme prend en compte la non-linéarité. Nous verrons que cette troncature, bien qu'arbitraire, peut cependant être suffisante pour une modélisation correcte du système non linéaire.

Pour les raisons définies précédemment, nous choisissons de développer chaque noyau de Volterra sur des fonctions orthonormées généralisées définies par (1.83). Le nombre de fonctions orthonormées généralisées est d'abord arbitrairement fixé à 4 pour chaque noyau de Volterra, ce qui fixe le nombre total de paramètres à 14.

On peut alors fixer la valeur des pôles des fonctions orthonormées généralisées à une valeur arbitraire. En l'absence de connaissance a priori sur le système non linéaire, le pôle de la première fonction est fixé à 10 et celui de la deuxième fonction à 20, celui de la troisième à 30 et le dernier à 40.

A ce point de l'étude, on remarque que le modèle de Volterra développé sur des fonctions orthonormées généralisées reste linéaire par rapport aux paramètres. La sortie $y(t)$ du modèle de Volterra peut donc s'écrire sous la forme du produit d'un régresseur $\varphi(t)$ par un vecteur paramètre θ :

$$y(t) = \varphi(t)\theta \quad (1.87)$$

La solution optimale θ_{opt} à un tel problème est donnée par la méthode classique des moindres carrés :

$$\theta_{opt} = (\varphi^T(t)\varphi(t))^{-1}\varphi^T(t)y(t) \quad (1.88)$$

Une fois l'identification paramétrique achevée, on utilise un autre signal d'entrée (figure 1.11(a)) et la réponse correspondante (figure 1.11(b)) du système non linéaire (1.86) afin de valider le modèle de Volterra obtenu.

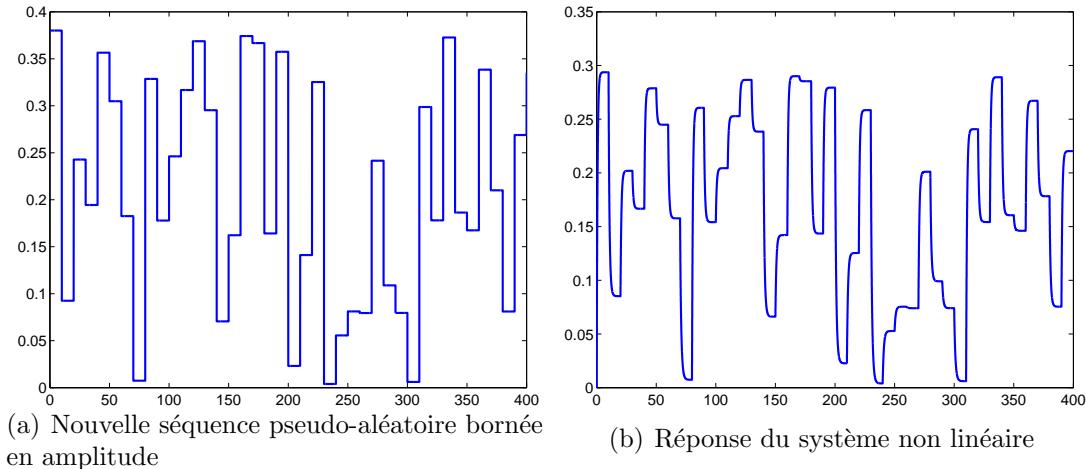


Fig. 1.11 - Signaux d'entrée et de sortie utilisés pour valider le modèle de Volterra

La figure 1.12(a) représente la sortie du système non linéaire (1.86) pour ce nouveau signal d'entrée et la sortie du modèle de Volterra décrit précédemment (pour les valeurs paramétriques optimales calculées par la méthode des moindres carrés).

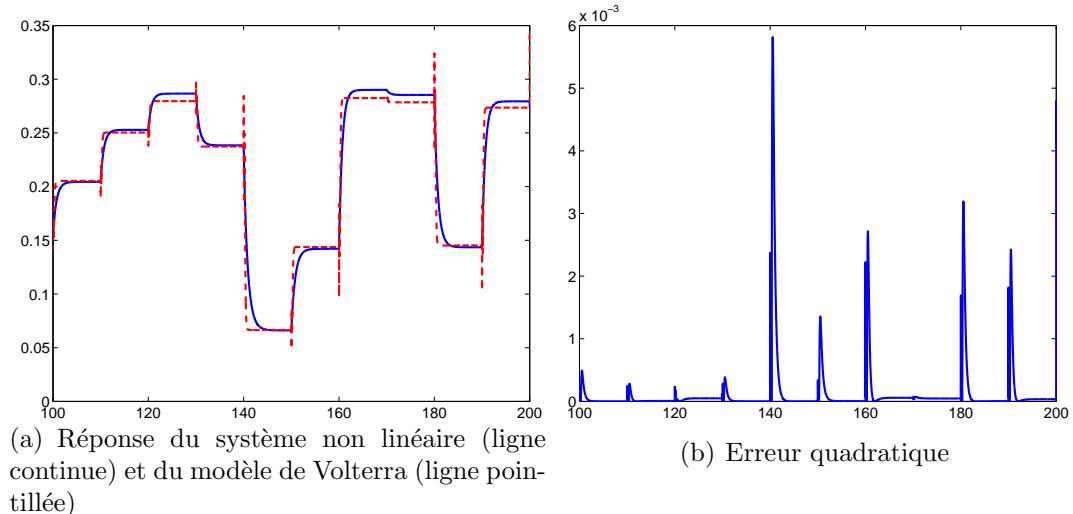


Fig. 1.12 - Modèle de Volterra tronqué à 2 noyaux développés sur 4 fonctions

La figure 1.12(b) représente l'erreur quadratique entre les deux courbes de la figure 1.12(a). Le critère quadratique J est égal à 8.95. La reconstruction est donc mauvaise, ce qui ne surprend pas l'utilisateur car les choix de structure, de nombre de fonctions et de valeur paramétriques du modèle de Volterra sont tous arbitraires.

Une structure complexe de modèle et un nombre important de paramètres ne suffisent donc pas à représenter correctement le système non linéaire considéré.

Une première solution consiste à complexifier le modèle de Volterra en augmentant soit le nombre de termes, soit le nombre de fonctions orthonormées généralisées utilisées pour modéliser chaque noyau. On peut par exemple utiliser 5 fonctions orthonormées généralisées au lieu de 4. Le nombre total de paramètres est dorénavant de 20.

La figure 1.13(a) représente la sortie du système non linéaire (1.86) et la sortie du modèle de Volterra complexifié. Le critère quadratique J est désormais égal à 8.04.

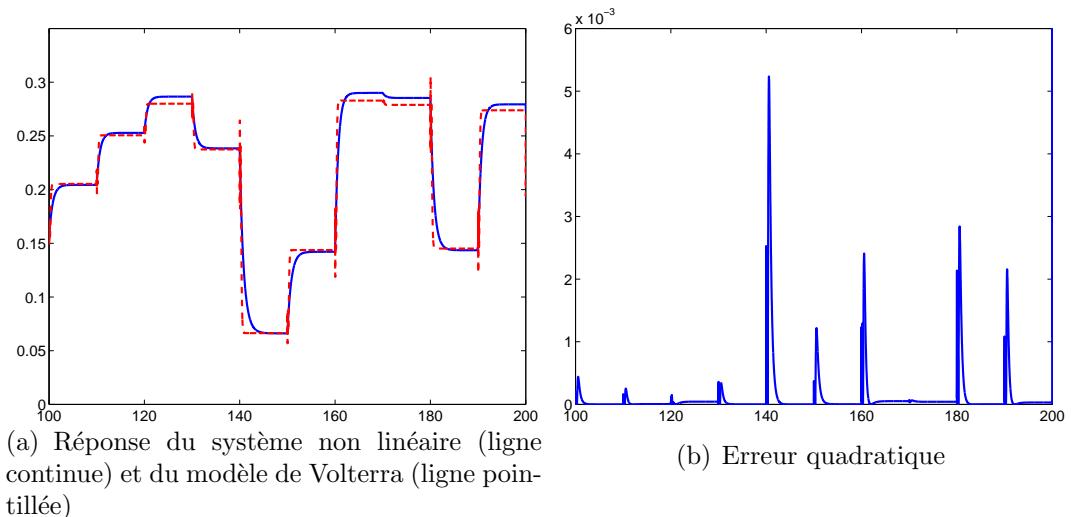


Fig. 1.13 - Modèle de Volterra tronqué à 2 noyaux développés sur 5 fonctions

Un modèle de Volterra plus complexe est donc plus précis mais reste incapable de représenter précisément la sortie du système non linéaire. La solution théorique consiste donc à augmenter indéfiniment le nombre de termes du modèle ou bien le nombre de fonctions orthonormées généralisées utilisées pour modéliser chaque noyau. Dans ces deux cas, le nombre de paramètres augmente et le calcul de la solution optimale par la méthode des moindres carrés devient problématique.

Une autre solution consiste à choisir d'autres valeurs de pôles tout en gardant la même structure de modèle de Volterra. En l'absence de connaissance a priori sur le système non linéaire considéré, le choix des valeurs paramétriques optimales doit se faire de manière empirique. Mais, en pratique, il arrive souvent que l'on dispose de la réponse du système non linéaire à différents signaux d'entrée. Ces réponses peuvent permettre à l'utilisateur d'estimer le temps de réponse global du système non linéaire et ainsi d'approcher beaucoup plus rapidement les valeurs optimales d'un ou plusieurs paramètres.

En conservant la même structure de modèle (2 noyaux de Volterra modélisés chacun sur 4 fonctions orthonormées généralisées) et en fixant les valeurs des pôles des fonctions à 1, 2, 3 et 4 on obtient la réponse de la figure 1.14(a).

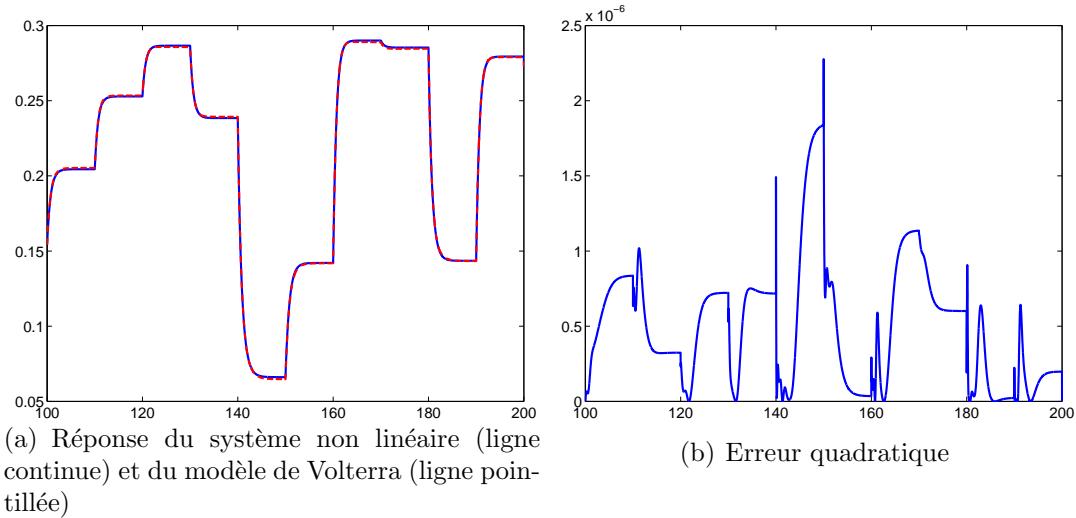


Fig. 1.14 - Modèle de Volterra tronqué à 2 noyaux développés sur 4 fonctions avec choix convenable des pôles

L'erreur quadratique entre les deux courbes de la figure 1.14(a) est donnée par la figure 1.14(b). Le critère quadratique J est égal à 0.022.

La qualité du modèle de Volterra est dans ce cas bien meilleure que précédemment. La connaissance a priori de la dynamique globale du système non linéaire peut donc aider l'utilisateur à fixer convenablement la structure du modèle de Volterra développé sur une base de fonctions orthonormées généralisées.

1.4.3.8 Discussion

Le modèle de Volterra dont les noyaux sont développés sur des bases de fonctions orthonormées généralisées apparaît donc comme un très bon moyen de représenter un très grand nombre de dynamiques différentes. Cependant, cette propriété intéressante entraîne un certain nombre de problèmes d'identification paramétrique :

- Problème du choix des ordres de troncature : la structure globale optimale du modèle de Volterra n'est pas connue d'avance par l'utilisateur. Dans la plupart des cas, une série de Volterra étant uniformément convergente, il n'est pas utile de conserver une infinité de termes (donc une infinité de noyaux) pour modéliser correctement un système non linéaire. Le problème du choix de l'ordre idéal de troncature se pose alors pour chaque système non linéaire étudié. Dans l'état actuel des connaissances, seule une connaissance a priori du système étudié permet de résoudre ce problème. De plus, une fois l'ordre de troncature fixé, l'utilisateur doit également fixer le nombre de fonctions orthonormées généralisées sur lesquelles sera développé chaque noyau de Volterra. Ici également, la connaissance a priori du comportement du système peut aider l'utilisateur à fixer des valeurs optimales du nombre de fonctions pour

chaque noyau.

- Problème du choix des pôles des fonctions orthonormées généralisées : un des avantages du modèle de Volterra développé sur bases de fonctions orthonormées généralisées est le grand nombre de pôles différents introduits dans les fonctions. Mais, dans de nombreux cas, l'utilisateur n'a pas de connaissance a priori sur les dynamiques du système étudié et doit fixer des valeurs de pôles a priori inexactes. Ceci pose des problèmes d'identification si le nombre de fonctions de chaque développement a été tronqué au départ par l'utilisateur : en effet, la procédure d'identification doit s'adapter à ces valeurs de pôles, fixées dès le départ, et l'algorithme d'identification par moindres carrés peut fournir un mauvais résultat d'identification des paramètres si les valeurs de pôles ne correspondent pas aux dynamiques réelles du système étudié.
- Problème du nombre de paramètres : ce troisième problème découle du précédent. En effet, pour pallier au problème du choix des pôles en l'absence de connaissance a priori sur les dynamiques du système non linéaire étudié, l'utilisateur peut choisir d'introduire le plus de dynamiques différentes possibles dans le modèle de Volterra, augmentant ainsi le nombre de fonctions sur lesquelles est développé chaque noyau de Volterra. Ceci provoque une forte augmentation du nombre total de paramètres du modèle de Volterra, sacrifiant ainsi l'objectif de parcimonie fixé au départ de notre étude.

Un des objectifs essentiels de cette étude est de conserver une certaine parcimonie paramétrique dans le modèle étudié, à la différence des multi-modèles, du modèle NARMAX et des réseaux artificiels de neurones évoqués précédemment. Une solution pour réduire le nombre de paramètres nécessaires est le développement des noyaux de Volterra non plus sur bases de fonctions orthonormées généralisées mais sur des fonctions de transfert.

1.4.4 Modélisation des noyaux de Volterra sur fonctions de transfert

1.4.4.1 Introduction

L'objectif de parcimonie est difficilement réalisable en conservant la modélisation des noyaux de Volterra établie à l'aide des fonctions orthonormées généralisées, en particulier lorsque connaissance a priori sur le comportement du système non linéaire est faible.

Une solution à ce problème consisterait à optimiser le choix de pôles des fonctions de la base orthonormée généralisée à l'aide d'un algorithme de programmation non linéaire. Cependant, en l'absence de connaissance sur la structure globale optimale du modèle de Volterra, cette optimisation devrait être effectuée pour différents ordres de troncature de la série de Volterra utilisée et des développements de chaque noyau de Volterra. Une telle procédure nécessiterait en outre un critère valable de comparaison des résultats d'optimi-

sation obtenus pour chaque structure de modèle employée. Le volume de calcul nécessaire deviendrait rapidement important et les résultats obtenus seraient difficilement exploitables.

Pourtant, l'idée d'optimiser par programmation non linéaire tous les pôles et les zéros des fonctions orthonormées généralisées afin de compenser le manque de connaissance a priori sur le système non linéaire paraît séduisante. Nous allons donc essayer d'effectuer cette optimisation mais en remplaçant les fonctions orthonormées généralisées par des fonctions de transfert classiques, dont la forme plus souple se prête davantage à la procédure d'optimisation des gains et des pôles.

Le développement théorique que nous allons présenter est d'un abord difficile à cause de calculs lourds et laborieux. Il conduit cependant à un modèle très parcimonieux des noyaux de Volterra, répondant ainsi à l'un des objectifs que nous nous sommes fixés.

1.4.4.2 Cas particulier : modèle de Wiener

On reprend le modèle de Wiener (figure 1.15), qui n'est autre qu'un cas particulier du modèle de Volterra, plus simple mais moins flexible.

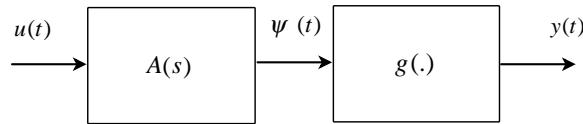


Fig. 1.15 - Modèle de Wiener

Nous allons utiliser une forme particulière de ce type de structure pour modéliser chaque noyau de Volterra.

Dans le modèle de Wiener, tout comme dans celui de Hammerstein, la partie linéaire est séparée de la partie non linéaire.

Nous allons commencer par restreindre la partie non linéaire du modèle de Wiener à un cas particulier, celui d'un gain g_1 :

$$g(\psi(t)) = g_1\psi(t) \quad (1.89)$$

La partie linéaire dynamique peut être la réponse impulsionnelle $h(t)$ d'une fonction $A(s)$. On peut alors se servir de ce cas particulier de modèle de Wiener pour représenter le premier noyau de Volterra de signal d'entrée $u(t)$ et de signal de sortie $y_1(t) = H_1[u(t)]$:

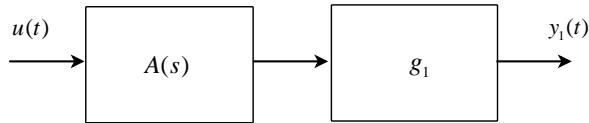


Fig. 1.16 - Modèle de Wiener représentant le premier noyau de Volterra

Si on veut maintenant représenter le noyau de Volterra d'ordre i , on utilise le produit de plusieurs réponses impulsionales, suivi d'un gain (figure 1.17).

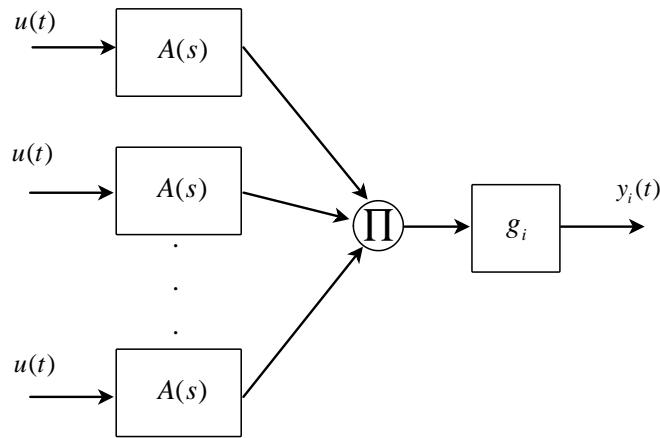


Fig. 1.17 - Modèle de Wiener représentant le noyau de Volterra d'ordre i

Dans cette forme de représentation, les paramètres du modèle restent les gains g_i utilisés dans chaque noyau de Volterra. On se rapproche donc du modèle de Volterra dans lequel chaque noyau est développé sur des fonctions orthonormées généralisées.

L'idée d'optimiser les pôles et les zéros des fonctions utilisées pour modéliser les noyaux au lieu de les fixer d'avance et d'optimiser simplement les gains du modèle conduit aux modifications suivantes : pour tous les noyaux de Volterra, on peut poser $g_i = 1$ et considérer ensuite que les termes $h(t)$ sont les réponses de fonctions de transfert du premier ordre du type :

$$A(s) = \frac{b_0}{a_0 + a_1 s} \quad (1.90)$$

Si l'on considère ensuite une seule fonction de transfert du type (1.90) pour chaque noyau de Volterra, on peut définir un modèle pour le système non linéaire, inspiré du modèle particulier de Wiener.

Par exemple, le modèle considéré pour le noyau d'ordre i serait représenté sur la figure 1.18.

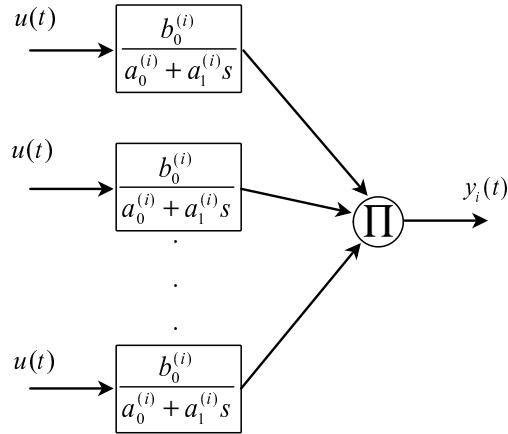


Fig. 1.18 - Noyau de Volterra d'ordre i modélisé par un produit de fonctions de transfert

Dans ce nouveau modèle, aucun paramètre n'est fixé au départ par l'utilisateur. Un algorithme de programmation non linéaire permettra l'évaluation de chacun des gains et des pôles de chacune des fonctions de transfert modélisant un noyau de Volterra.

Il s'agit maintenant d'essayer de généraliser cette forme de modèle afin de pouvoir représenter le plus grand nombre possible de systèmes non linéaires.

1.4.4.3 Généralisation : utilisation des séries de Fliess

a) Introduction

Le modèle 1.18 présenté au paragraphe précédent peut être vu comme un cas particulier de modèle de Wiener, lui-même considéré comme cas particulier de modèle de Volterra. Par contre, ce type de modèle répond bien à l'objectif d'identification globale des paramètres que nous nous étions fixé afin que l'utilisateur n'ait pas décider lui-même des valeurs de certains paramètres sans connaissance a priori sur le système non linéaire.

Un moyen de généraliser cette forme de modélisation des noyaux de Volterra sur des fonctions de transfert est d'utiliser les séries génératrices non commutatives évoquées précédemment.

Nous avons vu qu'une méthode de résolution du système non linéaire (1.86) consiste à utiliser la représentation temporelle de la série génératrice (1.49), développée en une

somme de séries génératrices représentant chacune un terme de la série de Volterra.

Reprendons la série génératrice (1.49) associée au système non linéaire (1.86). Elle s'écrit sous la forme :

$$g = h(x_0) + \sum_{v \geq 0} \sum_{j_0, \dots, j_v=0}^1 L_{j_v} \dots L_{j_1} h(x_0) z_{j_0} \dots z_{j_{v-1}} z_{j_v}$$

Afin de résoudre le système (1.86), il est possible de développer la série génératrice (1.49) qui lui est associée en une somme de termes de la forme :

$$g = \sum_{i=0}^{\infty} g_i \quad (1.91)$$

où g_i regroupe tous les termes de la série g ayant exactement i occurrences en z_1 . Chaque terme g_i est lui-même une série génératrice qui représente le terme d'ordre i du développement en série de Volterra de la sortie $y(t)$ du système non linéaire. On peut alors montrer que tous les termes g_i du développement admettent, dans le cas du système (1.86), une représentation de la forme :

$$g_i = (1 - a_{i0}z_0)^{-p_{i0}} z_1 (1 - a_{i1}z_0)^{-p_{i1}} z_1 \dots z_1 (1 - a_{ii}z_0)^{-p_{ii}} \quad (1.92)$$

où $a_{i0}, a_{i1}, \dots, a_{ii}$ sont des nombres complexes et $p_{i0}, p_{i1}, \dots, p_{ii}$ sont des entiers naturels.

Notre objectif est de pouvoir modéliser chacun des noyaux de Volterra sous la forme d'une somme de produits de réponses à des fonctions de transfert. Pour ce faire, nous allons utiliser la représentation temporelle de chaque terme g_i de la somme (1.91).

Considérons la série rationnelle

$$g_1 = (1 - a_0z_0)^{-1} z_1 (1 - a_1z_0)^{-1} \quad (1.93)$$

qui peut se développer sous la forme

$$g_1 = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_0^i a_1^j z_0^i z_1 z_0^j \quad (1.94)$$

L'expression $z_0^i z_1 z_0^j$ peut être développée en explicitant les opérateurs z_0 et z_1 (1.51) :

$$z_0^i z_1 z_0^j \Leftrightarrow \int_0^t (d\tau_0)^i \int_0^t u(\tau_1) d\tau_1 \int_0^t (d\tau_0)^j \quad (1.95)$$

Cette intégrale itérée est égale à

$$\int_0^t \frac{(t - \tau_1)^i \tau_1^j}{i! j!} u(\tau_1) d\tau_1 \quad (1.96)$$

Par conséquent, la série rationnelle g_1 est la représentation symbolique de

$$\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_0^i a_1^j \int_0^t \frac{(t-\tau_1)^i \tau_1^j}{i! j!} u(\tau_1) d\tau_1 \quad (1.97)$$

Cette expression peut être vue comme le développement d'un produit de deux exponentielles et peut donc être écrite sous la forme

$$\int_0^t e^{a_0(t-\tau)} e^{a_1} d\tau \quad (1.98)$$

Ce résultat peut être généralisé sans difficulté à tous les termes de la série génératrice g .

La représentation temporelle générale de chaque terme g_i est la suivante :

$$\int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_i} f_{a_0}^{p_0}(t-\tau_i) \dots f_{a_{i-1}}^{p_{i-1}}(\tau_2-\tau_1) f_{a_i}^{p_i}(\tau_1) \prod_{j=1}^i u(\tau_j) d\tau_j \quad (1.99)$$

où les fonctions $f_a^p(t)$ s'expriment de la manière suivante :

$$f_a^p(t) = \left(\sum_{j=0}^{p-1} \frac{C_{p-1}^j}{j!} a^j t^j \right) e^{at} \quad (1.100)$$

b) Séparation de l'intégrale multiple

Afin de modéliser le noyau de Volterra d'ordre n sous la forme d'une somme de produits de réponses de fonctions de transfert, il est nécessaire de séparer l'intégrale multiple (1.99), que nous noterons $y_n(t)$, en une somme de produits d'intégrales de convolution simples.

Théorème 1.3 *Séparation de l'intégrale multiple $y_n(t)$ en une somme de produits d'intégrales simples $y_{n,r,i}(t)$*

L'intégrale multiple (1.99) où les fonctions $f_a^p(t)$ ont pour expression (1.100) peut être réécrite sous la forme :

$$y_n(t) = \int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} D_0 D_1 \dots D_{n-1} e^{(a_0-a_1)(t-\tau_n)} e^{(a_1-a_2)(t-\tau_{n-1})} \dots e^{a_{n-1}(t-\tau_1)} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (1.101)$$

où chaque terme D_i est fonction de t , τ_i , a_i et p_i ($i = 1, \dots, n$).

Cette intégrale multiple peut se décomposer en une somme de $P_n = p_0 \prod_{i=1}^{n-1} \frac{p_i(p_i+1)}{2}$ intégrales multiples $y_{n,r}(t)$ de la forme :

$$y_{n,r}(t) = \int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} \alpha_{ijk} \prod_{i=1}^n (t - \tau_i)^{q_{n,i,r}} e^{(a_0-a_1)(t-\tau_1) + (a_1-a_2)(t-\tau_{n-1}) + \dots + (a_{n-1}-a_n)(t-\tau_n)} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (1.102)$$

Chacune de ces P_n intégrales multiples $y_{n,r}(t)$ est séparable en un produit de n intégrales simples $y_{n,r,i}(t)$ de la forme suivante :

$$y_{n,r,i}(t) = \begin{cases} \int_0^t \alpha_{n,i} (t - \tau)^{q_{n,n-i,r}} e^{(a_i-a_{i+1})(t-\tau)} u(\tau) d\tau, & \text{si } 0 \leq i \leq n-2 \\ \int_0^t \alpha_{n,i} (t - \tau)^{q_{n,n-i,r}} e^{a_i(t-\tau)} u(\tau) d\tau, & \text{si } i = n-1 \end{cases} \quad (1.103)$$

où les termes $\alpha_{n,i}$ et $q_{n,i,r}$ sont des constantes réelles qui dépendent des termes p_i et a_i de chaque fonction (1.100) de l'intégrale multiple (1.99).

Preuve : voir annexe.

Chacune des intégrales simples $y_{n,r,i}(t)$ est une réponse monodimensionnelle à l'entrée $u(t)$. Dans les conditions de stabilité classiques ($a_i - a_{i+1} \geq 0$ et $a_i \geq 0$), on peut appliquer la transformée de Laplace à l'expression (1.103), et on obtient le résultat suivant :

$$Y_{n,r,i}(s) = \begin{cases} \frac{\alpha_{n,i}}{(s-(a_i-a_{i+1}))^{q_{n,n-i,r}+1}} U(s) & \text{si } 0 \leq i \leq n-2 \text{ et } 0 \leq j \leq p_i - 1 \\ \frac{\alpha_{n,i}}{(s-a_i)^{q_{n,n-i,r}+1}} U(s) & \text{si } i = n-1 \text{ et } 0 \leq j \leq p_i - 1 \end{cases} \quad (1.104)$$

On peut donc utiliser la représentation par blocs pour dessiner le système non linéaire ainsi modélisé.

c) Cas particulier : $n = 3$

Si l'on considère le cas particulier de la modélisation du noyau de Volterra d'ordre 3 sous la forme d'une somme de produits de fonctions de transfert, on doit effectuer la séparation d'une intégrale multiple $y_3(t)$ d'ordre 3 en une somme de produits d'intégrales simples $y_{3,r,i}(t)$.

L'intégrale multiple $y_3(t)$ où les fonctions $f_a^p(t)$ ont pour expression (1.100) peut être réécrite sous la forme :

$$y_3(t) = \int_0^t \int_0^{\tau_2} \int_0^{\tau_3} D_0 D_1 D_2 e^{(a_0-a_1)(t-\tau_3)} e^{(a_1-a_2)(t-\tau_2)} e^{a_2(t-\tau_1)} \prod_{i=1}^3 u(\tau_i) d\tau_i \quad (1.105)$$

où chaque terme D_i est fonction de t , τ_i , a_i et p_i ($i = 1, \dots, 3$).

On peut ensuite séparer cette intégrale multiple en une somme de $P_3 = p_0 \prod_{i=1}^2 \frac{p_i(p_i+1)}{2}$ intégrales multiples $y_{3,r}(t)$ de la forme :

$$y_{3,r}(t) = \int_0^t \alpha_{3,0}(t-\tau_3)^{q_{3,3,r}} e^{(a_0-a_1)(t-\tau_3)} u(\tau_3) d\tau_3 \int_0^t \alpha_{3,1}(t-\tau_2)^{q_{3,2,r}} e^{(a_1-a_2)(t-\tau_2)} u(\tau_2) d\tau_2 \\ \int_0^t \alpha_{3,2}(t-\tau_1)^{q_{3,1,r}} e^{a_2(t-\tau_1)} u(\tau_1) d\tau_1 \quad (1.106)$$

Chacune de ces P_3 intégrales multiples $y_{3,r}(t)$ est séparable en un produit de 3 intégrales simples $y_{3,r,i}(t)$ de la forme suivante :

$$y_{3,r,i}(t) = \begin{cases} \int_0^t \alpha_{3,i}(t-\tau)^{q_{3,3-i,r}} e^{(a_i-a_{i+1})(t-\tau)} u(\tau) d\tau, & \text{si } 0 \leq i \leq 1 \\ \int_0^t \alpha_{3,i}(t-\tau)^{q_{3,3-i,r}} e^{a_i(t-\tau)} u(\tau) d\tau, & \text{si } i = 2 \end{cases} \quad (1.107)$$

où les termes $\alpha_{3,i}$ et $q_{3,i,r}$ sont des constantes réelles qui dépendent des termes p_i et a_i de chaque fonction (1.100) de l'intégrale multiple $y_3(t)$.

Preuve : voir annexe.

Chacune des intégrales simples $y_{3,r,i}(t)$ est une réponse monodimensionnelle à l'entrée $u(t)$. Dans les conditions de stabilité classiques ($a_i - a_{i+1} \geq 0$ et $a_i \geq 0$), on peut appliquer la transformée de Laplace à l'expression (1.107), ce qui amène au résultat suivant :

$$Y_{3,r,i}(s) = \begin{cases} \frac{\alpha_{3,i}}{(s-(a_i-a_{i+1})^{q_{3,3-i,r}}+1)} U(s) & \text{si } 0 \leq i \leq 1 \\ \frac{\alpha_{3,i}}{(s-a_i)^{q_{3,3-i,r}}} U(s) & \text{si } i = 2 \end{cases} \quad (1.108)$$

d) Représentation schématique pour $n = 3$

Dans le cas où $n = 3$, le modèle obtenu pour le noyau de Volterra d'ordre 3 avec un ensemble de $P_3 = 18$ schémas blocs interconnectés est représenté sur la figure 1.19.

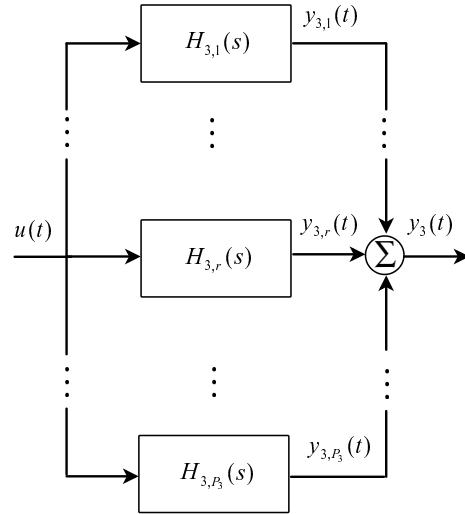


Fig. 1.19 - Modèle du noyau d'ordre 3

La structure de chaque schéma-bloc $H_{3,r}$ est relativement simple : il ne s'agit que de mettre en parallèle 3 blocs représentant chacun une fonction de transfert d'ordre plus ou moins élevé (figure 1.20).

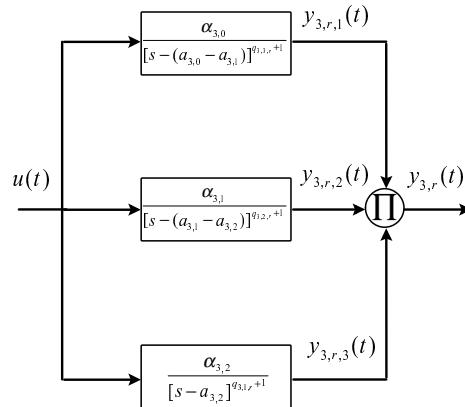


Fig. 1.20 - Structure d'un schéma-bloc

Nous avons adopté ici la notation $a_{3,i}$ au lieu de a_i pour les pôles des fonctions afin de différencier les pôles considérés pour chaque noyau de Volterra modélisé.

Nous verrons que dans le cas général, le noyau d'ordre n peut être modélisé par un

ensemble interconnecté de P_n schémas blocs. Chacun de ces schémas-blocs mettra en parallèle n blocs représentant chacun une fonction de transfert d'ordre plus ou moins élevé.

La complexité du modèle vient du nombre de schémas-blocs nécessaires à la modélisation d'un noyau de Volterra. Dans le cas où $n = 3$, 18 schémas-blocs de structure identique (mais dont la valeur des paramètres est différente) sont nécessaires.

e) Représentation schématique. Cas général

Dans le cas général, on peut représenter le modèle obtenu pour le noyau de Volterra d'ordre n par un ensemble de P_n schémas blocs interconnectés (figure 1.21). La valeur de P_n , et donc le nombre de schémas-blocs nécessaires, dépend bien sûr des valeurs des termes p_i , $i = 0, \dots, n - 1$.

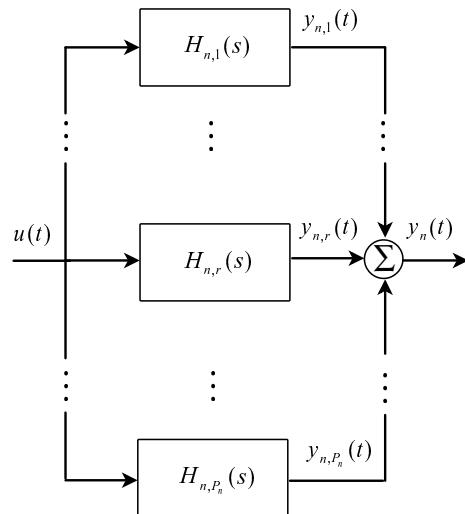


Fig. 1.21 - Modèle du noyau d'ordre n

Bien que sa représentation schématique paraisse lourde, la structure de chaque schéma-bloc $H_{n,r}$ est pourtant simple : il ne s'agit que de mettre en parallèle n blocs représentant chacun une fonction de transfert d'ordre plus ou moins élevé (figure 1.22).

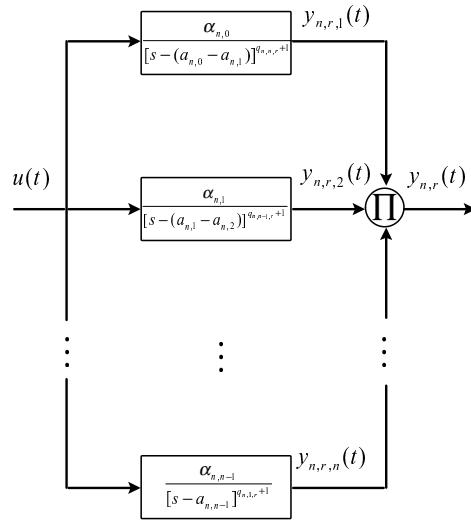


Fig. 1.22 - Structure d'un schéma-bloc

Nous avons également adopté ici la notation $a_{n,i}$ au lieu de a_i pour différencier les pôles des fonctions du modèle du noyau de Volterra d'ordre n .

1.4.4.4 Identification et discussion

a) Introduction

Les noyaux de Volterra sont modélisés par des sommes de produits de réponses au signal d'entrée $u(t)$. La modélisation de la série de Volterra utilisée pour représenter le système non linéaire se ramène donc à un problème d'identification des paramètres du modèle.

L'utilisation d'une série de Volterra pour modéliser un système non linéaire suppose que l'on puisse identifier les termes de la série à partir d'un signal d'entrée $u(t)$ quelconque mais suffisamment riche pour pouvoir sensibiliser l'ensemble des paramètres. Dans le cas du système non linéaire (1.86), l'utilisation d'une séquence pseudo aléatoire bornée en amplitude (figure 1.11(a)) se révèle être un moyen idéal pour effectuer une identification globale des paramètres de tous les termes du modèle de Volterra.

Ce type d'excitation comporte un nombre suffisamment important de variations d'amplitude pour sensibiliser tous les termes de la série de Volterra.

Un avantage du modèle de Volterra proposé est son faible nombre de paramètres. On peut voir sur les figures 1.21 et 1.22 que le modèle du noyau d'ordre n comprend seulement $n + 1$ paramètres : les n termes $a_{n,0}, \dots, a_{n,n-1}$ des fonctions et le nombre p_n de termes de

chaque développement D_n (qui permet de calculer le nombre P_n de "blocs"). Si ce nombre p_n est fixé dès le départ pour que la structure du modèle soit adaptée au système non linéaire considéré, le modèle de Volterra tronqué à ses n premiers termes comprend alors $\frac{n(n+1)}{2}$ paramètres.

Considérons à nouveau l'exemple académique d'un système non linéaire quadratique de représentation d'état suivante :

$$\begin{cases} \dot{x}(t) = -x(t) - x^2(t) + u(t) \\ y(t) = x(t) \end{cases} \quad (1.109)$$

Rappelons que la réponse d'un tel système au signal d'entrée défini par la figure 1.11(a) est donnée par la figure 1.23.

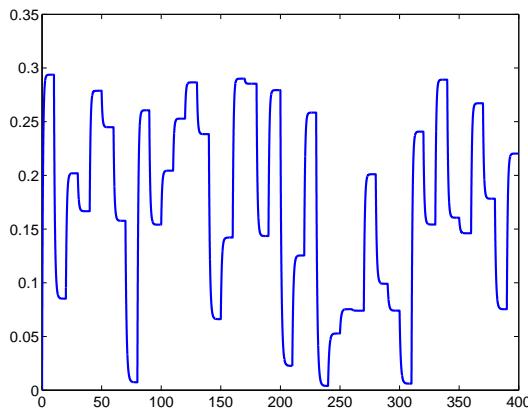


Fig. 1.23 - Réponse du système non linéaire

La série de Volterra est tronquée à ses deux premiers termes. Le premier noyau est modélisé par :

$$h_1(t) = \int_0^t \alpha_{1,0}(t-\tau) e^{a_{1,0}(t-\tau)} u(\tau) d\tau \quad (1.110)$$

et sa réponse représentée sur la figure 1.24

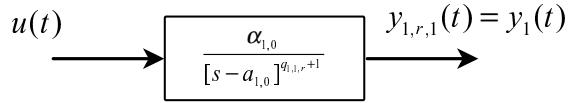


Fig. 1.24 - Modélisation du premier noyau

Le terme p_0 est fixé à 1. Par conséquent, le nombre P_1 de termes de la somme (définie dans l'annexe) est égal à 1. L'indice r ne prend que la valeur 1 et donc $q_{1,1,r} = 1$.

Pour le deuxième noyau, on choisit de fixer $p_0 = 1$ et $p_1 = 2$. Le nombre P_2 de termes de la somme est égal à 3. L'indice r prend donc les valeurs 1 à 3. Le deuxième noyau est ainsi modélisé par une somme de $P_2 = 3$ blocs dont la forme est représentée sur la figure (1.25) avec $r = 1, \dots, P_2 = 3$.

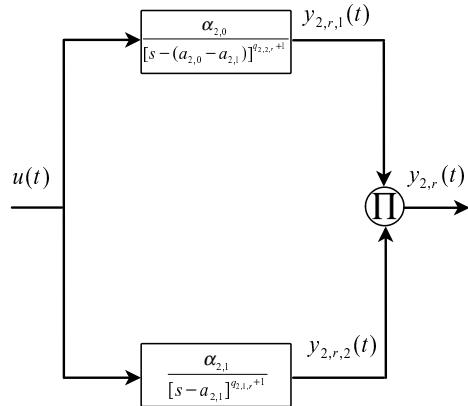


Fig. 1.25 - Modélisation du deuxième noyau

Le modèle de Volterra global est constitué de la somme du modèle du premier noyau et du modèle du deuxième noyau. Les paramètres du modèle global sont ici les pôles $a_{1,0}$ pour le bloc unique du premier noyau, $a_{2,0}$ et $a_{2,1}$ dans chacun des blocs du deuxième noyau. Les gains $\alpha_{i,j}$ ($i = 1, 2$ et $j = 1, \dots, i - 1$) des fonctions ne sont pas considérés comme des paramètres indépendants car les calculs présentés dans les sections précédentes montrent qu'ils dépendent des pôles $a_{i,j}$ ($i = 1, 2$ et $j = 1, \dots, i - 1$) et uniquement de ces pôles. On notera la parcimonie du modèle global qui, malgré sa complexité, est entièrement caractérisé par un ensemble de paramètres très restreint.

térisé par 3 paramètres.

Dans notre cas, le modèle de Volterra global n'est pas linéaire par rapport aux paramètres. Il est donc impossible d'utiliser un algorithme d'identification utilisant la méthode des moindres carrés. La solution adoptée ici est l'utilisation d'un algorithme de programmation non linéaire de type Levenberg-Marquardt [Marquardt, 1963], qui réalise un compromis entre la stabilité de la méthode du gradient et la rapidité de la méthode de Gauss-Newton.

b) Principe de l'identification paramétrique

Un algorithme d'identification non linéaire est basé sur une recherche itérative de l'optimum paramétrique. Si l'on appelle θ le vecteur de paramètres à identifier, l'estimée $\hat{\theta}$ de ce vecteur sera calculée itérativement. A l'itération $i + 1$, l'estimée, notée $\hat{\theta}_{i+1}$, est donnée par :

$$\hat{\theta}_{i+1} = \hat{\theta}_i + \alpha f_i \quad (1.111)$$

où f_i est une direction de recherche calculée à partir de l'information sur le critère quadratique $J(\theta)$ obtenue à l'itération i :

$$J(\theta) = (y(\theta) - \hat{y}(\hat{\theta}_i))^T (y(\theta) - \hat{y}(\hat{\theta}_i)) \quad (1.112)$$

Le terme α est une constante positive destinée à assurer la convergence de l'algorithme. Cet incrément αf_i est différent suivant les méthodes utilisées. La méthode du *gradient* est basée sur le développement de la dérivée première du critère $J(\theta)$:

$$\hat{\theta}_{i+1} = \hat{\theta}_i - \lambda \frac{\partial J(\theta)}{\partial \theta} \Big|_{\theta=\hat{\theta}_i} \quad (1.113)$$

Le terme $\frac{\partial J(\theta)}{\partial \theta}$ est appelé gradient du critère. La constante λ assure la stabilité de l'algorithme.

Partant du même principe que la méthode du gradient, la méthode de *Newton* développe le critère jusqu'au deuxième ordre et fait intervenir le gradient $\frac{\partial J(\theta)}{\partial \theta}$ et le hessien $\frac{\partial^2 J(\theta)}{\partial \theta \partial \theta^T}$. L'algorithme de Newton s'écrit :

$$\hat{\theta}_{i+1} = \hat{\theta}_i - \left[\frac{\partial^2 J(\theta)}{\partial \theta \partial \theta^T} \Big|_{\theta=\hat{\theta}_i} \right]^{-1} \left[\frac{\partial J(\theta)}{\partial \theta} \Big|_{\theta=\hat{\theta}_i} \right] \quad (1.114)$$

La méthode du gradient permet une convergence lente mais stable vers l'optimum paramétrique tandis que la méthode de Newton converge rapidement au détriment de la stabilité. La méthode dite de *Levenberg-Marquardt* permet de profiter des avantages des deux algorithmes précédents par un compromis entre rapidité de convergence et stabilité. L'algorithme de Levenberg-Marquardt est défini de la manière suivante :

$$\widehat{\theta}_{i+1} = \widehat{\theta}_i - \left[\lambda_i I + \frac{\partial^2 J(\theta)}{\partial \theta \partial \theta^T} \Big|_{\theta=\widehat{\theta}_i} \right]^{-1} \left[\frac{\partial J(\theta)}{\partial \theta} \Big|_{\theta=\widehat{\theta}_i} \right] \quad (1.115)$$

Le paramètre λ_i est un paramètre d'ajustement. étant donné que l'on cherche à minimiser le critère $J(\theta)$, deux cas peuvent se présenter. Si $J(\widehat{\theta}_{i+1}) < J(\widehat{\theta}_i)$ alors l'algorithme converge et on doit diminuer le terme λ_{i+1} par rapport au terme λ_i afin de rendre le hessien prépondérant et de tendre vers l'algorithme de Newton, qui convergera plus rapidement. Mais si $J(\widehat{\theta}_i) < J(\widehat{\theta}_{i+1})$, l'algorithme ne converge pas et l'on doit augmenter le terme λ_{i+1} par rapport au terme λ_i afin de tendre vers l'algorithme du gradient et d'assurer ainsi la stabilité. On augmente la valeur du terme λ_{i+1} jusqu'au retour de la convergence avant de passer à l'itération suivante.

Ces algorithmes nécessitent le calcul du gradient, effectué de la manière suivante :

$$\frac{\partial J(\theta)}{\partial \theta} = -2\sigma^T(\theta)\varepsilon(\theta) \quad (1.116)$$

où $\varepsilon(\theta) = y(\theta) - \widehat{y}(\widehat{\theta})$ est l'erreur et $\sigma(\theta) = \frac{\partial \widehat{y}(\theta)}{\partial \theta}$ est le vecteur des fonctions de sensibilité paramétrique.

Le hessien est très souvent approché par l'expression suivante :

$$\frac{\partial^2 J(\theta)}{\partial \theta \partial \theta^T} \simeq 2\sigma(\theta)\sigma^T(\theta) \quad (1.117)$$

L'algorithme de Levenberg-Marquardt doit être initialisé en donnant une valeur de départ θ_0 aux paramètres du vecteur θ . L'inconvénient majeur de cette méthode est qu'elle ne garantit pas la convergence globale des paramètres vers l'optimum. En effet, le problème peut admettre un certain nombre d'optima locaux et, selon la valeur initiale des paramètres, l'algorithme pourra converger vers un minimum local. Certaines méthodes ont été développées pour assurer la convergence de l'algorithme, comme l'utilisation d'algorithmes génétiques afin de calculer des valeurs paramétriques qui assurent une valeur du critère proche de l'optimum global. On peut alors utiliser ces valeurs paramétriques obtenues pour initialiser l'algorithme de Levenberg-Marquardt et être certain qu'il va converger vers l'optimum global.

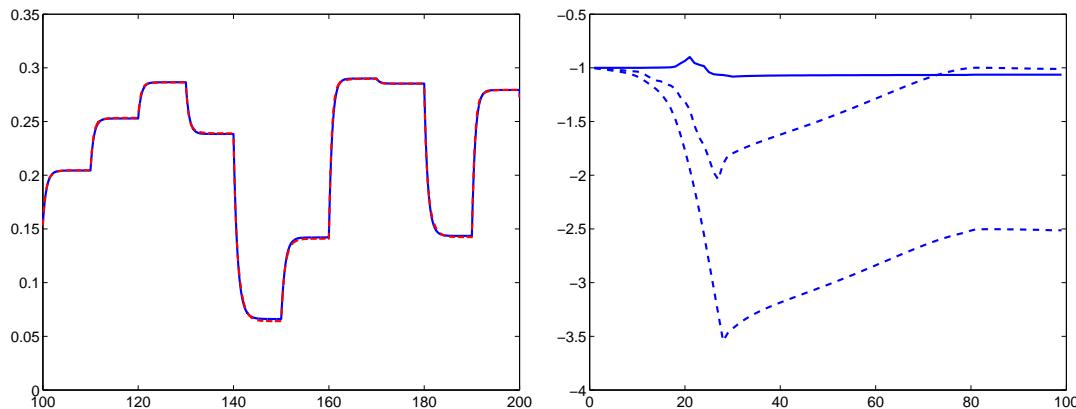
c) Discussion

Reprendons l'exemple du système non linéaire (1.109). Le modèle de Volterra est tronqué à ses deux premiers noyaux et on initialise les paramètres à la valeur -1 . Les fichiers de mesure du signal d'entrée $u(t)$ et de sortie $y(t)$ comportent 40000 points. La période d'échantillonnage est égale à 0.01 s. Après 100 itérations, les valeurs des 3 paramètres obtenues sont :

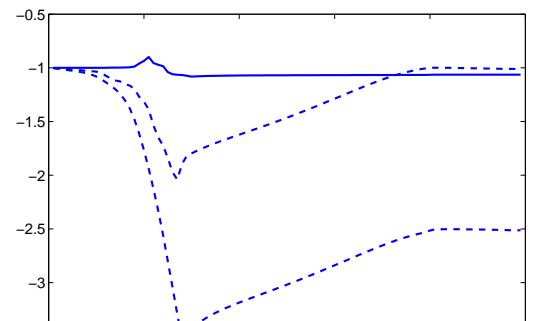
$$\begin{aligned} a_{1,0} &= -1.0635 \\ a_{2,0} &= -1.0097 \\ a_{2,1} &= -2.5121 \end{aligned} \quad (1.118)$$

et le critère quadratique est $J = 0.1104$.

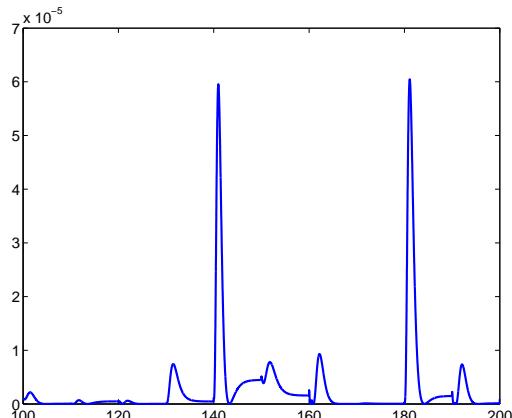
La figure 1.26(a) compare la sortie du système non linéaire (1.109) et la sortie du modèle de Volterra pour ces 3 valeurs paramétriques.



(a) Sortie du système non linéaire (ligne continue) et du modèle de Volterra (ligne pointillée)



(b) Evolution du paramètre $a_{1,0}$ (courbe continue), $a_{2,0}$ (courbe en pointillés longs) et $a_{2,1}$ (courbe en pointillés fins)



(c) Erreur quadratique

Fig. 1.26 - Résultats de modélisation et d'identification paramétrique

On constate une bonne adéquation entre les deux courbes de la figure 1.26(a).

La figure 1.26(b) représente l'évolution des paramètres en fonction du nombre d'itérations.

On s'aperçoit que la valeur du premier paramètre $a_{1,0}$ converge bien mais qu'on ne peut rien affirmer pour les valeurs des 2 autres paramètres. Par conséquent, malgré le

fait que les deux courbes de la figure 1.26(a) soient très proches l'une de l'autre, rien ne garantit que l'algorithme de Levenberg-Marquardt nous ait fourni les valeurs optimales des paramètres. En effet, un changement d'initialisation des paramètres conduit à d'autres valeurs du critère J . Si l'on initialise les paramètres à la valeur -2 , on obtient $J = 0.4446$; si l'on initialise les paramètres à la valeur -10 , on obtient $J = 1.0521$. Ces initialisations conduisent donc à des minima locaux et les modèles ainsi obtenus fournissent donc une réponse au signal d'entrée $u(t)$ plus ou moins satisfaisante. On peut donc se rapprocher de l'optimum global suivant l'initialisation paramétrique mais seule une démarche empirique permet d'obtenir (ou pas) une initialisation paramétrique correcte.

Le plus gros avantage du modèle de Volterra proposé est son faible nombre de paramètres. Cette parcimonie entraîne malgré tout une complexité importante du modèle de Volterra, ce qui peut rendre délicate la procédure d'identification des paramètres.

D'autres contraintes doivent être considérées : d'une part, les termes a_i et $a_i - a_{i+1}$, $i = 1, \dots, n$ dans les fonctions qui modélisent le noyau d'ordre n (figure 1.22) doivent respecter les conditions de stabilité :

$$\begin{cases} a_{n,i} - a_{n,i+1} < 0, & i = 0, \dots, n-2 \\ a_{n,i} < 0, & i = n-1 \end{cases} \quad (1.119)$$

D'autre part, les valeurs des gains $\alpha_{n,i}$ de chaque fonction dépendent des valeurs des pôles $a_{n,i}$ et sont donc "fixées" par les valeurs des pôles obtenues lors de la procédure d'identification. Dans la plupart des cas (comme dans le cas que nous venons de voir), les contraintes sur les valeurs des pôles se répercutent sur les valeurs des gains et conduisent à des minima locaux du critère. Cette convergence vers des minima locaux est également due à l'initialisation paramétrique. Ainsi, malgré une parcimonie paramétrique intéressante, on n'obtient le plus souvent que des solutions sous-optimales du problème non linéaire.

1.4.5 Modèle simplifié des noyaux de Volterra sur fonctions de transfert

1.4.5.1 Introduction

Bien que parcimonieux, le modèle de Volterra inspiré des séries de Fliess proposé précédemment est assujetti à un certain nombre de contraintes qui peuvent conduire à des solutions inacceptables du problème non linéaire. Si l'on considère des applications pratiques, la complexité du modèle et les contraintes imposées sur ses paramètres peuvent se révéler autant d'inconvénients qui prennent le pas sur sa parcimonie. Cependant, toute l'étude effectuée montre bien que les séries de Volterra peuvent être développées sur des fonctions de transfert.

Nous allons donc utiliser ce résultat pour concevoir un autre modèle plus flexible [Bibes *et al.*, 2003a] : il sera soumis à moins de contraintes et disposera de davantage de degrés de liberté dans ses paramètres. Le but est également de conserver une certaine

parcimonie du modèle de Volterra précédent afin de satisfaire au mieux l'objectif de départ.

1.4.5.2 Résultat préliminaire

Considérons une famille de fonctions $\psi_j(t)$, $j = 1, \dots, k$ unidimensionnelles linéaires. La réponse $y_{\psi_j}(t)$ de la fonction $\psi_j(t)$ au signal d'entrée $u(t)$ est donnée par l'équation suivante :

$$y_{\psi_j}(t) = \int_0^\infty \psi_j(\tau) u(t - \tau) d\tau \quad (1.120)$$

Soit $y_i(t)$ le produit de toutes les réponses $y_{\psi_j}(t)$:

$$y_i(t) = \prod_{j=1}^i y_{\psi_j}(t) \quad (1.121)$$

On développe cette expression :

$$y_i(t) = \prod_{j=1}^i \int_0^\infty \psi_j(\tau_j) u(t - \tau_j) d\tau_j \quad (1.122)$$

Si les variables τ_j sont indépendantes, on peut transformer ce produit d'intégrales simples en une intégrale multiple :

$$y_i(t) = \int_0^\infty \dots \int_0^\infty \prod_{j=1}^i \psi_j(\tau_j) u(t - \tau_j) d\tau_j \quad (1.123)$$

Si l'on effectue alors l'analogie avec l'expression (1.15), on peut poser :

$$h_i(\tau_1, \dots, \tau_i) = \prod_{j=1}^i \psi_j(\tau_j) \quad (1.124)$$

Cela revient à dire que sous certaines conditions, une fonction multidimensionnelle $h_i(\tau_1, \dots, \tau_i)$ peut être représentée par un produit de fonctions unidimensionnelles $\psi_j(\tau_j)$.

L'expression (1.123) peut alors être réécrite sous la forme :

$$y_i(t) = \int_0^\infty \dots \int_0^\infty h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i u(t - \tau_j) d\tau_j \quad (1.125)$$

Nous allons utiliser ce résultat afin de simplifier le modèle de Volterra développé sur fonctions de transfert.

1.4.5.3 Modèle de Volterra simplifié sur fonctions de transfert

Nous venons de voir que le noyau de Volterra $h_i(\tau_1, \dots, \tau_i)$ d'ordre i peut être représenté par un produit de i fonctions unidimensionnelles $\psi_j(\tau_j)$. L'idée générale est donc de représenter chaque noyau de Volterra du modèle par un tel produit. A ce point et sans autre renseignement, toute la difficulté réside dans le choix de la forme des fonctions unidimensionnelles $\psi_j(\tau_j)$.

Cependant, nous disposons d'un modèle de départ, défini dans la section précédente. Nous avons vu que le noyau de Volterra d'ordre n peut être modélisé par une somme de P_n produits de n intégrales simples de la forme :

$$y_{n,r,i}(t) = \begin{cases} \int_0^t \alpha_{n,i}(t-\tau)^{q_{n,n-i,r}} e^{(a_i-a_{i+1})(t-\tau)} u(\tau) d\tau, & \text{si } 0 \leq i \leq n-2 \\ \int_0^t \alpha_{n,i}(t-\tau)^{q_{n,n-i,r}} e^{a_i(t-\tau)} u(\tau) d\tau, & \text{si } i = n-1 \end{cases} \quad (1.126)$$

Dans cette forme de modèle, la constante $\alpha_{n,i}$ est dépendante de la valeur de la constante a_i et la constante $q_{n,n-i,r}$ dépend des ordres p_i de développement des fonctions :

$$\begin{cases} 0 \leq q_{n,1,r} \leq p_{n-1} - 1 \\ 0 \leq q_{n,i,r} \leq (p_{n-i} - 1)(p_{n-i+1} - 1) \end{cases} \text{ pour } i = 2, \dots, n \quad (1.127)$$

Nous avons vu que ce modèle est soumis à des contraintes : les différences $a_i - a_{i+1}$ doivent obéir à la condition de stabilité et les valeurs des gains $\alpha_{n,i}$ sont fixées par les valeurs des pôles a_i .

Nous proposons donc de modifier le modèle (1.126) précédemment présenté de telle manière qu'il soit à la fois plus "libre" (moins soumis à des contraintes paramétriques) et plus facile à manipuler (en particulier à identifier), tout en gardant une relative parcimonie paramétrique.

La nouvelle forme de modélisation du noyau de Volterra d'ordre n que nous présentons est le produit de n intégrales simples de la forme :

$$\tilde{y}_{i,n}(t) = \int_0^t \sum_{k=0}^{p_{i,n}-1} \gamma_{k,i,n}(t-\tau)^k e^{a_{n,i}(t-\tau)} u(\tau) d\tau, \text{ pour } 1 \leq i \leq n \quad (1.128)$$

où les termes $\gamma_{k,i,n}$ sont des constantes (indépendantes des termes $a_{n,i}$). Le terme $p_{i,n}$ est également une constante.

Le noyau de Volterra d'ordre n est donc modélisé par l'expression :

$$\begin{aligned} y_n(t) &= \prod_{i=1}^n \tilde{y}_{i,n}(t) \\ &= \prod_{i=1}^n \int_0^t \sum_{k=0}^{p_{i,n}-1} \gamma_{k,i,n}(t-\tau)^k e^{a_{n,i}(t-\tau)} u(\tau) d\tau \end{aligned} \quad (1.129)$$

Si l'on compare les expressions (1.126) et (1.128), on remarque bien évidemment certaines similitudes mais également des différences, introduites dans un but bien précis.

Tout d'abord, il ne faut pas perdre de vue que l'on cherche à améliorer la flexibilité du modèle (1.126). Les termes $\alpha_{n,i}$ de chaque dénominateur des fonctions de transfert utilisées dans ce modèle dépendent des termes a_i . Le nombre total de paramètres est donc restreint car les termes $\alpha_{n,i}$, qui ne sont pas considérés comme des paramètres du modèle, ne sont pas identifiés indépendamment des paramètres a_i . Un moyen d'améliorer la flexibilité du modèle global est de transformer ces termes $\alpha_{n,i}$ en véritables paramètres $\gamma_{k,i,n}$ qui seront identifiés indépendamment des paramètres a_i . Ces paramètres se rajoutent aux paramètres $a_{n,i}$. On peut croire que cela tend à diminuer la parcimonie du modèle et augmenter la difficulté d'identification paramétrique : ces deux problèmes seront discutés par la suite.

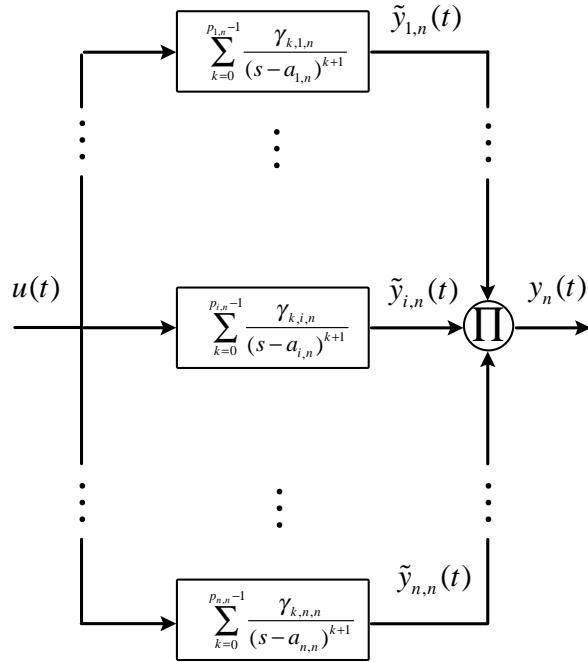
La seconde modification est la disparition des différences de pôles $a_i - a_{i+1}$ au profit d'un seul pôle $a_{n,i}$ pour chaque fonction considérée. Cette modification permet de libérer le nouveau modèle de Volterra d'une des contraintes principales du précédent : la condition de stabilité $a_{n,i} < 0$ ne s'applique plus qu'aux seuls pôles des fonctions de transfert.

Enfin, l'introduction d'une somme dont l'indice k porte sur l'exposant du terme $t - \tau$ ne modifie que formellement l'intégrale simple (1.126) : en effet, la valeur du terme k dépend de la valeur du terme $p_{i,n}$, tout comme la valeur des termes $q_{n,i,r}$ ne dépendait que de la valeur des termes p_{n-i} dans le modèle précédent. Les valeurs des termes p_{n-i} (pour le premier modèle) et $p_{i,n}$ (pour le deuxième modèle) sont données par l'utilisateur afin de fixer la structure du modèle de Volterra dès le départ : ces termes indiquent l'ordre des fonctions utilisées pour modéliser un noyau de Volterra.

Sous les conditions habituelles de stabilité, on peut appliquer la transformée de Laplace à l'expression (1.128) et on obtient :

$$\tilde{Y}_{i,n}(s) = \sum_{k=0}^{p_{i,n}-1} \frac{\gamma_{k,i,n}}{(s - a_{n,i})^{k+1}} U(s) \quad (1.130)$$

Le nouveau modèle de Volterra proposé pour le noyau d'ordre n est représenté schématiquement sur la figure 1.27.


 Fig. 1.27 - Représentation du noyau d'ordre n

On remarque les similitudes entre la modélisation du noyau de Volterra d'ordre n proposée précédemment (figure 1.22) et cette forme de modélisation. Toutefois, il faut noter que le modèle précédent du noyau de Volterra d'ordre n se compose d'une somme de P_n schémas-blocs (figure 1.22) alors que le nouveau modèle du noyau de Volterra d'ordre n présenté ici ne comporte que le seul schéma-bloc 1.27. La complexité du nouveau modèle est ainsi fortement diminuée car chaque noyau de Volterra est modélisé par un seul produit de réponses de fonctions à l'entrée $u(t)$ au lieu d'une somme de produits de réponses à l'entrée $u(t)$.

On introduit maintenant les notations simplificatrices suivantes :

$$d_{i,n}(t - \tau) = \sum_{k=0}^{p_{i,n}-1} \gamma_{k,i,n}(t - \tau)^k e^{a_{n,i}(t-\tau)} \quad (1.131)$$

Par calcul de la transformée de Laplace de cette expression, on obtient :

$$D_{in}(a_{n,i}, p_{i,n}) = \sum_{k=0}^{p_{i,n}-1} \frac{\gamma_{k,i,n}}{(s - a_{n,i})^{k+1}} \quad (1.132)$$

Le modèle de Volterra global tronqué à l'ordre n représentant le système non linéaire considéré est le suivant :

$$y(t) = \sum_{j=1}^n y_j(t) = \sum_{j=1}^n \prod_{i=1}^j \tilde{y}_{i,j}(t) = \sum_{j=1}^n \prod_{i=1}^j \int_0^t d_{i,j}(t-\tau) u(\tau) d\tau \quad (1.133)$$

On utilise les notations introduites pour schématiser le nouveau modèle de Volterra tronqué à l'ordre n pour représenter un système non linéaire :

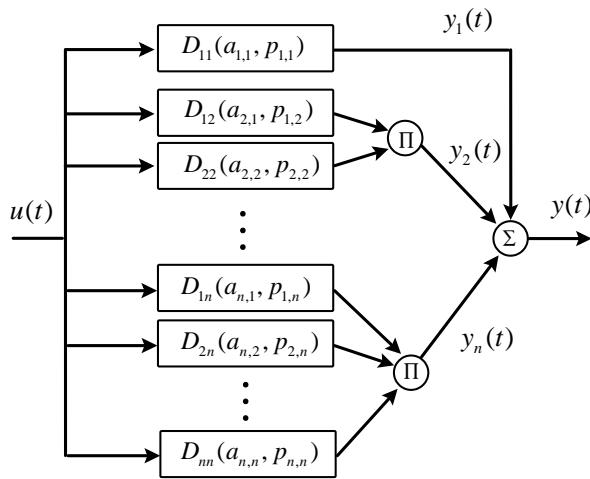


Fig. 1.28 - Modèle global développé sur des fonctions de transfert

1.4.5.4 Identification et discussion

Reprendons le cas du système non linéaire de représentation d'état :

$$\begin{cases} \dot{x}(t) = -x(t) - x^2(t) + u(t) \\ y(t) = x(t) \end{cases} \quad (1.134)$$

Afin de comparer les deux structures de modèles de Volterra, nous allons reprendre les mêmes signaux d'entrée $u(t)$ et de sortie $y(t)$ que ceux définis par les figures 1.11(a) et 1.11(b).

La structure du modèle de Volterra est fixée de la même manière que précédemment, en tronquant la série à ses deux premiers termes. Le modèle du premier noyau est développé sur un seul terme (donc $p_{1,1} = 1$) :

$$D_{11}(a_{1,1}, p_{1,1}) = \frac{\gamma_{0,1,1}}{s - a_{1,1}} \quad (1.135)$$

Le deuxième noyau est modélisé par le produit de deux développements D_{ji} , chacun tronqué à ses deux premiers termes ($p_{1,2} = 2$ et $p_{2,2} = 2$) :

$$\begin{aligned} D_{12}(a_{2,1}, p_{1,2}) &= \frac{\gamma_{0,2,1}}{s-a_{2,1}} + \frac{\gamma_{1,2,1}}{(s-a_{2,1})^2} \\ D_{22}(a_{2,2}, p_{2,2}) &= \frac{\gamma_{0,2,2}}{s-a_{2,2}} + \frac{\gamma_{1,2,2}}{(s-a_{2,2})^2} \end{aligned} \quad (1.136)$$

Les paramètres de cette forme de modèle de Volterra sont donc au nombre de 8 : le gain $\gamma_{0,1,1}$ et le pôle $a_{1,1}$ du modèle du premier noyau et les gains $\gamma_{0,2,1}$, $\gamma_{1,2,1}$, $\gamma_{0,2,2}$ et $\gamma_{1,2,2}$ ainsi que les pôles $a_{2,1}$ et $a_{2,2}$ du modèle du deuxième noyau.

Malgré ce nombre de paramètres plus important que celui du modèle précédent, l'objectif de parcimonie est relativement respecté. Nous verrons plus tard que l'on peut encore diminuer ce nombre de paramètres en modifiant légèrement le modèle de Volterra présenté.

Le modèle de Volterra global n'est toujours pas linéaire par rapport aux paramètres. La solution à nouveau adoptée ici est l'utilisation d'un algorithme d'identification non linéaire de type Levenberg-Marquardt.

Les paramètres du modèle sont initialisés de la manière suivante :

$$\begin{aligned} a_{1,1} &= -1; a_{2,1} = -1; a_{2,2} = -1 \\ \gamma_{0,1,1} &= 1; \gamma_{0,2,1} = 2; \gamma_{1,2,1} = 3; \gamma_{0,2,2} = 4; \gamma_{1,2,2} = 5 \end{aligned} \quad (1.137)$$

D'autres initialisations sont possibles mais on a vu que toutes ne mènent pas forcément à l'optimum paramétrique. Nous allons voir que cette initialisation conduit à un résultat tout à fait convenable.

Les fichiers de mesure du signal d'entrée $u(t)$ et de sortie $y(t)$ comportent 40000 points. La période d'échantillonnage est égale à 0.01 s. Après 100 itérations, les valeurs des 8 paramètres sont :

$$\begin{aligned} a_{1,1} &= -1.0841 \\ a_{2,1} &= -1.3270 \\ a_{2,2} &= -1.2682 \\ \gamma_{0,1,1} &= 1.0276 \\ \gamma_{0,2,1} &= 0.0020 \\ \gamma_{1,2,1} &= -0.2635 \\ \gamma_{0,2,2} &= 0.2499 \\ \gamma_{1,2,2} &= 4.7715 \end{aligned} \quad (1.138)$$

et le critère quadratique est $J = 0.0432$. En comparaison avec la valeur obtenue pour le modèle de Volterra précédent ($J = 0.1104$), celle-ci est du même ordre de grandeur.

La figure 1.29(a) compare la sortie du système non linéaire (1.134) et la sortie du modèle de Volterra pour ces 8 valeurs paramétriques. Les deux courbes sont quasiment superposées.

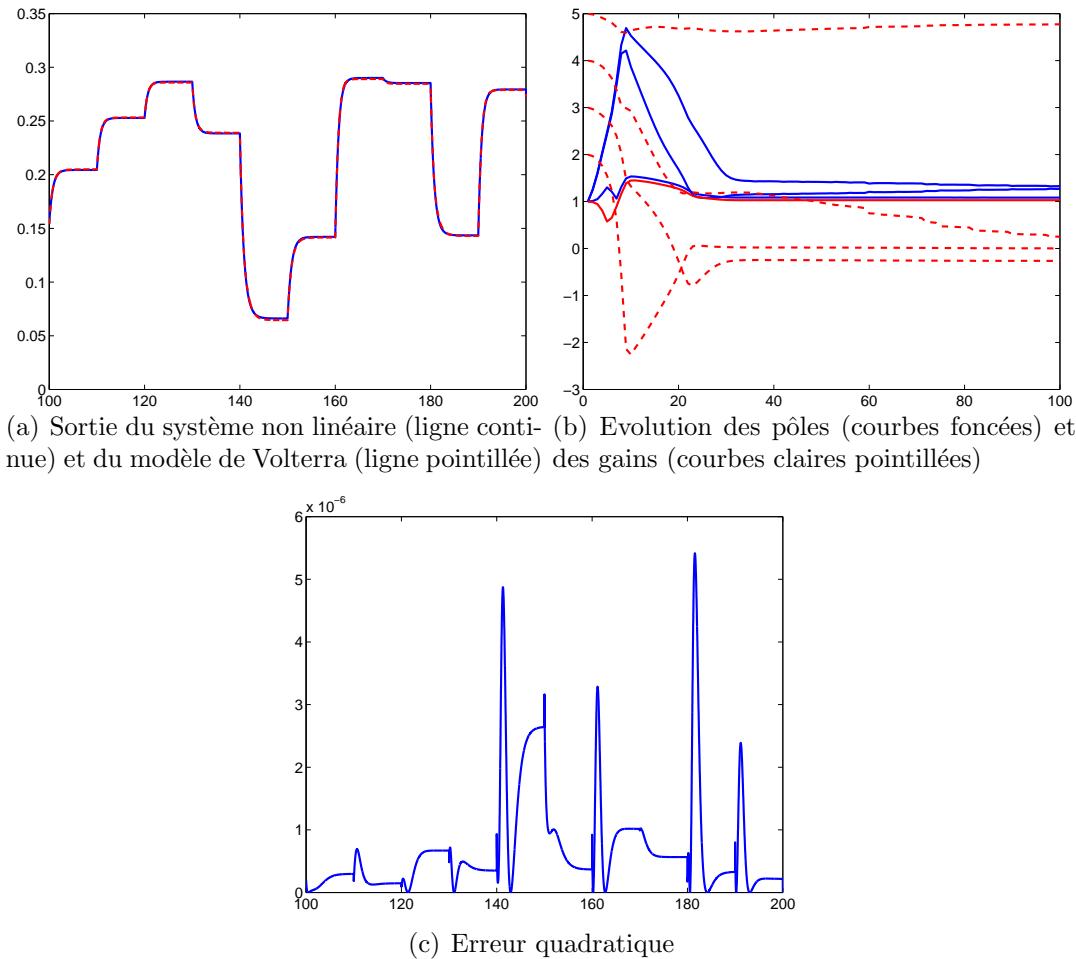


Fig. 1.29 - Résultats de modélisation et d'identification paramétrique

La figure 1.29(b) représente l'évolution des paramètres en fonction du nombre d'itérations.

Tous les paramètres ne convergent pas vers des valeurs stables. L'obtention de l'optimum paramétrique n'est absolument pas garantie mais néanmoins, l'estimation donne un résultat tout à fait satisfaisant.

1.4.5.5 Conclusion

Le modèle de Volterra développé sur fonctions de transfert est plus facilement identifiable et d'une précision tout à fait comparable à celle du modèle de Volterra inspiré directement des séries de Fliess, malgré un nombre de paramètres plus important. La relative souplesse de ce nouveau modèle compense donc la perte de parcimonie par rapport au modèle précédent. Nous allons maintenant essayer de diminuer à nouveau ce nombre de paramètres en introduisant un modèle de Volterra développé autour d'une valeur moyenne.

1.5 Développement autour d'une composante continue

1.5.1 Introduction

L'approche classique de représentation des systèmes non linéaires par les séries de Volterra consiste à développer le modèle autour d'un point de fonctionnement particulier relatif à chaque système non linéaire considéré. Dans le cas où les conditions initiales de fonctionnement du système sont nulles, on se contente habituellement de considérer que la composante continue est nulle et que les variations du signal d'entrée $u(t)$ et de sortie $y(t)$ s'effectuent autour de cette valeur moyenne.

Nous avons vu que le premier noyau de Volterra modélise la partie linéaire du système considéré. Dès que l'amplitude des signaux d'entrée et de sortie atteint une certaine valeur, le système commence à avoir un comportement non linéaire que le premier noyau de Volterra seul ne peut modéliser. Il faut donc ajouter un second noyau de Volterra au modèle pour tenir compte du comportement non linéaire du système. L'augmentation progressive de l'amplitude des signaux d'entrée et de sortie nécessite donc l'ajout progressif de noyaux dans le modèle de Volterra pour tenir compte des non-linéarités introduites. Et cela jusqu'au moment où les non-linéarités du système sont trop importantes pour être convenablement modélisées par un nombre fini de noyaux de Volterra. On a alors atteint les limites de cette forme de modélisation.

Nous venons de mettre l'accent sur un inconvénient du modèle classique de Volterra, qui peut avoir des conséquences importantes sur le nombre de paramètres du modèle global : si l'amplitude des signaux autour d'une valeur moyenne nulle devient importante, le modèle de Volterra classique devra comporter un nombre important de noyaux afin de représenter correctement les non-linéarités du système. Nous avons étudié plusieurs formes de modélisation des noyaux de Volterra qui comportent chacune un certain nombre de paramètres. Il va de soi que plus le nombre de noyaux de Volterra nécessaires est important, plus le nombre de paramètres pour les modéliser va devenir important. On peut ainsi aboutir à un modèle surparamétrisé qui pourra causer certaines difficultés dans l'identification globale du vecteur des paramètres.

Nous proposons ici un moyen de pallier à ce problème : plutôt que de développer le modèle de Volterra autour d'une valeur moyenne nulle, nous allons développer ce modèle autour de la valeur moyenne u_0 du signal d'entrée $u(t)$ et ainsi séparer la partie statique de la partie dynamique du modèle de Volterra. Seule la partie dynamique autour de la composante continue sera identifiée. Ainsi, la structure du modèle de Volterra nécessaire à une modélisation correcte du système non linéaire sera moins complexe et le nombre de paramètres nécessaires pourra être diminué [Bibes *et al.*, 2003b].

Remarquons que si l'on se contente d'ajouter la composante continue du signal de sortie $y(t)$ du système à la partie dynamique en sortie de notre modèle de Volterra, on ne

reconstruit pas correctement le signal de sortie $y(t)$ du système non linéaire. Cela parce qu'on ne prend pas en compte les termes "de couplage" (à savoir les termes comprenant à la fois la partie statique et la partie dynamique du signal $u(t)$) pour reconstruire $y(t)$. Il ne suffit donc pas d'identifier la partie dynamique puis de lui rajouter la partie statique pour reconstruire correctement un signal en sortie de modèle. D'où l'intérêt de développer notre modèle de Volterra autour d'une composante continue non nulle du signal d'entrée $u(t)$.

1.5.2 Expression du modèle développé autour d'une composante continue

On considère à nouveau le système non linéaire

$$\begin{cases} \dot{x}(t) = f(x(t)) + u(t)g(x(t)) \\ y(t) = h(x(t)) \end{cases} \quad (1.139)$$

où $x(t) \in \mathbb{R}^n$ est le vecteur d'état, $u(t) \in \mathbb{R}$ le signal d'entrée et $y(t) \in \mathbb{R}$ le signal de sortie. Les fonctions f , g et h sont des fonctions non linéaires de $x(t)$.

On décompose le signal d'entrée $u(t)$ de la manière suivante :

$$u(t) = u_0 + \Delta u(t) \quad (1.140)$$

où u_0 est un terme constant qui représente la partie statique du signal d'entrée et $\Delta u(t)$ est un terme variant dans le temps qui représente la partie dynamique du signal d'entrée $u(t)$.

La série de Volterra tronquée à l'ordre n représentant le système non linéaire (1.139) a été définie de la manière suivante :

$$\hat{y}(t) = \sum_{i=1}^n \underbrace{\int_0^t \dots \int_0^t}_{i} h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i [u(t - \tau_j)] d\tau_j \quad (1.141)$$

Si l'on applique la décomposition de l'entrée $u(t)$ à ce modèle, on obtient :

$$\hat{y}(t) = \sum_{i=1}^n \underbrace{\int_0^t \dots \int_0^t}_{i} h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i [u_0 + \Delta u(t - \tau_j)] d\tau_j \quad (1.142)$$

1.5.3 Bases de fonctions orthonormées

Proposition 1.3 : *Le développement en série de Volterra de la sortie du système (1.139) autour d'une composante continue u_0 s'écrit, dans le cas où les noyaux de Volterra sont eux-mêmes développés sur une base de fonctions orthonormées généralisée, sous la forme :*

$$\hat{y}(t) = \sum_{i=1}^n \left(\sum_{\nu_1=1}^{M_i} \dots \sum_{\nu_i=1}^{\nu_{i-1}} \xi_{\nu_1 \dots \nu_i} \right. \\ \left. \left[u_0^i \prod_{j=1}^i \phi_{\nu_j}(0) + \sum_{j=1}^i u_0^{i-j} \mathcal{J}_{i,j}(\Delta u(t)) \right] \right) \quad (1.143)$$

où

$$\mathcal{J}_{i,j}(\Delta u(t)) = \sum_{k=1}^{C_i^{i-j}} \left(\prod_{k=1}^{i-j} \phi_{\nu_k}(0) \right) \left(\prod_{l=1}^j \mathcal{I}_{\nu_l}(\Delta u(t)) \right) \quad (1.144)$$

avec $\forall k \in [1, i-j], \forall l \in [1, j]$:

$$\begin{cases} \{\nu_k\} \cup \{\nu_l\} = \{1, \dots, i+j\} \\ \{\nu_k\} \cap \{\nu_l\} = \emptyset \end{cases} \quad (1.145)$$

et où l'expression $\mathcal{I}_{\nu_l}(u(t))$ est la réponse de la fonction orthonormée $\phi_{\nu_l}(t)$ à l'entrée $u(t)$:

$$\mathcal{I}_{\nu_l}(u(t)) = \int_0^t \phi_{\nu_l}(\tau) u(t-\tau) d\tau \quad (1.146)$$

Preuve : le développement du noyau de Volterra $h_i(\tau_1, \dots, \tau_i)$ sur une base de fonctions orthonormées $\{\phi_\nu\}$ s'écrit :

$$\hat{h}_i(\tau_1, \dots, \tau_i) = \sum_{\nu_1=0}^{M_i} \dots \sum_{\nu_i=0}^{\nu_{i-1}} \xi_{\nu_1 \dots \nu_i} \prod_{j=1}^i \phi_{\nu_j}(\tau_j) \quad (1.147)$$

Par conséquent, l'expression (1.142) peut être développée de la manière suivante :

$$\hat{y}(t) = \sum_{i=1}^n \left(\sum_{\nu_1=1}^{M_i} \dots \sum_{\nu_i=1}^{\nu_{i-1}} \xi_{\nu_1 \dots \nu_i} \right. \\ \left. \left[\prod_{j=1}^i \int_0^t \phi_{\nu_j}(\tau_j) [u_0 + \Delta u(t-\tau_j)] d\tau_j \right] \right) \quad (1.148)$$

On développe le calcul de l'expression entre crochets :

$$\prod_{j=1}^i \int_0^t \phi_{\nu_j}(\tau_j) [u_0 + \Delta u(t-\tau_j)] d\tau_j \\ = \prod_{j=1}^i \left(u_0 \int_0^t \phi_{\nu_j}(\tau_j) d\tau_j + \int_0^t \phi_{\nu_j}(\tau_j) \Delta u(t-\tau_j) d\tau_j \right) \quad (1.149)$$

Le premier terme $\int_0^t \phi_{\nu_j}(\tau_j) d\tau_j$ correspond à la réponse d'une fonction orthonormée ϕ_{ν_j} à un signal d'entrée $u(t) = 1, t \geq 0$. On peut donc considérer ce terme comme un gain et l'écrire $\phi_{\nu_j}(0)$.

Le deuxième terme $\int_0^t \phi_{\nu_j}(\tau_j) \Delta u(t-\tau_j) d\tau_j$ correspond à la réponse d'une fonction orthonormée ϕ_{ν_j} à la partie dynamique de l'entrée. On peut donc noter ce terme $\mathcal{I}_{\nu_j}(\Delta u(t))$, conformément à la notation introduite précédemment.

Le calcul devient :

$$\begin{aligned} & \prod_{j=1}^i \int_0^t \phi_{\nu_j}(\tau_j) [u_0 + \Delta u(t - \tau_j)] d\tau_j \\ &= \prod_{j=1}^i (u_0 \phi_{\nu_j}(0) + \mathcal{I}_{\nu_j}(\Delta u(t))) \end{aligned} \quad (1.150)$$

On pose maintenant

$$\begin{cases} \int_0^t \phi_{\nu_j}(\tau_j) d\tau_j = \phi_{\nu_j}(0) = a_j \\ \int_0^t \phi_{\nu_j}(\tau_j) \Delta u(t - \tau_j) d\tau_j = \mathcal{I}_{\nu_j}(\Delta u(t)) = b_j \end{cases} \quad (1.151)$$

de manière à pouvoir appliquer la généralisation de la loi du binôme de Newton :

$$\begin{aligned} \prod_{j=1}^i (u_0 a_j + b_j) &= u_0^i \prod_{j=1}^i a_j + \\ &\sum_{j=1}^i u_0^{i-j} \left[\sum_{k_1=1}^{C_i^{i-j}} \left(\prod_{k_1=1}^{i-j} a_{k_1} \right) \left(\prod_{k_2=1}^j b_{k_2} \right) \right] \end{aligned} \quad (1.152)$$

avec $\{k_1\} \cup \{k_2\} = \{1, \dots, k+j\}$ et $\{k_1\} \cap \{k_2\} = \emptyset$.

Cette généralisation nous permet donc d'obtenir, en remplaçant les termes a_j et b_j par leur expression, le résultat que nous souhaitions démontrer :

$$\begin{aligned} \hat{y}(t) &= \sum_{i=1}^N \left(\sum_{\nu_1=1}^{M_i} \dots \sum_{\nu_i=1}^{\nu_{i-1}} \xi_{\nu_1 \dots \nu_i} \right. \\ &\left. \left[u_0^i \prod_{j=1}^i \phi_{\nu_j}(0) + \sum_{j=1}^i u_0^{i-j} \mathcal{J}_{i,j}(\Delta u(t)) \right] \right) \end{aligned} \quad (1.153)$$

■

1.5.4 Fonctions de transfert

Proposition 1.4 : *Le développement en série de Volterra de la sortie du système (1.139) autour d'une composante continue u_0 s'écrit, dans le cas où les noyaux de Volterra sont eux-mêmes développés sur des fonctions de transfert, sous la forme :*

$$\hat{y}(t) = \sum_{i=1}^n \left(u_0^i \prod_{j=1}^i d_{j,i}(0) + \sum_{j=1}^i u_0^{i-j} \mathcal{K}_{i,j}(\Delta u(t)) \right) \quad (1.154)$$

où

$$\mathcal{K}_{i,j}(\Delta u(t)) = \sum_{k=1}^{C_i^{i-j}} \left(\prod_{l=1}^{i-j} d_{k,l}(0) \right) \left(\prod_{l=1}^j \mathcal{I}_{l,i}(\Delta u(t)) \right) \quad (1.155)$$

avec $\forall k \in [1, i-j]$, $\forall l \in [1, j]$

$$\begin{cases} \{k\} \cup \{l\} = \{1, \dots, i+j\} \\ \{k\} \cap \{l\} = \emptyset \end{cases} \quad (1.156)$$

et où l'expression $\mathcal{I}_{l,i}(u(t))$ est la réponse du développement $d_{l,i}$ à l'entrée $u(t)$:

$$\mathcal{I}_{l,i}(u(t)) = \int_0^t d_{l,i}(\tau) u(t - \tau) d\tau \quad (1.157)$$

Preuve : en considérant le développement du noyau de Volterra $h_i(\tau_1, \dots, \tau_i)$ sur des fonctions de transfert (1.133), l'expression (1.142) peut être développée de la manière suivante :

$$\hat{y}(t) = \sum_{i=1}^n \underbrace{\int_0^t \dots \int_0^t}_{i} \prod_{j=1}^i d_{j,i}(\tau_j) \prod_{j=1}^i [u_0 + \Delta u(t - \tau_j)] d\tau_j \quad (1.158)$$

où l'expression du terme $d_{j,i}$ est donnée par (1.131).

On développe ce calcul :

$$\hat{y}(t) = \sum_{i=1}^n \underbrace{\int_0^t \dots \int_0^t}_{i} \prod_{j=1}^i d_{j,i}(\tau_j) \prod_{j=1}^i [u_0 + \Delta u(t - \tau_j)] d\tau_j \quad (1.159)$$

Le premier terme $\int_0^t d_{j,i}(\tau_j) d\tau_j$ correspond à la réponse d'un développement $d_{j,i}$ de fonctions de transfert à un signal d'entrée $u(t) = 1$, $t \geq 0$. On peut donc considérer ce terme comme un gain et l'écrire $d_{j,i}(0)$.

Le deuxième terme $\int_0^t d_{j,i}(\tau_j) \Delta u(t - \tau_j) d\tau_j$ correspond à la réponse d'un développement $d_{j,i}$ à la partie dynamique de l'entrée. On peut noter ce terme $\mathcal{I}_{j,i}(\Delta u(t))$.

Le calcul devient :

$$\hat{y}(t) = \sum_{i=1}^n \prod_{j=1}^i [u_0 d_{j,i}(0) + \mathcal{I}_{j,i}(\Delta u(t))] \quad (1.160)$$

On peut alors poser

$$\begin{cases} \int_0^t d_{j,i}(\tau_j) d\tau_j = d_{j,i}(0) = a_j \\ \int_0^t d_{j,i}(\tau_j) \Delta u(t - \tau_j) d\tau_j = \mathcal{I}_{j,i}(\Delta u(t)) = b_j \end{cases} \quad (1.161)$$

et appliquer la loi du binôme de Newton définie en (1.152) pour obtenir, en remplaçant les termes a_j et b_j par leur expression, le modèle de Volterra sur fonctions de transfert et développé autour de la composante continue de l'entrée :

$$\hat{y}(t) = \sum_{i=1}^n \left(u_0^i \prod_{j=1}^i d_{j,i}(0) + \sum_{j=1}^i u_0^{i-j} \mathcal{K}_{i,j}(\Delta u(t)) \right) \quad (1.162)$$

■

1.5.5 Identification du modèle sur fonctions de transfert et application

Afin de comparer les performances du modèle défini en (1.133) et de ce même modèle développé autour d'une composante continue (1.154), nous allons appliquer ce dernier

modèle au système non linéaire de représentation d'état :

$$\begin{cases} \dot{x}(t) = -x(t) - x^2(t) + u(t) \\ y(t) = x(t) \end{cases} \quad (1.163)$$

Tout comme le modèle défini en (1.133), le modèle développé autour d'une composante continue n'est pas linéaire par rapport aux paramètres. On utilise donc à nouveau un algorithme d'identification non linéaire de type Levenberg-Marquardt.

On reprend les mêmes signaux d'entrée $u(t)$ et de sortie $y(t)$ que ceux définis par les figures 1.11(a) et 1.11(b).

La structure du modèle de Volterra est fixée de la manière suivante : on tronque la série à ses deux premiers termes. Le modèle du premier noyau est développé sur un seul terme (donc $p_{1,1} = 1$) et le deuxième noyau est également développé sur un seul terme (donc $p_{1,2} = 1$ et $p_{2,2} = 1$) :

$$\begin{aligned} D_{11}(a_{1,1}, p_{1,1}) &= \frac{\gamma_{0,1,1}}{s-a_{1,1}} \\ D_{12}(a_{2,1}, p_{1,2}) &= \frac{\gamma_{0,2,1}}{s-a_{2,1}} \\ D_{22}(a_{2,2}, p_{2,2}) &= \frac{\gamma_{0,2,2}}{s-a_{2,2}} \end{aligned} \quad (1.164)$$

En comparaison avec le modèle de Volterra (1.133) développé sur des fonctions de transfert, le nombre de paramètres de ce modèle est réduit de 8 à 6 : les pôles $a_{1,1}$, $a_{2,1}$ et $a_{2,2}$ ainsi que les gains $\gamma_{0,1,1}$, $\gamma_{0,2,1}$ et $\gamma_{0,2,2}$.

Les paramètres du modèle sont initialisés de la manière suivante :

$$\begin{aligned} a_{1,1} &= -1; a_{2,1} = -1; a_{2,2} = -1 \\ \gamma_{0,1,1} &= 1; \gamma_{0,2,1} = 1.1; \gamma_{0,2,2} = 1.2 \end{aligned} \quad (1.165)$$

Cette initialisation paramétrique nous permet d'atteindre une valeur acceptable des paramètres, sans garantie qu'il s'agisse de l'optimum.

Le principe de cette méthode consiste à identifier uniquement la partie dynamique du modèle. On n'utilise que les parties dynamiques des signaux d'entrée $u(t)$ et $y(t)$. Pour cela, il suffit de retirer la valeur moyenne $u_0 = 0.2059$ de $u(t)$ et la valeur moyenne $y_0 = 0.1692$ de $y(t)$. Cela revient à ramener ces signaux à une composante continue nulle. Les signaux d'entrée et de sortie utilisés sont ceux des figures 1.30(a) et 1.30(b).

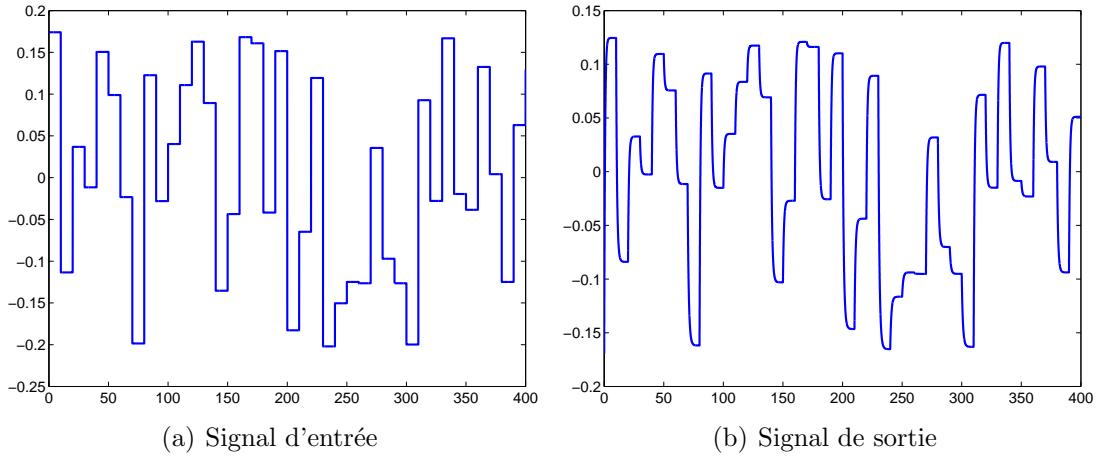


Fig. 1.30 - Partie dynamique des signaux

Après 100 itérations, les valeurs des 6 paramètres obtenues sont :

$$\begin{aligned}
 a_{1,1} &= -1.3070 \\
 a_{2,1} &= -1.1917 \\
 a_{2,2} &= -1.1917 \\
 \gamma_{0,1,1} &= 0.9855 \\
 \gamma_{0,2,1} &= -0.7806 \\
 \gamma_{0,2,2} &= 0.8544
 \end{aligned} \tag{1.166}$$

et le critère quadratique vaut $J = 0.1114$. En comparaison avec les valeurs obtenues pour les modèles précédents, celui-ci est du même ordre de grandeur. On a donc obtenu les mêmes performances avec 2 paramètres de moins que le modèle précédent.

Une fois la partie dynamique du système correctement identifiée, on utilise les valeurs des paramètres et de la composante continue u_0 du signal d'entrée $u(t)$ pour reconstruire la partie statique du modèle de Volterra global défini par (1.154).

La figure 1.31(a) illustre le cas d'une mauvaise reconstruction du signal de sortie ne tenant pas compte des termes de couplage (à savoir les termes comprenant à la fois la partie statique u_0 et la partie dynamique $\Delta u(t)$ du signal $u(t)$).

La figure 1.31(b) compare la sortie du système non linéaire et la sortie du modèle de Volterra correctement reconstruite pour les 6 valeurs paramétriques.

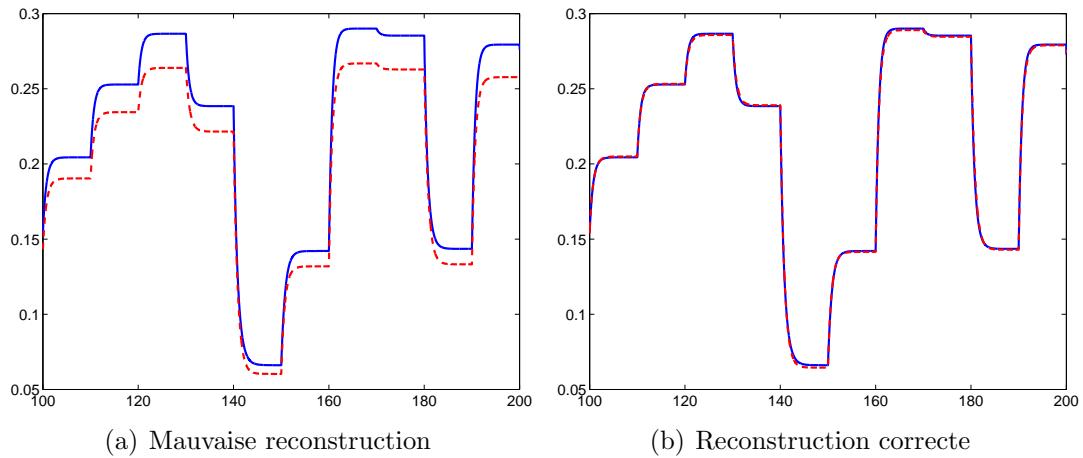


Fig. 1.31 - Sortie du système non linéaire (ligne continue) et du modèle de Volterra (ligne pointillée)

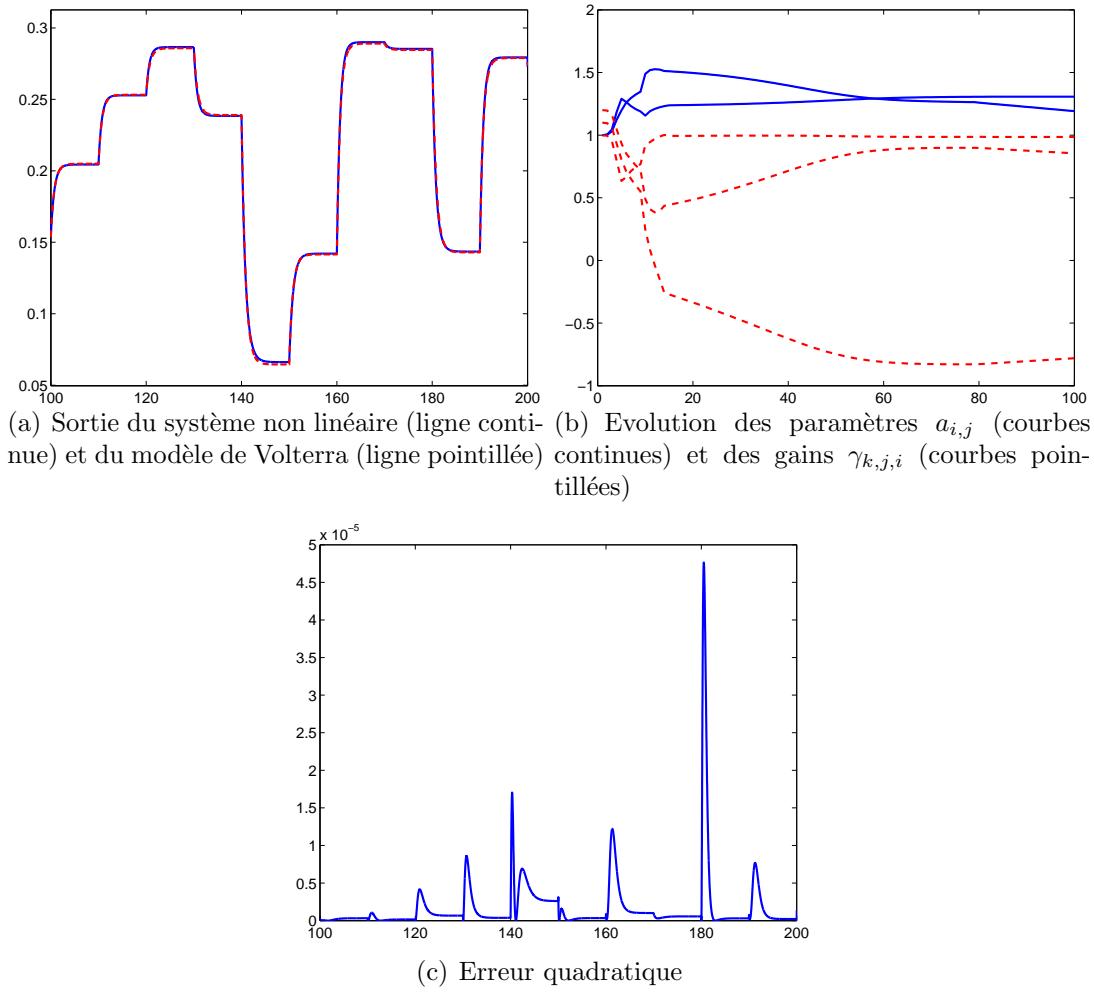


Fig. 1.32 - Résultats de modélisation et d'identification paramétrique

La figure 1.32(a) compare la sortie du système non linéaire (1.134) et la sortie du modèle de Volterra pour ces 6 valeurs paramétriques. Les deux courbes sont quasiment superposées.

La figure 1.32(b) représente l'évolution des paramètres en fonction du nombre d'itérations.

Dans ce cas aussi, tous les paramètres ne convergent pas vers des valeurs stables. L'obtention de l'optimum paramétrique n'est pas garantie mais, malgré cela, les valeurs des paramètres obtenues donnent un résultat tout à fait satisfaisant.

1.5.6 Conclusion

La séparation du signal d'entrée $u(t)$ en une partie statique u_0 et une partie dynamique $\Delta u(t)$ permet à l'utilisateur de diminuer la complexité et le nombre de paramètres du modèle de Volterra utilisé. La méthode proposée consiste à modéliser et identifier seulement la partie dynamique du modèle de Volterra. On peut ensuite reconstruire le signal $y(t)$ complet en sortie du modèle de Volterra par l'ajout à la partie dynamique identifiée $\Delta y(t)$ de la partie statique y_0 et des termes "de couplage" qui dépendent à la fois de la partie statique et de la partie dynamique de l'entrée $u(t)$.

Cette procédure permet donc d'utiliser un modèle de Volterra plus simple et plus parcimonieux dans ses paramètres. Le fait de tronquer la série de Volterra à un nombre moins important de termes implique un nombre de paramètres de modélisation des noyaux moins important. On répond ainsi davantage à l'objectif de parcimonie paramétrique fixé au début de cette étude.

1.6 Comparaison des 4 modèles

Les courbes des figures 1.33 et 1.34 reprennent l'ensemble des résultats de modélisation relatifs aux systèmes non linéaires quadratiques présentés dans ce chapitre.

La figure 1.33 reprend le cas de la modélisation des noyaux de Volterra par développement sur une base de fonctions généralisée. Le nombre total de paramètres du modèle est égal à 14. Si l'on choisit correctement les pôles des fonctions de la base, le modèle est suffisant pour représenter correctement le système non linéaire quadratique (critère quadratique $J = 0.022$).

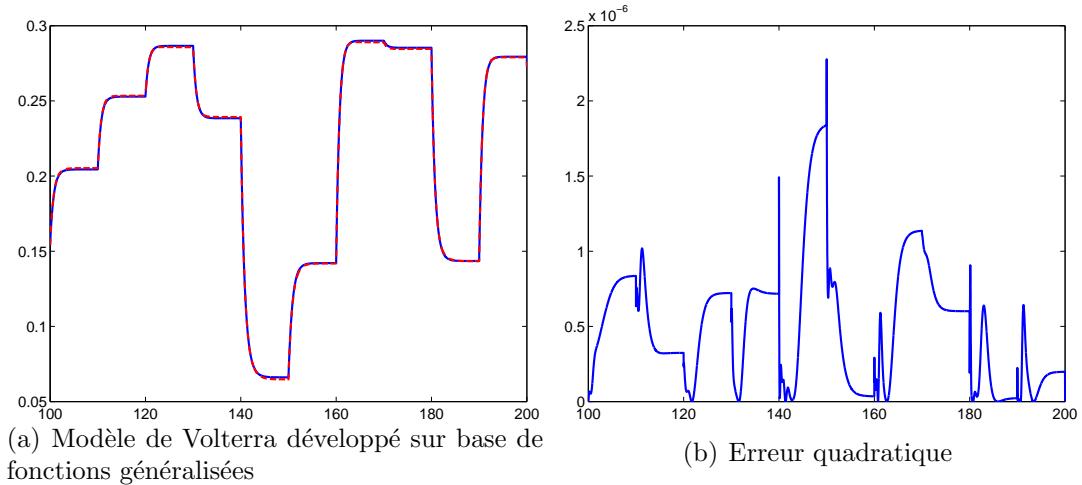


Fig. 1.33 - Résultats de modélisation sur bases de fonctions généralisée

Un des objectifs du chapitre est de réduire le nombre de paramètres du modèle tout en conservant une précision satisfaisante du modèle. La figure globale 1.34 reprend les 3 formes de modélisation sur fonctions de transfert présentées dans le chapitre.

Le premier modèle développé sur fonctions de transfert (figure 1.34(a)) réduit considérablement le nombre de paramètres (égal à 3) et assure une bonne précision (critère quadratique $J = 0.1104$). Cependant, la complexité de ce modèle rend difficile la procédure d'identification paramétrique.

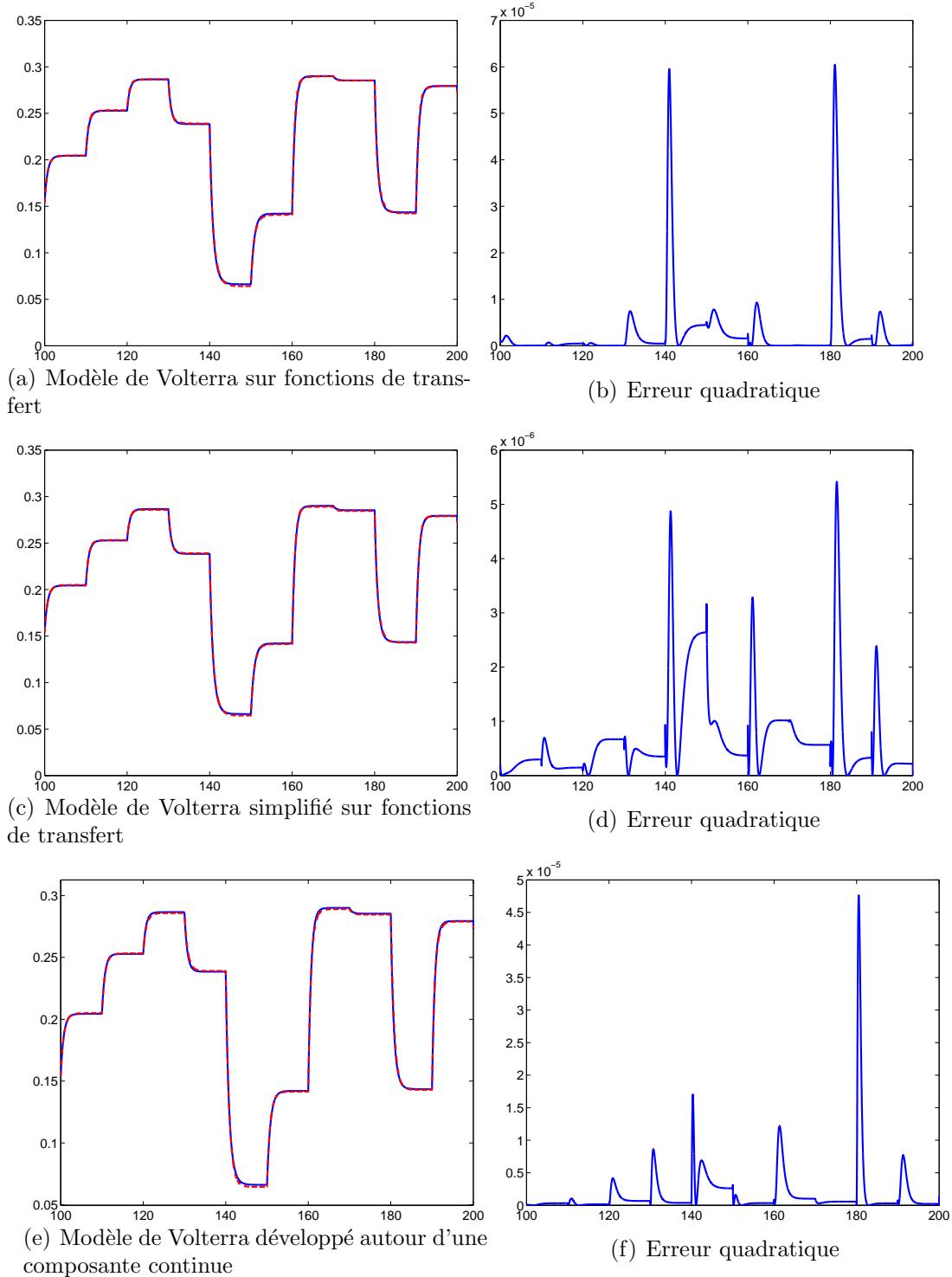


Fig. 1.34 - Comparaison des résultats de modélisation sur fonctions de transfert

Le second modèle développé sur fonctions de transfert (figure 1.34(c)) est une simplification du premier. Le nombre total de paramètres augmente (il est égal à 8) mais reste inférieur à celui du modèle développé sur une base de fonctions généralisée. La précision

reste correcte (critère quadratique $J = 0.0432$).

Le troisième modèle développé autour d'une composante continue (figure 1.34(e)) est introduit pour diminuer le nombre de paramètres du second. On passe ainsi de 8 paramètres à 6 et la précision reste correcte (critère quadratique $J = 0.1114$). Cette méthode de modélisation peut également être appliquée aux autres modèles pour améliorer leur parcimonie.

1.7 Conclusion du chapitre

La modélisation de systèmes non linéaires est un domaine de recherche qui comporte de nombreuses pistes encore inexplorées. Parmi les nombreuses possibilités de modélisation, nous avons opté pour la représentation du comportement d'un système non linéaire par les séries de Volterra. Cet outil de modélisation permet en effet de modéliser le comportement d'une assez large variété de systèmes non linéaires et peut être vu comme une généralisation de la relation de convolution, outil classique de modélisation des systèmes linéaires.

Les noyaux de Volterra peuvent rarement être exprimés de manière analytique et l'on doit avoir recours à leur modélisation. Nous avons évoqué un choix de modèles sur différentes bases de fonctions. L'utilisation de tels modèles montre toute la souplesse et l'adaptabilité des modèles de Volterra à de nombreux systèmes non linéaires. Cependant, l'absence de connaissance sur un système non linéaire ou la présence de non-linéarités trop importantes font ressortir un inconvénient majeur de cette forme de modélisation : l'augmentation du nombre de termes de la série de Volterra va de pair avec l'augmentation du nombre de paramètres nécessaires à une modélisation correcte du système. Le nombre total de paramètres peut ainsi varier du simple au double. Cette absence de parcimonie rend le modèle de Volterra difficile à utiliser et engendre des problèmes d'identification.

Afin de pallier à cet inconvénient, nous avons proposé un modèle des noyaux de Volterra développé sur fonctions de transfert. Cependant, la complexité de ce nouveau modèle entraîne également dans ce cas des difficultés d'identification. Nous avons donc envisagé de simplifier le modèle développé sur fonctions de transfert et présenté une autre forme de modèle sensiblement différente, moins complexe et plus flexible, tout en conservant une parcimonie paramétrique meilleure que dans le cas du développement des noyaux de Volterra sur bases de fonctions.

Afin d'améliorer à nouveau cette parcimonie, nous avons introduit une méthode qui consiste à séparer la partie statique de la partie dynamique des signaux d'entrée et de sortie utilisés pour l'identification des paramètres. L'identification paramétrique de la partie dynamique seule permet de simplifier davantage le modèle de Volterra nécessaire et ainsi de réduire le nombre de paramètres.

Cet ensemble de modélisations et d'améliorations permet d'obtenir un modèle de Volterra suffisamment souple et parcimonieux. Nous allons donc ensuite pouvoir envisager son inversion afin de reconstruire le signal d'entrée $u(t)$ d'un système non linéaire à partir de mesures du signal de sortie $y(t)$ et de la connaissance de la structure et des paramètres du modèle de Volterra utilisé pour représenter son comportement.

Chapitre 2

Inversion d'un modèle de Volterra

2.1 Introduction

La détermination d'une grandeur inconnue à partir de mesures est un problème qui continue de se poser dans de nombreux domaines scientifiques (physique, chimie, biologie,...). Selon les cas, l'évaluation de la grandeur inconnue peut s'effectuer de manière directe ou de manière indirecte. Les problèmes qui en découlent seront alors traités différemment.

Dans le cas d'une mesure directe, la précision avec laquelle sera déterminée la grandeur inconnue est avant tout un problème lié à la précision de l'instrumentation utilisée : il s'agit de recueillir des mesures qui soient les plus précises et les plus informatives possible afin de pouvoir quantifier la grandeur considérée dans les meilleures conditions possibles.

Dans le cas où la mesure ne peut pas s'effectuer directement, une solution est la mesure d'une autre grandeur dont on sait qu'elle a un lien avec la grandeur inconnue à déterminer. En plus d'une instrumentation précise nécessaire à la mesure de cette deuxième grandeur, on doit également avoir une idée assez précise du lien qui unit les deux grandeurs, celle que l'on mesure et celle que l'on veut déterminer. Ce lien peut être établi en utilisant les lois physiques, chimiques ou mathématiques qui régissent le phénomène étudié (qui met en jeu les grandeurs considérées). Dans la majeure partie des cas, on doit se contenter d'une approximation plus ou moins satisfaisante du lien entre les deux grandeurs à cause d'un manque de connaissance de l'ensemble des phénomènes physiques ou chimiques mis en jeu.

La détermination d'une grandeur inconnue à partir de la mesure d'une grandeur corrélée et du lien qui unit ces deux grandeurs porte le nom de problème inverse.

Ce chapitre est constitué de trois grandes sections. La première section dresse un état de l'art non exhaustif des problèmes inverses. Les deux sections suivantes décrivent l'apport de ce mémoire quant aux techniques d'inversion du modèle de Volterra. La deuxième section est consacrée à l'inversion du modèle de Volterra par régularisation de Tikhonov. Une telle démarche a déjà été proposée par [Inglada, 2000] dans le cadre de la topographie d'un milieu sous-marin. Différentes techniques d'inversion du modèle de Volterra ont été présentées dans un cadre théorique rigoureux inspiré de l'approche de [Schetzen, 1980]. Cependant, ces techniques n'ont été ensuite appliquées qu'à un transfert linéaire, suffisant pour correctement représenter les phénomènes physiques mis en jeu. La méthode d'inversion du modèle de Volterra proposée dans ce mémoire sera appliquée à deux systèmes chimiques aux non-linéarités avérées. La troisième section se démarque de l'approche précédente en introduisant deux techniques d'inversion du modèle de Volterra par retour d'entrée reconstruite.

2.2 Les problèmes inverses

2.2.1 Introduction

Dans le cadre général des sciences expérimentales, l'expérimentateur est souvent amené à mesurer précisément une quantité donnée. Dans certains cas, cette quantité est directement accessible et mesurée par l'intermédiaire d'un simple capteur. On citera, dans le domaine général de l'électricité, le cas de la tension ou de l'intensité d'un courant électrique, mesurables facilement à l'aide d'un voltmètre ou d'un ampèremètre. Dans le domaine général de la chimie, le pH d'une solution est facilement mesurable à l'aide d'un pHmètre.

Mais il existe de nombreux cas où une grandeur, qu'elle soit physique ou chimique, n'est pas directement accessible à la mesure. On doit dans ce cas se contenter de mesurer d'autres quantités directement observables et, quand c'est possible, trouver un lien entre la ou les grandeur(s) directement mesurable(s) et la grandeur non directement mesurable. Ce lien, qui peut être analytique et que nous nommerons modèle, peut permettre d'obtenir la grandeur non mesurable en la reconstruisant.

Jusqu'ici, nous avons employé le terme de "modèle" pour qualifier le lien entre une grandeur d'entrée et une grandeur de sortie (dans le cas mono-entrée mono-sortie). Nous allons garder cette définition et donc faire l'analogie entre sortie du modèle et grandeur directement mesurable d'une part, et entrée du modèle et grandeur non directement mesurable d'autre part. Par conséquent, étant donnés le modèle et la sortie, la détermination de l'entrée est ce que l'on appelle l'inversion ou encore, la résolution d'un problème inverse. Le champ d'application de ce type de problème est très vaste : géophysique [Scales et Smith, 1996], thermique [Ghannam, 2000], médecine [Louis, 1992] [Louis, 1997], topographie [Inglada, 2000], la liste étant loin d'être exhaustive. Suivant le domaine d'application considéré, on utilise également le terme déconvolution ou restauration (cas du traitement d'image et de signal) [Demoment, 1987].

Pour reconstruire une grandeur que l'on considère comme l'entrée d'un modèle, il faut donc pouvoir mesurer la sortie du modèle, qui peut être vue comme une conséquence de l'entrée. Un problème inverse peut donc être vu comme la reconstruction d'une cause à partir de la mesure d'une ou plusieurs de ses conséquences et de la connaissance du modèle qui les relie.

2.2.2 Cadre général : les problèmes mal posés

2.2.2.1 Introduction

Un grand nombre de problèmes en sciences expérimentales nous amène à déterminer une grandeur non observable u à partir d'un ensemble fini de mesures d'une ou plusieurs grandeurs observables y et de la connaissance du modèle qui les relie :

$$H(u, y) = 0 \quad (2.1)$$

H est un opérateur (linéaire ou non linéaire) qui décrit les relations théoriques entre u et y . Il est en général caractérisé par un vecteur de paramètres θ .

Le problème direct consiste à calculer la grandeur y connaissant la grandeur u et le modèle H . Il s'agit d'un problème d'identification paramétrique et de calcul de réponse bien connu des automaticiens.

Le problème inverse consiste à calculer u connaissant les mesures y et, parfois, le modèle H . C'est un problème plus délicat car il est souvent mal posé (nous verrons ce que cela veut dire) et la solution obtenue peut être extrêmement sensible aux erreurs et aux perturbations sur les mesures des grandeurs observables y . N'importe quel dispositif expérimental donne lieu à des incertitudes dues essentiellement à la précision finie des mesures d'une part, et à la corruption des mesures par du bruit d'autre part. Il faut donc tenir compte de ces imprécisions dans la résolution du problème inverse.

La résolution des problèmes inverses fait donc appel à la notion mathématique de problème mal posé au sens de Hadamard [Hadamard, 1923].

2.2.2.2 Cas linéaire

On garde les définitions de u , grandeur (ou vecteur de grandeurs) non directement mesurable dans un ensemble F , et de y , grandeur (ou vecteur de grandeurs) observable dans un ensemble G dont on possède un nombre fini de mesures. On suppose également que ces deux grandeurs sont liées par un opérateur H linéaire de la manière suivante :

$$Hu = y \quad (2.2)$$

Définition : d'après le mathématicien français Hadamard [Hadamard, 1923], un problème est dit bien posé s'il admet une solution et si cette solution est unique et stable.

a) Existence

La solution d'un problème existe si

$$\forall y \in G, \exists u \in F : Hu = y \quad (2.3)$$

b) Unicité

La solution d'un problème est unique si

$$\left. \begin{array}{l} Hu_1 = y \\ Hu_2 = y \end{array} \right\} \Rightarrow u_1 = u_2 \quad (2.4)$$

c) Stabilité

La solution d'un problème est stable si une faible variation de y engendre une faible variation de la solution u . Autrement exprimé :

$$\delta y \ll 1 \Rightarrow \delta u \ll 1 \quad (2.5)$$

On peut également dire dans ce cas que la dépendance de u par rapport à y est continue. L'exigence de continuité est reliée à celle de stabilité ou de robustesse de la solution. La continuité est cependant une condition nécessaire, mais pas suffisante, de stabilité. Un problème bien posé pourra ainsi être mal conditionné. Cette notion est caractérisée par le calcul d'un nombre de condition qui, suivant sa valeur, renseigne l'utilisateur sur la stabilité numérique du problème.

Selon Hadamard [Hadamard, 1923], un problème est dit mal posé si au moins une des 3 conditions définies précédemment n'est pas vérifiée.

2.2.2.3 Application des notions précédentes au modèle non linéaire de Volterra

Toutes les notions définies précédemment peuvent s'appliquer dans le cas où l'opérateur H qui relie la grandeur u à la grandeur y est non linéaire. Nous nous limiterons au cas où H est une fonctionnelle de Volterra. La relation entre u et y est la suivante :

$$H[u] = y \quad (2.6)$$

Les conditions d'existence, d'unicité et de stabilité du problème inverse sont les mêmes que dans le cas linéaire. Nous supposerons en outre que la relation 2.6 est stricte au sens BIBO : l'entrée et la sortie sont bornées.

a) Existence

La solution du problème inverse non linéaire existe si

$$\forall y \in G, \exists u \in F : H[u] = y \quad (2.7)$$

b) Unicité

La solution du problème inverse non linéaire est unique si

$$\left. \begin{array}{l} H[u_1] = y \\ H[u_2] = y \end{array} \right\} \Rightarrow u_1 = u_2 \quad (2.8)$$

c) Stabilité

Dans le cadre des hypothèses faites sur le système non linéaire direct 2.6, la solution du problème inverse non linéaire est stable si une faible variation de y engendre une faible variation de la solution u . Autrement exprimé :

$$\delta y \ll 1 \Rightarrow \delta u \ll 1 \quad (2.9)$$

Un problème inverse non linéaire est dit mal posé si au moins une de ces 3 conditions n'est pas vérifiée. Sa résolution au sens habituel du terme est alors impossible. Il faut envisager d'autres techniques de résolution.

2.2.2.4 Quelques exemples de problèmes inverses

On rencontre de nombreux exemples de problèmes inverses dans des domaines très variés des sciences expérimentales, en particulier celui de l'imagerie. L'intérêt des techniques d'imagerie (tomographie à rayons X, à ultrasons, à émission de positrons, tomographie d'impédance, imagerie par résonance magnétique nucléaire,...) est qu'elles permettent de voir (ou bien de reconstruire) ce qui ne peut être vu directement, et cela sans détruire ce qui est observé (cas de l'imagerie médicale) et sans danger pour l'observateur (cas des problèmes thermiques). L'objet dont on veut caractériser l'intérieur peut être le corps humain (imagerie médicale : échographie, scanner, rayons X), l'écorce terrestre (prospection pétrolière par des méthodes sismiques ou magnétiques, hydrogéologie) ou une étoile (astronomie). Il peut aussi s'agir de reconstruire des images (restauration d'images floues dans le cas du télescope Hubble) ou des topographies inaccessibles (fonds sous-marins).

Du point de vue mathématique, ces problèmes se répartissent en deux grands groupes : les problèmes linéaires qui se ramènent à la résolution d'équations intégrales de première espèce, et les problèmes non linéaires qui aboutissent le plus souvent à l'estimation de paramètres dans des équations différentielles ou aux dérivées partielles. Cette seconde catégorie peut être séparée en deux sous-catégories selon que l'on cherche à reconstruire un vecteur (dimension finie) ou une fonction.

De manière générale, on rencontre des problèmes inverses dans les domaines particuliers suivants :

- **Thermique** : afin de déterminer la répartition de la température dans un matériau hétérogène, on écrit la loi de conservation de l'énergie ainsi que la loi de Fourier qui relie le flux de chaleur au gradient de température. Ces deux équations permettent d'obtenir l'équation de la chaleur en milieu hétérogène. Cette équation aux dérivées partielles doit ensuite être complétée par des conditions aux limites et une condition initiale de température. Dans ce cas précis, le problème direct consiste à déterminer la température T connaissant différents coefficients physiques (densité du fluide étudié, chaleur spécifique et conductivité thermique), ainsi que la source de chaleur f . Ce problème est bien connu, tant du point de vue théorique (existence et unicité de la solution) que du point de vue numérique. Plusieurs problèmes inverses peuvent se poser : la détermination de la température initiale étant donnée une mesure de la

température à un instant t_f (problème linéaire) ou bien la détermination de certains des coefficients de l'équation de la chaleur étant donnée une mesure partielle de la température (problème non linéaire).

- **Hydrogéologie** : l'hydrogéologie (science qui étudie entre autres les nappes phréatiques) est une source abondante de problèmes inverses. Il est en effet difficile d'accéder aux couches du sous-sol pour mesurer les propriétés aqueuses des roches. Un problème actuel est le contrôle des polluants dans les nappes d'eau souterraines. Un milieu poreux est constitué d'une matrice rocheuse, comprenant des pores qui laissent passer l'eau. Il est impossible de décrire l'écoulement d'un fluide dans un tel milieu hétérogène dans la mesure où l'on doit prendre en compte des échelles spatiales allant du centimètre (dimension d'un pore) au kilomètre (modèle d'une région), et sachant que la disposition des pores est a priori inconnue. On utilise alors des modèles simplifiés (dont le plus courant est la loi de Darcy) afin de relier la hauteur de l'eau dans le milieu (appelée *charge piézométrique* et notée $h(x, y, z, t)$) à la vitesse de filtration (notée $\vec{q}(x, y, z, t)$). Cette loi établit que la vitesse de filtration est proportionnelle à l'opposé du gradient hydraulique. On exprime ensuite la loi de conservation de la masse (le milieu étant par hypothèse incompressible). On obtient alors une équation aux dérivées partielles à laquelle on rajoute une condition initiale (h donné à $t = 0$) et des conditions aux limites (conditions de Dirichlet ou de Neumann). Les problèmes de transport de contaminants font intervenir, en plus de l'écoulement, l'évolution de la concentration des espèces (composés chimiques, hydrocarbures, radionucléides) contenues dans l'écoulement. Ce phénomène met en jeu trois mécanismes : la convection (imposée par la vitesse de filtration \vec{q}), la diffusion moléculaire et la dispersion cinématique. La quantité étudiée est la concentration $C(x, y, z, t)$ du polluant qui obéit à une équation de type convection-diffusion. Le problème direct est constitué par l'équation aux dérivées partielles et l'équation de type convection-diffusion. Ce problème couplé est en général non linéaire. On peut mesurer la concentration en polluant en un certain nombre de points et à des instants discrets. Un problème inverse est alors de chercher la conductivité hydraulique connaissant ces mesures. En pratique, il est rare que l'on ait suffisamment de mesures et le problème est donc sous-déterminé.
- **Sismique** : la prospection pétrolière par des méthodes sismiques donne lieu à un problème inverse largement étudié en raison de l'intérêt économique rattaché à sa solution. Il s'agit en réalité d'une famille de problèmes inverses dont le but commun est de déterminer les propriétés élastiques du sous-sol (densité, vitesses de propagation des ondes élastiques) à partir de mesures des champs de déplacement, ou de pression, en surface. Lors d'une campagne sismique, une source (en général une explosion) provoque un ébranlement des roches formant le sous-sol. L'écho est enregistré par une série de capteurs placés en surface. Cette expérience est répétée pour plusieurs positions de la source (de plusieurs centaines à plusieurs milliers). On obtient de cette manière une très grande quantité de données. Le but est d'estimer les propriétés du milieu étant donné un modèle de propagation. Il existe un grand nombre de méthodes spécifiques pour traiter ce problème. Il existe également

plusieurs modèles physiques pouvant rendre compte de l'expérience décrite ci-dessus à des degrés divers d'approximation. Un des cas les plus simples consiste à supposer que la région étudiée se compose d'un fluide (expérience de sismique sous-marine). On peut démontrer dans ce cas que la propagation des ondes est régie par l'équation des ondes acoustiques. La quantité mesurée est un champ scalaire de pression. Il est commode de faire l'hypothèse que le domaine d'étude est le demi-espace $\{z > 0\}$. On suppose donc que la terre est plane et infinie mais ces approximations sont justifiées par l'échelle considérée, qui est de l'ordre du kilomètre, l'axe Oz étant orienté vers le bas. Le problème direct est alors, connaissant la source d'ondes, leur vitesse de propagation et la densité du milieu, de reconstruire le champ scalaire de pression. Ce problème direct a été abondamment étudié, ses propriétés numériques sont bien connues ainsi que des méthodes efficaces pour sa résolution numérique. Le problème inverse consiste à déterminer la vitesse de propagation des ondes, leur source et la densité du milieu (qu'il n'est pas réaliste de supposer connue) à partir des mesures, c'est-à-dire de la connaissance de la pression en chaque point du milieu. Ce problème est non linéaire puisque sa solution dépend de façon non linéaire de la vitesse de propagation des ondes et de la densité du milieu. Dans la réalité, il faut tenir compte d'un paramètre supplémentaire : l'expérience est répétée en déplaçant le dispositif sources-récepteurs. L'ensemble des "tirs" fournit une immense quantité de données, rendant ainsi le problème inverse surdéterminé. Une information importante à exploiter est que tous ces enregistrements proviennent du même sous-sol.

- **Imagerie médicale** : les sciences médicales fournissent un grand nombre de problèmes inverses, dont l'importance pratique n'échappera à personne. Dans chacun des cas que nous allons évoquer, la grandeur à reconstruire se situe bien évidemment à l'intérieur du corps humain.
 - **tomographie par rayons X** : la tomographie par rayons X est la principale technique utilisée par les scanners. Un tube à rayons X est monté sur un portique qui entoure le patient. Les rayons émis sont mesurés par des détecteurs placés en face de l'émetteur. On considère souvent la situation bidimensionnelle, où le domaine représente une "section transverse" du patient. On suppose que les rayons suivent une ligne droite et sont atténués à la traversée des tissus proportionnellement à leur intensité et à la distance parcourue (loi de Bouger). Les rayons X suivent des lignes droites qui seront paramétrisées par leur vecteur normal et leur distance à l'origine. Le problème direct consiste à déterminer l'intensité mesurée au détecteur connaissant celle à l'émetteur ainsi que la fonction d'atténuation du patient. Le problème inverse consiste à déterminer la fonction d'atténuation connaissant l'intensité mesurée à l'émetteur et au récepteur. En pratique, il faut que les données soient mesurées de façon uniforme sur un cercle autour du patient. Ce n'est pas tout le temps réalisable et la fonction d'atténuation ne peut donc pas toujours être reconstruite de manière stable. Par ailleurs, la formule de reconstruction fait intervenir la dérivée des mesures, qui montre également son caractère instable, la dérivation amplifiant le bruit des mesures.

- **échographie** : cette méthode d’investigation présente l’avantage d’être sans risque pour le patient. Les sources sont de brèves impulsions d’une onde acoustique à très haute fréquence ; les mesures sont des échos acoustiques et l’on recherche les discontinuités de la vitesse de propagation dans le milieu. Le problème direct est de calculer une onde diffractée u_s connaissant le potentiel q (et l’onde plane incidente u_i) tandis que le problème inverse est de retrouver q à partir de mesures de u_s effectuées loin de l’obstacle. Par rapport aux exemples précédents, le problème est posé ici dans le domaine fréquentiel. Lorsque l’onde traverse le patient, elle est réfléchie par les changements de la densité et des paramètres élastiques des tissus.

2.2.2.5 Conclusion

Cette liste d’exemple est loin d’être exhaustive. Le but était de montrer que la théorie des problèmes inverses est très étendue et que des applications pratiques nombreuses et variées existent.

Les méthodes de résolution des problèmes inverses sont elles aussi relativement variées suivant les problèmes considérés et les approches adoptées pour les résoudre.

2.2.3 Méthodes algébriques déterministes

Un problème inverse est dit mal posé s’il ne satisfait pas à au moins une des conditions suivantes : existence d’une solution, unicité de cette solution et stabilité de cette solution.

La nécessité d’approfondir des problèmes qui ne sont pas mathématiquement bien posés, mais cependant d’un grand intérêt pour les sciences expérimentales, est à l’origine de deux branches de l’analyse : la théorie de l’inversion généralisée et celle de la régularisation.

2.2.3.1 Moindres carrés et inversion généralisée

Lorsque la difficulté de résolution du problème inverse provient de la non-unicité de la solution, la théorie de l’inversion généralisée permet de trouver une solution quel que soit le cas envisagé. La résolution de l’équation linéaire $y = Hu$ présente 3 possibilités : soit il y a une solution au problème, soit il y a une infinité de solutions au problème, soit il n’y a pas de solution au problème.

Dans le premier cas, on doit pouvoir directement calculer l’inverse de H pour obtenir la solution u . Dans le deuxième cas, il faut choisir une solution optimale au problème parmi un ensemble de solutions suivant un critère défini au préalable. Dans le troisième cas, on ne peut pas trouver de solution exacte mais on essaie de définir une solution approximative qui minimise un critère de différence entre y et Hu . Dans le cas où le critère

choisi est la norme euclidienne, la solution est dite solution des moindres carrés.

Considérons l'équation

$$y = Hu \quad (2.10)$$

où $y \in \mathbb{R}^m$ et $u \in \mathbb{R}^n$. Plaçons nous dans le cas général où H est une matrice de dimension $m \times n$.

L'inverse généralisée d'un tel système est notée H^+ et satisfait les conditions suivantes [Albert, 1972] :

$$\begin{aligned} HH^+H &= H \\ H^+HH^+ &= H^+ \\ (HH^+)^T &= HH^+ \\ (H^+H)^T &= H^+H \end{aligned} \quad (2.11)$$

La solution du système linéaire s'écrit alors

$$u = H^+y \quad (2.12)$$

La résolution d'un tel système met en évidence 4 cas. Selon les dimensions m et n et le rang de la matrice H , on définira les matrices inverse, pseudo-inverse à gauche et pseudo-inverse à droite.

La matrice inverse est la solution d'un problème qui possède autant d'inconnues que de contraintes. Selon les cas, il n'est pas pour autant certain qu'il existe une solution.

La matrice pseudo-inverse à gauche est une solution d'un problème sur-déterminé, qui contient des informations redondantes. Elle minimise l'erreur quadratique moyenne.

La matrice pseudo-inverse à droite est une solution d'un problème sous-déterminé, qui ne contient pas suffisamment d'information pour garantir une solution unique. Elle permet d'obtenir une solution particulière qui minimise la norme quadratique.

Ces deux pseudo-inverses, qu'on appelle généralement matrice inverse généralisée au sens défini par Moore et Penrose [Albert, 1972] [Ben-Israel et Greville, 2003], sont définies par

$$H^+ = \lim_{\delta \rightarrow 0} (H^T H + \delta^2 I)^{-1} H^T = \lim_{\delta \rightarrow 0} H^T (H H^T + \delta^2 I)^{-1} \quad (2.13)$$

Si les colonnes de H sont linéairement indépendantes, on peut poser $\delta = 0$.

a) Cas où $m=n$ et $\text{rang}(H)=n$

C'est le cas d'une matrice carrée ($m = n$) et non singulière ($\det(H) \neq 0$). Dans ce cas unique, $H^+ = H^{-1}$ est la matrice inverse de H . La matrice H^{-1} vérifie la propriété

$$H^{-1}H = HH^{-1} = I_n \quad (2.14)$$

où I_n est la matrice identité de dimension n .

b) Cas où $m < n$ et $\text{rang}(H) = m$

Il s'agit du cas où le problème est sous-déterminé : il y a plus d'inconnues que de contraintes. Dans ce cas, la matrice inverse généralisée est de la forme $H^+ = H^T(HH^T)^{-1}$ et vérifie la propriété $HH^+ = I_m$. Il existe une infinité de solutions au problème inverse. On note U l'ensemble des solutions u^+ de l'équation $u = H^+y$.

Parmi cet ensemble de solution, on retient la solution \hat{u} qui minimise une norme préétablie :

$$\hat{u} = \min \{ \|u\|, u = H^+y \} \quad (2.15)$$

c) Cas où $m > n$ et $\text{rang}(H) = n$

Il s'agit du cas où le problème est surdéterminé : il y a plus de contraintes que d'inconnues. La matrice inverse généralisée est de la forme $H^+ = (H^TH)^{-1}H^T$ et vérifie la propriété $HH^+ = I_n$. Il n'existe pas de solution exacte au problème inverse et l'on doit définir une quasi-solution \hat{u} de l'équation $y = Hu$ qui minimise une distance $\Delta(y, Hu)$ entre y et Hu :

$$\hat{u} = \arg \min_u \{ \Delta(y, Hu) \} \quad (2.16)$$

Lorsque la distance choisie est une distance euclidienne $\Delta(y, Hu) = \|y - Hu\|^2$, \hat{u} est la solution au sens des moindres carrés :

$$\hat{u} = \arg \min_u \{ \|y - Hu\|^2 \} \quad (2.17)$$

Cet ensemble coïncide avec l'ensemble des solutions de l'équation normale :

$$[H^TH]\hat{u} = H^Ty \quad (2.18)$$

d) Cas où $\text{rang}(H) = k < \min(m, n)$

Dans ce cas, la matrice H est singulière ($\det(H) = 0$). Pour l'inverser, il faut faire appel à une décomposition en valeurs singulières ou encore à des méthodes itératives.

i) Décomposition en valeurs singulières

On considère l'équation $y = Hu$ où $\text{rang}(H) < \min(m, n)$. On note $\{v_j, j = 1, \dots, k\}$ les vecteurs propres de la matrice $H^T H$, $\{u_i, i = 1, \dots, k\}$ les vecteurs propres de la matrice HH^T et λ_j^2 les valeurs propres correspondantes.

On a alors les relations suivantes :

$$\begin{aligned} H^T H v_j &= \lambda_j^2 v_j, \quad j = 1, \dots, k \\ HH^T u_i &= \lambda_i^2 u_i, \quad i = 1, \dots, k \\ Hv_j &= \lambda_j u_j, \quad j = 1, \dots, k \\ H^T u_i &= \lambda_i v_i, \quad i = 1, \dots, k \end{aligned} \tag{2.19}$$

et la matrice H peut se décomposer de la manière suivante :

$$H = U \Lambda V^T \tag{2.20}$$

où U , Λ et V sont des matrices définies par

$$\begin{aligned} U &= (u_1 \ u_2 \ \cdots \ u_k) \\ V &= (v_1 \ v_2 \ \cdots \ v_k) \\ \Lambda &= \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_k) \end{aligned} \tag{2.21}$$

avec $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k$ et $k \leq \inf(m, n)$.

La solution inverse généralisée au sens de Moore et Penrose est définie par

$$u^+ = H^+ y \text{ avec } H^+ = V \Lambda^+ U \tag{2.22}$$

avec

$$\Lambda^+ = \text{diag}\{\alpha_i\}, \left\{ \begin{array}{ll} \alpha_i = \frac{1}{\lambda_i} & \text{si } \lambda_i \neq 0 \\ \alpha_i = 0 & \text{si } \lambda_i \simeq 0 \end{array} \right. \tag{2.23}$$

ii) Méthodes itératives

Ces méthodes permettent de calculer itérativement une solution u de norme minimale au problème $Hu = y$. Elles sont couramment utilisées en raison de leur simplicité et de la facilité d'incorporation de contraintes déterministes. Elles nécessitent cependant un gros volume de calcul à cause du grand nombre d'itérations nécessaires à l'approximation optimale de la solution. Il est également nécessaire de fixer une règle d'arrêt qui peut reposer sur l'estimation de l'erreur résiduelle. Un seuil d'arrêt est fixé au départ.

La méthode de reconstitution itérative la plus ancienne est celle de Van Cittert (qui date des années 1930) [Demoment, 1987]. D'autres auteurs ont ajouté des modifications afin d'accélérer la convergence vers la solution optimale et d'améliorer la solution reconstruite. On citera les méthodes de Baily et de Landweber [Demoment, 1987], la méthode de Jansson [Jansson, 1984] et la méthode du gradient conjugué [Sullivan, 1990].

2.2.3.2 Régularisation

La régularisation d'un problème mal posé consiste à le transformer en un problème bien posé : étant donné les mesures et la connaissance du modèle dont on dispose, on cherche à définir une solution unique et à assurer la stabilité de cette solution vis à vis des erreurs de mesure [Tikhonov et Arsénine, 1976].

Si l'on considère à nouveau le problème

$$y = Hu \quad (2.24)$$

où l'opérateur H est linéaire ou non linéaire, alors ce problème est bien posé au sens de Tikhonov s'il répond aux trois conditions énumérées précédemment.

Dans le cas d'un problème linéaire, les principales difficultés sont la sur-détermination et les valeurs singulières proches de 0 de la matrice H .

Suivant les connaissances dont on dispose, une estimation \hat{u} de la grandeur u à reconstruire est correcte si le terme $\hat{y} = H\hat{u}$ est suffisamment proche de la grandeur mesurée y . La qualité de la grandeur reconstruite peut donc être quantifiée en calculant l'adéquation entre la grandeur mesurée y et la grandeur calculée Hu sous la forme d'un critère :

$$J(u) = \Delta(y, Hu) \quad (2.25)$$

où Δ représente la distance entre y et Hu . Cette distance peut être la norme euclidienne et le critère devient alors le critère quadratique classique :

$$J(u) = \|y - Hu\|^2 \quad (2.26)$$

La minimisation de ce critère conduit à la solution \hat{u} du problème inverse :

$$\hat{u} = \arg \min_u \{ J(u) \} \quad (2.27)$$

Cependant, on s'aperçoit rapidement que cette minimisation conduit rarement à une estimation correcte \hat{u} de la grandeur u à reconstruire. Nous avons déjà vu en effet que si le rang de la matrice H de dimension $m \times n$ est inférieur à n , il existe une infinité de solutions \hat{u} . On utilise alors un critère de comparaison. Ce critère ne garantit en aucun cas la stabilité de la solution obtenue car le mauvais conditionnement de la matrice H peut la conduire à se comporter comme un filtre passe-haut qui va amplifier les plus petites variations de la grandeur mesurée y . La condition de stabilité définie plus haut n'est donc pas assurée.

La grandeur mesurée y ne donne aucun renseignement sur certains aspects de la grandeur u à reconstruire. Il est donc nécessaire d'ajouter des informations additionnelles sur la grandeur u qui permettront de sélectionner la meilleure reconstruction parmi plusieurs possibles. Une manière d'effectuer cette sélection consiste à introduire un second terme $\Omega(u)$ au critère général (2.25) afin d'ajouter une information supplémentaire concernant la

grandeur u à reconstruire et ainsi de *régulariser* le problème. On appelle ce terme ajouté *terme de pénalisation* ou de *régularisation*. Le critère $J(u)$ pénalisé devient :

$$J(u) = \Delta(y, Hu) + \Omega(u) \quad (2.28)$$

Si l'on reprend le cas du critère (2.26), l'ajout du terme $\Omega(u)$ le transforme en un critère des moindres carrés de norme minimale, car il ne s'agit plus ici de simplement minimiser un seul terme mais de choisir parmi plusieurs possibilités une solution \hat{u} qui rendra le terme $\Omega(u)$ minimal. On peut choisir *a priori* une valeur constante particulière J_0 du critère J_u : à partir du moment où la valeur du critère est inférieure ou égale à J_0 , on considérera qu'on a obtenu une solution acceptable au problème *régularisé* [Tikhonov et Arsénine, 1976], [Ljung *et al.*, 1992].

Un choix possible parmi les plus simples pour $\Omega(u)$ consiste à prendre la norme euclidienne de la grandeur u à reconstruire :

$$\Omega(u) = \|u\|^2 \quad (2.29)$$

Dans le cas où l'on connaît *a priori* une solution par défaut u_d au problème inverse, on peut l'intégrer au terme $\Omega(u)$:

$$\Omega(u) = \|u - u_d\|^2 \quad (2.30)$$

Plus généralement, dans beaucoup de cas, il ne suffit pas seulement de minimiser la norme de la grandeur u à reconstruire mais d'un opérateur L (linéaire ou non) qui agit sur cette grandeur. Il faut donc intégrer cet opérateur dans $\Omega(u)$:

$$\Omega(u) = \|L(u)\|^2 \quad (2.31)$$

Dans le cas simple où l'opérateur L est linéaire, on peut écrire :

$$\Omega(u) = \|Lu\|^2 = uL^T Lu \quad (2.32)$$

Typiquement, l'opérateur L (dans le cas linéaire) est la matrice identité. Mais on peut également choisir une approximation de la dérivée première L_1 :

$$L_1 = \frac{1}{\Delta T} \begin{pmatrix} -1 & 1 & \cdots & 0 \\ 0 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & -1 \end{pmatrix} \quad (2.33)$$

Il est également possible d'utiliser une approximation de la dérivée seconde :

$$L_2 = \frac{1}{(\Delta T)^2} \begin{pmatrix} 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \ddots & \vdots \\ \vdots & 0 & 1 & \ddots & 1 \\ 0 & \ddots & \ddots & \ddots & -2 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \quad (2.34)$$

Dans d'autres cas, on pourra utiliser une combinaison linéaire des dérivées

$$\Omega(u) = \sum_{k=1}^q \alpha_k \|L_k u\|^2 \quad (2.35)$$

où L_k est une matrice d'approximation de la dérivée d'ordre k et les termes α_k sont des constantes positives.

Il existe de nombreuses méthodes de régularisation qui essaient de traiter à la fois le problème de la fidélité aux données mesurées (correspondant au premier terme du critère à minimiser) et celui de la fidélité à l'information a priori (correspondant au deuxième terme du critère à minimiser). Ces deux problèmes étant bien souvent en conflit, il s'agit alors de trouver une solution \hat{u} qui soit un bon compromis entre les nécessités dues aux grandeurs mesurées et aux données a priori [Ljung *et al.*, 1992].

a) Régularisation de Tikhonov

La régularisation de Tikhonov est une des méthodes les plus connues et de nombreux développements existent de nos jours [Tikhonov et Arsenine, 1976], [Chavent et Kunisch, 1994], [Johansen, 1997]. Le critère à minimiser $J(u)$ est défini de la manière suivante :

$$J(u) = \|y - Hu\|^2 + \lambda \|L(u)\|^2 \quad (2.36)$$

où le terme λ est appelé coefficient de régularisation. Ce coefficient sert ici à pondérer l'influence des connaissances a priori dont on dispose sur le problème à résoudre. La solution \hat{u}_λ du problème inverse $y = Hu$ est définie de la manière suivante :

$$\hat{u}_\lambda = \arg \min_u \{\|y - Hu\|^2 + \lambda \|L(u)\|^2\} \quad (2.37)$$

On obtient donc ici une famille de solutions \hat{u}_λ paramétrée par le coefficient de régularisation λ . Si ce paramètre est très grand, l'effet du terme $\|y - Hu\|^2$ est négligeable devant celui du terme $\|L(u)\|^2$. On néglige les données mesurées et on tient compte en priorité de la connaissance a priori sur le problème. La solution ainsi obtenue peut être correcte du point de vue de la minimisation du critère mais en inadéquation avec les données mesurées. Il faut par ailleurs être certain de l'exactitude des connaissances a priori dont on dispose. C'est dans ce cas l'expertise de l'utilisateur qui est requise.

D'autre part, si le paramètre de régularisation est trop faible, on néglige la connaissance a priori en donnant une plus grande importance aux grandeurs mesurées. Cependant, plus la valeur du paramètre de régularisation diminue et plus on se rapproche d'un éventuel mauvais conditionnement du problème. Il ne faut pas perdre de vue que le second terme a été ajouté afin de transformer le problème mal posé en un problème bien posé.

Afin de calculer analytiquement la solution du problème, on dérive le critère (2.36) par rapport à la grandeur à reconstruire :

$$\frac{\partial J}{\partial u} = 2\lambda L(u) \frac{\partial L(u)}{\partial u} - 2H^T(y - Hu) \quad (2.38)$$

Ceci conduit à calculer la solution de l'équation :

$$2\lambda L(u) \frac{\partial L(u)}{\partial u} - 2H^T(y - Hu) = 0 \quad (2.39)$$

On voit bien dans ce cas que si l'on pose $\lambda = 0$, on retrouve la solution des moindres carrés. Si λ n'est pas nul, le terme additionnel $\lambda L(u) \frac{\partial L(u)}{\partial u}$ modifie les valeurs propres de la matrice $H^T H$ seule. A partir du moment où la matrice $H^T H + \lambda L(u) \frac{\partial L(u)}{\partial u}$ n'est plus singulière, le problème a une solution unique.

b) Choix du coefficient de régularisation

Dans de nombreux problèmes inverses, un réglage précis du coefficient de régularisation λ est inutile, la solution \hat{u}_λ n'étant sensible qu'à des variations importantes de λ . Il est alors possible, avec un peu d'expérience, de choisir empiriquement λ . Dans le cas contraire, il existe quelques méthodes de réglage automatique à partir des données mesurées.

i) Contrôle de l'énergie de l'erreur résiduelle

Une des idées les plus intuitives et les plus anciennes pour régler la valeur de λ est de considérer ce coefficient comme un multiplicateur de Lagrange [Demoment *et al.*]. Reprenons l'expression générale du critère pénalisé :

$$J(u) = \Delta(y, Hu) + \Omega(u) \quad (2.40)$$

Dans le cadre de la régularisation de Tikhonov, ce critère peut s'exprimer sous la forme :

$$J(u) = \|y - Hu\|^2 + \lambda \|L(u)\|^2 \quad (2.41)$$

avec ici $\Delta(y, Hu) = \|y - Hu\|^2$ et $\Omega(u) = \lambda \|L(u)\|^2$.

Si l'on considère λ comme un multiplicateur de Lagrange, le problème de régularisation est équivalent à l'énoncé suivant :

$$\hat{u}_\lambda = \arg \min_u \{\Omega(u)\} \text{ sous la contrainte } \Delta(y, Hu) = c \quad (2.42)$$

Le degré de régularisation est fixé par la valeur de c . Plus la valeur de c est faible, plus on accorde d'importance aux données mesurées y car la condition $\Delta(y, Hu) = c$ devient davantage contraignante. Par conséquent, la notion de compromis entre fidélité aux données mesurées et degré de pénalisation de la solution du problème apparaît déjà dans ce

contexte.

Lorsque le terme Δ est quadratique, on préconise souvent de choisir c égal à la dimension de la grandeur u à reconstruire.

ii) Méthode de la courbe en "L"

On peut également utiliser une méthode alternative qui a fait ses preuves dans des problèmes inverses de la forme

$$J(u) = \|y - H\hat{u}_\lambda\|^2 + \lambda\Omega(u) \quad (2.43)$$

et dans le cas précis où $\Omega(u)$ est quadratique : la méthode de la courbe en L [Hansen, 2001].

Cette méthode consiste à tracer en échelle logarithmique le terme de régularisation $\Omega(\hat{u}_\lambda)$ en fonction du critère des moindres carrés $\|y - H\hat{u}_\lambda\|^2$ en faisant varier le coefficient de régularisation λ .

Cette courbe a en général l'allure caractéristique d'un L et la valeur de λ correspondant à l'angle de ce L fournit un bon compromis entre les exigences contradictoires de fidélité aux grandeurs mesurées et de fidélité aux connaissances a priori.

On remarque que toutes les méthodes de choix du coefficient de régularisation λ exposées ici n'ont de justification claire que dans le cas de critères quadratiques. De même que pour le choix du terme de régularisation $\Omega(u)$, les justifications sont données de manière empirique. Chaque solution de régularisation est à considérer selon la nature même du problème inverse traité.

2.2.4 Méthodes probabilistes

Une autre classe de méthodes de résolution des problèmes inverses est la classe des méthodes probabilistes. L'approche générale de ces méthodes consiste d'abord à prendre en compte explicitement le plus grand nombre d'erreurs possible (erreur de mesure, de modélisation et, éventuellement, de discrétisation du problème), ensuite à prendre en compte les informations *a priori* dont on peut disposer et, enfin, une fois la solution du problème inverse estimée, à caractériser l'incertitude même sur cette solution.

Un outil mathématique très utile pour représenter des erreurs de toute sorte est la théorie des probabilités. En effet, une loi de probabilité permet de caractériser l'incertitude sur les mesures et le manque de connaissance sur une grandeur.

Parmi les méthodes probabilistes de résolution des problèmes inverses du type $y = Hu$, on distingue :

- Les méthodes utilisant les moments : on cherche à lier les moments (d'ordre 1 et 2) des grandeurs mesurées y aux moments de la grandeur à calculer u . On obtient ainsi un lien entre les deux grandeurs qu'il suffit d'inverser pour calculer u .
- Les méthodes statistiques classiques utilisant le maximum de vraisemblance : on prend en compte explicitement le caractère incertain des mesures en le caractérisant par une loi de probabilité conditionnelle $p(y | u)$. On considère ensuite $p(y | u)$ comme une fonction de la grandeur u que l'on nomme *fonction de vraisemblance* $V(u)$. On définit alors un critère basé sur cette fonction, par exemple le maximum de vraisemblance, qui permet d'obtenir une estimation de u .
- Les méthodes basées sur *l'entropie* : le principe ressemble à celui du maximum de vraisemblance. On définit cette fois la fonction entropie $S(u)$ que l'on va essayer de maximiser sous certaines contraintes correspondant au lien entre u et y .
- L'approche statistique bayésienne où non seulement on prend en compte l'incertitude sur les mesures de la grandeur y mais on attribue également une loi de probabilité à la grandeur u qui traduit l'information a priori dont on dispose. On peut ainsi combiner ces deux informations tout comme on le fait dans les méthodes de régularisation vues précédemment.

2.2.4.1 Utilisation des moments

Cette approche considère que les grandeurs observée y et mesurée u sont des fonctions aléatoires [Mohammad-Djafari, 1999b]. On se limite donc à caractériser les lois de probabilités de ces grandeurs par leurs moments jusqu'à l'ordre deux. Il suffit ensuite d'établir un lien entre les moments de la grandeur observée et ceux de la grandeur inconnue à reconstruire, puis d'essayer d'inverser ce lien.

Nous allons prendre l'exemple classique du filtrage optimal de Wiener appliqué au problème de la déconvolution des signaux.

On considère le problème suivant (cf figure 2.1) :

$$y(t) = h(t) * u(t) + b(t) \quad (2.44)$$

où les fonctions $y(t)$, $u(t)$ et $b(t)$ sont aléatoires et représentent respectivement la grandeur mesurée, la grandeur inconnue à reconstruire et le bruit.

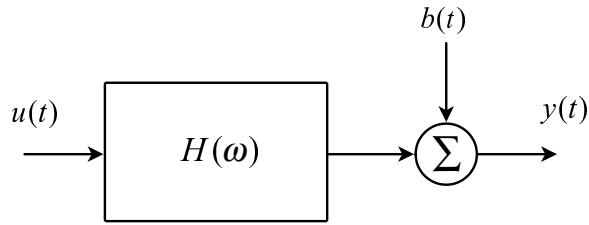


Fig. 2.1 - Modèle d'observation

Les moments d'ordre 1 de ces grandeurs sont leur moyenne (ou espérance mathématique) :

$$E[y(t)], \quad E[u(t)] \text{ et } E[b(t)] \quad (2.45)$$

Rappelons que la moyenne $E[x(t)]$ d'une grandeur $x(t)$ est calculée à chaque instant t . Si cette moyenne ne dépend pas de t , le signal est dit stationnaire au premier ordre. Si le signal est "ergodique" (ses moyennes statistiques sont égales à ses moyennes temporelles, propriété généralement admise), on peut remplacer le calcul sur différentes réalisations à un instant t donné par une moyenne temporelle.

$$E[x(t)] = \frac{1}{2T} \lim_{t \rightarrow \infty} \int_{-T}^T x(t) dt \quad (2.46)$$

Les moments d'ordre 2 de ces grandeurs sont leurs fonctions d'autocorrélation temporelle ($R_{yy}(\tau)$ et $R_{uu}(\tau)$) et leurs fonctions d'intercorrélation temporelle ($R_{bu}(\tau)$ et $R_{yu}(\tau)$) définis de la manière suivante :

$$\begin{aligned} R_{yy}(\tau) &= E[y(t)y(t + \tau)] \\ R_{uu}(\tau) &= E[u(t)u(t + \tau)] \\ R_{bu}(\tau) &= R_{ub}(-\tau) = E[b(t)u(t + \tau)] \\ R_{yu}(\tau) &= R_{uy}(-\tau) = E[y(t)u(t + \tau)] \end{aligned} \quad (2.47)$$

On fait l'hypothèse que $u(t)$ et $b(t)$ sont indépendants (non corrélés), ce qui nous permet d'obtenir les relations suivantes :

$$\begin{aligned} E[y(t)] &= h(t) * E[u(t)] + E[b(t)] \\ R_{yy}(\tau) &= h(t) * h(t) * R_{uu}(\tau) + R_{bb}(\tau) \\ R_{yu}(\tau) &= h(t) * R_{uu}(\tau) \end{aligned} \quad (2.48)$$

La transformée de Fourier de certaines des précédentes relations donne :

$$\begin{aligned} S_{yy}(\omega) &= |H(\omega)|^2 S_{uu}(\omega) + S_{bb}(\omega) \\ S_{yu}(\omega) &= H(\omega) S_{uu}(\omega) \\ S_{uy}(\omega) &= H^*(\omega) S_{uu}(\omega) \end{aligned} \quad (2.49)$$

L'objectif du filtrage optimal est de fournir une solution $\hat{u}(t)$ obtenue par filtrage linéaire de $y(t)$:

$$\hat{u}(t) = w(t) * y(t) \quad (2.50)$$

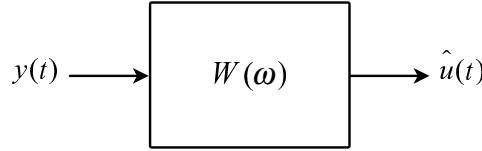


Fig. 2.2 - Déconvolution par filtrage de Wiener

Il s'agit donc d'obtenir la réponse impulsionnelle du filtre $w(t)$ telle que l'erreur quadratique moyenne $E[(u(t) - \hat{u}(t))^2]$ soit minimale. Après diverses manipulations mathématiques, on obtient l'expression de la transformée de Fourier de $w(t)$:

$$W(\omega) = \frac{H^*(\omega)S_{uu}(\omega)}{|H(\omega)|^2 S_{uu}(\omega) + S_{bb}(\omega)} = \frac{1}{H(\omega)} \frac{|H(\omega)|^2}{|H(\omega)|^2 + \frac{S_{bb}(\omega)}{S_{uu}(\omega)}} \quad (2.51)$$

Un autre problème réside en la détermination de $S_{yy}(\omega)$ et $S_{uy}(\omega)$, qui nécessite la connaissance a priori de $S_{uu}(\omega)$ et de $S_{bb}(\omega)$. On peut faire l'hypothèse que le rapport de ces deux dernières grandeurs est égal à l'inverse du rapport signal sur bruit (hypothèse peu réaliste) ou bien au carré du module de la fonction de transfert d'un filtre passe-haut.

Une fois obtenue l'expression de $W(\omega)$, on peut estimer la solution $\hat{u}(t)$ du problème inverse en calculant l'expression temporelle du filtre de Wiener par transformée de Fourier inverse.

2.2.4.2 Critère du maximum de vraisemblance

La méthode du maximum de vraisemblance repose sur une idée de base assez intuitive : la meilleure estimée d'une grandeur à reconstruire u est celle qui permet d'obtenir la plus grande probabilité d'obtention de la grandeur mesurée y . Cette méthode prend donc en compte explicitement le caractère incertain de la grandeur mesurée en la caractérisant par une loi de probabilité $p(y | u)$ [Demoment *et al.*].

On considère ensuite cette loi de probabilité comme une fonction $V(u) = p(y | u)$ appelée fonction vraisemblance.

La solution \hat{u} du problème inverse maximise cette fonction vraisemblance :

$$\hat{u} = \arg \max_u \{p(y | u)\} \quad (2.52)$$

Pour des raisons calculatoires (nous verrons plus tard que la loi de probabilité $p(y | u)$ s'exprime souvent sous la forme d'une exponentielle), il est souvent préférable de maximiser le logarithme népérien de la fonction vraisemblance :

$$\hat{u} = \arg \max_u \{\ln p(y | u)\} \quad (2.53)$$

Le logarithme népérien étant une fonction monotone, le maximum de $\ln p(y | u)$ coïncide avec le maximum de $p(y | u)$.

On note qu'il est évidemment possible d'obtenir la solution \hat{u} du problème inverse en estimant non pas le maximum du logarithme népérien mais le minimum de son opposé :

$$\hat{u} = \arg \min_u \{-\ln p(y | u)\} \quad (2.54)$$

Il est en effet parfois plus simple de minimiser une grandeur plutôt que de la maximiser.

L'estimateur du maximum de vraisemblance possède quelques propriétés intéressantes : c'est un estimateur à variance minimale (on parle d'efficacité de cet estimateur) et si le nombre de mesures de la grandeur observable y augmente alors l'estimateur converge asymptotiquement vers la solution du problème inverse.

Nous allons maintenant nous intéresser à la forme que peut prendre la loi de probabilité $p(y | u)$.

a) Loi gaussienne

On considère le modèle général $y = H(u) + b$ et on suppose que le bruit b peut être modélisé par un vecteur aléatoire centré, blanc et gaussien. On le représente alors par une loi normale (ou de Gauss) $N(0, \sigma_b^2 I)$ d'espérance nulle et de variance σ_b^2 .

On en déduit alors la loi de probabilité :

$$p(y | u) = K \exp \left(\frac{-1}{2\sigma_b^2} [y - H(u)]^T [y - H(u)] \right) \quad (2.55)$$

où K est une constante non nulle.

On peut donc calculer une estimation de la grandeur u au sens du maximum de vraisemblance :

$$\hat{u} = \arg \min_u \{-\ln p(y | u)\} \quad (2.56)$$

On obtient l'expression suivante :

$$\hat{u} = \arg \min_u \{\|y - H(u)\|^2\} \quad (2.57)$$

On remarque que, dans ce cas précis, on aboutit à l'estimateur des moindres carrés.

b) Loi gaussienne généralisée

On suppose maintenant que chaque mesure b_i du bruit b obéit à une loi gaussienne généralisée :

$$p(b_i) = A \exp [-\beta |b_i|^p], \quad \beta > 0, \quad 1 \leq p \leq 2 \quad (2.58)$$

où A est une constante.

Si on fait l'hypothèse que b est un bruit blanc, l'estimation de la grandeur u au sens du maximum de vraisemblance a pour expression analytique :

$$\hat{u} = \arg \min_u \left\{ \sum_{i=1}^n |y_i - [H(u)]_i|^p \right\} \quad (2.59)$$

On peut écrire cette expression sous la forme :

$$\hat{u} = \arg \min_u \{ \|y - H(u)\|^p \} \quad (2.60)$$

Lorsque $p = 2$, on retrouve à nouveau l'estimation au sens des moindres carrés.

c) Loi Gamma

A présent, voyons le cas où chaque mesure b_i du bruit b obéit à une loi gamma :

$$p(b_i) = K_i b_i^{-\alpha} \exp (-\beta b_i) \quad (2.61)$$

On fait également l'hypothèse que le bruit est blanc. On en déduit la loi gamma associée au logarithme népérien de $p(y | u)$:

$$\ln p(y | u) = K + \sum_{i=1}^n \alpha \log(y_i - [H(u)]_i) + \beta(y_i - [H(u)]_i) \quad (2.62)$$

La solution $\hat{u} = \arg \min_u \{-\ln p(y | u)\}$ n'a pas d'expression analytique dans ce cas. Elle peut toutefois être calculée par un algorithme itératif.

En conclusion, un des défauts de l'approche du maximum de vraisemblance est qu'elle fournit des résultats rarement satisfaisants pour la résolution de problèmes inverses dans le cadre algébrique où le nombres de grandeurs à reconstruire est du même ordre de grandeur, voire plus grand, que le nombre de mesures. Une solution est l'approche du *maximum de vraisemblance pénalisé* qui consiste à définir la solution \hat{u} du problème inverse de la manière suivante :

$$\hat{u} = \arg \min_u \{ -\ln p(y | u) + \phi(u) \} \quad (2.63)$$

où $\phi(u)$ est une fonction de pénalisation qui permet d'obtenir une solution beaucoup plus satisfaisante. Ce principe de pénalisation est dans l'idée semblable à celui de régularisation vu dans la section précédente.

2.2.4.3 Critère du maximum d'entropie

a) Définitions

La notion d'entropie peut être introduite de la manière suivante [Mohammad-Djafari, 1999b], [Mohammad-Djafari, 1999a] : plus un événement est rare et plus le gain d'information obtenu par sa réalisation est grand. On considère donc une variable aléatoire discrète X produisant des réalisations $x = \{x_1, x_2, \dots, x_n\}$ auxquelles on attribue les probabilités $p = \{p_1, p_2, \dots, p_n\}$ pour représenter une information partielle sur la variable X . On définit ensuite la quantité

$$I_i = \ln \left(\frac{1}{p_i} \right) \quad (2.64)$$

comme étant la quantité d'information apportée par la réalisation x_i de la variable X . L'utilisation du logarithme rend additif le gain total d'information obtenu par la réalisation de plusieurs événements indépendants.

On définit alors l'entropie $S(p)$ d'un processus par la somme pondérée des informations apportées par chaque réalisation (définition de l'entropie donnée par Shannon) :

$$S(p) = \sum_{i=1}^n p_i \ln \left(\frac{1}{p_i} \right) = - \sum_{i=1}^n p_i \ln p_i \quad (2.65)$$

L'entropie S peut être vue comme une mesure d'incertitude de la distribution $p = \{p_1, p_2, \dots, p_n\}$.

Ce concept peut être généralisé en définissant le terme $-\ln(p_i/q_i)$ comme étant le gain d'information, sur une probabilité *a priori* q_i , apporté par la connaissance de la probabilité p_i de réalisation d'un événement x_i . On définit alors :

$$S(p, q) = \sum_{i=1}^n p_i \ln \left(\frac{p_i}{q_i} \right) \quad (2.66)$$

comme étant l'entropie croisée (ou relative) de la distribution $p = \{p_1, p_2, \dots, p_n\}$ par rapport à la distribution a priori $q = \{q_1, q_2, \dots, q_n\}$.

Moyennant quelques précautions de calcul, on peut généraliser ces définitions au cas continu et définir l'entropie par :

$$S(p) = - \int p(x) \ln p(x) dx \quad (2.67)$$

et l'entropie croisée par

$$S(p, q) = \int p(x) \ln \frac{p(x)}{q(x)} dx \quad (2.68)$$

On définit maintenant la meilleure estimée $\hat{\Phi}$ d'une fonction $\Phi(X)$ de la variable aléatoire discrète X au sens du minimum de l'erreur quadratique moyenne :

$$\hat{\Phi} = E[\Phi(X)] = \sum_{i=1}^n p_i \Phi(x_i) \quad (2.69)$$

Il s'agit ici d'un problème direct et bien posé. Mais il existe en général beaucoup de distributions p qui satisfont cet ensemble de contraintes. Le problème inverse considéré est donc mal posé au sens de la non-unicité de sa solution. Le principe du maximum d'entropie permet alors de choisir une solution : la meilleure solution sera celle qui a l'entropie maximale (c'est-à-dire celle qui satisfait toutes les contraintes) et qui est la moins compromettante vis-à-vis de toute autre information inconnue (information apportée par aucune des réalisations de la variable aléatoire).

Le problème inverse se formule donc ainsi :

$$\begin{aligned} &\text{maximiser } S(p) = - \sum_{i=1}^n p_i \ln p_i \\ &\text{sous les contraintes } \sum_{i=1}^n p_i \Phi_k(x_i) = d_k, \quad k = 1, \dots, m \end{aligned} \quad (2.70)$$

Dans le cas de l'entropie croisée, le problème se formule ainsi : étant donnée une distribution de probabilités a priori $q = \{q_1, q_2, \dots, q_n\}$, il s'agit de déterminer la distribution de probabilités a posteriori $p = \{p_1, p_2, \dots, p_n\}$ qui minimise le terme

$$S(p, q) = \sum_{i=1}^n p_i \ln \left(\frac{p_i}{q_i} \right) \quad (2.71)$$

et qui satisfait les contraintes

$$\sum_{i=1}^n p_i \Phi_k(x_i) = d_k, \quad k = 1, \dots, m \quad (2.72)$$

Ces deux définitions de problèmes inverses peuvent être étendues au cas continu. Le premier problème (2.70) est défini en continu par :

$$\begin{aligned} &\text{maximiser } S(p) = - \int p(x) \ln p(x) dx \\ &\text{sous les contraintes } \int \Phi_k(x) p(x) dx = d_k, \quad k = 1, \dots, m \end{aligned} \quad (2.73)$$

Dans le cas de l'entropie croisée, le deuxième problème (2.71) est défini ainsi : étant donnée une densité de probabilité a priori $p(x)$, il s'agit de déterminer la densité de probabilité a posteriori $q(x)$ qui minimise l'entropie croisée

$$S(p, q) = \int p(x) \ln \frac{p(x)}{q(x)} dx \quad (2.74)$$

et qui satisfait les m contraintes

$$\int \Phi_k(x)p(x)dx = d_k, \quad k = 1, \dots, m \quad (2.75)$$

b) Application à des problèmes inverses

L'approche de base (maximum d'entropie classique) consiste à considérer la grandeur à reconstruire u comme une distribution de probabilités et les données mesurées y comme un ensemble de contraintes sur cette distribution de probabilités. En général, ces contraintes ne sont pas suffisantes pour définir une solution unique au problème inverse.

On utilise donc le principe du maximum d'entropie pour choisir une solution parmi l'ensemble des solutions admissibles défini par :

$$\{u : J = \|y - H(u)\| \leq J_0\} \quad (2.76)$$

On choisit dans cet ensemble la solution qui maximise l'entropie $S(u)$ définie par :

$$S(u) = - \sum_{j=1}^n u_j \ln u_j \quad (2.77)$$

On peut définir l'entropie de manière plus générale :

$$S(u) = - \sum_{j=1}^n \left[u_j \ln \left(\frac{u_j}{m_j} \right) + (u_j - m_j) \right] \quad (2.78)$$

où m est une solution par défaut (a priori) que l'on considère également comme une distribution de probabilités.

2.2.4.4 Approche bayésienne des problèmes inverses

L'approche bayésienne pour la résolution de problèmes inverses permet de prendre en compte et de traiter de la même manière l'information a priori sur la grandeur à reconstruire u et sur les grandeurs mesurées y [Mohammad-Djafari, 1998], [Mohammad-Djafari, 1999b]. On tient compte non seulement de l'incertitude des mesures en leur attribuant une loi de probabilité mais également de l'incertitude concernant l'information a priori au travers d'une loi de probabilité a priori que l'on attribue à la grandeur inconnue du problème. La règle de Bayes permet de tenir compte en les fusionnant de ces deux sources d'information. Ainsi, le cadre bayésien offre les réponses les plus cohérentes et les plus complètes à des problèmes laissés ouverts dans les autres approches. D'une certaine manière, l'approche bayésienne généralise la régularisation déterministe.

a) Introduction

La démarche correspondant à l'approche bayésienne peut se scinder en différentes étapes. La toute première de ces étapes consiste bien sûr à expliciter le problème lui-même. On définit un modèle H reliant la grandeur mesurée y et la grandeur à reconstruire u :

$$y = H(u) + b \quad (2.79)$$

Le terme b étant le bruit qui entache inévitablement y .

On peut ensuite définir de manière explicite les connaissances *a priori* dont on dispose (et si on peut en disposer) sur u et sur le bruit de mesure b . Cette étape est cruciale pour la suite du problème car la qualité de l'estimation \hat{u} de u dépend directement de la qualité (et parfois de la quantité) des informations que l'on aura pu rassembler.

La deuxième étape de la démarche consiste en l'attribution d'une loi de probabilité *a priori* $p(u, \theta_u)$ à la grandeur inconnue du problème afin de traduire la connaissance dont on dispose sur cette grandeur avant même d'essayer de l'estimer. Cette loi de probabilités peut dépendre d'un vecteur de paramètres θ_u .

On attribue également une loi de probabilité conditionnelle $p(y | u, \theta_y)$ à y afin de traduire l'incertitude sur les mesures. Ces incertitudes sont dues en partie au manque de précision de l'appareil de mesure et au bruit environnant qui s'ajoute aux mesures. De même que la précédente, cette loi de probabilité peut dépendre d'un vecteur de paramètres θ_y . L'ensemble $\theta = [\theta_u, \theta_y]$ est appelé vecteur des hyperparamètres du problème.

La troisième étape de la démarche consiste à utiliser la règle de Bayes pour combiner les deux sources d'information : l'information contenue dans y et celles contenues dans la loi de probabilité *a priori* sur u . On calcule ainsi une loi de probabilité *a posteriori* :

$$p(u | y, \theta) = \frac{p(y | u, \theta_y)p(u, \theta_u)}{p(y, \theta)} \quad (2.80)$$

avec

$$p(y, \theta) = \int p(y | u, \theta_y)p(u, \theta_u)du \quad (2.81)$$

Cette loi de probabilité $p(u | y, \theta)$ dépend des hyperparamètres du problème et contient toute l'information disponible sur la grandeur inconnue u . Elle permet donc de choisir la meilleure estimée \hat{u} de u suivant un critère prédéfini. Une fois cette solution choisie, il est impératif de lui attribuer un degré de confiance et de déterminer sa sensibilité vis-à-vis des erreurs de mesure et de modélisation. L'approche bayésienne nous permet de répondre à ces attentes car on dispose de la loi de probabilité *a posteriori* sur u qui permet de caractériser l'erreur d'estimation.

Tous les problèmes de cette approche résident dans le choix des lois de probabilités *a priori* $p(u, \theta_u)$, $p(y | u, \theta_y)$ et des hyperparamètres θ , ainsi que dans le calcul effectif de la loi de probabilité *a posteriori* $p(u | y, \theta)$.

b) Un exemple simple : le cas linéaire gaussien

Afin de mieux comprendre la démarche exposée en introduction, nous allons l'appliquer à un cas simple : celui où le modèle reliant u et y est linéaire et où l'on peut attribuer des lois de probabilités gaussiennes à u et y . Le problème est donc le suivant :

$$y = Hu + b \quad (2.82)$$

où b représente le bruit de mesure (bruit additif dans ce cas particulier). La matrice H est une matrice de dimension $[m \times n]$.

Faisons d'abord l'hypothèse que l'on connaît de la grandeur u *a priori* sa moyenne $E[u] = u_0$ et sa matrice de covariance $E[(u - u_0)(u - u_0)^T] = R_u = \sigma_u^2 P_0$. On peut alors en déduire que la loi de probabilité attribuée à u est une loi gaussienne :

$$p(u) = A \exp \left[\frac{-1}{2} (u - u_0) R_u^{-1} (u - u_0)^T \right] \quad (2.83)$$

que l'on peut aussi écrire

$$p(u) = A \exp \left[\frac{-1}{2\sigma_u^2} (u - u_0) P_0^{-1} (u - u_0)^T \right] \quad (2.84)$$

avec

$$A = (2\pi)^{\frac{-n}{2}} |R_u|^{\frac{-1}{2}} = (2\pi\sigma_u^2)^{\frac{-n}{2}} |P_0|^{\frac{-1}{2}} \quad (2.85)$$

On suppose ensuite que la valeur moyenne $E[b]$ du bruit est nulle et que sa matrice de covariance est définie par $E[bb^T] = R_b = \sigma_b^2 I$. Cela signifie bien sûr que le bruit est blanc et non corrélé et qu'il n'y a pas d'erreur systématique lorsqu'on le mesure. On peut donc attribuer au bruit b une loi gaussienne :

$$p(b) = A_1 \exp \left[\frac{-1}{2\sigma_b^2} uu^T \right] \quad (2.86)$$

avec

$$A_1 = (2\pi)^{\frac{-n}{2}} |R_b|^{\frac{-1}{2}} = (2\pi\sigma_b^2)^{\frac{-n}{2}} \quad (2.87)$$

En faisant de plus l'hypothèse que le bruit b est indépendant de la grandeur inconnue u , on peut déduire la loi de probabilité conditionnelle :

$$p(y | u) = B \exp \left[\frac{-1}{2} (y - Hu)^T R_b^{-1} (y - Hu) \right] \quad (2.88)$$

que l'on peut également écrire

$$p(y | u) = B \exp \left[\frac{-1}{2\sigma_b^2} (y - Hu)^T (y - Hu) \right] \quad (2.89)$$

avec

$$B = (2\pi)^{-\frac{m}{2}} |R_b|^{-\frac{1}{2}} = (2\pi\sigma_b^2)^{-\frac{m}{2}} \quad (2.90)$$

On peut maintenant utiliser la règle de Bayes :

$$p(u | y) = \frac{p(y | u)p(u)}{p(y)} \quad (2.91)$$

et on obtient au final :

$$p(u | y) = C \exp \left[\frac{-1}{2} J(u) \right] \quad (2.92)$$

avec

$$\begin{aligned} J(u) &= (y - Hu)^T R_b^{-1} (y - Hu) + (u - u_0)^T R_u^{-1} (u - u_0) \\ &= \frac{1}{\sigma_b^2} [(y - Hu)^T (y - Hu) + \lambda (u - u_0)^T P_0^{-1} (u - u_0)] \end{aligned} \quad (2.93)$$

où $\lambda = \frac{\sigma_b^2}{\sigma_u^2}$.

On peut montrer a posteriori que la loi de probabilité $p(u | y)$ est gaussienne de valeur moyenne \hat{u} et de matrice de covariance \hat{P} :

$$\begin{cases} \hat{u} = (H^T H + \lambda P_0^{-1})^{-1} H^T y \\ \hat{P} = \sigma_b^2 (H^T H + \lambda P_0^{-1})^{-1} \end{cases} \quad (2.94)$$

On a ainsi obtenu une loi de probabilité a posteriori et l'on peut en déduire une estimée \hat{u} qui n'est autre que la valeur moyenne a posteriori et qui peut aussi être calculée par

$$\hat{u} = \arg \max_u \{p(u | y)\} = \arg \min_u \{J(u) = Q(u) + \lambda \Omega(u)\} \quad (2.95)$$

avec

$$\begin{cases} Q(u) = (y - Hu)^T (y - Hu) \\ \Omega(u) = (u - u_0)^T P_0^{-1} (u - u_0) \end{cases} \quad (2.96)$$

On peut aussi remarquer que si l'on pose $P_0^{-1} = D^T D$, on obtient :

$$J(u) = \|y - Hu\|^2 + \lambda \|D(u - u_0)\|^2 \quad (2.97)$$

On retrouve donc la notion de solution régularisée et de régularisation quadratique avec cependant quelques différences dans le choix des distances et du coefficient de régularisation.

Dans l'approche de régularisation déterministe, le choix des distances $\Delta_1(y, Hu)$ et $\Delta_2(u, u_0)$ était arbitraire alors que l'approche bayésienne permet de définir rigoureusement ces termes comme étant les conséquences des hypothèses faites respectivement sur la loi de probabilité attribuée au bruit b et sur la loi de probabilité a priori attribuée à u .

Dans l'approche de régularisation déterministe, le coefficient de régularisation λ est déterminé de manière empirique alors que l'approche bayésienne le définit comme étant le rapport de la variance du bruit b par la variance de u . Si l'on connaît ces deux variances, la valeur de λ est déterminée mais ce n'est bien sûr pas le cas en pratique. Par contre, cette expression indique bien que plus la variance du bruit est importante, plus la valeur de λ doit être grande pour obtenir un résultat satisfaisant.

L'approche bayésienne permet de quantifier le degré de confiance que l'on attribue à la solution estimée \hat{u} en offrant la possibilité de calculer la matrice de covariance a posteriori $E[(u - \hat{u})^2]$ de la solution.

c) Choix de la loi de probabilité *a priori*

Le problème de la conversion d'une information a priori en une loi de probabilité est un problème encore largement ouvert. La principale difficulté est qu'une information a priori se présente rarement directement sous une forme probabiliste. On peut par exemple savoir par avance que la grandeur u à reconstruire est positive ou bien bornée entre 0 et 1. Il faut ensuite pouvoir construire des lois de probabilité qui puissent incorporer une ou plusieurs de ces informations.

Les méthodes existantes peuvent être regroupées en trois grandes classes.

Certaines reposent sur la théorie des *groupes de transformation* pour déterminer la mesure de référence "naturelle" pour la grandeur à reconstruire. On utilise ce genre de méthode lorsqu'on sait peu de choses, c'est-à-dire lorsque l'information a priori se limite à une connaissance qualitative sur la nature de la grandeur à reconstruire.

D'autres méthodes reposent sur des principes informationnels. Il s'agit principalement des méthodes dites "à maximum d'entropie" dans lesquelles on recherche une distribution qui soit la plus proche (au sens d'une distance) d'une distribution de référence (souvent choisie par l'approche précédente) tout en vérifiant une information incomplète connue a priori sous la forme de contraintes sur la loi recherchée. Cette approche n'est véritablement applicable que lorsque l'information a priori est faite de contraintes linéaires sur la distribution recherchée.

Il existe enfin une dernière classe de méthodes, celle des constructions faites "à la main". Elles permettent d'incorporer dans une distribution a priori des propriétés locales essentielles que doit posséder la grandeur à reconstruire. Ces méthodes de construction reposent en grande partie sur l'expérience personnelle et demandent beaucoup de savoir-faire.

d) Calcul de la loi de probabilité *a posteriori*

Nous avons vu que dans le cas d'un modèle linéaire gaussien H , la loi de probabilité a posteriori est également gaussienne et donc définie par ses deux premiers moments (moyenne et matrice de covariance). On dispose donc d'expressions analytiques qui nous permettent de calculer l'estimée \hat{u} de la grandeur u et de quantifier le degré de confiance que l'on peut lui attribuer.

Dans le cas général, le calcul analytique de la loi de probabilité a posteriori $p(u | y)$ peut devenir plus délicat et on doit souvent se contenter de l'approximer par une loi gaussienne ou bien de se limiter au calcul de sa moyenne et de sa matrice de covariance (quand c'est possible). On peut également essayer de définir un estimateur ponctuel à partir de cette loi, comme le maximum a posteriori :

$$\hat{u} = \arg \max_u \{p(u | y)\} = \arg \min_u \{-\ln p(u | y)\} \quad (2.98)$$

ou bien la moyenne a posteriori

$$\hat{u} = \int u p(u | y) du \quad (2.99)$$

Le calcul du maximum a posteriori est plus souvent utilisé car il ne nécessite qu'une optimisation alors que le calcul de la moyenne a posteriori fait intervenir la résolution d'une intégrale de dimension élevée. Dans le cas du modèle linéaire gaussien, cela ne pose pas de problème car ces deux calculs aboutissent au même résultat. Mais ils peuvent fournir des résultats très différents dans le cadre général. Il convient donc d'être prudent dans les hypothèses de départ et dans les approximations faites ensuite pour la loi de probabilité.

2.2.5 Conclusion

Les différentes approches évoquées dans cette partie permettent la résolution de nombreux problèmes inverses bien ou mal posés. Pour des raisons de clarté et de simplicité, nous n'avons évoqué jusqu'à maintenant que des approches d'inversion destinées à résoudre des problèmes inverses dans le cadre linéaire. Cependant, la majeure partie des systèmes naturels étant non linéaires, les méthodes d'inversion employées dans le cas de systèmes linéaires doivent être étendues dans ce cadre plus général. De telles démarches de généralisation des méthodes d'inversion à des problèmes non linéaires constituent encore de nos jours une partie importante de la recherche fondamentale.

2.3 Inversion du modèle de Volterra par régularisation de Tikhonov

2.3.1 Introduction

Nous avons vu qu'une méthode d'inversion bien connue et couramment employée est la régularisation de Tikhonov [Tikhonov et Arsenine, 1976], [Chavent et Kunisch, 1994],

[Johansen, 1997]. Dans le cas d'un modèle linéaire $y = Hu$, cette méthode consiste à minimiser un critère composite $J(u)$ (2.36). Le propos de cette partie est d'adapter cette technique d'inversion en l'appliquant à un modèle non linéaire : le modèle de Volterra décrit dans le premier chapitre de ce mémoire [Bibes *et al.*, 2003c], [Bibes *et al.*, 2004]. Cette approche d'inversion du modèle de Volterra a déjà été proposée par [Inglada, 2000] dans le cadre théorique. Cependant, le modèle obtenu est quasi-linéaire et l'estimation de la grandeur inconnue revient à une inversion de filtre linéaire. Les différents systèmes étudiés dans notre travail de thèse possèdent des non-linéarités importantes. Les méthodes d'inversion proposées seront donc appliquées à des systèmes aux non-linéarités avérées, participant ainsi à l'originalité de notre approche.

2.3.2 Principe de la méthode

2.3.2.1 Modèle de Volterra

Reprendons la série de Volterra décrite dans le premier chapitre :

$$y(t) = \sum_{i=1}^{\infty} \left(\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i u(t - \tau_j) d\tau_j \right) \quad (2.100)$$

Cette série peut s'écrire de manière plus synthétique :

$$y(t) = H[u(t)] \quad (2.101)$$

Si la (ou les) non-linéarité(s) du système considéré sont relativement douces, la série de Volterra peut être tronquée à ses premiers termes et utilisée pour modéliser la sortie $y(t)$ du système. Le modèle $\hat{y}(t)$ de la sortie s'écrit :

$$\hat{y}(t) = \sum_{i=1}^n \left(\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i u(t - \tau_j) d\tau_j \right) \quad (2.102)$$

On notera donc :

$$\hat{y}(t) = \hat{H}[u(t)] = \sum_{i=1}^n H_i[u(t)] \quad (2.103)$$

Nous avons décrit deux manières de modéliser les noyaux de Volterra en les développant soit sur des fonctions de transfert, soit sur une base de fonctions orthonormées multidimensionnelles $\{\Phi_{m_1 \dots m_i}\}$.

Dans toute la suite de cette partie, c'est cette dernière méthode de modélisation que nous utiliserons. En ce qui concerne les fonctions de la base, le choix le plus judicieux est celui des fonctions orthonormées généralisées unidimensionnelles définies de la manière suivante :

$$\Phi_n(s) = \frac{\sqrt{2\operatorname{Re}\{p_n\}}}{s + p_n} \prod_{k=1}^{n-1} \frac{s - \overline{p_k}}{s + p_k} \quad (2.104)$$

Chaque terme $H_i[u(t)]$ sera modélisé ainsi :

$$\widehat{H}_i[u(t), \theta_i] = \sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{m_{i-1}} b_{m_1 \dots m_i} I_{m_1}(t) \dots I_{m_i}(t) \quad (2.105)$$

où le terme $I_{m_i}(t)$ représente la réponse de la fonction $\Phi_{m_i}(s)$ à l'entrée $u(t)$ du système non linéaire :

$$I_{m_i}(t) = \int_0^t \phi_{m_i}(\tau) u(t - \tau) d\tau \quad (2.106)$$

et où les paramètres $b_{m_1 \dots m_i}$ composent le vecteur θ_i des paramètres qui caractérisent le modèle du noyau de Volterra d'ordre i .

Le modèle global de Volterra utilisé pour représenter le système non linéaire est donc le suivant :

$$\widehat{y}(t, \theta) = \widehat{H}[u(t), \theta] = \sum_{i=1}^n \left(\sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{m_{i-1}} b_{m_1 \dots m_i} I_{m_1}(t) \dots I_{m_i}(t) \right) \quad (2.107)$$

Le vecteur θ rassemble tous les paramètres de modélisation de chaque noyau de Volterra : $\theta = [\theta_1, \dots, \theta_n]$.

2.3.2.2 Inversion du modèle de Volterra

A partir de la connaissance du modèle non linéaire $\widehat{y}(t, \theta)$ qui relie la grandeur d'entrée $u(t)$ à la grandeur de sortie $y(t)$ et d'un vecteur fini y^* de mesures de la sortie $y(t)$, on cherche à obtenir une reconstruction correcte \widehat{U} de l'entrée $u(t)$ du système par la méthode de régularisation de Tikhonov. Remarquons ici que la qualité de la reconstruction peut être caractérisée par l'erreur quadratique entre l'entrée réelle et l'entrée reconstruite mais que l'appréciation de la qualité de reconstruction reste à la discréption de l'utilisateur.

Comme on l'a vu précédemment, la méthode consiste en la minimisation d'un critère composite défini comme suit :

$$J_\lambda(u) = \left\| y - \widehat{H}[u, \theta] \right\|^2 + \lambda \|L(u)\|^2 \quad (2.108)$$

où λ est appelé paramètre de régularisation.

La solution reconstruite \widehat{U}_λ au sens de la minimisation du critère $J_\lambda(u)$ est donnée par

$$\widehat{U}_\lambda = \arg \min_u \left\{ \|y - \widehat{H}[u, \theta]\|^2 + \lambda \|L(u)\|^2 \right\} \quad (2.109)$$

a) Cas linéaire

Dans le cas d'un système causal linéaire mono-entrée mono-sortie, la relation liant l'entrée $u(t)$ à la sortie $y(t)$ du système peut être écrite sous la forme d'un produit de convolution :

$$y(t) = \int_0^t h(t-\tau)u(\tau)d\tau \quad (2.110)$$

où $h(t)$, réponse impulsionnelle du système linéaire, est supposée connue.

L'ensemble des mesures y^* de la sortie $y(t)$ du système est effectué sur un horizon temporel fini. En d'autres termes, chaque fichier de mesures de la sortie $y(t)$ ne comporte qu'un nombre fini N de points et représente la sortie $y(t)$ avec une périodicité finie donnée. C'est à l'utilisateur de s'assurer que ce nombre fini de mesures est suffisant pour représenter avec précision la sortie $y(t)$ du système. La dimension du vecteur y^* est donc fixée dès le départ par l'utilisateur, qui choisit lui-même l'horizon des mesures et la période d'échantillonnage T_e .

Ce passage d'une dimension a priori infinie vers une dimension finie N de la sortie $y(t)$ justifie donc la discrétisation du modèle linéaire (2.110). Une représentation discrète de la sortie peut être donnée par :

$$y_k = \sum_{i=0}^I h_{k-i} u_i = \sum_{i=0}^I h_i u_{k-i} \quad (2.111)$$

où $1 \leq k \leq N$.

Les termes h_i sont les coefficients de la réponse impulsionnelle discrète du modèle. Leur expression dépend de la méthode de discrétisation utilisée. Dans l'immense majorité des cas, la réponse impulsionnelle est supposée finie :

$$h_i \simeq 0, \quad i > I, \quad i \in \mathbb{N} \quad (2.112)$$

Le vecteur y^* des mesures peut donc s'écrire

$$y^* = HU \quad (2.113)$$

avec

$$\begin{aligned} y^* &= [y_1, \dots, y_N]^T \\ U &= [u_1, \dots, u_N]^T \end{aligned} \quad (2.114)$$

où $u_i = u(iT_e)$ pour $i = 1, \dots, N$.

H est une matrice $N \times N$ donnée par

$$H = \begin{bmatrix} h_0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ h_1 & h_0 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ h_I & & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & \ddots & h_0 & 0 \\ 0 & \cdots & \cdots & h_I & \cdots & h_1 & h_0 \end{bmatrix} \quad (2.115)$$

Sous cette forme, le problème change de nature car il revient à reconstruire une entrée U de dimension finie. Dans le cas classique où le paramètre de régularisation λ est nul, l'entrée reconstruite $\widehat{U}_{\lambda=0}$ est la solution du problème au sens des moindres carrés :

$$\widehat{U}_{\lambda=0} = (H^T H)^{-1} H^T y^* \quad (2.116)$$

Toutes les difficultés de ce problème viennent d'un éventuel mauvais conditionnement de la matrice $H^T H$. L'inversion d'une matrice $H^T H$ mal conditionnée et un ensemble de mesures y^* de la sortie $y(t)$ entaché d'erreur pourront conduire à une mauvaise reconstruction \widehat{U} de l'entrée U . D'où l'intérêt de la méthode de régularisation de Tikhonov qui conduit, dans le cas linéaire, à la solution suivante :

$$\widehat{U}_\lambda = [H^T H + \lambda(LU)^T LU]^{-1} H^T y^* \quad (2.117)$$

où L est soit la matrice identité, soit la matrice des dérivées seconde et λ le paramètre de régularisation.

b) Cas non linéaire : modèle de Volterra

Dans le cas d'un système causal mono-entrée mono-sortie non linéaire par rapport à l'entrée, la relation liant l'entrée $u(t)$ à la sortie $y(t)$ du système peut également être écrite sous la forme d'un modèle de Volterra (2.107). Ce modèle est toujours linéaire par rapport aux paramètres, ce qui facilite l'identification du vecteur θ , mais il devient non linéaire par rapport à l'entrée $u(t)$.

La reconstruction de l'entrée $u(t)$ par minimisation du critère composite $J_\lambda(u)$ (2.108) nécessite alors un algorithme d'optimisation non linéaire. Nous allons utiliser l'algorithme de Levenberg-Marquardt (1.115) déjà décrit et employé dans le chapitre précédent. Cet algorithme, ainsi que la plupart des algorithmes de programmation non linéaire, nécessite la dérivation par rapport à l'entrée $u(t)$ du critère $J_\lambda(u)$ afin d'obtenir le vecteur gradient $\frac{\partial J_\lambda(u)}{\partial u(t)}$ et une approximation de la matrice hessienne $\frac{\partial^2 J_\lambda(u)}{\partial u^2(t)}$. L'expression du critère com-

posite $J_\lambda(u)$ (2.108) montre que le calcul du vecteur gradient et de la matrice hessienne requiert le calcul des fonctions de sensibilité par rapport à l'entrée $u(t)$:

$$\frac{\partial H[u(t)]}{\partial u(t)} \quad (2.118)$$

D'après l'expression du modèle de Volterra (2.107), ce calcul des fonctions de sensibilité nécessite à son tour le calcul de chaque dérivée de la forme

$$\frac{\partial I_{m_1}(t)}{\partial u(t)}, \dots, \frac{\partial I_{m_i}(t)}{\partial u(t)} \quad (2.119)$$

et de la forme

$$\frac{\partial I_{m_i}(t)I_{m_j}(t)}{\partial u(t)} \quad (2.120)$$

Le modèle de Volterra développé sur une base de fonctions orthonormées généralisée (2.107) est suffisamment explicite pour permettre d'effectuer analytiquement le calcul des fonctions de sensibilité mais le problème du calcul et de la simulation numérique du vecteur gradient et de la matrice hessienne se pose : la dimension du vecteur y^* des mesures de la sortie $y(t)$ peut considérablement alourdir la somme de calculs numériques nécessaire à la résolution du problème inverse.

Afin de diminuer la somme de calculs, l'idée est d'adopter une démarche de discréétisation semblable à celle présentée dans le cas linéaire : nous choisissons de simuler chaque fonction ϕ_n de la base orthonormée généralisée par l'utilisation de la convolution discrète et des coefficients φ_i de la réponse impulsionale discrète de la fonction $\Phi_n(s)$.

L'expression de la réponse d'une fonction généralisée à l'entrée $u(t)$

$$I_n(t) = \int_0^t \phi_n(\tau)u(t-\tau)d\tau \quad (2.121)$$

devient, après discréétisation

$$I_n = \phi_n U \quad (2.122)$$

où I_n et U sont des vecteurs de N valeurs données en fonction de la période d'échantillonnage T_e choisie :

$$\begin{aligned} I_n &= [I_n(kT_e)], \quad k = 0, \dots, N-1 \\ U &= [u(kT_e)], \quad k = 0, \dots, N-1 \end{aligned} \quad (2.123)$$

Le terme ϕ_n est une matrice $N \times N$, analogue à la matrice (2.115) définie dans le cas linéaire, et donnée par

$$\phi_n = \begin{bmatrix} \varphi_0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \varphi_1 & \varphi_0 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ \varphi_I & & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & \ddots & \varphi_0 & 0 \\ 0 & \cdots & \cdots & \varphi_I & \cdots & \varphi_1 & \varphi_0 \end{bmatrix} \quad (2.124)$$

où les coefficients φ_i sont les coefficients de la réponse impulsionale discrète de la fonction $\Phi_n(s)$.

La réponse impulsionale de la fonction $\Phi_n(s)$ est finie :

$$\varphi_i \simeq 0, \quad i > I, \quad i \in \mathbb{N} \quad (2.125)$$

Une fois cette discréttisation effectuée, le modèle de Volterra global discréttisé, tronqué à l'ordre n , est donc défini par :

$$\hat{y}(\theta) = \hat{H}[U, \theta] = \sum_{i=1}^n \left(\sum_{m_1=0}^{M_i} \dots \sum_{m_i=0}^{m_{i-1}} b_{m_1 \dots m_i} (\phi_{m_1} U) \odot \dots \odot (\phi_{m_i} U) \right) \quad (2.126)$$

L'opération \odot est le produit de Hadamard entre deux matrices A et B défini de la manière suivante :

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}, B = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix}, A \odot B = \begin{bmatrix} a_1 b_1 & a_2 b_2 \\ a_3 b_3 & a_4 b_4 \end{bmatrix} \quad (2.127)$$

Cette discréttisation du modèle de Volterra présente deux avantages. D'abord, la simulation numérique du modèle non linéaire ne comporte que des produits matriciels, ce qui permet d'alléger la charge de calculs. Ensuite, le calcul du vecteur gradient est grandement simplifié puisque le calcul des dérivées (2.119) devient

$$\frac{\partial I_{m_1}}{\partial U} = \phi_{m_1}, \dots, \frac{\partial I_{m_i}}{\partial U} = \phi_{m_i} \quad (2.128)$$

et celui des dérivées (2.120) devient

$$\begin{aligned} \frac{\partial (I_{m_i} \odot I_{m_j})}{\partial U} &= \frac{\partial I_{m_i}}{\partial U} \odot I_{m_j} + \frac{\partial I_{m_j}}{\partial U} \odot I_{m_i} \\ &= \phi_{m_i} \odot I_{m_j} + \phi_{m_j} \odot I_{m_i} \end{aligned} \quad (2.129)$$

Le vecteur gradient est donc donné par l'expression

$$\frac{\partial J_\lambda(U)}{\partial U} = -2\varepsilon \frac{\partial \hat{H}[U, \theta]}{\partial U} + 2\lambda L^T L U \quad (2.130)$$

où

$$\varepsilon = y^* - \hat{H}[U, \theta] \quad (2.131)$$

La matrice hessienne est obtenue par l'approximation de Gauss-Newton :

$$\frac{\partial^2 J_\lambda(U)}{\partial U^2} \simeq 2 \left(\frac{\partial \hat{H}[U, \theta]}{\partial U} \right)^T \frac{\partial \hat{H}[U, \theta]}{\partial U} + 2\lambda L^T L \quad (2.132)$$

Une fois ces calculs effectués, la reconstruction du signal d'entrée U est donc assurée par la minimisation du critère composite $J_\lambda(U)$ (2.108). Un des inconvénients de l'algorithme ci-dessus est qu'il ne garantit pas l'optimalité de la solution, à savoir une entrée reconstruite \hat{U}_λ qui minimise globalement le critère $J_\lambda(U)$. La reconstruction optimale \hat{U}_λ de l'entrée U dépend du paramètre de régularisation λ et de l'initialisation de l'algorithme de Levenberg-Marquardt. Le problème d'initialisation de l'algorithme a déjà été discuté au premier chapitre. Le problème du choix du paramètre λ est inhérent à la méthode de régularisation de Tikhonov.

2.3.3 Application en simulation

Nous allons maintenant appliquer cette méthode d'inversion du modèle de Volterra par régularisation de Tikhonov au système non linéaire dont la représentation d'état est la suivante :

$$\begin{cases} \dot{x}(t) = -x(t) - x^2(t) + u(t) \\ y(t) = x(t) \end{cases} \quad (2.133)$$

La non-linéarité est introduite par le terme quadratique $x^2(t)$. Le modèle de Volterra utilisé pour représenter la relation entrée/sortie de ce système est développé sur ses 2 premiers noyaux. Ces 2 noyaux sont à leur tour modélisés par un développement sur 4 fonctions orthonormées généralisées. Les pôles des fonctions généralisées sont fixées aux valeurs -0.5 , -1 , -1.5 et -2 . Le nombre total de paramètres est égal à 14. La période d'échantillonnage est fixée à 0.01s. La figure 2.3(a) représente la sortie du système non linéaire (2.133) et la sortie du modèle de Volterra proposé.

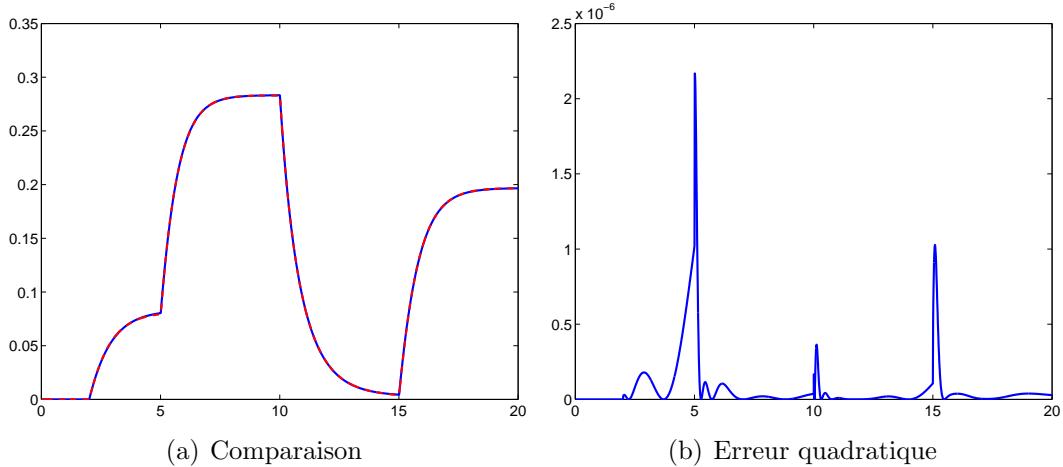


Fig. 2.3 - Comparaison entre sortie du système non linéaire (ligne continue) et sortie du modèle de Volterra (ligne pointillée)

L'erreur quadratique entre les deux courbes, présentée sur la figure 2.3(b), est très faible. L'adéquation entre le système et le modèle de Volterra est donc suffisamment bonne pour envisager l'inversion du modèle de Volterra afin de correctement reconstruire le signal d'entrée $u(t)$ du système non linéaire connaissant sa sortie $y(t)$.

a) Cas déterministe

Afin de valider la méthode d'inversion et le modèle de Volterra choisi, on commence par effectuer une minimisation du critère $J_\lambda(u)$ (2.108) sans paramètre de régularisation, soit $\lambda = 0$, et sans ajouter de bruit au vecteur y^* des sorties mesurées. Les résultats obtenus sont les suivants :

La figure 2.4(a) compare l'entrée U du système non linéaire et l'entrée reconstruite $\hat{U}_{\lambda=0}$ par la méthode décrite précédemment.

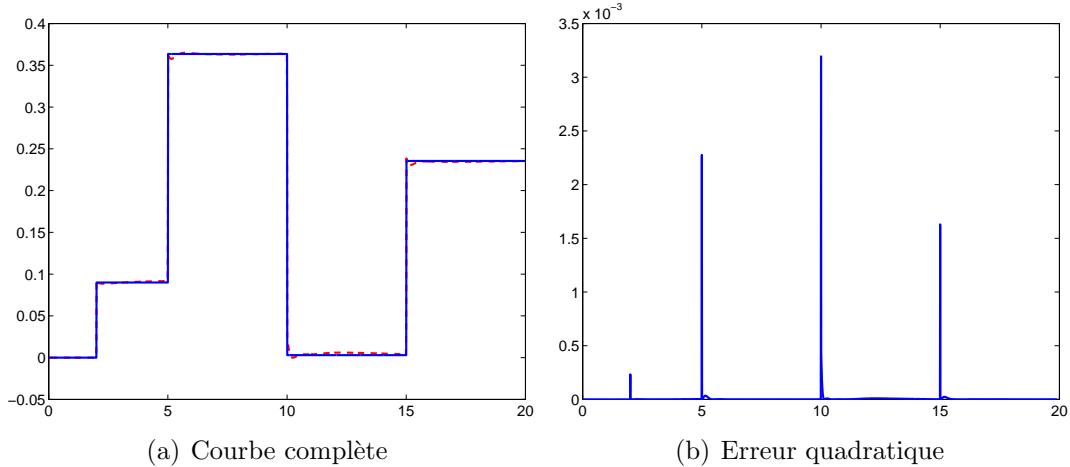


Fig. 2.4 - Entrée du système non linéaire (ligne continue) et entrée reconstruite (ligne pointillée)

La figure 2.4(b) nous donne l'erreur quadratique entre les deux courbes précédentes.

Ce résultat permet donc de valider le bon fonctionnement de l'algorithme de reconstruction de l'entrée d'un système non linéaire par régularisation de Tikhonov.

b) Cas stochastique

A la différence du signal de sortie $y(t)$ précédent, tout signal de sortie mesuré est inévitablement entaché d'un bruit de mesure. Cette constatation s'étend à tous les types de signaux mesurés. Ce bruit de mesure a évidemment une influence sur la qualité de la reconstruction de l'entrée $u(t)$ du système non linéaire et cela même si le modèle de Volterra est suffisamment précis.

On considère donc dans cette partie un signal $y(t)$ simulé auquel on a ajouté un bruit blanc gaussien. Le rapport signal sur bruit est fixé à 100. Le modèle de Volterra défini précédemment (2 noyaux, 14 paramètres) permet un bon compromis entre l'erreur de modélisation (due aux troncatures) et l'erreur de variance (due au bruit) comme le montre la figure 2.5.

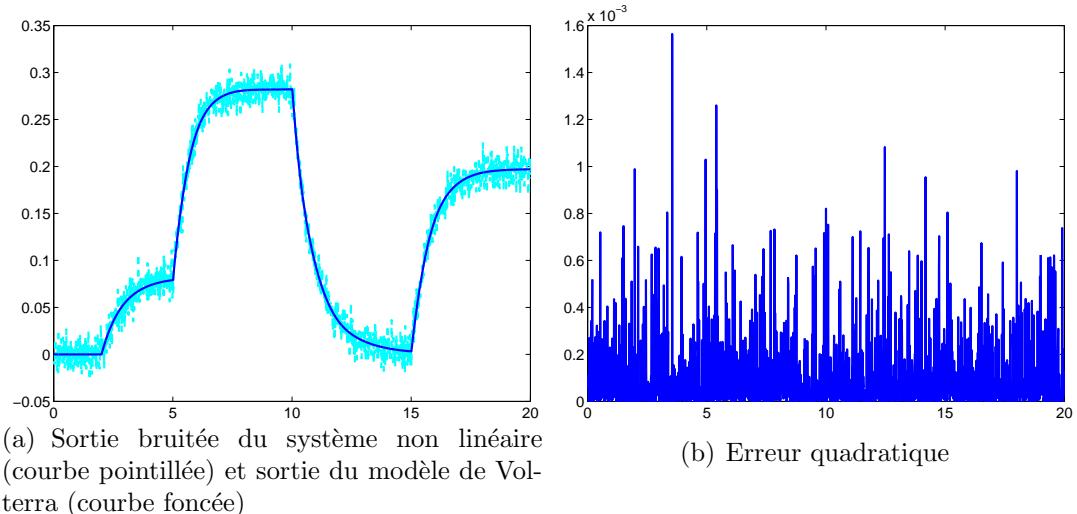


Fig. 2.5 - Comparaison entre les sorties du système non linéaire et du modèle de Volterra

Nous conservons donc le même modèle de Volterra dans les différentes études qui suivent : d'abord, l'étude de l'influence du paramètre de régularisation λ ; ensuite l'étude de l'influence de l'amplitude du bruit de mesure ; et enfin l'étude de l'influence du choix de l'opérateur L de régularisation qui, dans notre étude, sera une matrice.

i) Influence du paramètre de régularisation λ

Afin d'étudier l'importance du paramètre de régularisation λ dans la reconstruction de l'entrée $u(t)$ du système non linéaire, nous allons fixer le rapport signal sur bruit à la valeur 100 et prendre comme opérateur L le plus simple qui soit : la matrice identité I .

On effectue donc la reconstruction de l'entrée $u(t)$ à partir de mesures de la sortie bruitée y^* et du modèle de Volterra pour différentes valeurs de λ échelonnées entre 10^{-5} et 1 (figure 2.6).

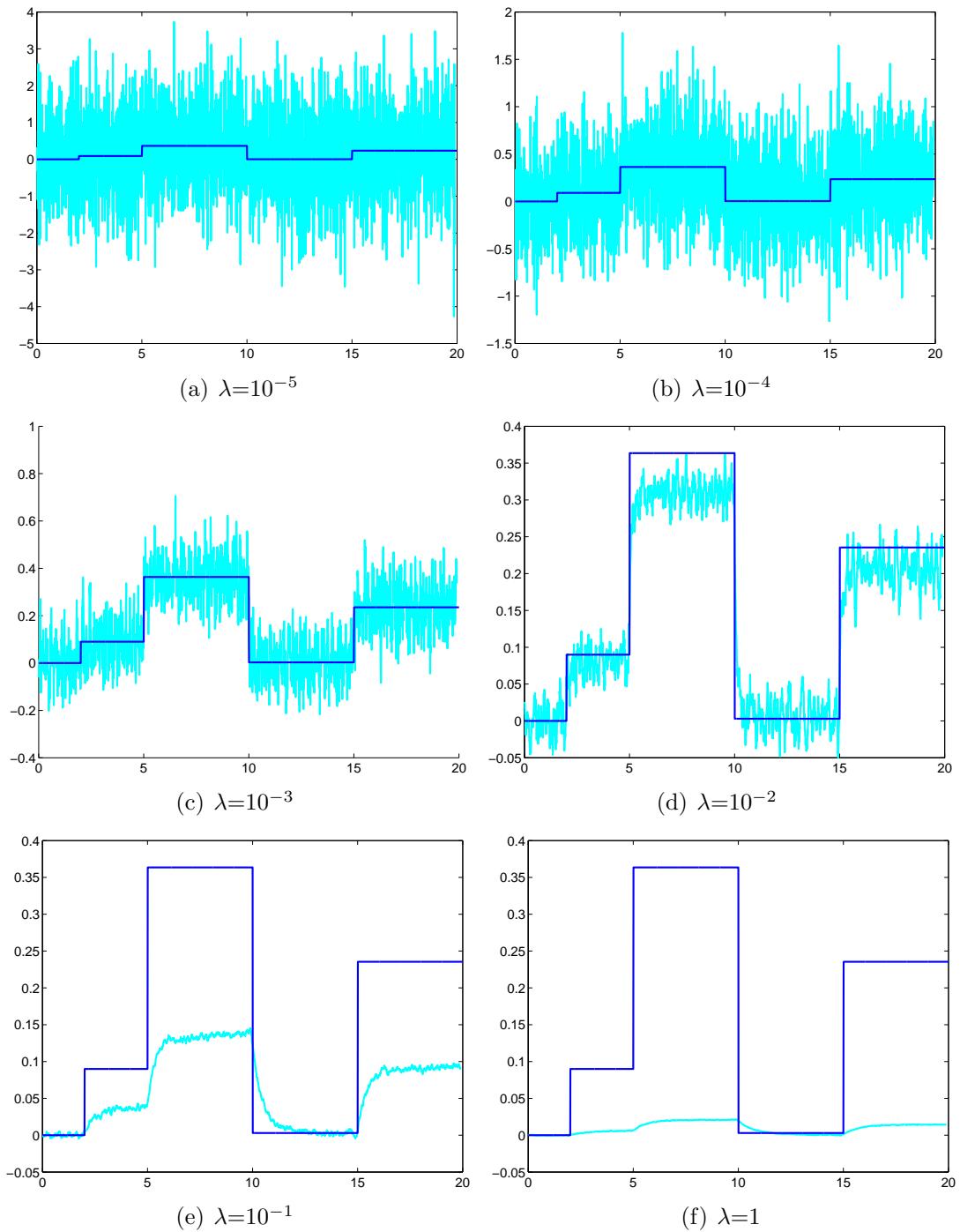


Fig. 2.6 - Reconstruction de l'entrée $u(t)$ pour différentes valeurs de λ

L'augmentation progressive de la valeur du paramètre de régularisation λ permet de constater l'influence du deuxième terme du critère composite $J_\lambda(u)$ (2.108) sur la qualité de la reconstruction \widehat{U}_λ de l'entrée $u(t)$.

De manière générale, on retrouve sur l'entrée reconstruite le bruit ajouté à la sortie $y(t)$ du système, mais d'autant plus amplifié que la valeur de λ est faible. Ceci montre

bien que, dans le cas où l'on ne régularise que très peu ou pas du tout, l'inversion du modèle de Volterra est équivalente à un filtrage passe-haut : les composantes en haute fréquence (c'est à dire le bruit) sont amplifiées. Il en résulte une reconstruction de l'entrée $u(t)$ fidèle au données mesurées mais fortement bruitée. On constate à nouveau que l'absence totale de régularisation est équivalente à reconstruire l'entrée par la méthode classique des moindres carrés.

L'ajout du terme de régularisation $\lambda \|L(u)\|^2$ et l'augmentation progressive de la valeur du paramètre λ permet de vérifier son influence sur la qualité de la reconstruction de l'entrée $u(t)$. Si la valeur de λ augmente trop, ce n'est plus l'adéquation aux données mesurées qui est privilégiée mais plutôt l'information a priori que l'on possède sur l'entrée $u(t)$. Il en résulte une atténuation importante du bruit sur la solution reconstruite mais on observe également que le signal reconstruit U n'est plus du tout en adéquation avec l'entrée réelle $u(t)$ du système non linéaire : la régularisation a occasionné une perte d'information par rapport aux données mesurées.

Ces observations indiquent donc qu'il est nécessaire de trouver un compromis entre l'importance accordée aux données et celle accordée à la régularisation de la solution. Ce compromis doit donc être effectué sur le choix de la valeur du paramètre λ .

Différents essais de reconstruction de l'entrée $u(t)$ montrent que la meilleure valeur du paramètre de régularisation λ se situe aux alentours de 0.005 comme le montre la figure 2.7. Ceci est conforme au résultat de l'approche bayésienne qui définit le coefficient de régularisation optimal comme étant le rapport $\lambda_{opt} = \sigma_b^2 / \sigma_u^2$ de la variance du bruit par la variance de l'entrée $u(t)$ à reconstruire. Dans notre application, ce rapport est égal à 0.0049.

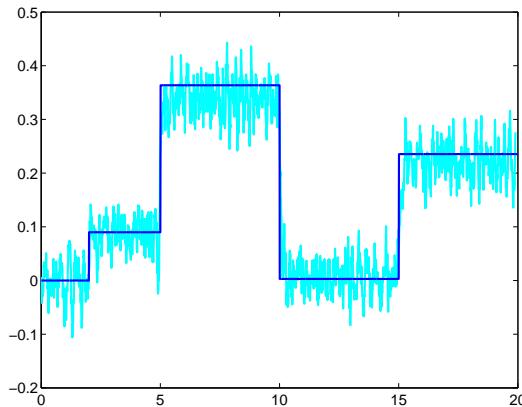


Fig. 2.7 - Reconstruction de l'entrée $u(t)$ pour $\lambda = 0.005$

On remarque que cette valeur λ_{opt} , qui offre pourtant un compromis entre le niveau de bruit de l'entrée reconstruite \widehat{U}_λ et sa régularisation, ne permet pas d'obtenir une reconstruction totalement satisfaisante car cette dernière reste relativement bruitée. Le niveau

de bruit des données mesurées a donc une importance capitale pour la reconstruction de l'entrée d'un système par inversion de modèle. Le paragraphe suivant propose donc d'observer l'influence du bruit de mesure sur la qualité de la reconstruction de l'entrée $u(t)$ du système non linéaire.

ii) Influence du bruit de mesure

Afin d'observer l'influence du bruit de mesure, nous allons ajouter aux données de sortie mesurées y^* un bruit d'amplitude croissante. Dans chaque cas, on fixe le paramètre de régularisation λ à une valeur optimale λ_{opt} et on garde $L = I$ comme opérateur de régularisation agissant sur l'entrée $u(t)$. On fait varier le rapport signal sur bruit de manière à obtenir des données de sortie faiblement bruitées (rapport S/B égal à 10000) jusqu'à des données fortement bruitées (rapport S/B égal à 20). L'ensemble de figures 2.8 représente l'entrée reconstruite pour ces différents niveaux de bruit.

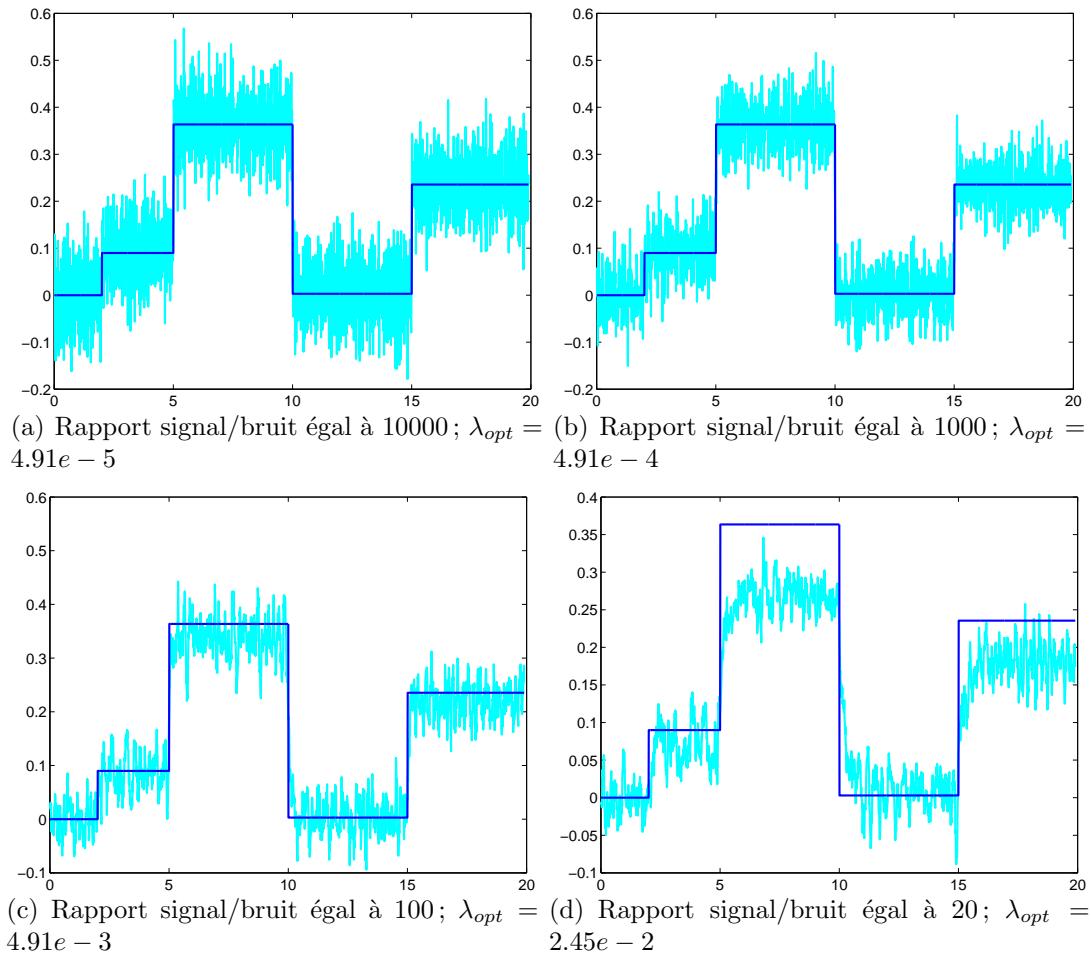


Fig. 2.8 - Reconstruction de l'entrée $u(t)$ pour différents niveaux de bruit de la sortie mesurée $y(t)$

Malgré une valeur du paramètre λ qui respecte au mieux le compromis entre adéquation aux données et régularisation de l'entrée reconstruite, on constate qu'un fort niveau de bruit ajouté aux données mesurées $y(t)$ dégrade considérablement le signal reconstruit jusqu'à le rendre inutilisable (cas d'un rapport signal sur bruit égal à 20 sur la figure 2.8(d)).

La mesure du signal de sortie doit donc être la plus précise possible. Cependant, dans certains cas de figure, il n'est pas possible d'éliminer complètement le bruit de mesure sans perdre d'information sur le signal de sortie $y(t)$. Le filtrage des données ne suffit pas à obtenir une entrée reconstruite suffisamment proche de la véritable entrée du système non linéaire. Il faut donc trouver un autre moyen de limiter l'amplification du bruit de mesure.

Une solution consiste à modifier le critère composite $J_\lambda(u)$ en choisissant une matrice de régularisation L autre que la matrice identité.

iii) Influence du choix de la matrice de régularisation L

Nous venons de voir que le problème de l'amplification du bruit de mesure ne peut être évité lors de l'inversion du modèle de Volterra (ou de n'importe quel modèle linéaire ou non linéaire). On peut cependant modifier la deuxième partie du critère composite $J_\lambda(u)$ pour agir sur l'entrée reconstruite \widehat{U}_λ et tenter de minimiser l'importance du bruit.

Jusqu'ici, la matrice de régularisation L considérée était la matrice identité. On se contentait donc de minimiser la norme $\|\cdot\|_2$ de l'entrée reconstruite U . Un autre choix possible est de minimiser non pas la norme de l'entrée reconstruite mais celle de la dérivée seconde de l'entrée reconstruite afin d'adoucir les variations brusques occasionnées par le bruitage des données mesurées.

Le critère composite $J_\lambda(u)$ devient donc :

$$J_\lambda(u) = \|y - H[u]\|^2 + \lambda \left\| \frac{d^2 u}{dt^2} \right\|^2 \quad (2.134)$$

étant donné que l'on travaille sur des mesures discrètes et sur un modèle de Volterra discrétilisé qui nous a conduit au critère composite discrétilisé suivant :

$$J_\lambda(U) = \|y - \widehat{H}[U, \theta]\|^2 + \lambda \|LU\|^2 \quad (2.135)$$

il est nécessaire de discrétiliser l'opérateur dérivée seconde $\frac{d^2}{dt^2}$ afin de l'appliquer au vecteur U de dimension finie.

Considérons une fonction $f(t)$ échantillonnée avec une période T_e sur un horizon de N points. Sur cet horizon, la fonction $f(t)$ est représentée par le vecteur fini :

$$[f_1, f_2, \dots, f_i, \dots, f_N] \quad (2.136)$$

avec

$$f_i = f(iT_e) \quad (2.137)$$

Selon la méthode d'Euler, une approximation de la dérivée seconde de la fonction $f(t)$ au temps $t = iT_e$ est donnée par :

$$\left(\frac{d^2 f}{dt^2} \right)_{t=iT_e} \simeq \frac{f((i-1)T_e) - 2f(iT_e) + f((i+1)T_e)}{T_e^2} \quad (2.138)$$

que l'on peut écrire plus simplement

$$\frac{f_{i-1} - 2f_i + f_{i+1}}{T_e^2} \quad (2.139)$$

On peut donc écrire une approximation de la dérivée seconde de l'entrée reconstruite U au temps $t = kT_e$ de la manière suivante :

$$\frac{U((k-1)T_e) - 2U(kT_e) + U((k+1)T_e)}{T_e^2} \quad (2.140)$$

ou, plus simplement

$$\frac{U_{k-1} - 2U_k + U_{k+1}}{T_e^2} \quad (2.141)$$

A partir de l'approximation de la dérivée seconde de chaque composante du vecteur U , on peut donc définir une matrice L d'approximation de la dérivée seconde de l'entrée reconstruite U :

$$L = \frac{1}{T_e^2} \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \ddots & & \vdots \\ 0 & \ddots & 1 & -2 & 1 & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & 1 & -2 & 1 \end{pmatrix} \quad (2.142)$$

On appelle plus communément matrice des dérivées secondes la matrice L ainsi définie.

Si l'on multiplie cette matrice par le vecteur U , on obtient un vecteur d'approximations de la dérivée seconde de chaque composante :

$$\frac{1}{T_e^2} \begin{pmatrix} U_1 - 2U_2 + U_3 \\ \vdots \\ U_{k-1} - 2U_k + U_{k+1} \\ \vdots \\ U_{N-2} - 2U_{N-1} + U_N \end{pmatrix} \quad (2.143)$$

On peut maintenant utiliser cette discréttisation de l'opérateur dérivée seconde dans le critère composite $J_\lambda(U)$ afin d'en observer les effets sur la qualité de la reconstruction

de l'entrée $u(t)$ du système non linéaire. On va dans un premier temps se placer dans les mêmes conditions de simulation que dans le cas des figures 2.6(a) à 2.6(f). Le rapport signal sur bruit reste égal à 100. Selon différentes valeurs du coefficient de régularisation λ , les résultats obtenus sont ceux de l'ensemble de figures 2.9.

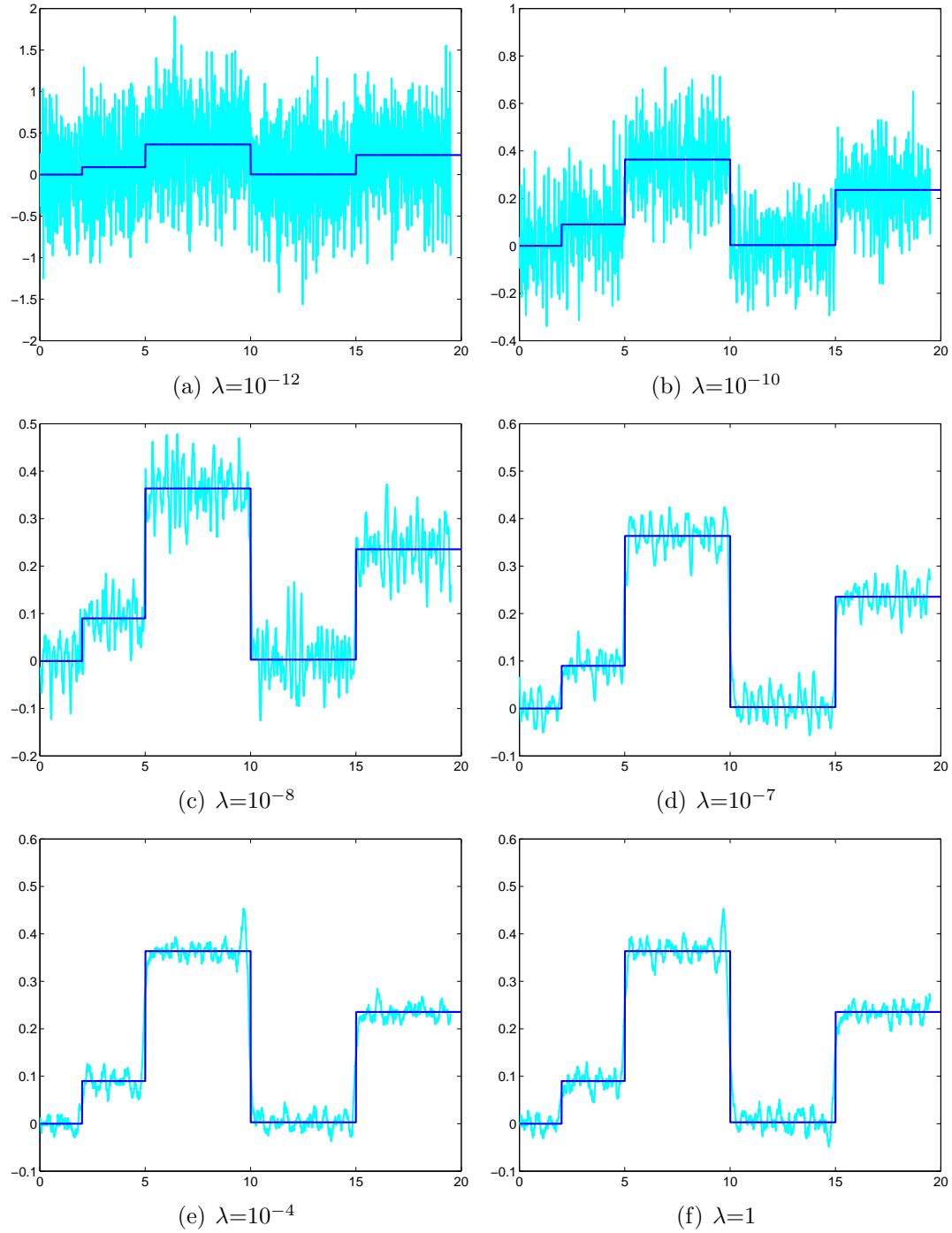


Fig. 2.9 - Reconstruction de l'entrée $u(t)$ pour différentes valeurs de λ et L matrice des dérivées secondes

On remarque tout d'abord que les valeurs considérées du coefficient de régularisation λ sont différentes des valeurs utilisées dans le cas de l'ensemble des figures 2.6(a) à 2.6(f). Ceci peut s'expliquer par la forme de la matrice L des dérivées seconde (2.142) appliquée au vecteur d'entrée discrétilisé U . En effet, les coefficients de cette matrice sont multipliés par le coefficient $\frac{1}{T_e^2}$ qui dépend de la période d'échantillonnage T_e . Puisque la période d'échantillonnage est $T_e = 0.01\text{s}$, la matrice L se retrouve multipliée par un coefficient $\frac{1}{T_e^2} = 10000$. Le deuxième terme du critère composite $J_\lambda(u)$ (2.134) prend donc beaucoup plus d'importance que le premier dans la reconstruction de l'entrée U . Afin de relativiser l'importance de ce second terme, il faudra donc considérer des valeurs de λ très faibles, de l'ordre de 10^{-4} .

Comparons maintenant la reconstruction de l'entrée U en utilisant la matrice identité (figures 2.6(a) à 2.6(f)) à celle qui utilise la matrice des dérivées seconde (figures 2.9(a) à 2.9(f)). On peut constater d'une part que l'effet de filtrage introduit par l'utilisation de la dérivée seconde de l'entrée reconstruite U est visible à partir d'une certaine valeur du paramètre λ située autour de la valeur 10^{-7} . Le terme $\frac{\lambda}{T_e^2}$ qui multiplie la matrice L est donc égal à 10^{-3} , valeur du même ordre de grandeur que celle optimale $\lambda = 0.005$ que nous avons considéré lors de l'utilisation de la matrice identité.

Mais, d'autre part, on constate que l'augmentation de la valeur du paramètre λ (figure 2.9(f)) n'occasionne pas de dégradation de la reconstruction de l'entrée comme c'était le cas lors de l'utilisation de la matrice identité.

2.3.4 Volume de calcul

Un problème qui n'a pas été abordé jusqu'à maintenant est celui du temps de calcul et de simulation nécessaire pour l'inversion du modèle de Volterra. En effet, suivant la dimension du vecteur U à reconstruire (généralement fixée par la dimension du vecteur y^* des données mesurées), ce temps peut devenir prohibitif, de l'ordre de plusieurs heures de calcul. Cette charge de calcul s'explique en grande partie par la dimension du vecteur y^* des points de mesure. De manière générale, il n'est pas rare de travailler sur des vecteurs de points dont la dimension atteint facilement 1000, voire davantage, selon l'horizon temporel et la période d'échantillonnage T_e choisis par l'utilisateur. Les matrices ϕ_n et L considérées pour simuler la sortie $\hat{y}(\theta)$ du modèle de Volterra et pour minimiser le critère composite $J_\lambda(U)$ sont donc également de dimension très importante. Par conséquent, effectuer une simulation de sortie de modèle de Volterra ainsi qu'une minimisation de critère sur toute l'étendue de l'horizon des mesures coûte très cher en temps de calcul, même pour les processeurs actuels dont la puissance de calcul augmente régulièrement. En outre, la charge de calcul nécessaire, si on utilise la méthode telle quelle, ne permet pas d'envisager la reconstruction d'un signal d'entrée en temps réel.

Une solution consiste donc à ne pas envisager la minimisation du critère $J_\lambda(U)$ sur l'ensemble N des points de mesure mais plutôt sur un horizon donné $I \ll N$ (figure 2.10).

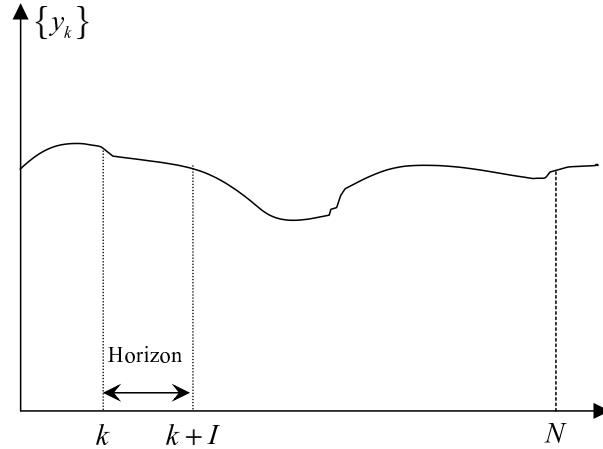


Fig. 2.10 - Horizon I de minimisation de critère

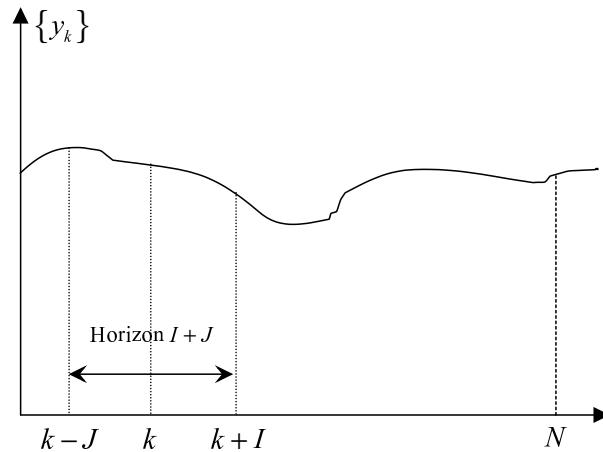


Fig. 2.11 - Horizon $J + I$ de reconstruction

A l'instant k , l'entrée a déjà été reconstruite pour $k-1$ échantillons et on la reconstruit pour l'échantillon k après minimisation du critère $J_\lambda(U)$ sur l'horizon $[k, k+I]$. On déplace ensuite l'horizon I en introduisant une nouvelle mesure, d'où le nom de reconstruction sur fenêtre glissante. Une fois qu'on a atteint la valeur $I = N$, le signal d'entrée $u(t)$ est reconstruit sur l'horizon entier.

La charge de calcul liée à l'optimisation sur I points est ainsi limitée.

Cependant, le nombre de points sur lesquels on reconstruit l'entrée augmente à chaque nouvelle mesure considérée. Par conséquent, l'horizon de simulation du modèle de Volterra

est en perpétuelle augmentation. Là aussi, une solution consiste à envisager la reconstruction de l'entrée sur un horizon donné $J + I < N$ dont la largeur est fixée par l'utilisateur (figure 2.11).

La figure 2.12 illustre cette technique de reconstruction par minimisation de critère sur une fenêtre glissante.

A l'instant k , l'entrée $u(t)$ a déjà été reconstruite sur l'horizon $[0, k - 1]$. On dispose donc d'un vecteur

$$\widehat{U}_{k-1} = [U(0), \dots, U(iT_e), \dots, U((k-1)T_e)] \quad (2.144)$$

On note ce vecteur plus simplement

$$\widehat{U}_{k-1} = [U_0, \dots, U_i, \dots, U_{k-1}] \quad (2.145)$$

L'étape suivante est de déterminer la reconstruction optimale U_{opt} de l'entrée $u(t)$ sur l'horizon $[k, k + I]$ par minimisation du critère composite :

$$J_{\lambda,k}(U) = \sum_{i=k}^{k+I} (y_i^* - \widehat{y}_i)^2 + \lambda \sum_{i=k}^{k+I} U_i^2 \quad (2.146)$$

où les termes y_i^* sont les mesures de la sortie $y(t)$ du système non linéaire sur l'horizon $[k, k + I]$.

Le terme $\widehat{y} = \widehat{H}[U, \theta]$ est la réponse simulée du modèle de Volterra à l'entrée U sur l'horizon $[k - J, k + I]$. Cette simulation n'est donc pas effectuée sur l'horizon global $[0, k + I]$ mais sur un horizon réduit afin de diminuer la charge de calcul nécessaire.

On utilise un algorithme de programmation non linéaire de type Levenberg-Marquardt pour minimiser le critère $J_{\lambda,k}(U)$. Le vecteur U est initialisé à la valeur 0 à chaque instant k .

La minimisation du critère $J_{\lambda,k}(U)$ permet d'obtenir la reconstruction U_{opt} de l'entrée $u(t)$ sur l'horizon $[k - J, k + I]$ mais on ne conserve que l'échantillon correspondant à l'instant k :

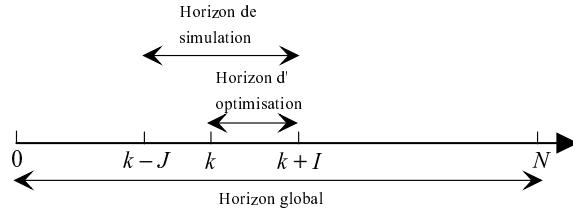
$$U_k = U_{opt}(k) \quad (2.147)$$

On dispose maintenant du vecteur

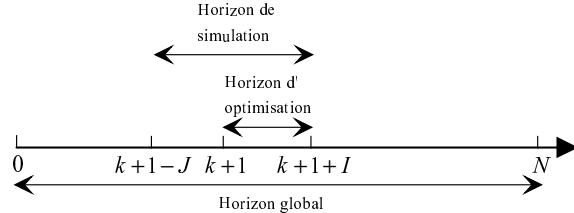
$$\widehat{U}_k = [U_0, \dots, U_i, \dots, U_k] \quad (2.148)$$

L'étape suivante consiste à incrémenter la valeur de k et minimiser le critère $J_{\lambda,k+1}(U)$.

Etape k :



Etape $k + 1$:



Etape $N - I$:

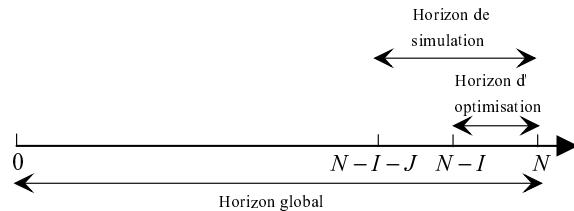


Fig. 2.12 - Reconstruction de l'entrée $u(t)$ sur une fenêtre glissante

Cette technique permet donc de réduire le temps de calcul global en limitant le temps consacré à la minimisation du critère composite $J_{\lambda,k}(U)$ et à la simulation de la sortie \hat{y} du modèle de Volterra. L'entrée $u(t)$ n'est plus reconstruite globalement sur l'horizon des mesures mais de manière itérative, point par point, en considérant un horizon de simulation réduit à $I + J$ points et un horizon de minimisation de critère réduit à I points. On peut ainsi passer de plusieurs heures de calcul à seulement quelques minutes. On peut encore diminuer ce temps en choisissant des valeurs de I et J faibles et ainsi envisager une exploitation en temps réel mais un choix de valeurs trop faibles peut dégrader la qualité de reconstruction de l'entrée à cause de l'étroitesse de l'horizon d'optimisation I et de simulation $[I + J]$. Il est donc nécessaire de respecter un compromis entre temps de calcul et qualité de reconstruction. Ce compromis peut empêcher d'implémenter cette méthode en temps réel pour des systèmes rapides.

Si les valeurs de I et J sont convenablement choisies, cette technique de reconstruction de l'entrée $u(t)$ n'affecte en rien la qualité de l'entrée reconstruite \hat{U} sur l'horizon global. On peut reprendre le cas où la valeur du paramètre de régularisation λ est fixé à la valeur 0.001 et la matrice L est celle des dérivées seconde (cf figure 2.13).

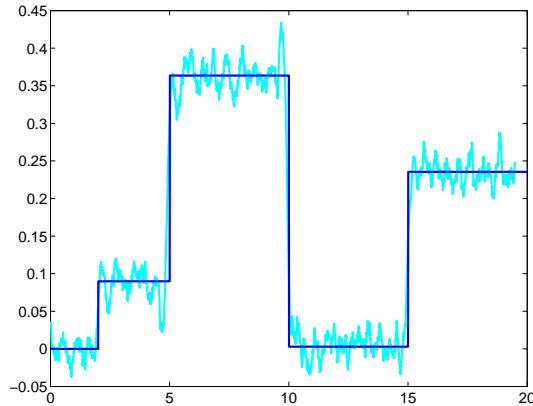


Fig. 2.13 - Reconstruction de l'entrée $u(t)$ sur une fenêtre glissante pour $\lambda = 0.001$

La méthode présentée permet donc de diminuer considérablement le volume de calcul nécessaire à la reconstruction du signal d'entrée.

2.3.5 Conclusion

L'étude de l'inversion d'un modèle par la méthode de régularisation de Tikhonov appliquée dans un cas non linéaire (celui du modèle de Volterra) fait ressortir un certain nombre de problèmes ouverts, dont la plupart concernent également les modèles linéaires.

Tout d'abord, la question de la forme du modèle employé et le fait qu'il puisse correctement représenter le système réel étudié doivent rester au centre des préoccupations de l'utilisateur car ces deux considérations doivent être antérieures au problème de l'inversion. Le modèle de Volterra considéré dans cette étude est un type particulier de modèle parmi d'autres. Nous verrons dans la partie de ce mémoire consacrée aux applications qu'il convient dans les deux cas considérés mais d'autres modèles pourraient bien sûr représenter tout aussi correctement le comportement des systèmes réels étudiés. Par contre, la méthode d'inversion par régularisation ne pourra peut-être pas s'appliquer à ces autres formes de modèle. Ceci met l'accent sur la difficulté à définir une méthode d'inversion globale, que ce soit dans le cas linéaire ou non linéaire.

Un autre problème ouvert aussi bien en linéaire qu'en non linéaire est la détermination du paramètre de régularisation optimal. Dans notre cas, cette détermination n'a pu se faire qu'à la suite de différents essais/erreurs et de comparaisons entre l'entrée réelle du système et l'entrée reconstruite. La présence de bruit sur les données mesurées ne facilite pas ce choix optimal du paramètre.

Un problème n'a pas été soulevé dans cette partie mais reste d'une importance capitale dans le cas non linéaire et ne se limite pas qu'au problème d'inversion de système. Il s'agit bien sûr de la minimisation de critère. La méthode employée dans notre cas est celle de

Levenberg-Marquardt. Les résultats obtenus sont relativement satisfaisants mais, comme on l'a vu dans la première partie du mémoire, cette méthode ne garantit pas un résultat optimal. La reconstruction de l'entrée réelle du système non linéaire après minimisation du critère régularisé nous fournit un résultat correct mais qui peut se révéler sous-optimal.

Cette étude a donc montré que la méthode d'inversion de modèle par régularisation de Tikhonov peut s'appliquer en utilisant un modèle de Volterra discrétilisé et une fenêtre d'optimisation glissante.

2.4 Inversion du modèle de Volterra par retour d'entrée reconstruite [George, 1959]

2.4.1 Introduction

Une alternative possible à la régularisation de Tikhonov appliquée à l'inversion d'un système non linéaire consiste en l'utilisation d'une représentation implicite de l'inverse du modèle non linéaire considéré. Les deux méthodes que nous allons décrire sont très proches l'une de l'autre dans leur principe et nécessitent toutes deux un retour additif de l'entrée reconstruite sur le signal mesuré utilisé pour reconstruire cette même entrée. De même que la régularisation de Tikhonov, ces deux méthodes ne nécessitent en aucun cas de procéder à une inversion explicite du modèle non linéaire considéré.

Ces deux approches sont relativement intuitives et permettent la reconstruction point par point d'un signal d'entrée en utilisant uniquement la sortie mesurée et un modèle du système non linéaire. La forme de modèle que nous utiliserons est une série de Volterra dont chaque noyau est approché par un développement sur des fonctions de transfert. Afin d'assurer une reconstruction de qualité, le modèle doit représenter le plus précisément possible la sortie du système non linéaire.

L'application de ces deux méthodes de reconstruction à un système non linéaire en simulation permettra de constater que les résultats de reconstruction obtenus sont précis et tout à fait comparables entre eux et à ceux obtenus par régularisation.

2.4.2 Première approche

2.4.2.1 Principe

Les conditions de départ sont les mêmes que dans la partie précédente : on considère un système non linéaire dont l'entrée est notée $u(t)$ et la sortie $y(t)$, modélisé par un développement tronqué en série de Volterra. Nous utiliserons dans toute la suite de cette partie le modèle de Volterra développé sur fonctions de transfert. Cette forme de modélisation a été étudiée au chapitre 1. L'ensemble des résultats présentés peuvent également

2.4. Inversion du modèle de Volterra par retour d'entrée reconstruite [George, 1959]

s'appliquer au modèle de Volterra développé sur bases de fonctions orthonormées.

Dans toute la suite de cette partie, on utilise la forme de modèle définie par la figure 1.28 du premier chapitre. Chaque terme $D_{ji}(a_{i,j}, p_{j,i})$ représente donc une fonction de transfert d'ordre $p_{j,i} \in \mathbb{N}$ (somme de fonctions de transfert d'ordre $k \in \mathbb{N}, k = 0, \dots, p_{j,i}$) et est donné par l'expression (1.132).

La première approche considérée afin d'obtenir une reconstruction acceptable \hat{u} de l'entrée $u(t)$ à partir de la sortie mesurée $y(t)$ et du modèle de Volterra $H[u(t)]$ d'un système non linéaire repose sur le schéma d'inversion en boucle fermée donné sur la figure 2.14.

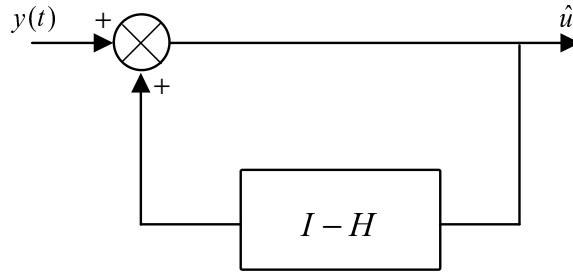


Fig. 2.14 - Schéma de reconstruction de l'entrée $u(t)$

La fonctionnelle H qui relie la sortie $y(t)$ du système non linéaire à son entrée $u(t)$ est définie par l'expression

$$y(t) = \sum_{i=1}^{\infty} H_i[u(t)] \quad (2.149)$$

avec

$$H_i[u(t)] = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i u(t - \tau_j) d\tau_j \quad (2.150)$$

On peut définir la fonctionnelle I de la manière suivante :

$$I[u(t)] = \sum_{i=1}^{\infty} I_i[u(t)] \quad (2.151)$$

On considère une fonction multidimensionnelle $x(t_1, \dots, t_i)$ et on lui applique la fonctionnelle I_i de la manière suivante :

$$I_i[x(t_1, \dots, t_i)] = \int_0^{t_1} \dots \int_0^{t_i} \delta(t_1 - \tau_1, \dots, t_i - \tau_i) x(t_1, \dots, t_i) \prod_{j=1}^i d\tau_j \quad (2.152)$$

La fonction δ est une impulsion de Dirac. Par définition, l'impulsion de Dirac est l'élément neutre du produit de convolution. L'expression (2.152) est donc égale à :

$$I_i[x(t_1, \dots, t_i)] = x(t_1, \dots, t_i) \quad (2.153)$$

Si l'on remplace maintenant la fonction x par le signal d'entrée u du système non linéaire et que l'on se place dans le cas particulier où $t_1 = \dots = t_i = t$, l'expression (2.152) peut s'écrire :

$$I_i[u(t, \dots, t)] = \prod_{j=1}^i u(t) = u^i(t) \quad (2.154)$$

La somme de deux fonctionnelles étant une fonctionnelle, on peut par conséquent développer le terme $(I - H)[u(t)]$ comme une fonctionnelle classique :

$$(I - H)[u(t)] = \sum_{i=1}^{\infty} (I_i - H_i)[u(t)] \quad (2.155)$$

avec

$$(I_i - H_i)[u(t)] = u^i(t) - H_i[u(t)] \quad (2.156)$$

L'expression de $H_i[u(t)]$ est donnée par (2.150).

Le schéma de la figure 2.14 est équivalent à la figure 2.15 où la fonctionnelle L relie l'entrée reconstruite \hat{u} à la sortie mesurée $y(t)$.

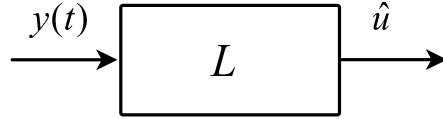


Fig. 2.15 - Schéma équivalent de reconstruction de l'entrée $u(t)$

La fonctionnelle L peut être développée de manière classique en une somme de fonctionnelles :

$$L[y(t)] = \sum_{i=1}^{\infty} L_i[y(t)] \quad (2.157)$$

Nous allons maintenant montrer que la fonctionnelle L (qui associe l'entrée reconstruite $\hat{u}(t)$ à la sortie $y(t)$ du système non linéaire) est l'inverse de la fonctionnelle de Volterra H (qui associe la sortie $y(t)$ à l'entrée $u(t)$ du système non linéaire).

2.4. Inversion du modèle de Volterra par retour d'entrée reconstruite [George, 1959]

Considérons d'abord le cas simple où la sortie mesurée $y(t)$ n'est pas bruitée. Le schéma de la figure 2.14 permet d'écrire la relation :

$$\hat{u} = y(t) + (I - H)[\hat{u}(t)] \quad (2.158)$$

Le schéma équivalent 2.15 nous donne la relation $\hat{u} = L[y(t)]$ que l'on peut introduire dans la relation (2.158) :

$$L[y(t)] = y(t) + (I - H)[L[y(t)]] \quad (2.159)$$

La fonctionnelle $I - H$ peut être développée en une somme de deux fonctionnelles :

$$\begin{aligned} L[y(t)] &= y(t) + I[L[y(t)]] - H[L[y(t)]] \\ &= y(t) + L[y(t)] - H[L[y(t)]] \end{aligned} \quad (2.160)$$

Après simplification, on obtient :

$$y(t) = H[L[y(t)]] \quad (2.161)$$

On en déduit donc que la fonctionnelle L est l'inverse de la fonctionnelle H . Le principe du schéma 2.14 peut donc être utilisé pour reconstruire l'entrée $u(t)$ du système non linéaire à partir de la connaissance du modèle de Volterra et de la mesure de la sortie $y(t)$.

2.4.2.2 Application à un système non linéaire

Afin de valider en simulation cette méthode d'inversion, on prend l'exemple d'un système non linéaire quadratique dont la représentation d'état est la suivante :

$$\begin{cases} \dot{x}(t) = -10^{-6}x(t) - x^2(t) + 10^{-6}u(t) \\ y(t) = 250x(t) \end{cases} \quad (2.162)$$

Ce système est différent de celui utilisé précédemment afin d'éviter certains problèmes numériques lors des simulations. On considère comme signal d'entrée la séquence pseudo-aléatoire variable en amplitude présenté sur la figure 2.16(a). La période d'échantillonnage est cette fois fixée à $T_e = 1s$.

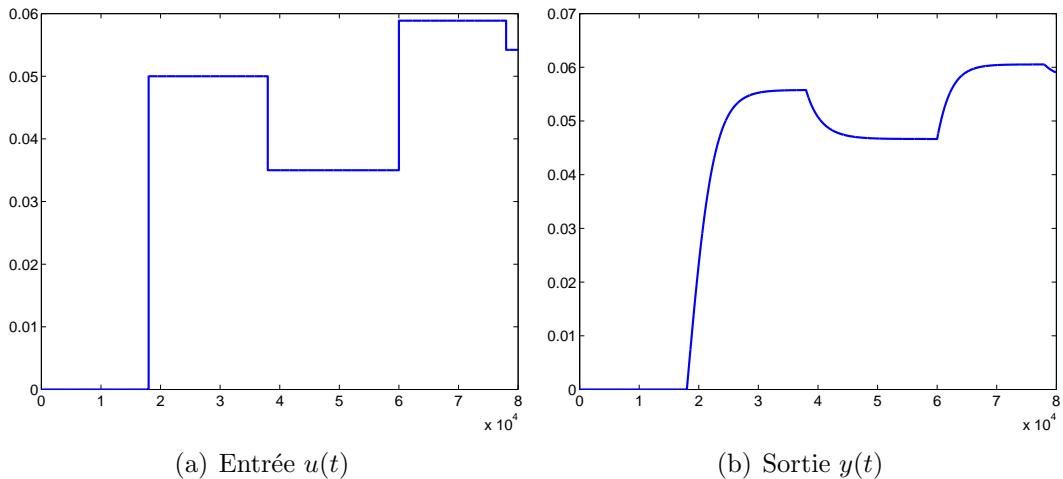


Fig. 2.16 - Simulation du système non linéaire (2.162)

La réponse du système non linéaire (2.162) est représentée sur la figure 2.16(b).

Afin de représenter correctement le lien entre ces deux signaux, le modèle de Volterra utilisé est développé sur ses deux premiers noyaux. Le premier noyau de Volterra est ensuite modélisé par un développement sur une fonction de transfert tandis que le deuxième est modélisé par un développement sur deux fonctions de transfert. Le nombre total de paramètres du modèle est égal à 8.

La sortie du modèle de Volterra et du système non linéaire (2.162) sont proches comme le montrent les figures 2.17(a) et 2.17(b).

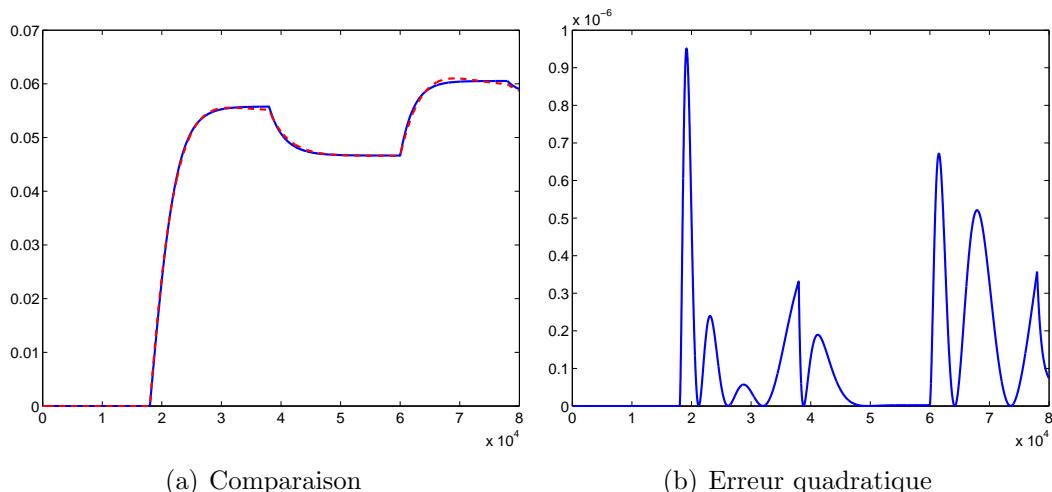


Fig. 2.17 - Sortie du système non linéaire (ligne continue) et du modèle de Volterra (ligne pointillée)

On dispose donc d'un modèle de Volterra suffisamment précis et d'une mesure non

2.4. Inversion du modèle de Volterra par retour d'entrée reconstruite [George, 1959]

bruitée de la sortie $y(t)$ du système non linéaire étudié. On peut donc appliquer la méthode d'inversion proposée afin de la valider.

Le modèle de Volterra tronqué à ses deux premiers noyaux étant suffisant pour correctement représenter la sortie du système non linéaire, on tronque également à ses deux premiers termes la fonctionnelle I définie par (2.151) :

$$\begin{aligned} I[u(t)] &= I_1[u(t)] + I_2[u(t)] \\ &= u(t) + u^2(t) \end{aligned} \quad (2.163)$$

Remarquons que l'augmentation de l'ordre de troncature du modèle de Volterra et de la fonctionnelle I n'apporte rien de plus à la précision des résultats obtenus par cette méthode.

Cette première méthode de reconstruction de l'entrée $u(t)$ donne les résultats figurant sur les courbes de la figure 2.18.

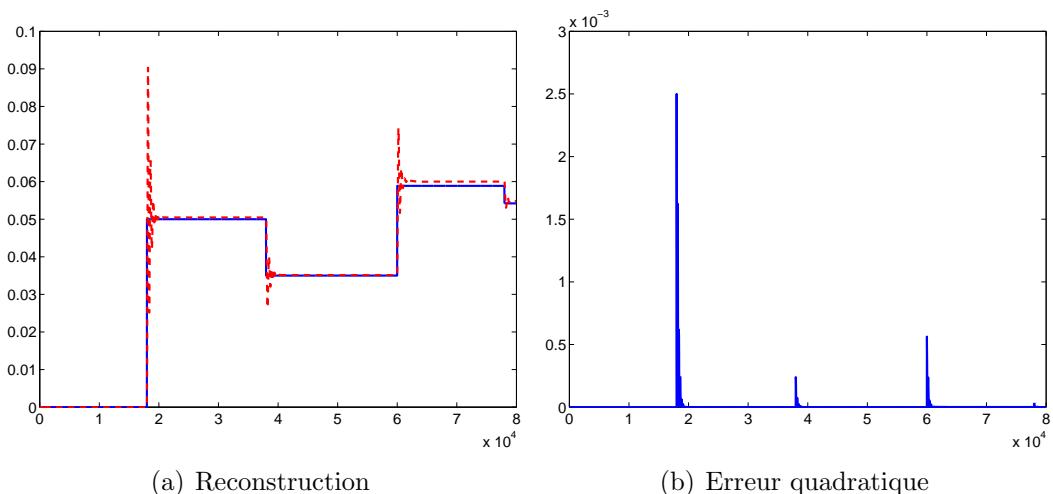


Fig. 2.18 - Comparaison de l'entrée du système non linéaire (courbe continue) avec l'entrée reconstruite (ligne pointillée)

La qualité de la reconstruction de l'entrée est donc satisfaisante. Toutefois, on remarque certaines différences entre l'entrée $u(t)$ réelle et l'entrée reconstruite \hat{u} malgré la précision du modèle de Volterra. De plus, l'apparition de pics dans la reconstruction de l'entrée peut gêner l'utilisateur si leur amplitude prend trop d'importance par rapport à celle du signal d'entrée réel. Une solution pourrait être le filtrage des composantes haute fréquence du signal reconstruit afin d'atténuer le plus possible ces "effets de bord" gênants.

Un avantage de cette approche par rapport à la méthode d'inversion par régularisation de Tikhonov est le faible temps de calcul qu'elle nécessite. En effet, le temps total nécessaire à la reconstruction de l'entrée $u(t)$ ne dépasse pas ici quelques dizaines de secondes. Cette différence s'explique en grande partie par l'utilisation nécessaire d'un algorithme

de programmation non linéaire, très coûteux en temps de calcul, qui ralentit donc la procédure de reconstruction de l'entrée par minimisation de critère. La première approche proposée dans cette partie ne possède pas cet inconvénient, ce qui permet de se rapprocher davantage de l'objectif de reconstruction de l'entrée du système non linéaire en temps réel.

Cette première application nous a donc permis de valider la méthode d'inversion proposée.

2.4.2.3 Influence du bruit de mesure sur la reconstruction

La sortie mesurée $y^*(t)$ du système non linéaire et un modèle de Volterra suffisamment précis permettent la reconstruction rapide de l'entrée par la méthode proposée. Cependant, nous avons déjà vu que toute mesure est inévitablement entachée de bruit. Afin de valider complètement la méthode proposée et de pouvoir la comparer à la méthode d'inversion par régularisation de Tikhonov, il est donc nécessaire d'étudier l'influence du bruit de mesure de la sortie sur la reconstruction \hat{u} de l'entrée $u(t)$.

On ajoute à la grandeur de sortie du système un bruit gaussien $b_2(t)$ (figure 2.19).

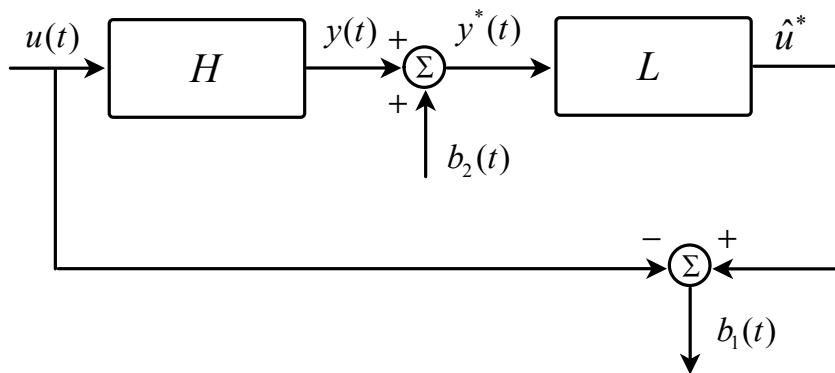


Fig. 2.19 - Ajout de bruit sur le signal de sortie du système

Le signal de sortie $y^*(t)$ utilisé pour reconstruire l'entrée est donc différent de celui utilisé en l'absence de bruit.

Le signal \hat{u}^* reconstruit par inversion du modèle de Volterra sera lui aussi différent à la fois du véritable signal d'entrée $u(t)$ et du signal reconstruit \hat{u} en l'absence de bruit. On peut noter ce signal reconstruit \hat{u}^* comme la somme du véritable signal d'entrée $u(t)$ et d'une erreur de reconstruction que l'on nomme $b_1(t)$ (figure 2.19).

$$\hat{u}^* = u(t) + b_1(t) \quad (2.164)$$

2.4. Inversion du modèle de Volterra par retour d'entrée reconstruite [George, 1959]

En l'absence de bruit de mesure, on a déjà montré que la reconstruction \hat{u} est identique à l'entrée $u(t)$ car la fonctionnelle L est l'inverse exacte de la fonctionnelle H .

$$\begin{aligned} y(t) &= H[u(t)] \\ \hat{u} &= u(t) = L[y(t)] \end{aligned} \quad (2.165)$$

En présence d'un bruit de mesure $b_2(t)$ ajouté au signal de sortie $y(t)$, la reconstruction \hat{u}^* ne sera pas exactement identique au signal d'entrée $u(t)$. On peut écrire :

$$\hat{u}^* = L[y^*(t)] = L[y(t) + b_2(t)] \quad (2.166)$$

On peut développer cette expression en la séparant en deux termes :

$$\hat{u}^* = L[H[u(t)] + b_2(t)] = u(t) + L[b_2(t)] \quad (2.167)$$

En considérant les expressions (2.164) et (2.167), on obtient finalement :

$$b_1(t) = L[b_2(t)] \quad (2.168)$$

Ceci confirme que l'erreur de reconstruction $b_1(t)$ dépend uniquement du bruit de mesure $b_2(t)$. Nous verrons par la suite quelle est l'influence de la fonctionnelle L (inverse exacte de H) sur l'amplitude de l'erreur de reconstruction en fonction de celle du bruit de mesure.

On reprend l'exemple du système non linéaire quadratique (2.162) mais en utilisant un signal de sortie bruité (rapport signal/bruit égal à 100) et on identifie à nouveau les paramètres du modèle de Volterra. La sortie bruitée du système quadratique et celle du modèle de Volterra sont présentées sur la figure 2.20.

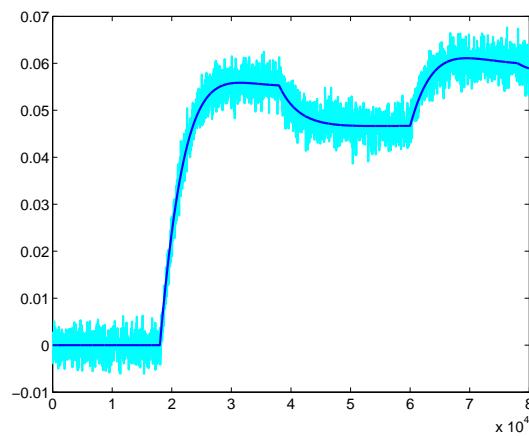


Fig. 2.20 - Sortie bruitée (courbe claire) et sortie du modèle de Volterra (courbe foncée)

Le résultat de la reconstruction \hat{u}^* de l'entrée $u(t)$ par la méthode présentée précédemment est donné sur la figure 2.21(a).

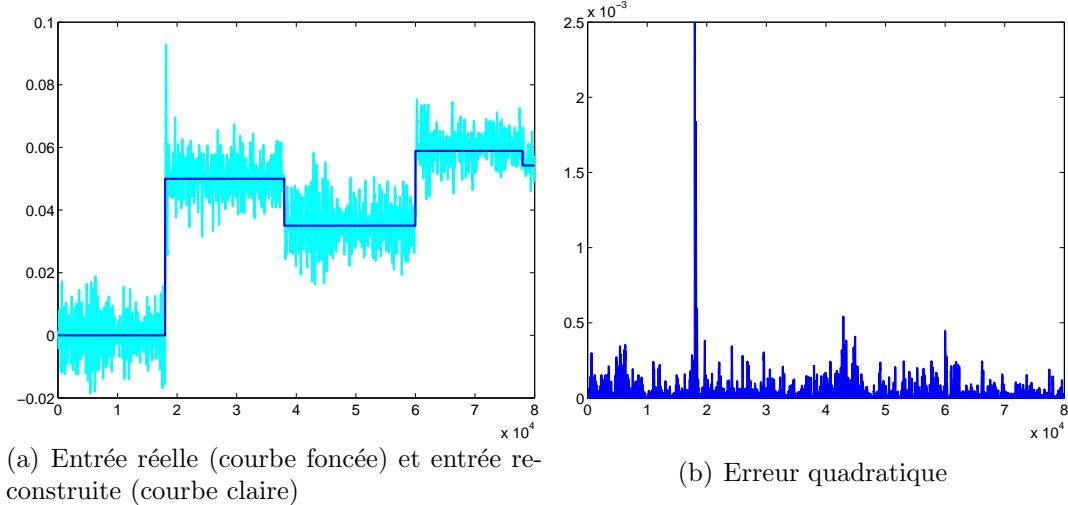


Fig. 2.21 - Comparaison de l'entrée du système non linéaire (courbe foncée) avec l'entrée reconstruite (courbe claire)

L'erreur quadratique entre les courbes de la figure 2.21(a) est représentée sur la figure 2.21(b).

La variance du bruit $b_2(t)$ ajouté au signal de sortie est égale à $5.38e - 6$ tandis que la variance du "bruit" qui entache le signal d'entrée reconstruit est égale à $4.28e - 5$. La procédure d'inversion a donc multiplié par un facteur d'ordre 10 le bruit de mesure ajouté au signal de sortie $y(t)$.

Malgré la présence de bruit et de pics de reconstruction, la qualité de reconstruction de l'entrée $u(t)$ est tout à fait correcte. Cette technique de reconstruction peut donc permettre de contourner le problème d'amplification de bruit dû à l'inversion de modèle.

2.4.3 Deuxième approche

2.4.3.1 Principe

On se place dans les mêmes conditions que la première approche : on utilise la forme de modèle de Volterra définie par la figure 1.28, où chaque terme $D_{ji}(a_{i,j}, p_{j,i})$ représente une fonction de transfert d'ordre $p_{j,i} \in \mathbb{N}$ (somme de fonctions de transfert d'ordre $k \in \mathbb{N}, k = 0, \dots, p_{j,i}$) et est donné par l'expression (1.132).

La deuxième approche de reconstruction de l'entrée $u(t)$ à partir de la sortie mesurée $y(t)$ et du modèle de Volterra $H[u(t)]$ repose également sur un schéma d'inversion en

2.4. Inversion du modèle de Volterra par retour d'entrée reconstruite [George, 1959]

boucle fermée présenté par la figure 2.22.

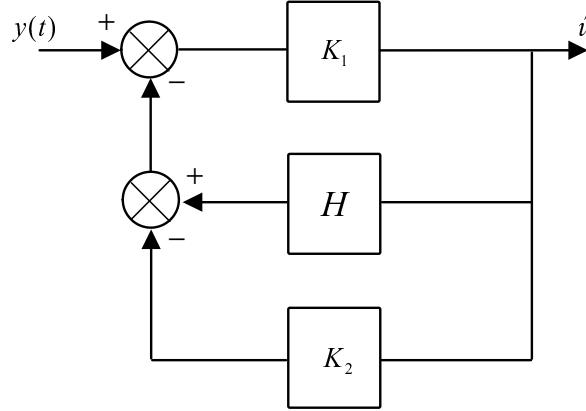


Fig. 2.22 - Schéma de reconstruction de l'entrée $u(t)$

Les termes K_1 et K_2 sont des constantes positives et l'on suppose que $K_1 = 1/K_2$. Nous verrons que le choix des valeurs de ces deux constantes est important pour la qualité de la reconstruction de l'entrée $u(t)$ et que ce choix peut s'apparenter à celui du paramètre λ de la méthode d'inversion par régularisation de Tikhonov.

La fonctionnelle H qui relie la sortie $y(t)$ du système non linéaire à son entrée $u(t)$ est définie par les expressions (2.149) et (2.150).

Ce schéma présente des caractéristiques similaires au schéma de la figure 2.14 étudié dans la section précédente. Nous verrons plus tard que, dans certains cas, ces deux schémas sont équivalents.

De même que la première approche étudiée, le schéma 2.22 est équivalent à la figure 2.23 où la fonctionnelle M relie l'entrée reconstruite \hat{u} à la sortie mesurée $y(t)$.

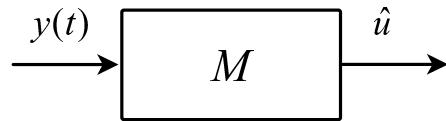


Fig. 2.23 - Schéma équivalent de reconstruction de l'entrée $u(t)$

La fonctionnelle M possède les mêmes propriétés que la fonctionnelle L et peut être développée en une somme de fonctionnelles :

$$M[y(t)] = \sum_{i=1}^{\infty} M_i[y(t)] \quad (2.169)$$

De même que dans l'approche précédente, on va montrer que la fonctionnelle M est l'inverse de la fonctionnelle de Volterra H .

On considère d'abord le cas où la sortie mesurée $y(t)$ n'est pas bruitée. D'après le schéma 2.22, l'entrée reconstruite peut s'écrire :

$$\hat{u} = K_1[y(t) - H[\hat{u}] + K_2[\hat{u}]] \quad (2.170)$$

Développons cette relation :

$$\hat{u} = K_1y(t) - K_1[H[\hat{u}]] + K_1K_2\hat{u} \quad (2.171)$$

Les termes K_1 et K_2 étant des coefficients, on a bien les relations $K_1[y(t)] = K_1y(t)$ et $K_2[\hat{u}] = K_2\hat{u}$.

étant donné que $K_1K_2 = 1$, on peut simplifier la relation (2.171) :

$$K_1y(t) = K_1[H[\hat{u}]] \quad (2.172)$$

On peut également simplifier par le coefficient K_1 et d'après le schéma équivalent 2.23, on a dans le cas d'un signal de sortie non bruité $y(t)$ la relation $\hat{u} = M[y(t)]$, que l'on peut introduire dans la relation (2.172) :

$$y(t) = H[M[y(t)]] \quad (2.173)$$

On en déduit que la fonctionnelle M est l'inverse de la fonctionnelle H . Le schéma 2.22 peut donc également être utilisé pour obtenir une reconstruction exacte \hat{u} de l'entrée $u(t)$ du système non linéaire à partir de la sortie $y(t)$ non bruitée et du modèle de Volterra.

2.4.3.2 Application à un système non linéaire

Afin de valider cette deuxième approche et de pouvoir la comparer à la première, on utilise les mêmes signaux d'entrée et de sortie (figures 2.16(a) et 2.16(b)) que pour la première approche. La période d'échantillonnage est toujours égale à 1s et la structure du modèle de Volterra reste la même : la série est tronquée à ses deux premiers termes et le nombre de paramètres est égal à 8.

Pour différents choix de valeur des coefficients K_1 et K_2 , la deuxième approche utilisée pour la reconstruction de l'entrée $u(t)$ donne les résultats figurant sur les courbes de la figure 2.24.

2.4. Inversion du modèle de Volterra par retour d'entrée reconstruite [George, 1959]

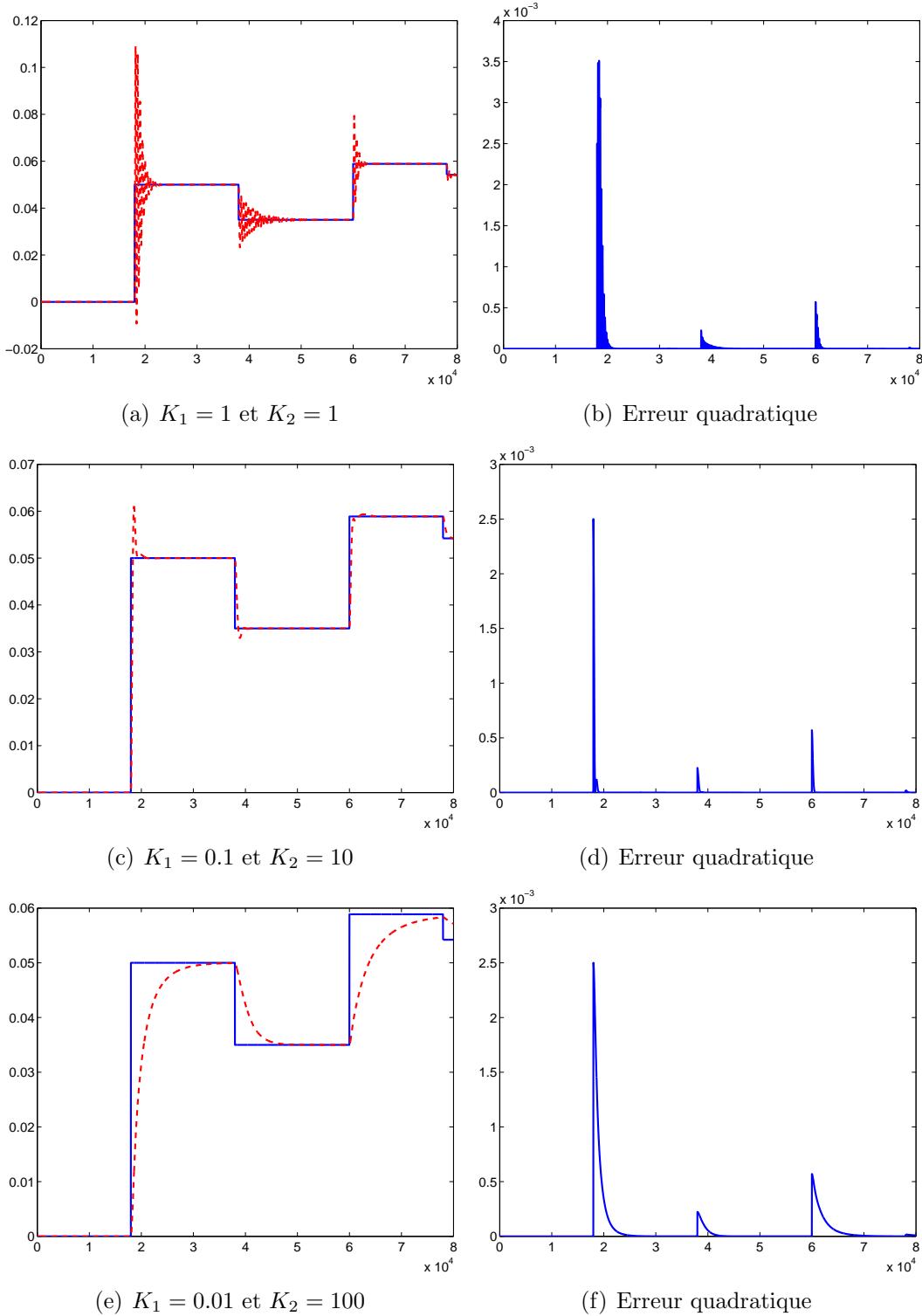


Fig. 2.24 - Entrée du système non linéaire (ligne continue) et entrée reconstruite (ligne pointillée) pour différentes valeurs de K_1 et K_2

La qualité de reconstruction varie selon les valeurs données aux coefficients K_1 et K_2 . Afin de mieux comprendre, on reprend l'expression (2.171) de l'entrée reconstruite et on

l'exprime de la manière suivante :

$$\hat{u} = K_1(y(t) - [H[\hat{u}]]) + K_1 K_2 \hat{u} \quad (2.174)$$

L'expression (2.174) est analogue à l'expression de l'entrée $u(t)$ reconstruite par la méthode de régularisation de Tikhonov. Le premier terme $K_1(y(t) - [H[\hat{u}]])$ représente la fidélité aux données dont on dispose, tandis que le second est un terme de régularisation de la solution \hat{u} . Faire varier les valeurs des coefficients K_1 et K_2 revient à accorder plus ou moins d'importance à l'un des deux termes de l'expression (2.174) : si la reconstruction est trop régularisée, on observe une perte d'information ; si le terme de régularisation est négligé, on observe des oscillations indésirables en haute fréquence. Les valeurs optimales de K_1 et K_2 semblent se situer autour de 0.1 et 10 respectivement. Cette seconde méthode d'inversion est donc comparable à celle par régularisation étudiée précédemment dans le chapitre.

On remarque cependant que le temps de calcul nécessaire à l'inversion du modèle de Volterra par cette deuxième méthode est équivalent à celui de la première méthode et reste largement inférieur au volume de calcul occasionné par la méthode d'inversion par régularisation de Tikhonov.

Cette application nous a donc permis de valider la deuxième méthode d'inversion proposée et de constater une analogie avec la méthode d'inversion par régularisation de Tikhonov.

2.4.3.3 Influence du bruit de mesure sur la reconstruction

En l'absence de bruit de mesure, la reconstruction \hat{u} est théoriquement identique à l'entrée $u(t)$ car la fonctionnelle M est l'inverse exacte de la fonctionnelle H .

$$\begin{aligned} y(t) &= H[u(t)] \\ \hat{u} &= u(t) = M[y(t)] \end{aligned} \quad (2.175)$$

De même que dans l'approche précédente, en présence d'un bruit de mesure $b_2(t)$ ajouté au signal de sortie $y(t)$, la reconstruction \hat{u}^* ne sera pas exactement identique au signal d'entrée $u(t)$. On peut écrire :

$$\hat{u}^* = M[y^*(t)] = M[y(t) + b_2(t)] \quad (2.176)$$

On peut développer cette expression :

$$\hat{u}^* = M[H[u(t) + b_2(t)]] \quad (2.177)$$

On sépare alors l'expression en 2 termes pour finalement obtenir :

$$\hat{u}^* = u(t) + M[b_2(t)] \quad (2.178)$$

2.4. Inversion du modèle de Volterra par retour d'entrée reconstruite [George, 1959]

Cette expression est tout-à-fait similaire à l'expression (2.167). On obtient finalement :

$$b_1(t) = M[b_2(t)] \quad (2.179)$$

De même que dans l'approche précédente, on a montré que l'erreur de reconstruction dépend de la fonctionnelle inverse M et du bruit de mesure $b_2(t)$.

En se plaçant dans les mêmes conditions que dans la première approche (même modèle de Volterra, bruit de mesure ajouté au signal de sortie $y(t)$), on obtient sur la figure 2.25(a) le résultat de reconstruction \hat{u}^* de l'entrée réelle $u(t)$ par la deuxième approche pour des valeurs optimales de K_1 et K_2 fixées à 0.1 et 10.

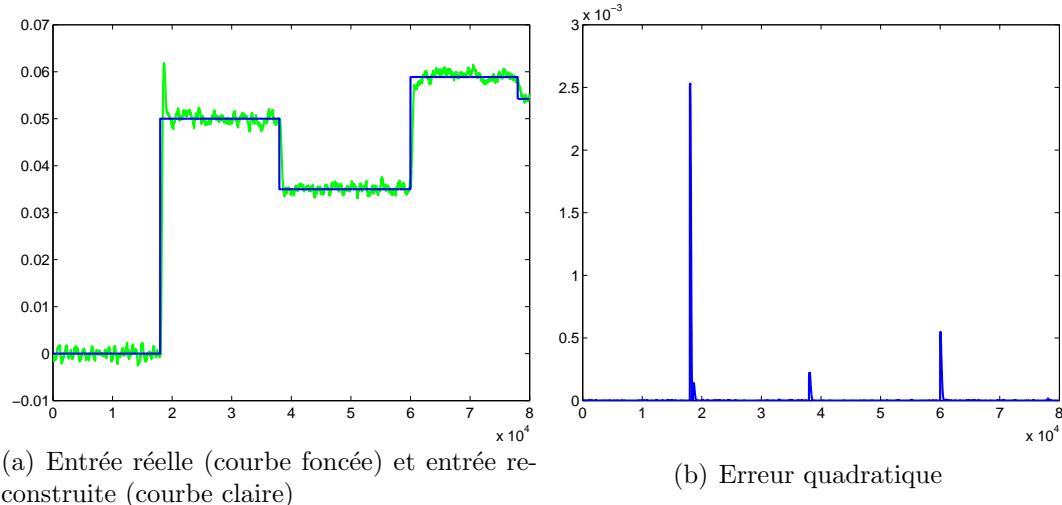


Fig. 2.25 - Comparaison de l'entrée du système non linéaire (courbe foncée) avec l'entrée reconstruite (courbe claire)

L'erreur quadratique entre les courbes de la figure 2.25(a) est représentée sur la figure 2.25(b).

La variance de l'erreur de reconstruction $b_1(t)$ est égale à $1.11e - 5$. Cette valeur est supérieure à celle de la variance du bruit de mesure $6.19e - 6$. Le bruit de mesure a donc été ici aussi amplifié par la procédure d'inversion du modèle de Volterra.

2.4.4 Comparaison des deux approches

Les deux approches proposées d'inversion du modèle de Volterra reposent sur deux schémas (2.14 et 2.22) en apparence différents mais qui peuvent se révéler équivalents dans un cas particulier.

Considérons le cas où les constantes K_1 et K_2 du schéma 2.22 sont égales à 1. Ce schéma est alors équivalent à celui de la figure 2.26.

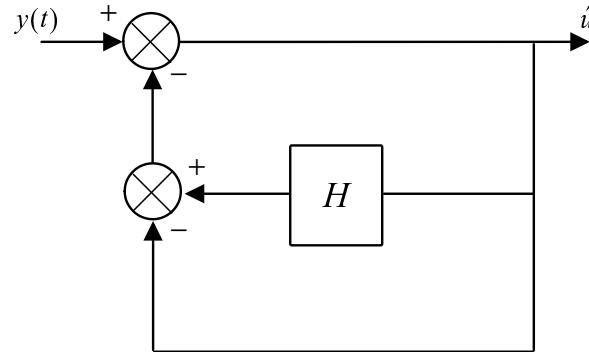


Fig. 2.26 - Schéma de reconstruction de l'entrée $u(t)$ par la deuxième approche avec $K_1 = K_2 = 1$

L'expression de l'entrée reconstruite \hat{u} est la suivante :

$$\hat{u} = y(t) + \hat{u} - H[\hat{u}] \quad (2.180)$$

Le schéma d'inversion de la première approche est celui de la figure 2.27.

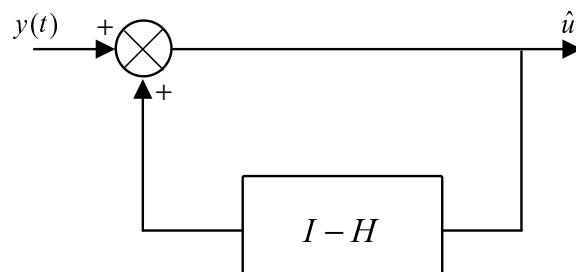


Fig. 2.27 - Schéma de reconstruction de l'entrée $u(t)$ par la première approche

L'expression de l'entrée reconstruite \hat{u} est dans ce cas :

$$\hat{u} = y(t) + I[\hat{u}] - H[\hat{u}] \quad (2.181)$$

Rappelons que l'expression de la fonctionnelle I est donnée par les expressions (2.151) et (2.152) :

$$I[u(t)] = \sum_{i=1}^{\infty} I_i[u(t)] \quad (2.182)$$

avec

$$I_i[u(t)] = \prod_{j=1}^i u(t) = u^i(t) \quad (2.183)$$

Par conséquent, les deux expressions (2.180) et (2.181) sont identiques si et seulement si on a l'égalité $I[\hat{u}] = \hat{u}$.

Cette égalité ne peut avoir lieu que si I est une fonctionnelle **linéaire**. Par conséquent, les deux schémas d'inversion 2.14 et 2.22 sont identiques uniquement dans le cas d'un système linéaire (et dans l'hypothèse où les constantes K_1 et K_2 sont égales à 1).

On a cependant constaté que, dans le cas d'un système non linéaire, les deux approches d'inversion fournissent des résultats satisfaisants et comparables.

2.4.5 Conclusion

Les deux approches d'inversion d'un modèle de Volterra que nous venons de présenter sont semblables dans leur principe mais ne sont véritablement équivalentes que dans le cas d'un système linéaire.

Si l'on considère un système non linéaire, les deux approches fournissent des résultats d'inversion comparables. La première méthode présente un avantage par rapport à la technique d'inversion par régularisation de Tikhonov : l'absence de paramètre de régularisation. On retrouve cependant sur le signal d'entrée reconstruit une erreur de reconstruction dont la variance est supérieure à celle du bruit de mesure. La seconde méthode permet une reconstruction de meilleure qualité que celle obtenue par la première méthode mais elle ne s'affranchit pas d'une des difficultés de la méthode de régularisation de Tikhonov, à savoir le choix des valeurs de certains paramètres afin de régulariser le signal reconstruit.

Un avantage commun aux deux méthodes proposées est le temps de calcul, sensiblement inférieur à celui nécessaire par la méthode d'inversion par régularisation de Tikhonov : il est égal à quelques dizaines de secondes et dépend de l'importance des fichiers de mesures utilisés et de la complexité du modèle de Volterra à inverser. Un des objectifs de ce travail étant la conservation de la simplicité des modèles de Volterra et de leur parcimonie paramétrique, le temps de calcul dépend à la fois du modèle et des fichiers de points de mesure. De plus, la procédure d'optimisation paramétrique, nécessaire à chaque itération lors de l'inversion par régularisation de Tikhonov et par conséquent très coûteuse en temps de calcul, est ici effectuée seulement une fois afin de modéliser correctement le système avant d'inverser le modèle. Ceci explique également la diminution notable du temps de

calcul nécessité par les deux méthodes d'inversion par retour d'entrée reconstruite.

2.5 Conclusion du chapitre

L'inversion du modèle d'un système réel est un problème largement étudié et élucidé dans le cas linéaire. On peut cependant considérer que la résolution de problèmes inverses linéaires n'est dans certains cas que la première étape d'un domaine de recherche plus vaste et plus général, celui de la résolution de problèmes inverses non linéaires.

Dans le premier chapitre de ce mémoire, nous avons proposé la représentation de systèmes non linéaires par les séries de Volterra, modèle non linéaire que l'on peut considérer comme une généralisation du produit de convolution utilisé pour représenter un système linéaire. L'inversion d'un produit de convolution a fait l'objet de nombreuses études et une solution particulièrement connue est la méthode d'inversion par régularisation de Tikhonov, qui permet entre autres d'atténuer les effets des perturbations dues au bruit de mesure.

Nous avons dans un premier temps proposé d'appliquer la méthode d'inversion par régularisation de Tikhonov au modèle de Volterra décrit dans le premier chapitre. Cette démarche correspond à l'approche de généralisation d'une méthode qui a fait ses preuves dans le domaine linéaire. La reconstruction de l'entrée du système étudié par minimisation d'un critère composite nécessite la simulation de la sortie du modèle et l'optimisation des paramètres par un algorithme de programmation non linéaire, deux procédures très coûteuses en temps de calcul. Nous avons donc utilisé une méthode de discréétisation du modèle de Volterra afin de diminuer le temps de calcul du vecteur gradient et de la matrice hessienne, couplée à un algorithme d'optimisation et de simulation sur fenêtre glissante, qui permettent de considérablement diminuer le temps de calcul global nécessaire à l'inversion du modèle de Volterra.

Le temps de calcul reste cependant trop important et une exploitation en temps réel de la méthode d'inversion du modèle de Volterra par régularisation de Tikhonov ne peut être envisagée que pour des systèmes lents tels certains procédés chimiques. Nous avons donc présenté et étudié deux autres méthodes d'inversion du modèle de Volterra par retour d'entrée reconstruite qui permettent d'éviter la procédure d'optimisation de critère quadratique et ainsi de gagner un temps de calcul important. Ces deux méthodes nécessitent cependant une modélisation précise du système étudié ainsi qu'une identification paramétrique rigoureuse, conditions qui sont difficilement remplies si la non-linéarité du système réel étudié est importante, provoquant des instabilités numériques lors de la procédure de reconstruction de la grandeur d'entrée.

Les applications présentées dans le chapitre suivant vont permettre d'appliquer et de tester les différentes méthodes d'inversion présentées dans ce chapitre et de constater les qualités et défauts de chacune dans le cas d'un système simulé (oxydation d'un composé organique par le procédé Fenton) et d'un système réel (élimination de composés organiques par digestion anaérobie).

Chapitre 3

Application à des procédés

3.1 Introduction

Les différents modèles de Volterra présentés dans le chapitre 1 et les méthodes d'inversion étudiées dans le chapitre 2 vont à présent être testés en simulation sur un procédé chimique d'oxydation de polluant puis validés sur un procédé biochimique expérimental de digestion anaérobie.

Les procédés chimiques et biochimiques ont été utilisés par l'homme bien avant que les notions de génie chimique ou biochimique ne soient apparues, même si les installations ou les appareils utilisés avaient fait l'objet d'une réflexion avant leur conception. Le génie chimique est né avec la notion de modélisation, nécessaire à la compréhension et à la reproduction de phénomènes chimiques ou biochimiques, et déjà largement employée dans le domaine de la physique. Afin de modéliser les opérations unitaires qui constituent un procédé complet, le génie chimique fait appel aux principes fondamentaux de conservation (matière, énergie, quantité de mouvement) ainsi qu'à d'autres branches scientifiques telles la thermodynamique et la mécanique des fluides, sans oublier les mathématiques. L'extension des concepts du génie chimique à des domaines connexes pouvant relever de la même méthodologie a permis de définir, entre autres, le génie biochimique. Ces catégories font partie de l'ensemble plus vaste du génie des procédés [Corriou, 2001].

Les deux études présentées dans ce chapitre relèvent donc du génie des procédés et plus particulièrement du génie des procédés de traitement des eaux. La première partie est consacrée à l'étude d'un simulateur d'oxydation de composé organique considéré comme polluant (l'atrazine) par le procédé Fenton. La deuxième partie est consacrée à l'étude d'un procédé biochimique réel de digestion anaérobie.

3.2 Généralités à propos du traitement des eaux

Le traitement des eaux, et en particulier la neutralisation de polluants dans les eaux résiduaires industrielles, est actuellement un domaine de recherche ouvert [ESIP, 2000]. Les activités industrielles génèrent selon le ou les types de fabrication des rejets polluants continus ou discontinus d'une extrême diversité. La finalité du traitement des eaux résiduaires industrielles est essentiellement la protection du milieu naturel, c'est-à-dire l'obtention d'une eau épurée qui satisfait aux normes de rejet édictées par la législation, mais aussi la réutilisation en usine de l'eau traitée, en particulier pour le refroidissement et certains lavages. Selon la nature ou l'importance de la pollution, différents procédés de neutralisation peuvent être envisagés et mis en oeuvre. Il faut également noter que, pour un même type de pollution, différentes solutions peuvent être envisagées [Boeglin, 1997].

D'une manière générale, une installation de traitement centralisée des rejets industriels comporte classiquement la succession de stades de traitement suivants :

- une série de traitements préliminaires, souvent liée au type d'activité industrielle : **rétention par dégrillage automatique et tamisage** des matières volumineuses

susceptibles de gêner les étapes ultérieures de traitement, **dessablage** assurant l'élimination des particules grossières par décantation ou encore **élimination des produits insolubles de faible densité** (huiles, graisses) par flottation.

- des traitements physico-chimiques, qui assurent la séparation physique des insolubles solides (matières en suspension) et liquides (substances huileuses, hydrocarbures...) de l'eau avec pour objectif une clarification plus ou moins poussée des rejets. Ce traitement peut, suivant les cas, constituer un stade intermédiaire ou un stade final du traitement. Il suppose un maintien du pH dans une zone assez étroite. Selon les circonstances, l'épuration physico-chimique peut être simultanée à d'autres processus comme la **neutralisation** des rejets, l'**oxydation** et la **réduction** de certains composés.
- des traitements biologiques, qui constituent le mode classique d'épuration de la pollution organique des eaux résiduaires industrielles. Les procédés les plus couramment mis en oeuvre pour la dépollution des rejets industriels sont de type **aérobie** (présence d'air ou d'oxygène), car la cinétique du processus s'avère beaucoup plus rapide et les rendements plus élevés qu'avec les traitements biologiques **anaérobies**. Selon les cas, on distingue les procédés aérobies utilisant une culture bactérienne libre en suspension de ceux qui utilisent une culture bactérienne fixée sur un support. Mais il faut noter que l'élimination de la pollution organique concentrée de certains rejets industriels biodégradables (en particulier dans l'industrie agroalimentaire) se fait de plus en plus par des procédés anaérobies de méthanisation pouvant également fonctionner avec une biomasse en suspension ou fixée sur un support.
- des traitements de finition, ou complémentaires, qui permettent d'obtenir une qualité d'effluent supérieure à celle obtenue par les procédés physico-chimiques et/ou biologiques classiques : il s'agit d'affiner l'eau en poussant l'épuration le plus loin possible par amélioration des performances des paramètres classiques et par l'action spécifique sur des paramètres qui ne sont que peu ou pas du tout affectés par les traitements classiques.

3.3 Etude en simulation : procédé d'oxydation de polluant

Le type de procédé de traitement des eaux que nous allons étudier ici en simulation correspond à un traitement physico-chimique de neutralisation de polluant par oxydation. Ce traitement peut constituer dans notre cas le stade final du procédé de neutralisation d'un polluant particulier ou bien un stade intermédiaire d'un procédé global d'épuration complète d'effluent industriel. Le procédé de traitement de l'eau mis en oeuvre est le procédé Fenton qui consiste en la neutralisation par oxydation d'un composé organique considéré comme polluant [Barbeni *et al.*, 1987], [Sedlak et Andren, 1991],

[DeLaat et Gallard, 1999].

3.3.1 Description du procédé étudié

A l'intérieur d'un réacteur chimique, on mélange de façon homogène à l'effluent à traiter un certain nombre de réactifs chimiques. L'effluent à traiter est de l'eau contenant une ou plusieurs sortes de composés organiques considérés comme polluants. Le procédé étudié consiste à neutraliser un composé organique particulier.

Dans cette étude, le composé organique considéré est l'atrazine (2-chloro-4-éthylamino-6-isopylamino-1,3,5-triazine), un herbicide systémique principalement utilisé pour maîtriser les mauvaises herbes gramoïdes et à feuilles larges dans les champs de maïs, de colza et de bleuet nain, et pour détruire les mauvaises herbes en général. L'atrazine peut avoir des effets sur la santé des personnes exposées à des concentrations supérieures à celles recommandées. Une exposition à des concentrations élevées d'atrazine dans l'eau potable peut provoquer des nausées et des étourdissements. L'atrazine est toxique pour les poissons et les invertébrés d'eau douce et plus particulièrement pour les plantes aquatiques. La famille des triazines est le pesticide le plus employé. Introduits en 1962, ce sont des produits assez bon marché et simples à utiliser. Ils sont utilisés massivement par les producteurs de maïs conventionnel qui traitent la plus grande partie de leurs surfaces à l'atrazine. Les triazines sont les substances les plus fréquemment rencontrées dans les nappes d'eau souterraines et dans les eaux de surfaces. En Bretagne, comme dans le Sud-Ouest et l'Île de France, il est courant de trouver, dans des prélèvements d'eau potable, des taux de triazine dix fois plus élevés que le seuil autorisé de 0,1 microgramme par litre. L'atrazine et toute la famille des triazines ont été bannies par l'Allemagne en mars 2001, et depuis le 28 septembre 2001 par la France. Cette décision devait prendre effet le 30 juin 2003.

La neutralisation d'un composé organique (par exemple l'atrazine) consiste en son oxydation par le radical hydroxyle OH° suivant la réaction chimique :



Le produit *Pi* obtenu après réaction pourra à nouveau réagir avec un radical hydroxyle OH° mais on ne le considère plus comme un polluant. Cette seule réaction chimique suffit donc à neutraliser un composé organique considéré comme polluant. Un autre problème, situé en amont de celui-ci, est la production de ces radicaux hydroxyles OH° extrêmement réactifs.

Fenton a montré en 1894 que l'ion ferreux, dont il existe cinq formes ($[Fe^{II}]^{2+}$, $[Fe^{II}OH]^+$, $[Fe^{II}(OH)_2]^\circ$, $[Fe^{II}(OH)_3]^-$ et $[Fe^{II}(OH)_4]^{2-}$), accélérerait l'oxydation de l'acide maléique par le peroxyde d'hydrogène H_2O_2 . Depuis cette date, l'étude de la décomposition du peroxyde d'hydrogène H_2O_2 par les ions ferreux et ferriques ainsi que l'oxydation de composés organiques par le système $Fe(II)/H_2O_2$ (connu sous le nom de réactif de Fenton) a fait

l'objet de nombreux travaux. Une des réactions produisant des radicaux hydroxyles est la suivante :



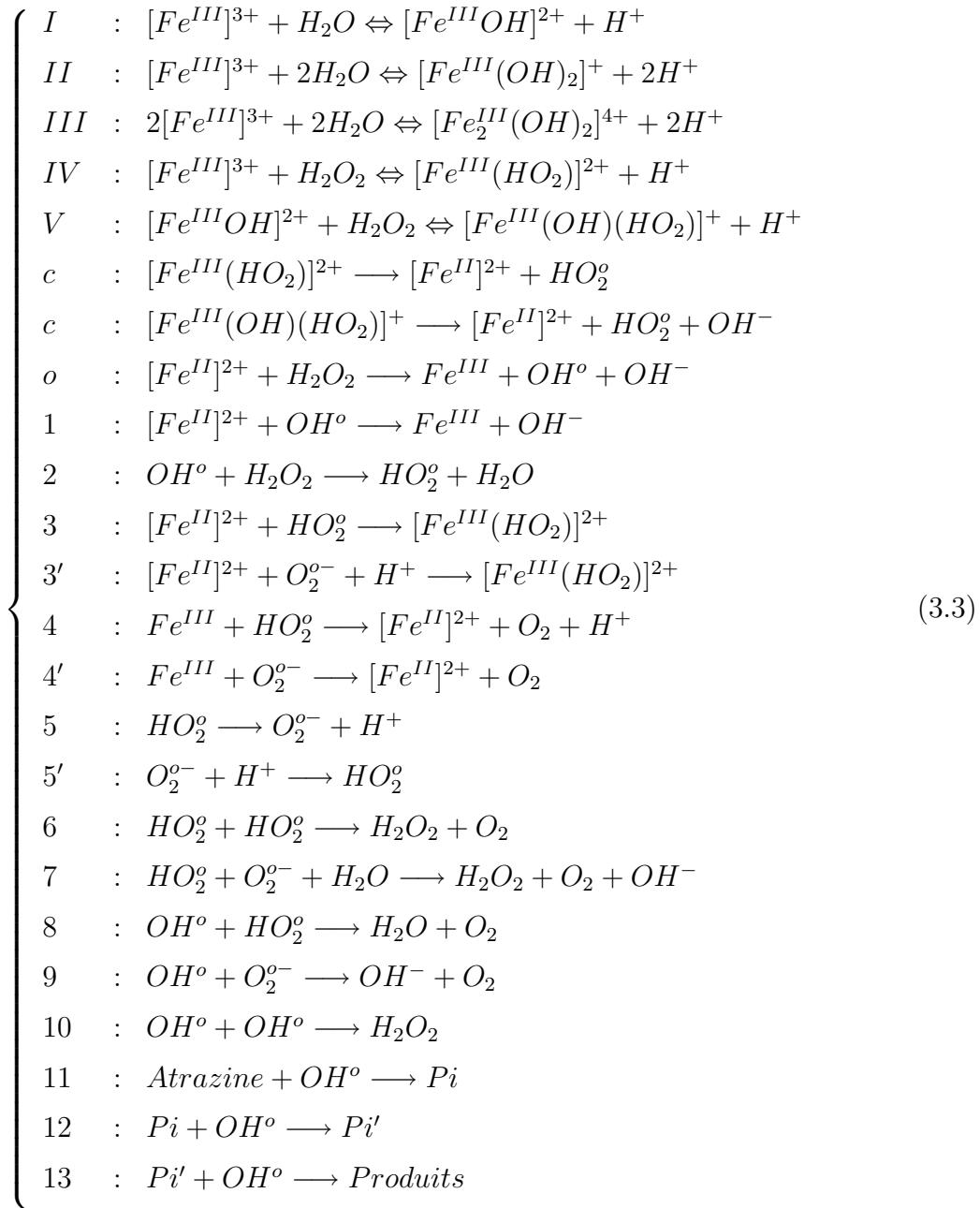
Les réactifs utilisés lors du procédé Fenton sont donc les ions ferreux $Fe(II)$ (ou ferriques $Fe(III)$) et le peroxyde d'hydrogène H_2O_2 qui, en réagissant entre eux, produisent les radicaux hydroxyles OH° nécessaires à l'oxydation du (ou des) composé(s) organique(s) considéré(s) comme polluant(s).

De nombreuses études ont montré qu'une irradiation UV-visible pouvait augmenter de façon importante les vitesses d'oxydation des composés organiques par le réactif de Fenton. Le système oxydant $Fe(II)/H_2O_2$ auquel se rajoute l'irradiation est donc communément appelé Photo-Fenton.

Des installations industrielles utilisant le procédé Fenton existent. Cependant, pour s'assurer que la concentration en polluant en sortie de réacteur respecte les normes de rejet, les réactifs en entrée sont introduits en excès. Cette étude s'inscrit dans un projet du programme Eaux du 12^{ème} Contrat de Plan état-Région de l'Université de Poitiers dont l'objectif à terme est d'optimiser le dosage en fer ferreux (ou ferrique) et en peroxyde d'hydrogène en entrée du réacteur chimique.

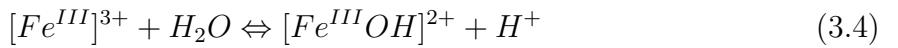
3.3.2 Modèle cinétique établi expérimentalement

A partir de connaissances expérimentales sur le mécanisme d'oxydation de l'atrazine par le procédé Photo-Fenton, un modèle cinétique du taux d'oxydation de l'atrazine par le système $Fe(II)/H_2O_2$ a pu être établi [Haber et Weiss, 1934], [Barb *et al.*, 1951], [Walling, 1975], [Gallard et DeLaat, 2000] :

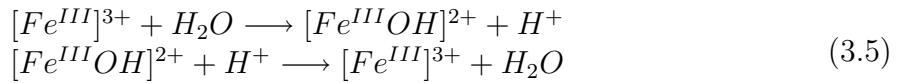


Ce modèle cinétique prend en compte les réactions d'hydrolyse de $Fe(III)$ (réactions I à III), les réactions de complexation de $Fe(III)$ avec H_2O_2 (réactions IV et V), la réaction de décomposition des hydroperoxydes de fer en $Fe(II)$ et HO_2^{\cdot} (réaction c) qui représente l'étape d'initiation du mécanisme de décomposition de H_2O_2 par $Fe(III)$, les réactions de propagation (réactions o, 1 à 5), les réactions de terminaison (réactions 6 à 10) et les réactions d'oxydation de l'atrazine et du produit Pi de l'oxydation (réactions 11 à 13).

Les 5 premières réactions de la liste (hydrolyse et complexation) sont des équilibres chimiques. En d'autres termes, si l'on prend le cas de la première réaction de la liste



les deux réactions suivantes ont lieu :



Les 19 autres réactions chimiques sont complètes. Chacune de ces réactions est caractérisée par une vitesse v de réaction et, plus particulièrement, par une constante de réaction k dont nous allons voir le principe de calcul.

On considère la réaction chimique complète suivante :



A et B sont les réactifs et C et D sont les produits de la réaction. La vitesse v de cette réaction chimique dépend de facteurs cinétiques tels la température et la concentration des réactifs. On peut la définir de la manière suivante :

$$v = k[A]^a[B]^b \quad (3.7)$$

Le terme k est la constante de la réaction. $a + b$ est appelé ordre de la réaction. k , a et b sont déterminés expérimentalement.

Dans le cas d'un équilibre chimique, on définit donc deux vitesses v et v' ainsi que deux constantes de réaction k et k' , une pour chaque "sens" de réaction :

$$\begin{cases} v = k[A]^a[B]^b \\ v' = k'[C]^c[D]^d \end{cases} \quad (3.8)$$

On définit ensuite la constante d'équilibre $K = \frac{k}{k'}$ en fonction de la concentration des réactifs et des produits ainsi que des coefficients a , b , c et d (loi de Guldberg et Waage) :

$$K = \frac{[C]^c[D]^d}{[A]^a[B]^b} \quad (3.9)$$

Dans le cas des équilibres chimiques, si la constante K est grande, on tend vers une réaction complète (disparition des réactifs A et B) tandis que si K est faible, on tend vers une réaction impossible (non apparition des produits C et D). Dans le cas des réactions complètes, la valeur de la constante k permet de calculer la vitesse d'apparition des produits ou de disparition des réactifs.

Un récapitulatif des constantes de chaque réaction du modèle cinétique établi expérimentalement est présenté en (3.10) et (3.11).

Constantes associées aux équilibres chimiques :

$$\begin{array}{lll} k_I = 2.291 \times 10^7 & k'_I = 1 \times 10^{10} & K_1 = 2.291 \times 10^{-3} \\ k_{II} = 4.571 \times 10^3 & k'_{II} = 1 \times 10^{10} & K_2 = 4.571 \times 10^{-7} \\ k_{III} = 1.122 \times 10^7 & k'_{III} = 1 \times 10^{10} & K_{2,2} = 1.122 \times 10^{-3} \\ k_{IV} = 3.1 \times 10^7 & k'_{IV} = 1 \times 10^{10} & K_{I_1} = 3.1 \times 10^{-3} \\ k_V = 2.0 \times 10^6 & k'_V = 1 \times 10^{10} & K_{I_2} = 2.0 \times 10^{-4} \end{array} \quad (3.10)$$

Constantes associées aux réactions chimiques :

$$\begin{array}{lll} k_c = 2.7 \times 10^{-3} & k_o = 63.0 & k_1 = 3.2 \times 10^8 \\ k_2 = 3.3 \times 10^7 & k_3 = 1.2 \times 10^6 & k'_3 = 1.0 \times 10^7 \\ k_4 = 1 \times 10^4 & k'_4 = 5 \times 10^7 & k_5 = 1.58 \times 10^5 \\ k'_5 = 1 \times 10^{10} & k_6 = 8.3 \times 10^5 & k_7 = 9.7 \times 10^7 \\ k_8 = 0.71 \times 10^{10} & k_9 = 1.01 \times 10^{10} & k_{10} = 5.2 \times 10^9 \\ k_{At} = (1.2 - 3) \times 10^9 & k_{Pi} \approx k_{At} & k'_{Pi} < 10^5 \end{array} \quad (3.11)$$

Une fois établi le modèle cinétique du taux d'oxydation de l'atrazine par le système $Fe(II)/H_2O_2$, on peut le transformer en un système d'équations différentielles du premier ordre traduisant la variation de concentration de chaque espèce et sous-espèce chimiques présentes à un moment ou à un autre lors du processus :

$$\left\{
 \begin{aligned}
 \frac{d[Fe(II)]}{dt} &= k_c([I_1] + [I_2]) - k_0.[Fe(II)].[H_2O_2] - k_1.[Fe(II)].[OH^o] \\
 &\quad - k_3.[Fe(II)].[HO_2^o] - k'_3.[Fe(II)].[O_2^{o-}] + k_4.[Fe(III)].[HO_2^o] \\
 &\quad + k'_4.[Fe(III)].[O_2^{o-}] \\
 \frac{d[Fe(III)_T]}{dt} &= -\frac{d[Fe(II)]}{dt} - k_{II}[Fe(III)] + k'_{II}[Fe(III)OH][H^+] \\
 \frac{d[H_2O_2]}{dt} &= -k_0.[Fe(II)].[H_2O_2] - k_2.[OH^o].[H_2O_2] + k_6.[HO_2^o].[HO_2^o] \\
 &\quad + k_7.[O_2^{o-}].[HO_2^o] + k_{10}.[OH^o].[OH^o] - k_{III}[Fe(III)][H_2O_2] \\
 &\quad + k'_{III}[Fe(III)HO_2][H^+] \\
 \frac{d[HO_2^o]}{dt} &= k_c([I_1] + [I_2]) + k_2.[OH^o].[H_2O_2] - k_3.[Fe(II)].[HO_2^o] \\
 &\quad - k_4.[Fe(III)].[HO_2^o] - k_5.[HO_2^o] + k'_5.[O_2^{o-}].[H^+] \\
 &\quad - 2k_6.[HO_2^o].[HO_2^o] - k_7.[O_2^{o-}].[HO_2^o] - k_8.[OH^o].[HO_2^o] \\
 \frac{d[O_2^{o-}]}{dt} &= -k'_3.[Fe(II)].[O_2^{o-}] - k'_4.[Fe(III)].[O_2^{o-}] + k_5.[HO_2^o] \\
 &\quad - k'_5.[O_2^{o-}].[H^+] - k_7.[O_2^{o-}].[HO_2^o] - k_9.[O_2^{o-}].[OH^o] \\
 \frac{d[OH^o]}{dt} &= k_0.[Fe(II)].[H_2O_2] - k_1.[Fe(II)].[OH^o] - k_2.[OH^o].[H_2O_2] \\
 &\quad - k_8.[OH^o].[HO_2^o] - k_9.[O_2^{o-}].[OH^o] - 2k_{10}.[OH^o].[OH^o] \\
 &\quad - k_{At}.[At].[OH^o] - k_{Pi}.[Pi].[OH^o] \\
 \frac{d[At]}{dt} &= -k_{At}.[At].[OH^o] \\
 \frac{d[Pi]}{dt} &= k_{At}.[At].[OH^o] - k_{Pi}.[Pi].[OH^o]
 \end{aligned}
 \right. \tag{3.12}$$

Dans la première équation différentielle de la liste, les concentrations $[I_1] = [Fe(III)(HO_2)^{2+}]$ et $[I_2] = [Fe(III)(OH)(HO_2)^{+}]$ des complexes en $Fe(III)$ dépendent des concentrations en $Fe(III)$ et H_2O_2 , et sont calculées analytiquement en utilisant les constantes d'équilibre de leur formation (KI_1 et KI_2) ainsi que celles des autres réactions de formation des différentes espèces de $Fe(III)$ (K_1 , K_2 et $K_{2,2}$).

A chaque instant de la réaction entre H_2O_2 , $Fe(III)$ et $Fe(II)$, les concentrations des différentes espèces de fer peuvent être calculées à partir des constantes d'équilibre :

$$\begin{aligned} K_1 &= \frac{[Fe(OH)^{2+}][H^+]}{[Fe^{3+}]} \\ K_2 &= \frac{[Fe(OH)_2^+][H^+]^2}{[Fe^{3+}]} \\ K_{2,2} &= \frac{[Fe_2(OH)_2^{4+}][H^+]^2}{[Fe^{3+}]^2} \\ KI_1 &= \frac{[I_1][H^+]}{[Fe^{3+}][H_2O_2]} \\ KI_2 &= \frac{[I_2][H^+]}{[Fe(OH)^{2+}][H_2O_2]} \end{aligned} \quad (3.13)$$

En considérant l'équation de conservation de la matière

$$[Fe]_T = [Fe(III)] + [I_1] + [I_2] + [Fe(II)] \quad (3.14)$$

où le terme $[Fe]_T$ représente la quantité totale de fer, et en exprimant chaque forme de fer ferrique par son équation en fonction de la forme $[Fe^{III}]^{3+}$, on obtient la relation suivante :

$$\begin{aligned} [Fe(III)] = & [Fe^{3+}] + \frac{K_1[Fe^{3+}]}{[H^+]} + \frac{K_2[Fe^{3+}]}{[H^+]^2} + \frac{2K_{2,2}[Fe^{3+}]^2}{[H^+]^2} + \frac{KI_1[Fe^{3+}][H_2O_2]}{[H^+]} \\ & + \frac{K_1KI_2[Fe^{3+}][H_2O_2]}{[H^+]^2} \end{aligned} \quad (3.15)$$

La racine positive de cette équation correspond à la concentration en $[Fe^{III}]^{3+}$ et dépend des concentrations initiales $[H^+]$, $[Fe(III)]$ et H_2O_2 :

$$[Fe^{3+}] = \frac{-\left(1 + \frac{K_1}{[H^+]} + \frac{K_2}{[H^+]^2} + \frac{KI_1[H_2O_2]}{[H^+]} + \frac{K_1KI_2[H_2O_2]}{[H^+]^2}\right) + \sqrt{\Delta Fe}}{\frac{4K_{2,2}}{[H^+]^2}} \quad (3.16)$$

avec

$$\Delta Fe = \left(1 + \frac{K_1}{[H^+]} + \frac{K_2}{[H^+]^2} + \frac{KI_1[H_2O_2]}{[H^+]} + \frac{K_1KI_2[H_2O_2]}{[H^+]^2}\right)^2 + 8[Fe(III)]\frac{K_{2,2}}{[H^+]^2} \quad (3.17)$$

On calcule ensuite les concentrations des autres formes de fer ferrique à partir des équations (3.13), dont $[I_1]$ et $[I_2]$ qui nous intéressent plus particulièrement :

$$\begin{aligned}[I_1] &= \frac{KI_1[Fe^{3+}][H_2O_2]}{[H^+]} \\ [I_2] &= \frac{KI_2K_1[Fe^{3+}][H_2O_2]}{[H^+]^2}\end{aligned}\tag{3.18}$$

avec $[H^+] = 10^{-pH}$.

L'influence du pH sur la vitesse de décomposition de l'atrazine a été étudiée dans un domaine de valeurs de pH compris entre 1 et 3 avec différentes concentrations initiales en fer ferreux et en peroxyde d'hydrogène H_2O_2 . Les résultats expérimentaux concernant l'évolution de la concentration en composé organique et du fer ferreux en fonction du temps de réaction montrent que la vitesse d'élimination de l'atrazine augmente lorsque le pH passe de 1 à 3, vraisemblablement en raison d'une différence de réactivité des radicaux hydroxyle OH° vis-à-vis de l'atrazine. D'autres résultats montrent que la vitesse de décomposition de H_2O_2 passe par une valeur optimale pour une valeur de pH située autour de 3. Le pH à l'intérieur du réacteur chimique sera donc maintenu autour de la valeur 3 durant tout le temps de réaction.

3.3.3 Modèle cinétique d'un réacteur chimique continu

3.3.3.1 Présentation

Le modèle cinétique du réacteur chimique utilisé par la suite s'inspire du système d'équations différentielles (3.12) pour décrire à chaque instant la concentration de 8 différentes espèces chimiques présentes lors du procédé d'oxydation : concentration en Fer II (variable X_1), concentration totale en Fer III (variable X_2), concentration en H_2O_2 (variable X_3), concentration en HO_2° (variable X_4), concentration en $O_2^{\circ-}$ (variable X_5), concentration en radicaux OH° (variable X_6), concentration en atrazine (variable X_7) et en produit dérivé de son oxydation (variable X_8).

Le modèle proposé (d'après [Gallard et DeLaat, 2000]) est donné par l'ensemble d'équations (3.19).

$$\left\{ \begin{array}{l} \dot{X}_1 = k_c([I_1] + [I_2]) - k_0.X_1.X_3 - k_1.X_1.X_6 \\ \quad - k_3.X_1.X_4 - k'_3.X_1.X_5 + k_4.[Fe(III)].X_4 \\ \quad + k'_4.[Fe(III)].X_5 + \frac{X_1^0.Q_{1in}}{V_R} - \frac{X_1.Q_{out}}{V_R} \\ \dot{X}_2 = -\dot{X}_1 + \frac{X_2^0.Q_{2in}}{V_R} - \frac{X_2.Q_{out}}{V_R} \\ \dot{X}_3 = -k_0.X_1.X_3 - k_2.X_6.X_3 + k_6.X_4.X_4 + k_7.X_5.X_4 \\ \quad + k_{10}.X_6.X_6 + \frac{X_3^0.Q_{3in}}{V_R} - \frac{X_3.Q_{out}}{V_R} \\ \dot{X}_4 = k_c([I_1] + [I_2]) + k_2.X_6.X_3 - k_3.X_1.X_4 \\ \quad - k_4.[Fe(III)].X_4 - k_5.X_4 + k'_5.X_5.[H^+] \\ \quad - 2k_6.X_4.X_4 - k_7.X_5.X_4 - k_8.X_6.X_4 \\ \dot{X}_5 = -k'_3.X_1.X_5 - k'_4.[Fe(III)].X_5 + k_5.X_4 \\ \quad - k'_5.X_5.[H^+] - k_7.X_5.X_4 - k_9.X_5.X_6 \\ \dot{X}_6 = k_0.X_1.X_3 - k_1.X_1.X_6 - k_2.X_6.X_3 - k_8.X_6.X_4 \\ \quad - k_9.X_5.X_6 - 2k_{10}.X_6.X_6 - k_{At}.X_7.X_6 \\ \quad - k_{Pi}.X_8.X_6 \\ \dot{X}_7 = -k_{At}.X_7.X_6 + \frac{X_7^0.Q_{4in}}{V_R} - \frac{X_7.Q_{out}}{V_R} \\ \dot{X}_8 = k_{At}.X_7.X_6 - k_{Pi}.X_8.X_6 - \frac{X_8.Q_{out}}{V_R} \end{array} \right. \quad (3.19)$$

Les concentrations des autres formes de fer ferrique I_1 et I_2 sont calculées avec les équations (3.18). La concentration en fer ferrique $[Fe(III)]$ est obtenue par le calcul (3.14). La concentration en ions H^+ est calculée d'après l'expression $[H^+] = 10^{-pH}$.

Les termes Q_{1in} , Q_{2in} , Q_{3in} et Q_{4in} représentent respectivement le débit d'entrée en fer ferreux $Fe(II)$, en fer ferrique $Fe(III)$, en peroxyde d'hydrogène H_2O_2 et en atrazine At . Le terme Q_{out} représente le débit global de sortie du réacteur avec la relation de conservation de débit :

$$Q_{out} = Q_{1in} + Q_{2in} + Q_{3in} + Q_{4in} \quad (3.20)$$

Les termes X_1^0 , X_2^0 , X_3^0 et X_7^0 représentent respectivement les concentrations en entrée en fer ferreux, en fer ferrique, en peroxyde d'hydrogène et en atrazine.

Le terme V_R représente le volume du réacteur.

Les non-linéarités du procédé d'oxydation de l'atrazine apparaissent de manière évidente sur le modèle proposé : elles sont matérialisées par des produits de variables d'état $X_i.X_j$. En outre, il faut tenir compte d'autres non-linéarités introduites par le calcul analytique des différentes formes de fer ferrique (3.14).

Ces constatations justifient l'utilisation d'un modèle de Volterra pour représenter les liens existant entre les différentes grandeurs d'entrée possibles (débit d'entrée en fer ferreux, en fer ferrique, en peroxyde d'hydrogène et en atrazine) et la grandeur de sortie qui nous intéresse : la quantité d'atrazine présente dans les effluents en sortie de procédé [Bibes *et al.*, 2003a].

La mesure de la quantité d'atrazine étant par ailleurs très coûteuse et très difficile, voire impossible, à mettre en oeuvre, l'inversion du modèle de Volterra reliant la quantité d'atrazine et une autre grandeur, dont un protocole de mesure existe (par exemple la concentration en peroxyde d'hydrogène), permettra éventuellement la reconstruction de la quantité d'atrazine. Cette solution pourra permettre d'accéder à une grandeur chimique à moindre coût, le procédé d'inversion de modèle et de reconstruction de la grandeur pouvant être entièrement géré informatiquement.

3.3.3.2 Utilisation du modèle cinétique du réacteur chimique

Afin de comparer dans un premier temps les deux procédés d'oxydation $H_2O_2/Fe(II)$ et $H_2O_2/Fe(III)$, nous allons nous placer dans un cas idéal : le volume global est fixé à 1 litre et on suppose que le liquide à l'intérieur du réacteur est parfaitement homogène, les réactifs et l'atrazine étant mélangés instantanément.

Considérons le procédé d'oxydation $H_2O_2/Fe(II)$. On commence par appliquer le modèle cinétique (3.19) dans les conditions suivantes : on suppose d'abord constante la concentration en atrazine dans l'effluent à traiter ($[At] = 8.10^{-7} mol/l$) ainsi que son débit ($Q_{4in} = 8.10^{-3} l/s$). Cette valeur de débit assure un temps de séjour de l'atrazine dans le réacteur à peu près égal à 20 minutes, ce qui est très suffisant pour l'ensemble des réactions chimiques.

La période d'échantillonnage Te est fixée à 10s et ce pour toute la suite de cette étude.

On fait varier le débit de peroxyde d'hydrogène (figure 3.1(a)) en maintenant constante sa concentration ($[H_2O_2] = 1.10^{-2} mol/l$) . La concentration en fer ferreux reste égale à $[Fe(II)] = 2.10^{-4} mol/l$ et son débit d'entrée est fixé à $Q_{1in} = 2.10^{-4} l/s$. La concentration en atrazine en sortie du procédé est présentée sur la figure 3.1(b).

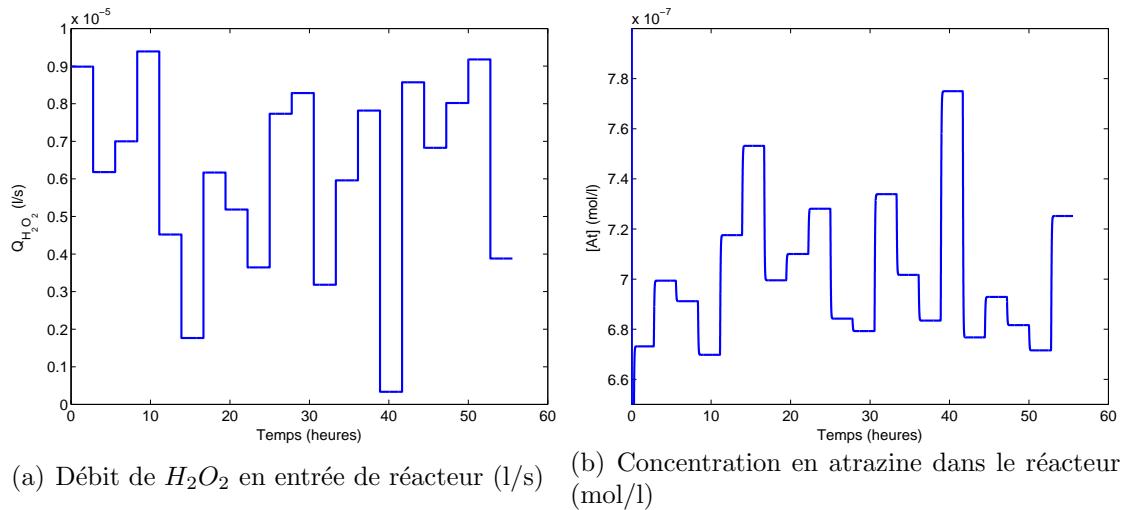


Fig. 3.1 - Procédé d'oxydation $H_2O_2/Fe(II)$ pour $[At]_{in} = 8.10^{-7} mol/l$ et $[Fe(II)] = 2.10^{-4} mol/l$

La variation de concentration en atrazine est l'inverse de celle en peroxyde d'hydrogène. Comme on l'a déjà vu, le procédé Fenton consiste ici en la création de radicaux OH° à partir du fer ferreux et du peroxyde d'hydrogène. La concentration en fer ferreux étant constante, chaque augmentation de la concentration en peroxyde d'hydrogène occasionne la création de davantage de radicaux OH° qui vont oxyder l'atrazine, faisant ainsi diminuer sa concentration. Le modèle cinétique du réacteur représente donc bien le comportement du procédé réel.

Comparons maintenant ces résultats à ceux du procédé d'oxydation $H_2O_2/Fe(III)$. Le débit d'atrazine reste constant en entrée, on conserve la même variation de débit de peroxyde d'hydrogène (figure 3.2(a)) et on introduit à la place du fer ferreux une certaine quantité de fer ferrique (même concentration $[Fe(III)] = 2.10^{-4} mol/l$). La concentration en atrazine en sortie du procédé est présentée sur la figure (3.2(b)).

3.3. Etude en simulation : procédé d'oxydation de polluant

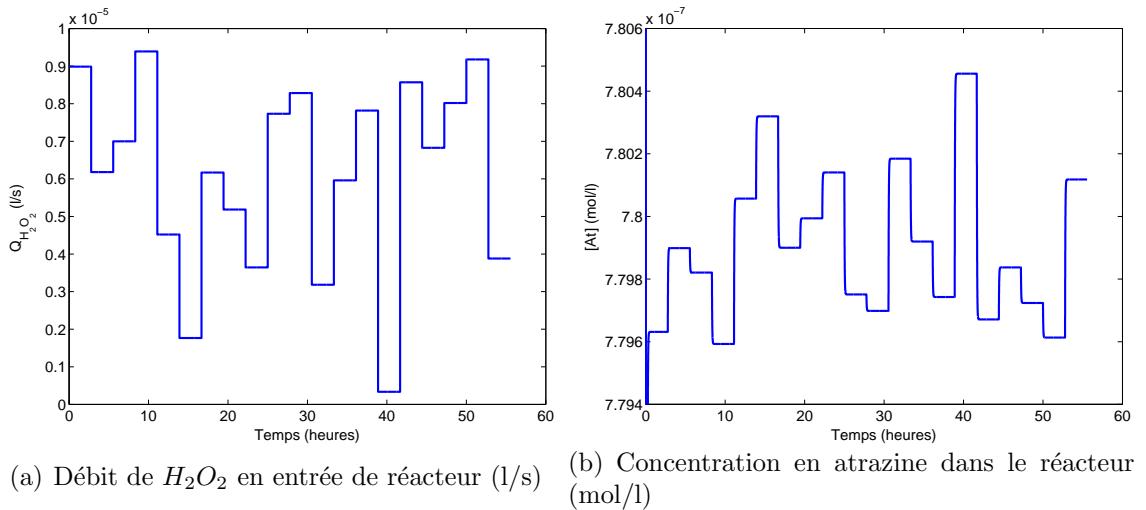


Fig. 3.2 - Procédé d'oxydation $H_2O_2/Fe(III)$ pour $[At]_{in} = 8.10^{-7} \text{ mol/l}$ et $[Fe(III)] = 2.10^{-4} \text{ mol/l}$

Les résultats obtenus semblent montrer qu'à concentration égale, l'action du fer ferreux est plus significative que celle du fer ferrique sur la production de radicaux OH° et l'oxydation de l'atrazine qui en résulte. Différentes mesures expérimentales confirment que la cinétique de décomposition du peroxyde d'hydrogène par le fer ferreux est beaucoup plus rapide qu'avec le fer ferrique [Gallard, 1998].

C'est pourquoi nous nous limiterons dans la suite de cette étude aux résultats d'oxydation de l'atrazine obtenus en utilisant le procédé Fenton $H_2O_2/Fe(II)$.

Afin d'étudier le procédé d'oxydation $H_2O_2/Fe(II)$ de l'atrazine dans des conditions relativement proches de celles qui seront utilisées sur le futur pilote expérimental (en cours de réalisation sur la plate-forme Eaux de l'Université de Poitiers), nous allons modifier les conditions de simulation. Le volume global du réacteur chimique est désormais fixé à 10 litres. On conserve également l'hypothèse de départ d'un réacteur parfaitement agité et d'un mélange homogène instantané des différentes espèces chimiques introduites.

On fait maintenant varier le débit d'atrazine à l'entrée du réacteur chimique (figure 3.3(a)) afin d'assurer un temps de séjour compris entre 5 et 10 minutes. Le débit et la concentration en peroxyde d'hydrogène sont maintenus constants ($Q_{3in} = 2.10^{-2} \text{ l/s}$ et $[H_2O_2] = 1.10^{-2} \text{ mol/l}$) ainsi que le débit et la concentration en fer ferreux ($Q_{1in} = 4.10^{-4} \text{ l/s}$ et $[Fe(II)] = 2.10^{-4} \text{ mol/l}$). La concentration en atrazine en sortie du procédé est présentée sur la figure 3.3(b).

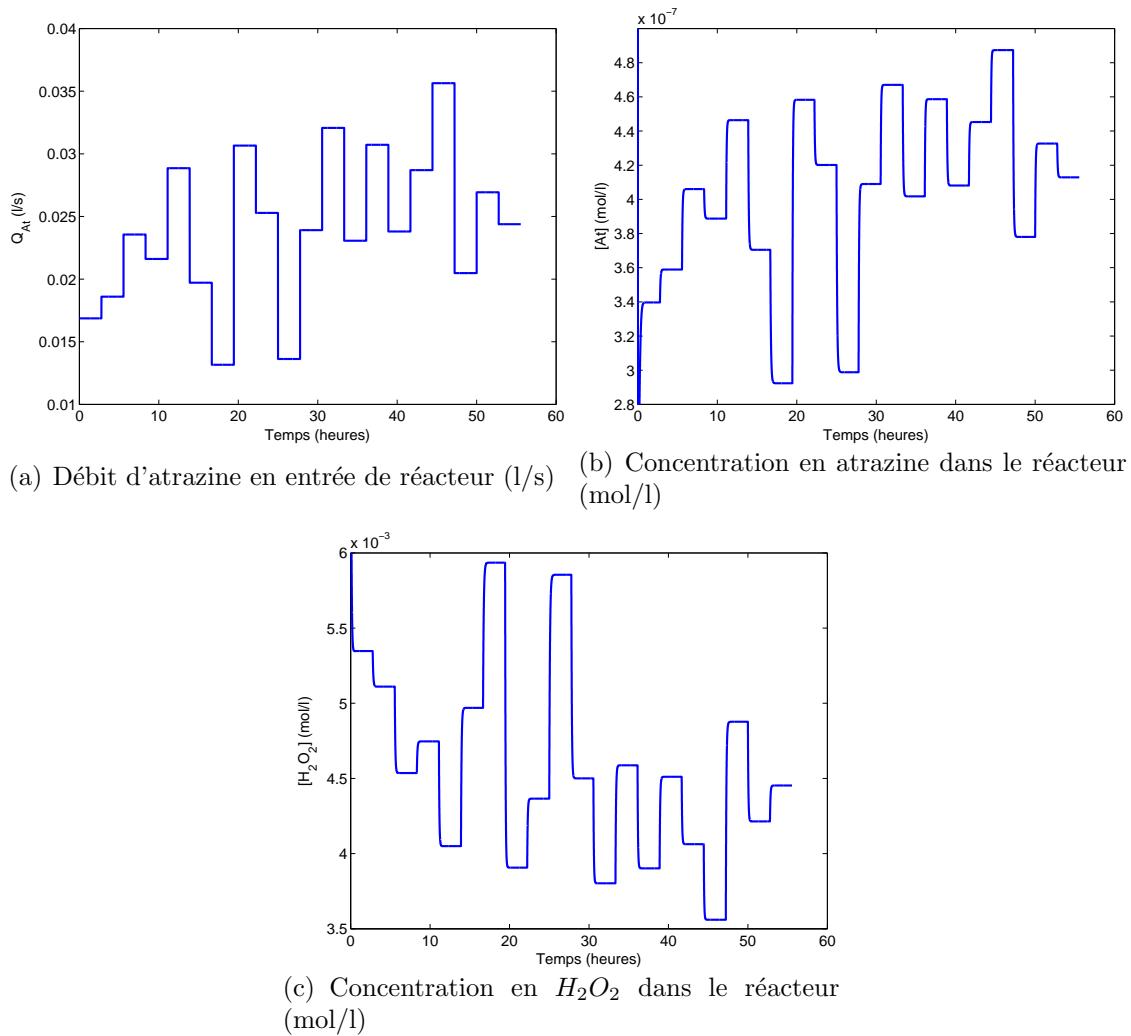


Fig. 3.3 - Procédé d'oxydation $H_2O_2/Fe(II)$ pour $[H_2O_2]_{in} = 1.10^{-2} \text{ mol/l}$ et $[Fe(II)]_{in} = 2.10^{-4} \text{ mol/l}$

On observe également la concentration en peroxyde d'hydrogène H_2O_2 dans le réacteur durant le procédé d'oxydation (figure 3.3(c)). Cela permet de constater que seule une partie du peroxyde d'hydrogène (entre 40% et 60%) a été utilisée afin de produire des radicaux OH° .

La concentration en atrazine après traitement chimique est donc divisée par un facteur moyen de 2 par rapport à la quantité d'atrazine présente dans l'effluent à traiter. Le débit des réactifs de Fenton restant constant, toute augmentation du débit d'atrazine à l'entrée du réacteur a une répercussion sur la quantité d'atrazine après traitement à partir du moment où une partie de la quantité de réactif a été utilisée lors du procédé.

Nous nous sommes bornés jusqu'ici à ne faire varier qu'une seule concentration à l'entrée du réacteur (ou bien le débit de peroxyde d'hydrogène, ou bien celui d'atrazine). La quantité considérée à chaque fois dans le réacteur est soit la concentration en atrazine

après traitement par le procédé d'oxydation $H_2O_2/Fe(II)$, soit la concentration restante en H_2O_2 . Dans ces conditions idéales, on peut donc considérer le réacteur chimique continu comme un système mono-entrée mono-sortie et ainsi justifier la représentation du lien entre la grandeur d'entrée et la grandeur de sortie par un modèle de Volterra.

Remarquons que le cas idéal du système mono-entrée mono-sortie dans lequel on se place ici limite les possibilités de commande du procédé car on ne pourra agir que sur le dosage d'un seul des deux réactifs, soit celui de peroxyde d'hydrogène, soit celui de fer. La commande du dosage optimal des deux réactifs nécessite un modèle à deux entrées et une sortie, ou plus généralement un modèle multi-entrées mono-sortie, que l'on pourrait envisager dans la suite directe de ce travail de recherche.

De plus, on a vu qu'il est impossible dans notre cas de mesurer directement la quantité d'atrazine présente dans l'effluent à traiter. La connaissance d'un modèle de Volterra reliant le débit d'atrazine (grandeur d'entrée) à la concentration en peroxyde d'hydrogène permet d'envisager l'inversion de ce modèle pour estimer par reconstruction le débit d'atrazine à l'entrée du réacteur. Une méthode satisfaisante d'analyse en continu de l'évolution de la concentration en H_2O_2 existe [Audebrand *et al.*, 2002]. Le temps nécessaire à cette analyse a été optimisé et réduit à quelques minutes, ce qui laisse envisager une utilisation en temps réel de ce type de méthode de mesure.

3.3.4 Application du modèle de Volterra

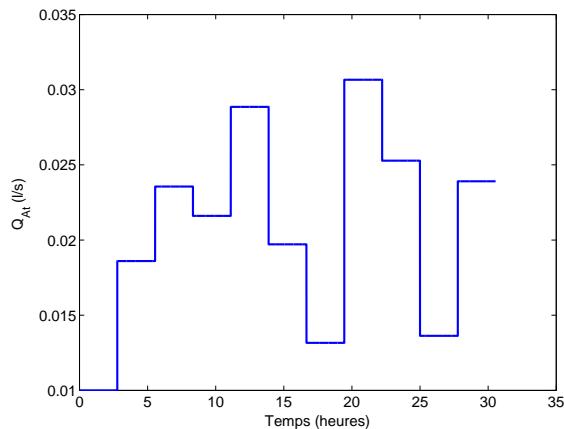


Fig. 3.4 - Débit d'atrazine en entrée du réacteur (l/s)

Considérons le cas du procédé d'oxydation $H_2O_2/Fe(II)$ pour une variation de débit d'atrazine en entrée représentée sur la figure 3.4.

Les variations de concentration en atrazine et en peroxyde d'hydrogène H_2O_2 données par le modèle cinétique du procédé sont représentées par les courbes continues de la figure

3.5.

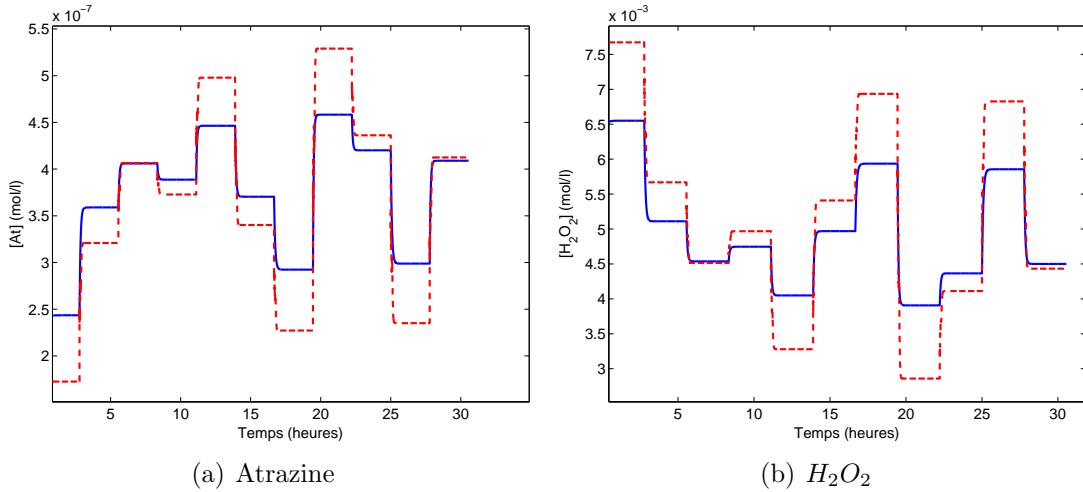


Fig. 3.5 - Variations de concentrations (mol/l) donnée par le modèle cinétique du réacteur (courbe continue) et le modèle linéaire (courbe pointillée)

On montre d'abord qu'un modèle linéaire ne suffit pas à représenter correctement le lien entre la variation d'atrazine en entrée et la variation d'atrazine et de peroxyde d'hydrogène dans le réacteur. A cet effet, le modèle de Volterra est tronqué à son premier terme et le noyau de Volterra (qui correspond dans ce cas à la réponse impulsionale du modèle linéaire) est développé sur 10 fonctions généralisées dont les valeurs des pôles sont fixées de -0.001 à -0.01 . Le modèle linéaire comporte donc 10 paramètres et les courbes de la figure 3.5 prouvent l'insuffisance d'un modèle linéaire. On note que l'ajout de fonctions dans chacun des développements n'apporte aucune amélioration au modèle linéaire.

On peut alors envisager la représentation par un modèle de Volterra du lien entre la variation d'atrazine en entrée et la variation d'atrazine dans le réacteur d'une part, puis la variation de peroxyde d'hydrogène dans le réacteur d'autre part. Dans chacun des deux cas, on utilise maintenant une série de Volterra tronquée à ses trois premiers termes. On utilise dans un premier temps la modélisation des noyaux de Volterra par développement sur une base de fonctions généralisée. Le premier noyau de Volterra est développé sur une fonction de la base. Le deuxième et le troisième noyau sont développés sur deux fonctions de la base. Les valeurs des pôles sont fixées à -0.005 et -0.01 . Dans chacun des deux cas, le modèle de Volterra global comporte 8 paramètres.

3.3. Etude en simulation : procédé d'oxydation de polluant

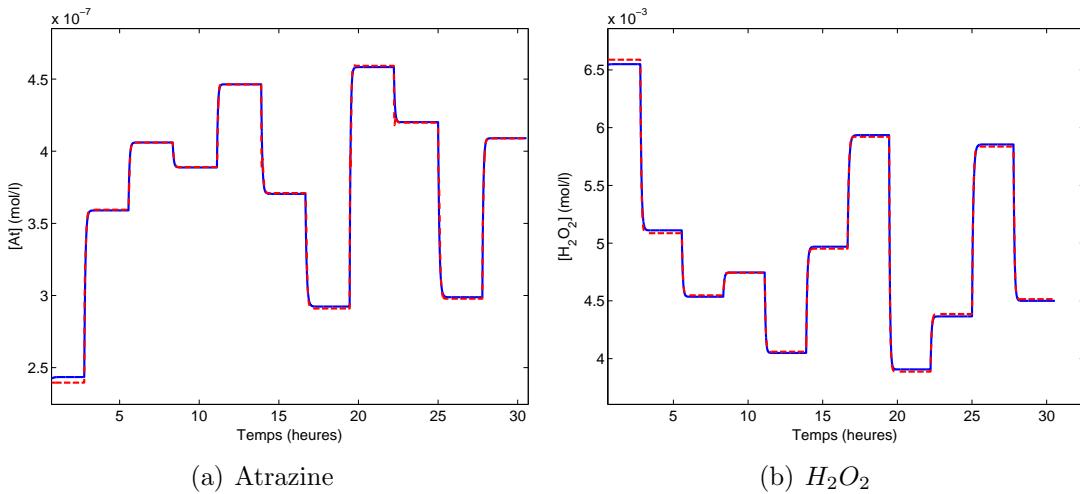


Fig. 3.6 - Variations de concentrations (mol/l) donnée par le modèle cinétique du réacteur (courbe continue) et par le modèle de Volterra (courbe pointillée)

Les sorties de chacun des deux modèles de Volterra sont représentées sur la figure 3.6 et comparées aux résultats correspondants donnés par le modèle cinétique (3.19) du réacteur chimique.

Considérons maintenant le cas d'une variation de débit de peroxyde d'hydrogène H_2O_2 en entrée du modèle cinétique représentée sur la figure 3.7(a).

La variation de concentration en atrazine donnée par le modèle cinétique du procédé et en sortie du modèle de Volterra est représentée sur la figure 3.7(b).

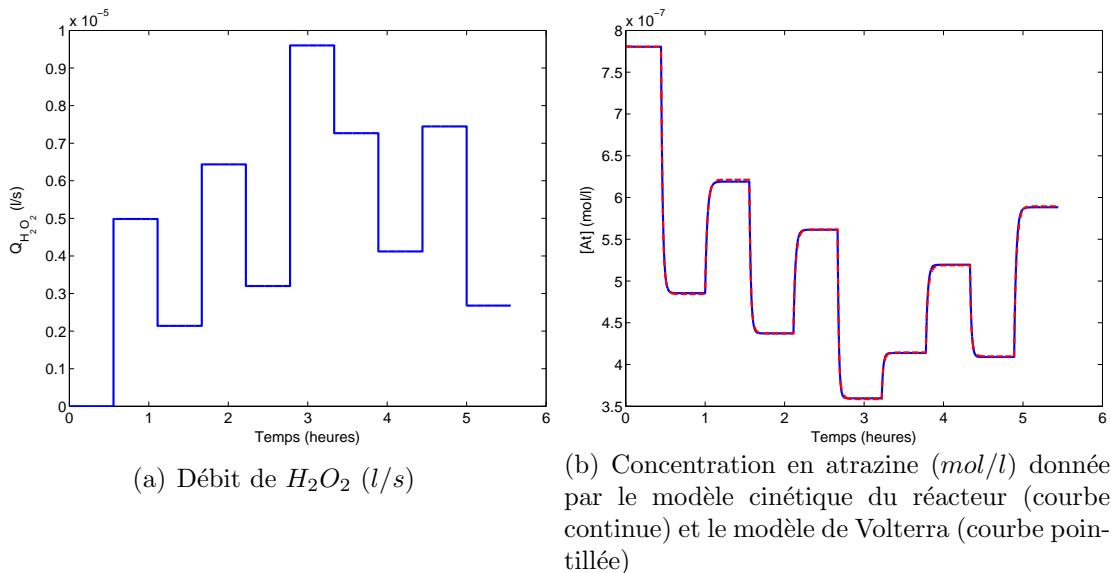


Fig. 3.7 - Variations données par le modèle cinétique du réacteur (courbe continue) et par le modèle de Volterra (courbe pointillée)

Le modèle de Volterra utilisé est tronqué à ses 3 premiers noyaux, développés respectivement sur une, deux et deux fonctions orthonormées généralisées. Les valeurs des pôles des fonctions sont fixées à -0.005 et -0.01 . Le nombre total de paramètres du modèle est égal à 8.

Divers modèles de Volterra peuvent donc être établis faisant le lien entre les variations de différentes espèces chimiques mises en jeu lors du procédé d'oxydation. Ces modèles de Volterra sont suffisamment précis et parcimonieux pour que l'on puisse envisager leur inversion par les deux méthodes présentées et étudiées dans le deuxième chapitre de ce mémoire : inversion du modèle de Volterra par régularisation de Tikhonov et par retour d'entrée reconstruite.

L'inversion du modèle de Volterra permet, à partir de la grandeur de sortie, de reconstruire la grandeur d'entrée. Dans le cadre du procédé d'oxydation $H_2O_2/Fe(II)$, la reconstruction d'une grandeur possède un double intérêt :

- la reconstruction de la quantité d'atrazine par l'inversion du modèle de Volterra qui la relie à la concentration en peroxyde d'hydrogène H_2O_2 : l'intérêt de cette reconstruction réside dans le fait que la quantité d'atrazine (ou, de manière générale, en composant organique polluant) est difficilement mesurable durant le procédé. Le procédé de mesure de la concentration en H_2O_2 est lui relativement simple en mettre en oeuvre. La connaissance du lien qui unit ces deux grandeurs permet donc, par l'inversion de ce lien, de s'affranchir du problème de mesure de la quantité d'atrazine.
- la commande du dosage des réactifs (peroxyde d'hydrogène ou fer ferreux) à l'entrée du réacteur : le but du procédé d'oxydation $H_2O_2/Fe(II)$ est la neutralisation de l'atrazine présente dans l'effluent à traiter. L'objectif final est l'optimisation de la quantité de réactif à ajouter en entrée de réacteur à partir de la connaissance de la quantité maximale d'atrazine que l'on désire en sortie du réacteur. L'intérêt est ici purement économique.

3.3.5 Inversion de modèle de Volterra par régularisation de Tikhonov

Afin de valider la méthode de reconstruction de la quantité d'atrazine par régularisation de Tikhonov, nous allons envisager l'inversion du modèle de Volterra reliant le débit d'atrazine à l'entrée (figure 3.4) et la concentration en H_2O_2 dans le réacteur (figure 3.6(b)). Le modèle de Volterra est tronqué à ses 3 premiers noyaux développés respectivement sur une, deux et deux fonctions orthonormées. Les pôles des fonctions sont fixés aux valeurs -0.01 et -0.05 . Le modèle possède au total 8 paramètres. Dans le cas où la mesure de la concentration en H_2O_2 n'est pas bruitée, le modèle de Volterra utilisé est suffisamment précis pour que l'on puisse envisager son inversion sans avoir à régulariser la quantité reconstruite ($\lambda = 0$). Le résultat de reconstruction est représenté sur la figure 3.8.

3.3. Etude en simulation : procédé d'oxydation de polluant

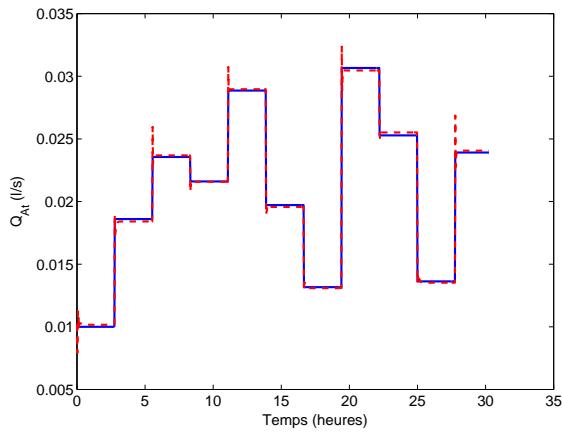


Fig. 3.8 - Débit d'atrazine (l/s) (courbe continue) et sa reconstruction (ligne pointillée)

Malgré l'absence de bruit de mesure, on constate l'apparition de pics de reconstruction d'amplitude non négligeable. La qualité de reconstruction est cependant correcte. L'augmentation du nombre de noyaux ou de fonctions modélisant ces noyaux n'apporte aucune amélioration ni à la précision du modèle ni à la qualité de la reconstruction du débit d'atrazine.

Considérons maintenant le cas où la mesure de la concentration en H_2O_2 est bruitée (rapport signal sur bruit égal à 100). La figure 3.9 montre la sortie du modèle de Volterra, dont les paramètres ont été réidentifiés pour représenter la concentration en peroxyde d'hydrogène dans le réacteur.

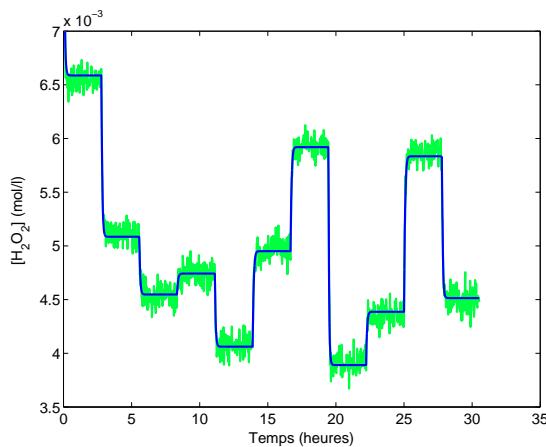


Fig. 3.9 - Concentration en H_2O_2 (mol/l) dans le réacteur (courbe claire) et en sortie du modèle de Volterra (courbe foncée)

Afin de reconstruire correctement la quantité d'atrazine à l'entrée du réacteur, il est maintenant nécessaire de régulariser la solution du problème inverse. La figure 3.10 re-

présente la reconstruction pour deux valeurs du paramètre de régulation ($\lambda = 10000$ et $\lambda = 100000$). Dans chacun des cas, la matrice de régulation utilisée est la matrice des dérivées secondes, qui offre les résultats de reconstruction les plus probants (cf le deuxième chapitre du mémoire).

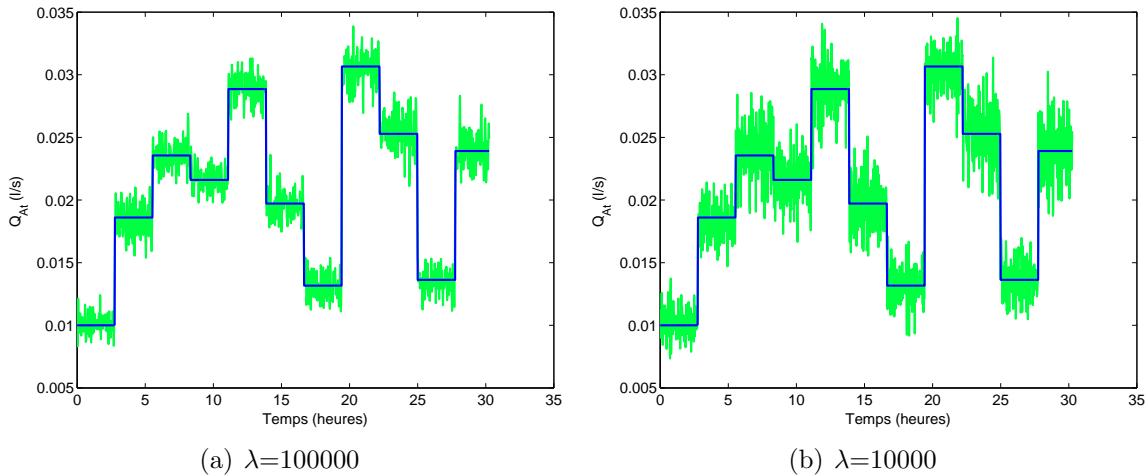


Fig. 3.10 - Comparaison entre le débit d'atrazine (l/s) (courbe foncée) et sa reconstruction (courbe claire) pour différentes valeurs de λ avec utilisation de la matrice des dérivées secondes

Le terme λ prend des valeurs très importantes, étant donné qu'il est proportionnel à l'inverse de la période d'échantillonnage. On constate un bon comportement par rapport à l'amplitude du bruit sans qu'il y ait pour autant une perte d'information sur le signal reconstruit.

Ces différents essais en présence de bruit de mesure permettent de valider la méthode d'inversion par régularisation dans le cas du modèle cinétique du procédé d'oxydation $H_2O_2/Fe(II)$.

Cet exemple de reconstruction de la quantité d'atrazine valide la méthode de reconstruction par régularisation de Tikhonov et confirme donc l'intérêt de cette méthode pour la reconstruction de grandeurs non directement mesurables à partir de la connaissance de grandeurs mesurables et d'un modèle qui les relie. On peut également constater l'influence de l'amplitude du bruit de mesure et de la précision du modèle de Volterra sur la qualité de reconstruction de la quantité d'atrazine.

3.3.6 Inversion de modèle de Volterra par retour d'entrée reconstruite

Les résultats de reconstruction par régularisation de Tikhonov montrent que la méthode employée est satisfaisante pour inverser un modèle de Volterra. Nous allons à présent

3.3. Etude en simulation : procédé d'oxydation de polluant

les comparer aux résultats obtenus par les deux méthodes d'inversion par retour d'entrée reconstruite décrites dans le deuxième chapitre du mémoire.

On utilise cette fois la modélisation des noyaux de Volterra par développement sur des fonctions de transfert afin de montrer que les méthodes d'inversion proposées fonctionnent quel que soit le choix de modélisation des noyaux de Volterra.

On considère toujours le cas du procédé d'oxydation $H_2O_2/Fe(II)$ pour une variation du débit d'atrazine en entrée représenté par la figure 3.4.

Dans tout ce qui suit, le modèle de Volterra est tronqué à ses 2 premiers noyaux, développés respectivement sur une et deux fonctions de transfert. Le modèle de Volterra global comporte 8 paramètres.

Dans le cas où les mesures de concentration ne sont pas bruitées, les variations de concentration en atrazine et en peroxyde d'hydrogène H_2O_2 données par le modèle cinétique du réacteur et le modèle de Volterra sont représentées sur la figure 3.11.

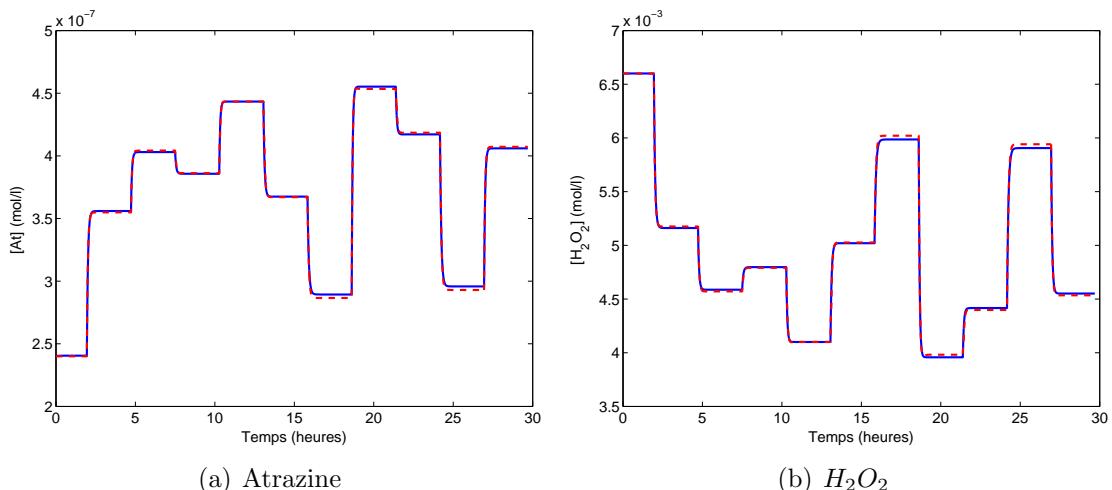


Fig. 3.11 - Variations de concentrations (mol/l) donnée par le modèle cinétique du réacteur (courbe continue) et le modèle de Volterra (courbe pointillée)

Si les mesures de concentration sont bruitées (rapport signal/bruit égal à 100), la figure 3.12 montre qu'après une nouvelle identification, le modèle de Volterra proposé reste valable pour représenter les variations de concentration en atrazine et en peroxyde d'hydrogène H_2O_2 données par le modèle cinétique du réacteur.

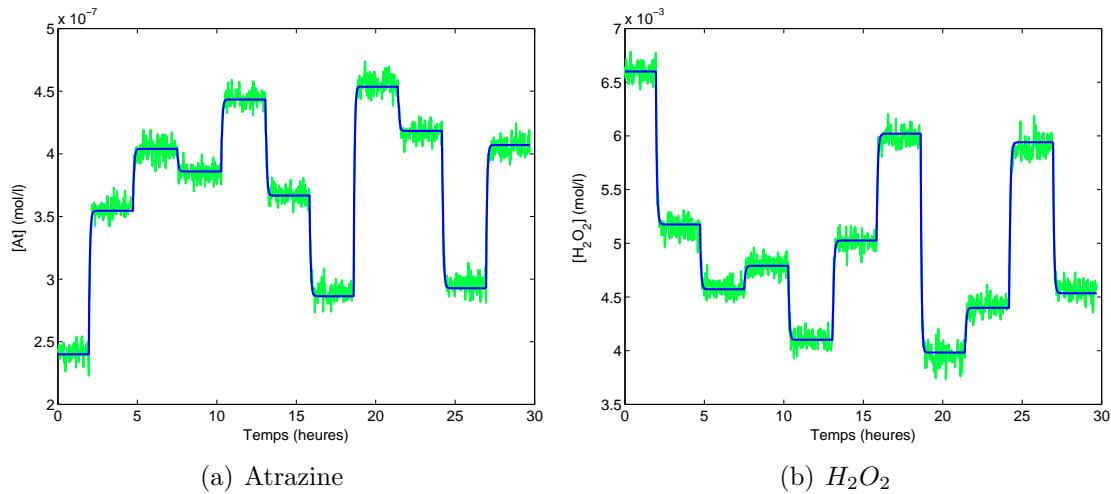


Fig. 3.12 - Variations de concentrations (mol/l) données par le modèle cinétique du réacteur (courbe claire) et le modèle de Volterra (courbe foncée)

Le modèle de Volterra développé sur des fonctions de transfert comporte autant de paramètres que celui développé sur base de fonctions généralisée mais sa complexité est moindre car il ne comporte que deux noyaux. Les résultats obtenus dans les deux cas sont pourtant tout à fait comparables.

Remarquons que la parcimonie du modèle de Volterra développé sur une base de fonctions généralisée n'est garantie que dans le cas d'un choix convenable des pôles des fonctions, ce qui était le cas dans l'étude précédente : les valeurs ont été fixées à -0.005 et -0.01 après divers essais.

Quoi qu'il en soit, le modèle développé sur fonctions de transfert est ici aussi suffisamment précis et parcimonieux pour que l'on puisse envisager son inversion.

On envisage directement la reconstruction du débit d'atrazine en entrée de réacteur (figure 3.4) par les deux méthodes d'inversion par retour d'entrée reconstruite dans le cas de mesures bruitées de la concentration en peroxyde d'hydrogène H_2O_2 dans le réacteur (rapport signal/bruit=100).

Les paramètres du modèle de Volterra sont à nouveau identifiés dans le cas où la mesure de la concentration en peroxyde d'hydrogène est bruitée (figure 3.12(b)). Lors de l'utilisation de la deuxième méthode, les valeurs des paramètres K_1 et K_2 sont respectivement fixées à 0.05 et 20 pour une reconstruction optimale. Les résultats de reconstruction sont représentés sur la figure 3.13.

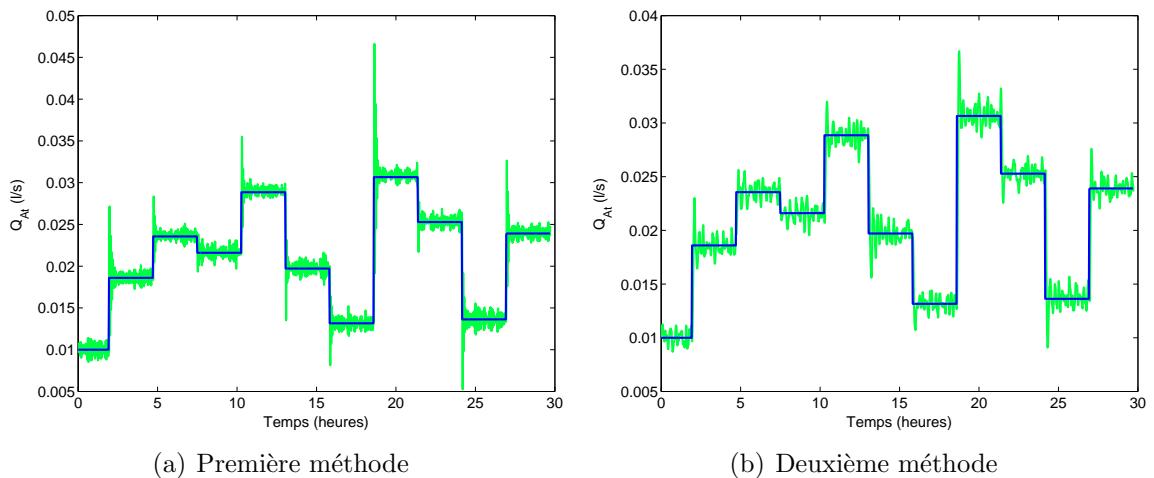


Fig. 3.13 - Débit d'atrazine (l/s) en entrée du réacteur (courbe foncée) et sa reconstruction (courbe claire)

Malgré la présence de pics de reconstruction, la qualité de reconstruction semble meilleure que celle obtenue dans le meilleur des cas par régularisation de Tikhonov (figure 3.10), ceci à cause de la faible amplitude de l'erreur de reconstruction. Ces résultats ont été obtenus sans avoir eu à tenir compte d'une éventuelle amplification du bruit de mesure ajouté à la grandeur de sortie. L'amplitude du bruit (ou erreur de reconstruction) que l'on retrouve sur le signal reconstruit est ici plus faible que l'amplitude du bruit ajouté à la mesure de la concentration en H_2O_2 utilisée. Chacune des deux méthodes d'inversion par retour d'entrée reconstruite joue implicitement le rôle d'un filtre passe-bas, atténuant l'amplitude du bruit.

Un autre avantage de ces deux méthodes d'inversion par rapport à la méthode par régularisation de Tikhonov est une diminution importante du temps de calcul nécessaire, qui est réduit à quelques dizaines de secondes au plus, suivant la taille du signal de sortie mesuré et la complexité du modèle de Volterra.

3.3.7 Conclusion

Cette étude d'un procédé chimique de traitement des eaux permet de valider dans le cadre de la simulation les modèles de Volterra et les différentes techniques d'inversion non linéaire proposés dans les deux premiers chapitres du mémoire.

La nature des non-linéarités inhérentes au procédé Fenton d'oxydation de composés organiques permet l'utilisation d'un modèle de Volterra. L'étape d'inversion du modèle de Volterra est d'autant plus rapide et efficace que le modèle possède une structure simple (i.e. un faible nombre de noyaux) et une bonne parcimonie paramétrique. L'objectif de reconstruction d'une concentration non mesurable à partir de la mesure de concentration d'une autre espèce chimique est donc atteint dans le cadre de la simulation. Partant des

constatations effectuées en simulation, une implémentation en temps réel du procédé de reconstruction de la concentration en composé organique polluant est envisageable.

L'étape suivante de cette étude consistera en l'application de ces méthodes de modélisation et d'inversion sur des données réelles.

3.4 Etude expérimentale : procédé de digestion anaérobie

Les traitements biologiques ou biochimiques constituent le mode classique d'épuration de la pollution organique des eaux résiduaires industrielles. Ces techniques d'épuration reposent sur les conditions qui permettent aux flores bactériennes de se développer et d'assurer la dégradation des matières organiques polluantes, éliminées dans la mesure où elles servent d'aliments aux bactéries. Le recours à l'épuration biologique ou biochimique dépend étroitement de la biodégradabilité des rejets industriels [Boeglin, 1997].

Les procédés d'épuration biologiques ou biochimiques les plus utilisés sont de type aérobies (présence d'air ou d'oxygène) car la cinétique du processus s'avère beaucoup plus rapide et les rendements d'épuration plus élevés qu'avec les traitements biologiques anaérobies.

Cependant, les procédés anaérobies, tout en assurant une élimination de la pollution organique carbonée, présentent quand même un certain nombre d'avantages par rapport aux traitements biologiques aérobies : une faible consommation d'énergie pour les besoins du procédé, une faible production de boues biologiques en excès et la récupération possible d'un biogaz (constitué en grande partie de méthane) pouvant être utilisé comme source d'énergie. Ce type de traitement, communément appelé "digestion anaérobie" est utilisé surtout pour des effluents très concentrés en pollution carbonée, de type industriel (brasserie, sucrerie, conserverie, distillerie,...).

3.4.1 Description du procédé étudié

La digestion anaérobie assure la conversion de la majeure partie du carbone organique contenu dans les effluents en un biogaz combustible composé en grande partie de méthane CH_4 . Vu le coût actuel de l'énergie, la digestion anaérobie est un procédé bien adapté au traitement d'effluents industriels concentrés en pollution organique et produits à des débits moyens. En contrepartie de certains avantages par rapport aux traitements aérobies classiques, la cinétique des réactions mises en jeu est lente (en particulier lors de l'étape de méthanisation), ce qui se traduit par un investissement de départ élevé compte tenu du volume important nécessaire des réacteurs biologiques anaérobies.

3.4. Etude expérimentale : procédé de digestion anaérobie

La digestion anaérobie s'effectue spontanément lorsqu'une pollution organique concentrée est maintenue en l'absence d'oxygène. Le processus de dégradation des composés organiques s'opère en deux étapes essentielles :

- une phase acide de liquéfaction des composés organiques aboutissant à la formation d'acides gras volatils (AGV). Cette phase fait intervenir des bactéries anaérobies qui sécrètent des enzymes extracellulaires capables d'hydrolyser les macromolécules organiques complexes (protéines, lipides, hydrates de carbone) de l'eau résiduaire en molécules plus petites, appelées acides volatils (acide acétique, propionique, butyrique,...) qui sont essentiellement gras et qui serviront de substrat aux bactéries méthanolégènes.
- une phase de gazéification ou méthanolégèse dont les produits finaux sont le méthane (CH_4) et le dioxyde de carbone (CO_2). L'acétogénèse permet d'abord la transformation des acides volatils en acide acétique et en hydrogène. La méthanolégèse assure ensuite par des bactéries anaérobies la transformation des acides volatils en produits gazeux finaux (méthane et dioxyde de carbone).

La digestion anaérobie est un processus lent à démarrer. La cinétique des réactions mises en jeu est principalement gouvernée par la méthanolégèse, étape relativement lente. Le fonctionnement global est fortement influencé par la sensibilité de la masse bactérienne au pH (la vitesse de production de méthane décroît rapidement en dehors d'une zone de pH allant de 6 à 8), à la température (chute de la production de méthane en dessous de $30^{\circ}C$), aux variations de pollution et à la présence en quantité trop importante de certains toxiques, comme les métaux lourds ou les sulfures. De manière générale, on vérifie la bonne marche du procédé de digestion anaérobie en mesurant la production de biogaz, elle-même fonction de la qualité du substrat entrant en fermentation lorsque toutes les conditions biologiques sont satisfaites.

On peut classer les différents digesteurs anaérobies mis en oeuvre industriellement en deux grandes catégories selon l'intervention dans la dégradation des polluants organiques de cultures bactériennes anaérobies libres ou fixées :

- les digesteurs à cultures libres dans lesquels la biomasse est en suspension. On assure un brassage puissant pour homogénéiser le milieu, le plus souvent par recyclage d'une fraction du biogaz produit. Pour optimiser le fonctionnement de la méthanisation, on peut procéder à la séparation des deux processus d'acidogénèse (réalisé dans un bassin) et de méthanisation (dans le réacteur proprement dit)
- les digesteurs à cultures fixées dans lesquels on s'efforce de fixer la culture bactérienne anaérobie sur un support solide. Le matériau est choisi en fonction de sa surface spécifique, afin de fixer le maximum de biomasse, et de son indice de vide, afin de limiter les risques de colmatage.

Le procédé étudié est un digesteur anaérobie en lit fixe à flux ascendant traitant des vinasses de distilleries de la région de Narbonne (figure 3.14) [Bernard *et al.*, 2001], [Steyer *et al.*, 2003].

Le réacteur est une colonne circulaire de 3.5m de hauteur et de 0.6m de diamètre, ce qui donne un volume total de 0.989m^3 . Le support utilisé pour fixer la biomasse est du *Cloisonyle™*, matériau dont la surface spécifique est de $180\text{m}^2/\text{m}^3$. Ce support occupant un certain volume, le volume utile du réacteur est de 0.948m^3 .

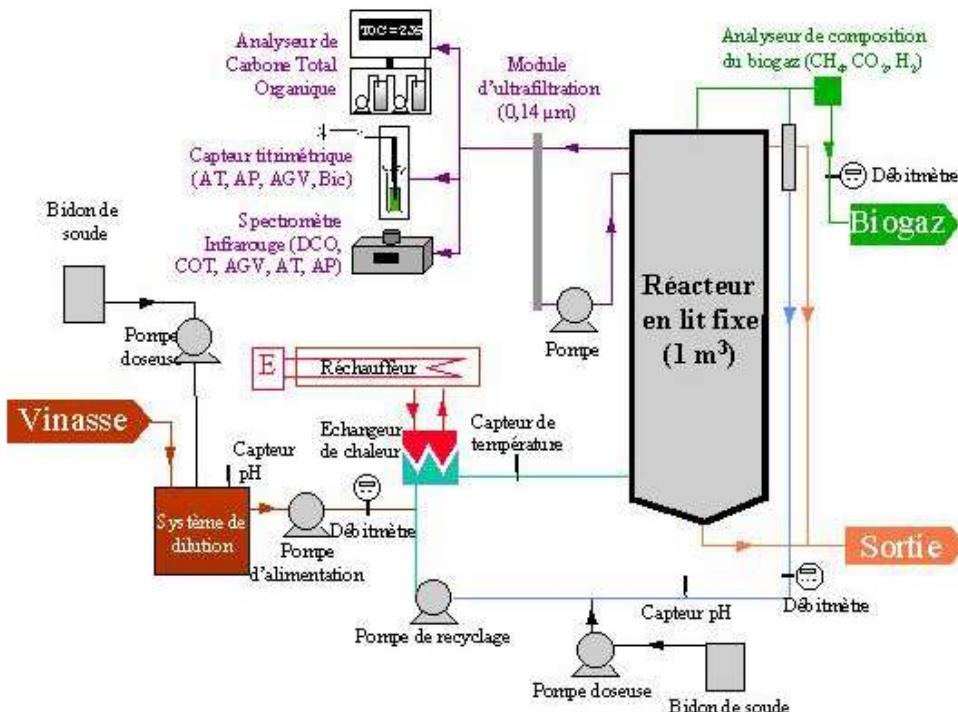


Fig. 3.14 - Digesteur anaérobio en lit fixe à flux ascendant

Le liquide à l'entrée du réacteur est obtenu par dilution de 20l de vinasses dans un réservoir de 200l. Ce réservoir d'alimentation est équipé de capteurs de niveau permettant d'obtenir une concentration donnée en matière carbonée à l'entrée du réacteur. Le pH est également mesuré (figure 3.15(a)) et régulé dans le réservoir d'alimentation par l'ajout de soude à l'aide d'une pompe doseuse. La température à l'intérieur du réacteur est maintenue à 35°C par une boucle de régulation constituée d'une résistance chauffante, d'un circuit d'eau chaude et d'une sonde de température.

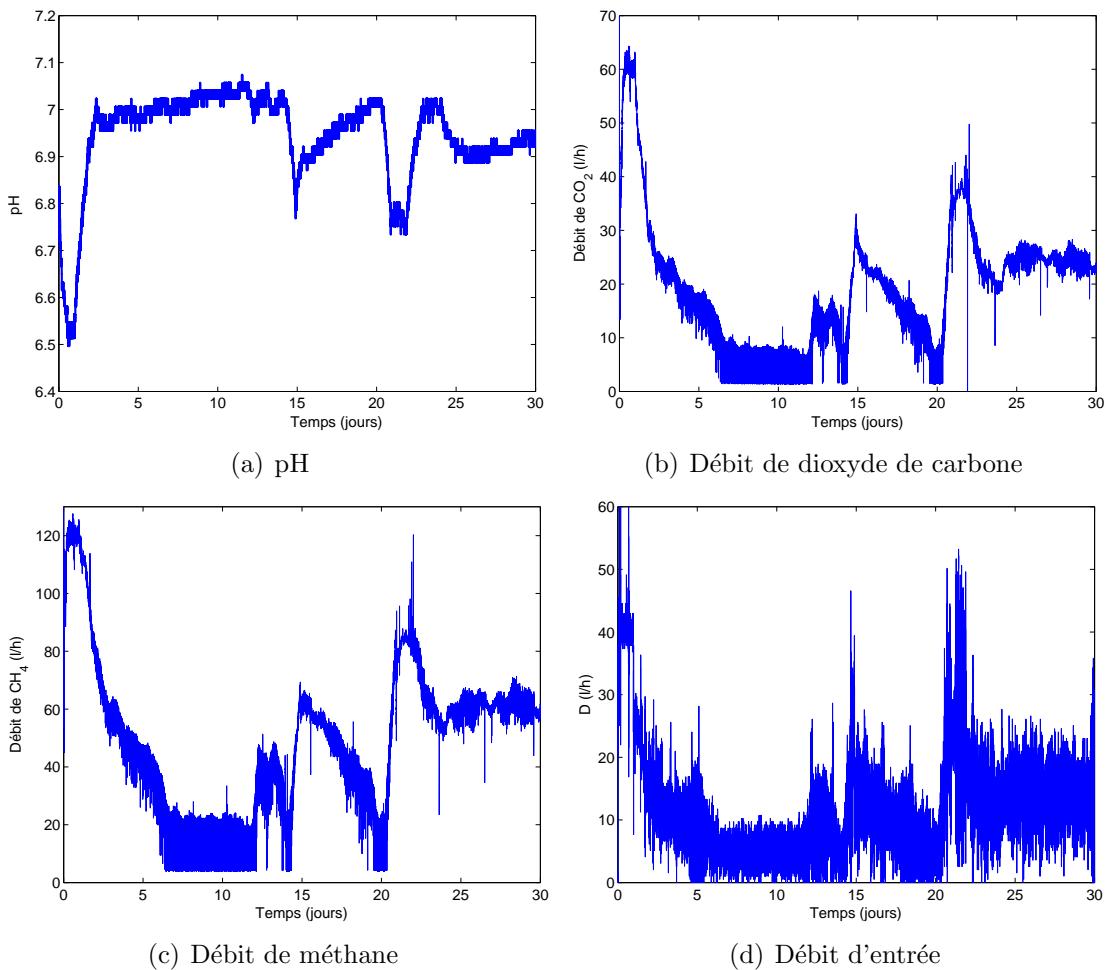


Fig. 3.15 - Mesures effectuées en ligne

Le débit de liquide à l'entrée du réacteur est mesuré à l'aide d'un capteur électromagnétique (figure 3.15(d)). Une fois dans le réacteur, le liquide monte jusqu'à la sortie en passant par le support sur lequel est fixée la biomasse. A la sortie du réacteur, un séparateur permet le dégazage du milieu liquide et la décantation des solides en suspension, qui seront ensuite recyclés. Le liquide est ensuite recyclé à l'entrée du réacteur à l'aide d'une pompe de recirculation. Un analyseur permet de mesurer le pourcentage (et donc le débit) d'hydrogène, de méthane (figure 3.15(c)) et de dioxyde de carbone (figure 3.15(b)) du gaz en sortie de réacteur.

D'autres mesures, habituellement obtenues hors-ligne et manuellement, sont effectuées en ligne dans le cadre spécifique de ce réacteur :

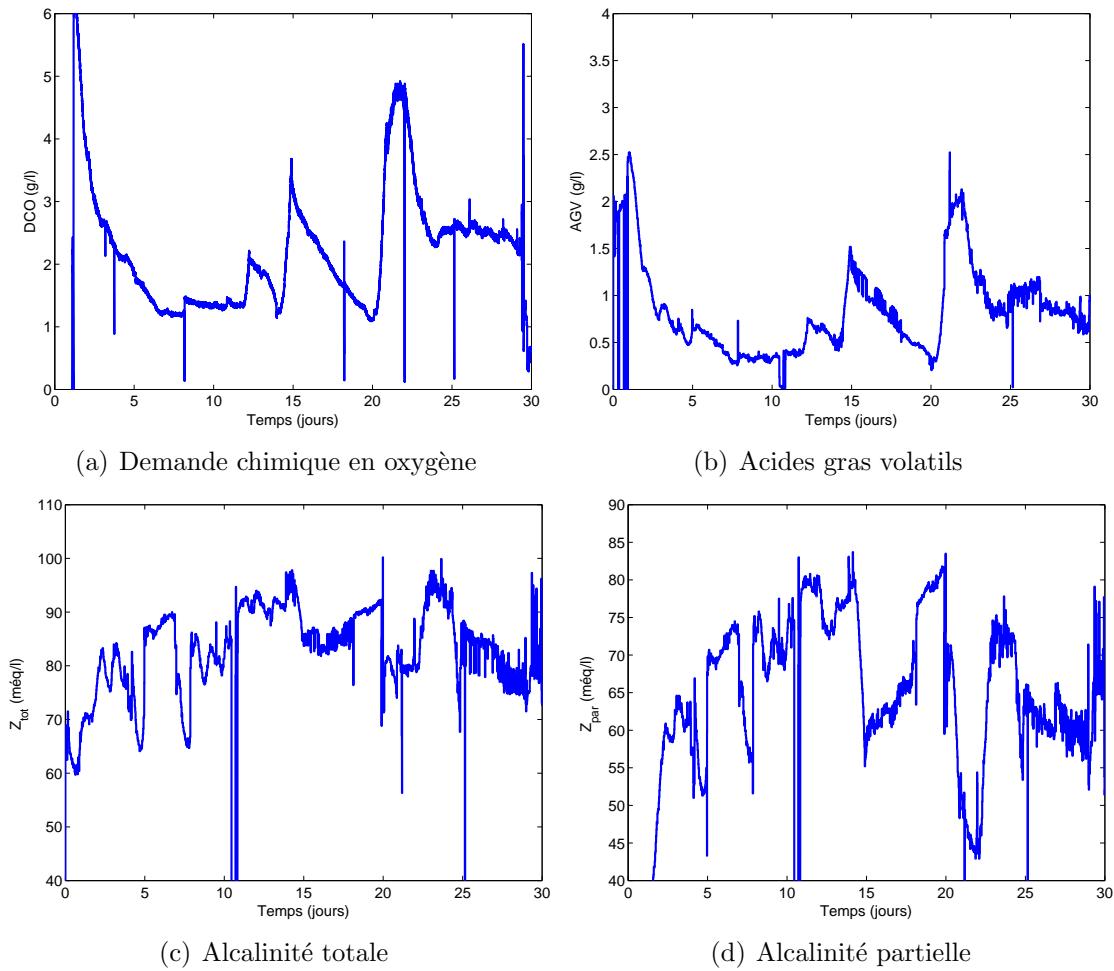


Fig. 3.16 - Autres mesures effectuées en ligne

- la concentration en acides gras volatils (AGV - figure 3.16(b)) mesurée à l'aide d'un capteur industriel ;
- la demande chimique en oxygène (DCO - figure 3.16(a)) mesurée par titrimétrie ;
- l'alcalinité totale (AT - figure 3.16(c)) et partielle (AP - figure 3.16(d)) mesurées par titrimétrie.

Une deuxième mesure de la DCO et de la concentration en AGV est effectuée par spectrométrie infrarouge. Ces mesures sont représentées sur la figure 3.17.

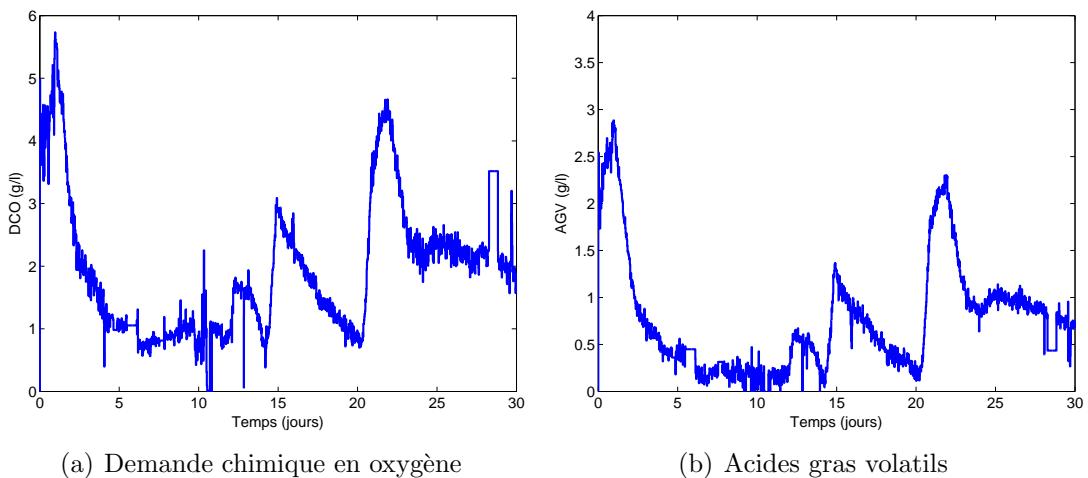


Fig. 3.17 - Mesures effectuées par spectrométrie infrarouge

3.4.2 Le modèle dynamique

La modélisation dynamique du processus de digestion anaérobie, sujet abordé depuis près de 30 ans, a d'abord été envisagée du point de vue de la précision. Les modèles proposées avaient pour objectif principal de représenter le mieux possible le comportement du réacteur. Les modèles mis au point étaient relativement simples et ne comportaient qu'une seule population de bactéries. Ils ont été complexifiés en y incluant davantage de populations de bactéries différentes et en décomposant le procédé total en plusieurs étapes intermédiaires. Il résulte de cette augmentation de la précision des modèles une complexité et une absence de parcimonie paramétrique qui rendent difficile, voire impossible, toute tentative d'utilisation à des fins de commande du procédé.

Le modèle dynamique développé par [Bernard *et al.*, 2001] suppose deux populations de bactéries différentes (X_1 et X_2) ainsi qu'une séparation du procédé de digestion anaérobiose en deux phases distinctes :

- une phase d'acidogénèse durant laquelle la population de bactéries acidogéniques X_1 consomme le substrat organique S_1 pour produire du dioxyde de carbone CO_2 et des acides gras volatils (S_2) selon la réaction



- une phase de méthanogénèse durant laquelle la seconde population de bactéries X_2 utilise les acides gras volatils comme substrat afin de croître et de produire du dioxyde de carbone CO_2 et du méthane CH_4 selon la réaction



Le carbone inorganique total C est stocké sous forme de bicarbonate et de CO_2 dissous. Une variable Z est introduite pour représenter l'alcalinité totale dans le digesteur.

Le modèle dynamique est donc constitué d'un ensemble d'équations différentielles du premier ordre qui représentent, pour un débit de liquide en entrée D , l'évolution des deux populations de bactéries (X_1 et X_2), des substrats organiques (S_1 et S_2), de l'alcalinité totale Z et de la concentration en carbone inorganique C :

$$\left\{ \begin{array}{l} \frac{dX_1}{dt} = (\mu_1(S_1) - \alpha D)X_1 \\ \frac{dX_2}{dt} = (\mu_2(S_2) - \alpha D)X_2 \\ \frac{dS_1}{dt} = D(S_{1in} - S_1) - k_1\mu_1(S_1)X_1 \\ \frac{dS_2}{dt} = D(S_{2in} - S_2) + k_2\mu_1(S_1)X_1 - k_3\mu_2(S_2)X_2 \\ \frac{dZ}{dt} = D(Z_{in} - Z) \\ \frac{dC}{dt} = D(C_{in} - C) + k_4\mu_1(S_1)X_1 + k_5\mu_2(S_2)X_2 - q_{CO_2} \end{array} \right. \quad (3.23)$$

Les termes S_{1in} , S_{2in} , Z_{in} et C_{in} représentent respectivement les concentrations en entrée de substrat organique, d'acides gras volatils (AGV), l'alcalinité et le carbone inorganique total.

Les taux de croissance bactérienne $\mu_1(S_1)$ et $\mu_2(S_2)$ sont représentés respectivement par le modèle de Monod et celui de Haldane .

$$\mu_1(S_1) = \mu_{1\max} \frac{S_1}{S_1 + K_{S_1}} \quad (3.24)$$

$$\mu_2(S_2) = \mu_{2\max} \frac{S_2}{S_2 + K_{S_2} + \frac{S_2^2}{K_{I_2}}} \quad (3.25)$$

Les paramètres $\mu_{1\max}$ et $\mu_{2\max}$ représentent le taux de croissance maximal de la biomasse acidogénique et méthanolélique respectivement. Les termes K_{S_1} et K_{S_2} sont des paramètres de saturation associés respectivement à S_1 et S_2 . Le terme K_{I_2} est une constante d'inhibition associée à S_2 .

Le terme α représente une fraction supposée constante des bactéries attachée au support. Cette fraction n'est pas affectée par le phénomène de dilution.

Le terme q_{CO_2} est le débit molaire en CO_2 . On le calcule de manière analytique en utilisant l'expression

$$q_{CO_2} = k_7(k_8 P_{CO_2} + Z - C - S_2) \quad (3.26)$$

où le terme P_{CO_2} est la pression partielle en CO_2 donnée par l'expression

$$P_{CO_2} = \frac{\Phi - \sqrt{\Phi^2 - 4k_8 P_T [CO_2]}}{2k_8} \quad (3.27)$$

avec

$$\Phi = k_8 P_T + [CO_2] + \frac{k_6 \mu_2(S_2) X_2}{k_7} \quad (3.28)$$

La concentration en dioxyde de carbone $[CO_2]$ est donnée par $[CO_2] = C + S_2 - Z$. Le terme P_T est la pression totale en sortie du réacteur.

Les valeurs identifiées de toutes les constantes définies figurent dans [Bernard *et al.*, 2001].

L'utilisation de ce modèle dynamique permet de comparer les mesures obtenues en ligne aux résultats de simulation. On suppose tout au long du procédé que la biomasse, après sa croissance initiale, est maintenue constante : la concentration X_1 en bactéries acidogènes et X_2 en bactéries méthanogènes reste les mêmes tout au long des 30 jours d'expérimentation (figure 3.18). Du fait de la croissance initiale des populations de bactéries, le volume total utilisé est de 350 litres sur les 948 utilisables à l'origine.

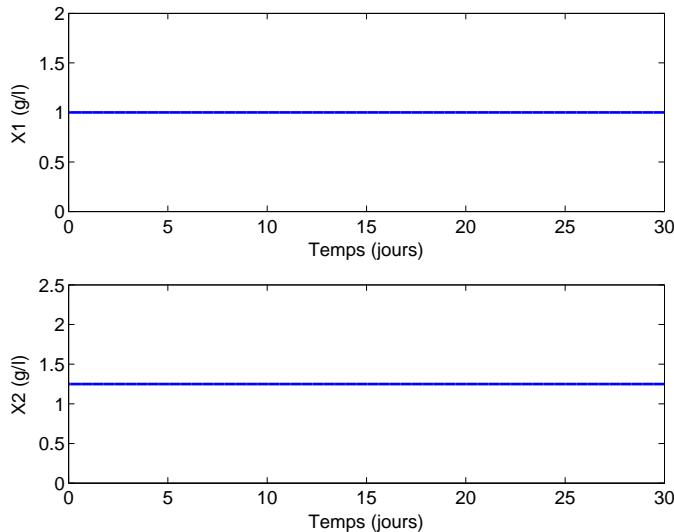


Fig. 3.18 - Concentrations en biomasse

La comparaison des résultats expérimentaux et des résultats fournis par le modèle dynamique est présentée sur l'ensemble de figures 3.19.

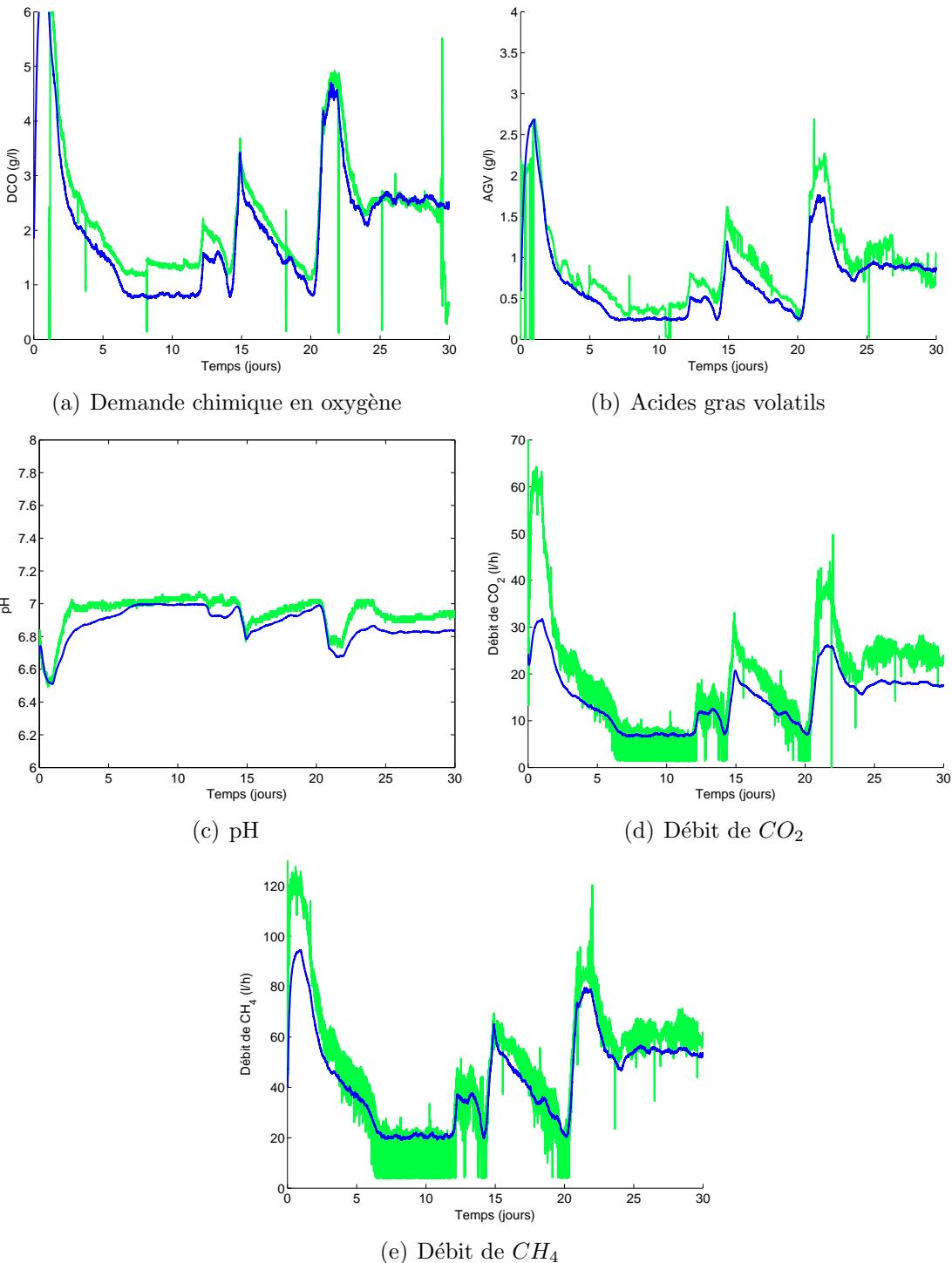


Fig. 3.19 - Comparaison des mesures obtenues en ligne (courbes claires) avec les résultats de simulation (courbes foncées)

Le modèle dynamique est donc valide sur un large domaine d'application du procédé de digestion anaérobie. Nous allons donc pouvoir comparer les résultats de simulation fournis par le modèle non linéaire de Volterra à ceux fournis par ce modèle dynamique en

plus des résultats expérimentaux afin de valider à la fois le modèle de Volterra présenté dans la première partie du mémoire et les techniques d'inversion de ce modèle décrites dans la deuxième partie du mémoire.

3.4.3 Application du modèle de Volterra au digesteur anaérobie

Le procédé de digestion anaérobie considéré ici présente, comme de nombreux autres procédés biologiques et biochimiques, un comportement dynamique fortement non linéaire dû en grande partie aux taux de croissance bactérienne $\mu_1(S_1)$ et $\mu_2(S_2)$ (3.24) et (3.25). Un tel comportement justifie l'utilisation de modèles non linéaires pour représenter le procédé dans son entier ou bien les relations entre les différentes grandeurs mesurées en ligne sur le pilote expérimental. Nous allons donc appliquer au digesteur anaérobie le modèle de Volterra décrit dans la première partie du mémoire [Bibes *et al.*, 2003b].

Si l'on envisage ensuite la commande du procédé de digestion anaérobie, il est important de pouvoir disposer en temps réel d'une mesure de grandeur d'entrée du réacteur. Bien souvent, en pratique, on ne dispose pas d'un capteur adéquat pour des raisons technologiques aussi bien qu'économiques. Et dans le cas où l'on dispose d'un capteur, il est souvent impossible de l'implémenter à l'entrée du réacteur. Une solution est alors d'effectuer des mesures de grandeurs en sortie du procédé et d'estimer la grandeur d'entrée en inversant le modèle qui la relie à une ou plusieurs des grandeurs mesurables. Sachant que la grandeur d'entrée peut être liée à d'autres grandeurs de sortie par un modèle de Volterra, nous allons appliquer les techniques d'inversion présentées dans la deuxième partie du mémoire afin de reconstruire la grandeur d'entrée souhaitée.

3.4.3.1 Modélisation

Les modèles de Volterra décrits dans le premier chapitre de ce mémoire sont des modèles mono-entrée mono-sortie. Il convient donc dans un premier temps de définir les grandeurs que nous considérons comme entrées et sorties du digesteur anaérobie. La description du réacteur en lit fixe (figure 3.14) amène naturellement à définir comme grandeur d'entrée le débit $D(t)$ de liquide pollué à l'entrée (figure 3.15(d)), avant son traitement par digestion anaérobie. Les autres grandeurs mesurées en ligne (débit de méthane, de dioxyde de carbone, demande chimique en oxygène, concentration en acides gras volatils, alcalinité totale, pH) pourront être considérées comme des grandeurs de sortie liées à la grandeur d'entrée. La demande chimique en oxygène (DCO) et la concentration en acides gras volatils (AGV) sont les principales mesures de la pollution et donc les témoins les plus importants de la dégradation de cette dernière par digestion anaérobie. On s'attachera donc particulièrement à la modélisation du lien entre débit d'entrée $D(t)$ et DCO et du lien entre $D(t)$ et AGV.

a) Identification de modèles de Volterra

Nous allons dans un premier temps utiliser le modèle dynamique (3.23) du procédé pour identifier différents modèles de Volterra entre la grandeur d'entrée $D(t)$ et les différentes grandeurs de sortie. La première étape est de créer un fichier d'entrée $D(t)$ (figure 3.20) et d'utiliser le modèle dynamique pour simuler les grandeurs de sortie (DCO, AGV, différents débits, alcalinité totale, pH).

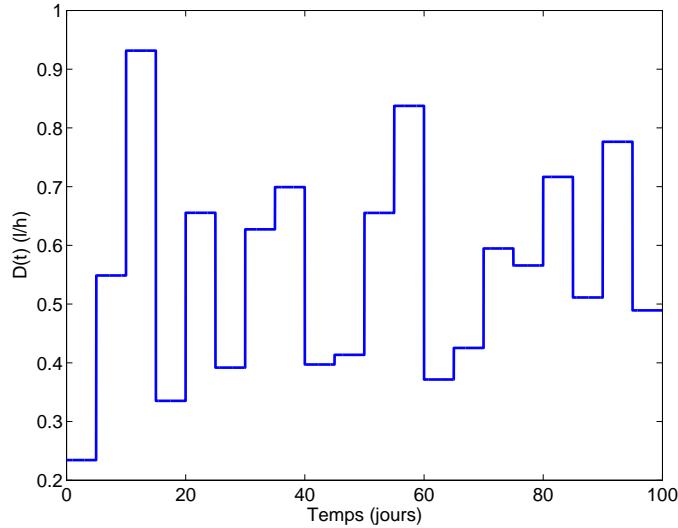


Fig. 3.20 - Débit d'entrée

Les conditions de simulation sont les suivantes : la période d'échantillonnage est fixée à 2 minutes (ce sera le cas dans toutes les simulations qui suivent). Les concentrations S_{1in} , S_{2in} , Z_{in} et C_{in} en entrée sont maintenues constantes ($S_{1in} = 6.86\text{g/l}$, $S_{2in} = 110.06\text{mmol/l}$, $Z_{in} = 138.27\text{meq/l}$, $C_{in} = 32.27\text{mmol/l}$).

Les valeurs des constantes utilisées dans le modèle dynamique (3.23) sont :

$$\begin{aligned}
 \alpha &= 0.5 \\
 k_1 &= 42.14 \\
 k_2 &= 116.5 \\
 k_3 &= 268 \\
 k_4 &= 50.6 \\
 k_5 &= 343.6 \\
 k_6 &= 453 \\
 k_7 &= 19.8 \\
 k_8 &= 16
 \end{aligned} \tag{3.29}$$

Les valeurs des constantes utilisées pour le calcul des taux de croissance bactérienne sont :

$$\begin{aligned}
 \mu_{1\max} &= 1.2 \\
 \mu_{2\max} &= 0.74 \\
 K_{S_1} &= 8.9 \\
 K_{S_2} &= 25 \\
 K_{I_2} &= 256
 \end{aligned} \tag{3.30}$$

On en déduit les taux de croissance bactérienne $\mu_1(S_1)$ et $\mu_2(S_2)$ (figure 3.21).

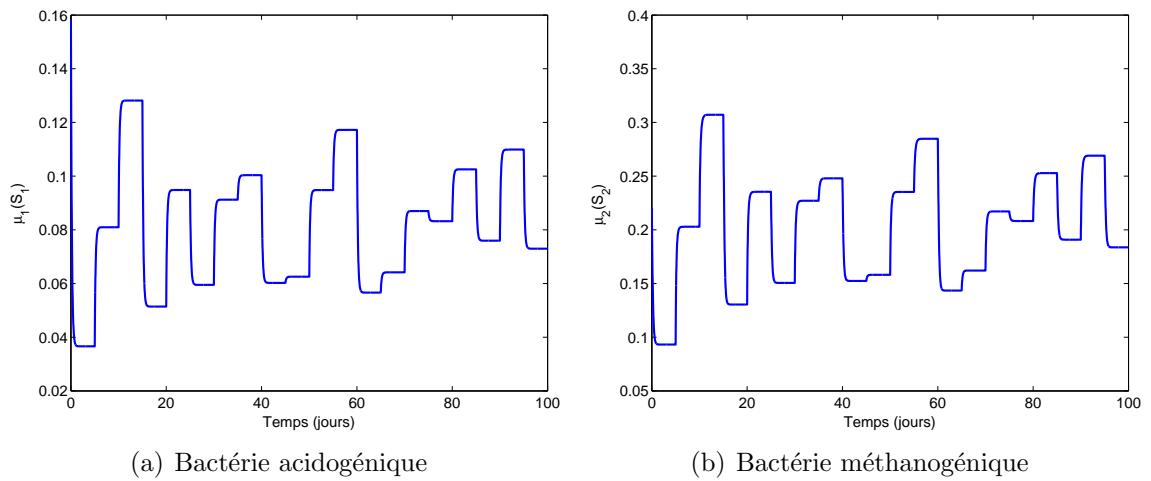


Fig. 3.21 - Taux de croissance bactérienne

Les différentes grandeurs de sortie obtenues par simulation (selon les conditions que l'on vient de décrire) sont données sur la figure 3.22.

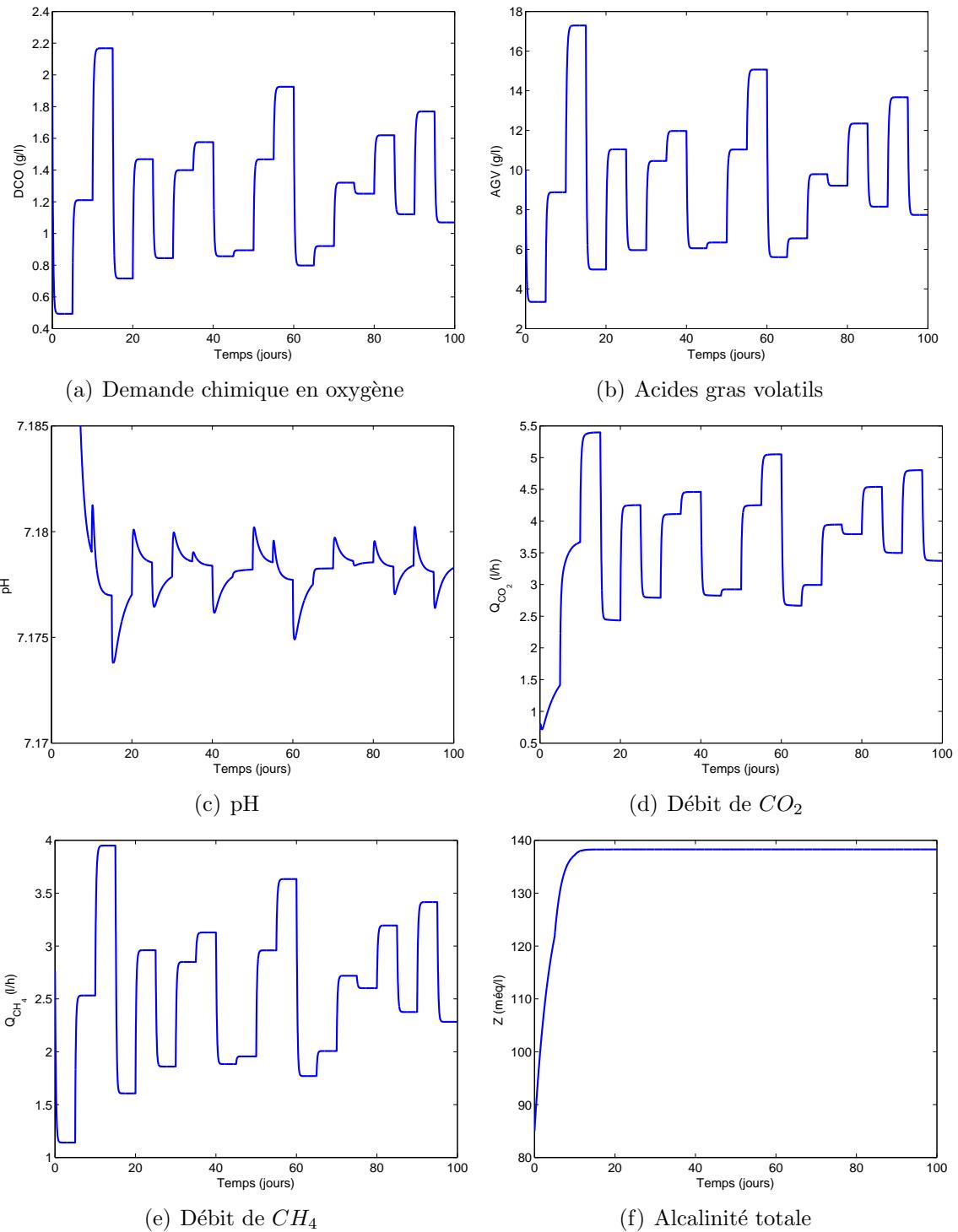


Fig. 3.22 - Fichiers de points utilisés pour l'identification de modèles de Volterra

Nous considérons des modèles de Volterra tronqués aux deux premiers termes. Chaque noyau est ensuite modélisé par un développement sur des fonctions de transfert. Le premier noyau est développé sur une seule fonction de transfert tandis que le deuxième noyau est développé sur deux fonctions de transfert. Dans chaque cas le modèle comporte donc

au total 8 paramètres (3 pôles $a_{1,1}$, $a_{2,1}$, $a_{2,2}$ et 5 gains $\gamma_{0,1,1}$, $\gamma_{0,2,1}$, $\gamma_{1,2,1}$, $\gamma_{0,2,2}$, $\gamma_{1,2,2}$) et peut être schématisé comme l'indique la figure 3.23.

Les différents modèles de Volterra identifiés possèdent tous la même structure. Nous allons voir que ce choix de structure garantit dans chacun des cas une précision suffisante des résultats. L'augmentation de la complexité du modèle (et donc la diminution de sa parcimonie paramétrique) n'améliore pas les résultats.

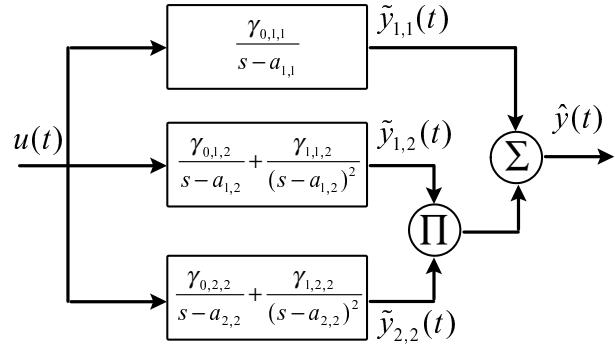


Fig. 3.23 - Structure du modèle de Volterra

Une fois l'identification paramétrique achevée (par programmation non linéaire), on dispose de 5 modèles de Volterra de la forme présentée par la figure (3.23) reliant le débit d'entrée $D(t)$ à la DCO , à la concentration en ACV , au débit de CO_2 , au débit de CH_4 et au pH respectivement. L'ensemble des figures 3.24 est une comparaison de la sortie de chaque modèle de Volterra avec les résultats de simulation du modèle dynamique.

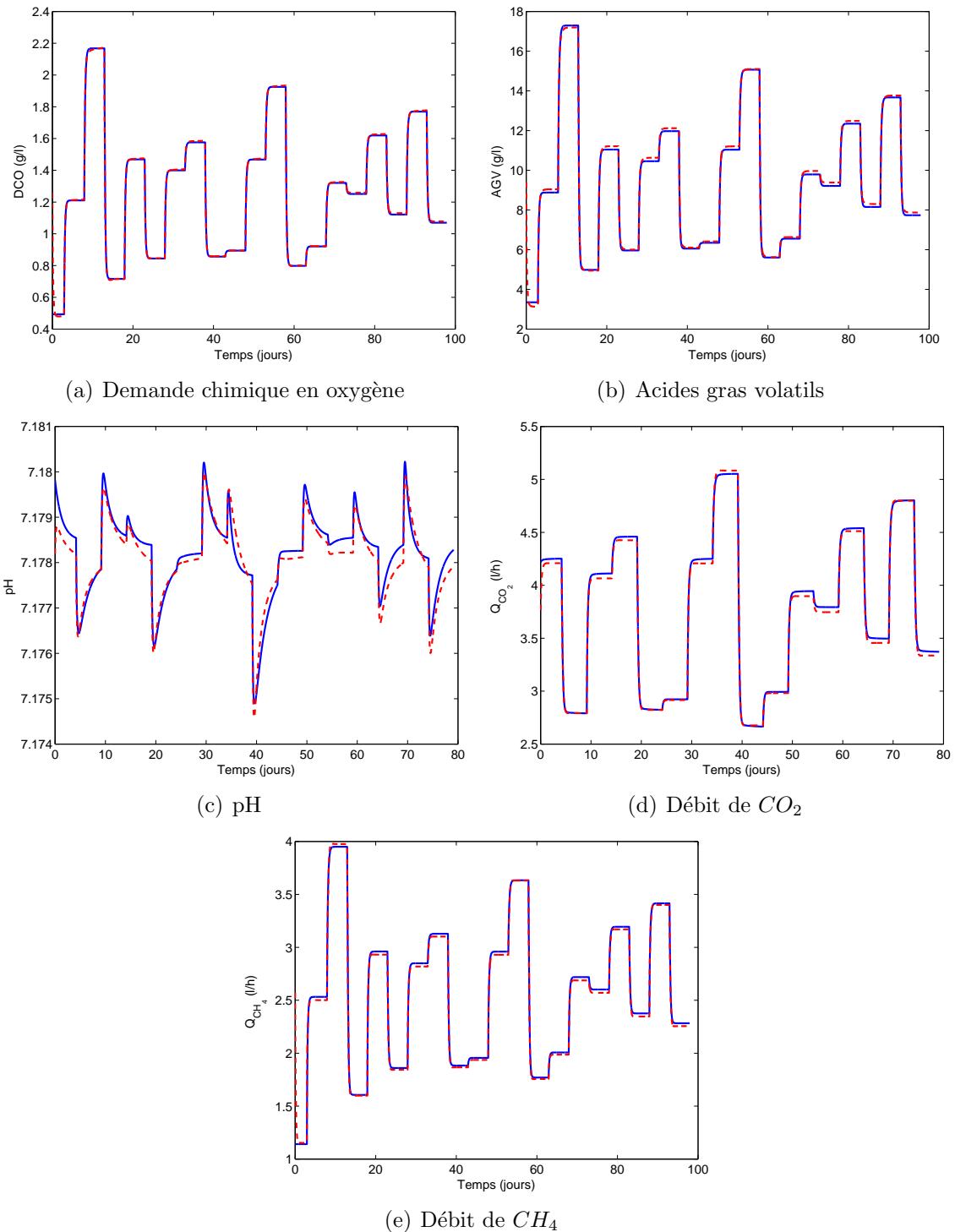


Fig. 3.24 - Comparaison entre sorties des modèles de Volterra (ligne pointillée) et sorties du modèle dynamique (ligne continue)

Dans chacun des 5 cas, le modèle de Volterra identifié fournit une bonne approximation de la sortie du modèle dynamique malgré un nombre de paramètres relativement faible. L'ajout d'un troisième terme dans le modèle de Volterra n'améliore pas la précision des

résultats obtenus, malgré un nombre plus important de paramètres.

Par contre, l'alcalinité Z (figure 3.22(f)) semble avoir un comportement indépendant de celui du débit d'entrée $D(t)$. Une fois passé un régime transitoire, la valeur de $Z(t)$ se stabilise autour d'une valeur constante. Sans comportement dynamique autour de cette valeur, il n'est pas possible de modéliser le lien entre $D(t)$ et $Z(t)$ par une série de Volterra.

b) Test sur les données mesurées

L'étape suivante consiste à valider les modèles de Volterra identifiés en comparant les sorties obtenues pour le débit d'entrée $D(t)$ (figure 3.15(d)) aux données expérimentales.

On applique donc le débit d'entrée $D(t)$ aux différents modèles de Volterra identifiés dans la partie précédente. L'ensemble de figures 3.25 compare les sorties de chacun des modèles de Volterra aux données expérimentales.

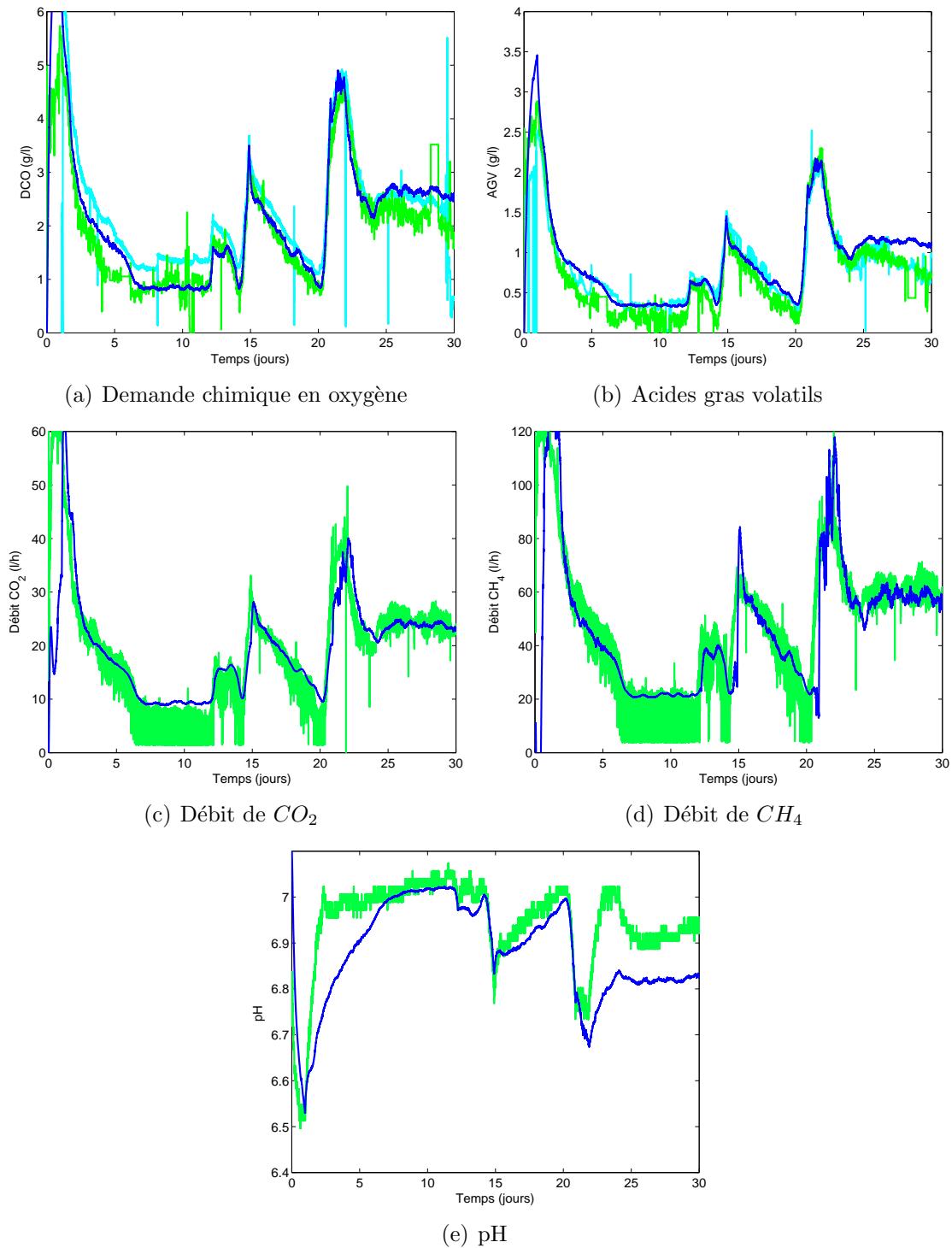


Fig. 3.25 - Validation expérimentale des modèles de Volterra (courbes claires : données expérimentales - courbe foncée : sortie du modèle de Volterra)

Dans chacun des cas, on constate une assez bonne adéquation entre mesures expérimentales et résultats de simulation des modèles de Volterra pour un même débit d'entrée. Quelques remarques sont cependant nécessaires :

- dans le cas de la DCO (figure 3.25(a)) et de la concentration en AGV (figure 3.25(b)), il a été nécessaire de multiplier la sortie du modèle de Volterra par un gain dont la valeur est 2.3 et 0.8 respectivement. Cette différence entre la sortie du modèle de Volterra et les données expérimentales peut s'expliquer : le modèle dynamique que nous avons utilisé pour identifier les modèles de Volterra a été établi à partir de données expérimentales datant du début d'utilisation du digesteur anaérobiose. A ce moment, le volume utile du réacteur était de 948l. Toutes les valeurs des paramètres du modèle ont été identifiés à partir de ces données. Par contre, les données expérimentales utilisées ici correspondent à une utilisation plus récente du digesteur. Les conditions expérimentales ont changé : entre autres changements, le volume utile du réacteur a diminué jusqu'à 350l à cause de la croissance bactérienne. De plus, les valeurs de certains paramètres ont dû (ou doivent) être à nouveau identifiées. Le modèle dynamique (3.23) qui a servi à identifier les paramètres des différents modèles de Volterra a donc été utilisé dans des conditions où il ne correspond pas exactement aux données expérimentales, ce qui explique la différence entre les mesures et les sorties du modèle de Volterra. Cette sensibilité du modèle de Volterra aux changements de conditions expérimentales pourrait être étudiée afin d'être utilisée à la mise en évidence de problèmes pratiques, tel le colmatage du réacteur.
- ce problème de gain n'apparaît pas dans le cas du débit de CO_2 et de CH_4 (courbes 3.25(c) et 3.25(d)).
- dans le cas du pH, on constate sur la courbe de mesures expérimentales (figure 3.15(a)) que sa variation est l'inverse de celle du débit d'entrée (figure 3.15(d)). Ceci justifie le gain unitaire négatif par lequel on doit multiplier la sortie du modèle de Volterra pour qu'elle corresponde au pH mesuré expérimentalement. On constate tout de même une différence de gain marquée à certains endroits entre les deux courbes de la figure 3.25(e). Cependant, la variation expérimentale de pH étant faible (le pH est régulé autour de la valeur optimale de 7 pour des conditions idéales de fonctionnement des cultures bactériennes), le modèle de Volterra peut suffire à prévoir cette variation en fonction de celle du débit d'entrée.
- dans tous les cas de figure, les résultats de simulation obtenus à l'aide des modèles de Volterra sont tout à fait comparables à ceux obtenus par le modèle dynamique (figures 3.19(a) à 3.19(e)).

Cet ensemble de résultats de comparaison permet ici de valider l'approche de représentation des liens entre le débit d'entrée et différentes grandeurs de sortie par modèles de Volterra. La précision de ces modèles est suffisante pour ensuite envisager leur inversion et permettre ainsi la reconstruction du débit d'entrée (ou de la DCO en entrée) à partir de la mesure de grandeurs de sortie.

3.4.3.2 Reconstruction de grandeurs par inversion du modèle de Volterra

a) Régularisation de Tikhonov

La méthode de régularisation de Tikhonov présentée dans le deuxième chapitre de ce mémoire est utilisée afin d'inverser les modèles de Volterra reliant le débit d'entrée $D(t)$ du digesteur anaérobiose aux différentes mesures de sorties (DCO, AGV, débit de CO_2 , débit de CH_4 , pH). Ainsi, les mesures de grandeurs les plus accessibles permettront, par inversion de modèle, de reconstruire une des grandeurs d'entrée, comme le débit (facilement mesurable) ou la DCO en entrée (dont le protocole de mesure est difficile, voire impossible à implémenter). La reconstruction du débit d'entrée $D(t)$ permettra de valider la méthode employée en comparant la grandeur reconstruite à la mesure du débit d'entrée.

Dans toute la partie qui suit, on utilise le modèle de Volterra développé sur base de fonctions orthonormées généralisée.

Considérons le débit d'entrée $D(t)$ représenté sur la figure 3.26(a). La figure 3.26(b) représente la sortie du modèle dynamique du digesteur (3.23) et la sortie du modèle de Volterra représentant le lien entre débit d'entrée et DCO en sortie : ce modèle de Volterra est tronqué à ses deux premiers noyaux, développés respectivement sur une et trois fonctions de la base orthonormée généralisée. Les pôles des fonctions sont tous fixés à 10. Le nombre total de paramètres est égal à 7. Les deux courbes de la figure 3.26(b) sont suffisamment proches pour envisager ensuite l'inversion du modèle de Volterra.

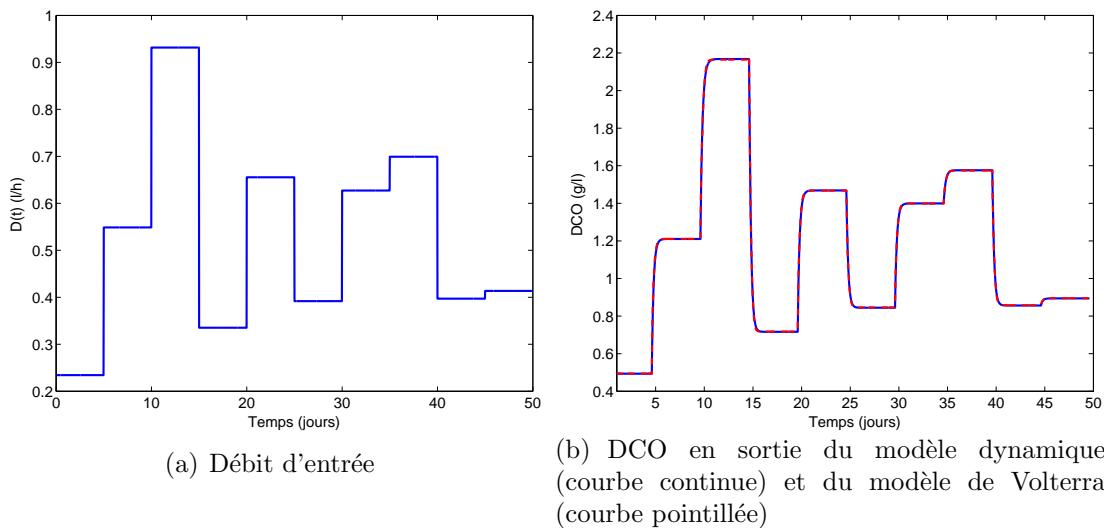


Fig. 3.26 - Modélisation du lien entre débit d'entrée et DCO en sortie

Dans le cas où aucun bruit de mesure ne s'ajoute à la DCO en sortie, aucune régularisation de l'entrée $D(t)$ reconstruite n'est nécessaire. On fixe donc la valeur du paramètre λ à 0. Le résultat de la reconstruction est représenté sur la figure 3.27.

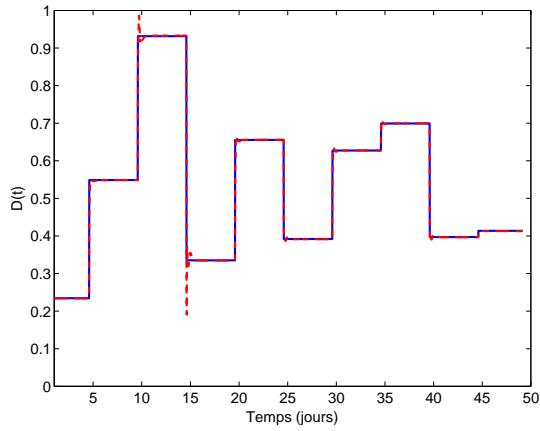


Fig. 3.27 - Comparaison entre $D(t)$ (courbe continue) et sa reconstruction (courbe pointillée)

Malgré la présence de plusieurs pics de reconstruction sur une partie du signal reconstruit, les deux courbes sont pratiquement superposées, attestant de la qualité de reconstruction du débit d'entrée $D(t)$.

On ajoute maintenant un bruit de mesure à la DCO en sortie (rapport signal/bruit égal à 100). Le modèle de Volterra est à nouveau identifié et l'on conserve le même débit d'entrée (figure 3.26(a)). La comparaison entre les deux courbes de la figure 3.28 montre que le modèle de Volterra est suffisamment précis pour modéliser correctement la DCO en sortie.

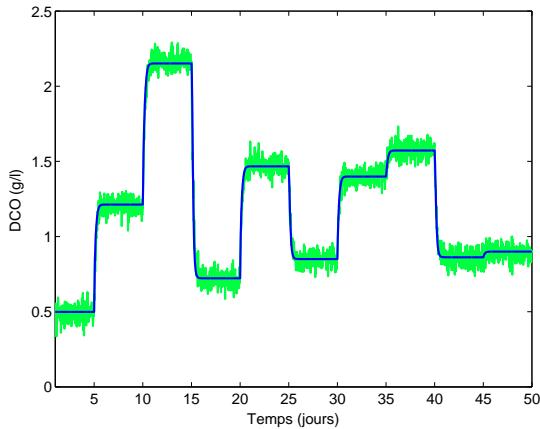


Fig. 3.28 - Comparaison entre DCO bruitée (courbe claire) et DCO en sortie du modèle de Volterra (courbe foncée)

Afin de reconstruire correctement le débit d'entrée, il est maintenant nécessaire de régulariser la solution du problème inverse. La figure 3.29 représente le débit d'entrée $D(t)$

reconstruit pour deux valeurs du paramètre de régularisation ($\lambda = 0.01$ et $\lambda = 0.1$). Dans chacun des deux cas, la matrice de régulation utilisée est la matrice des dérivées secondes.

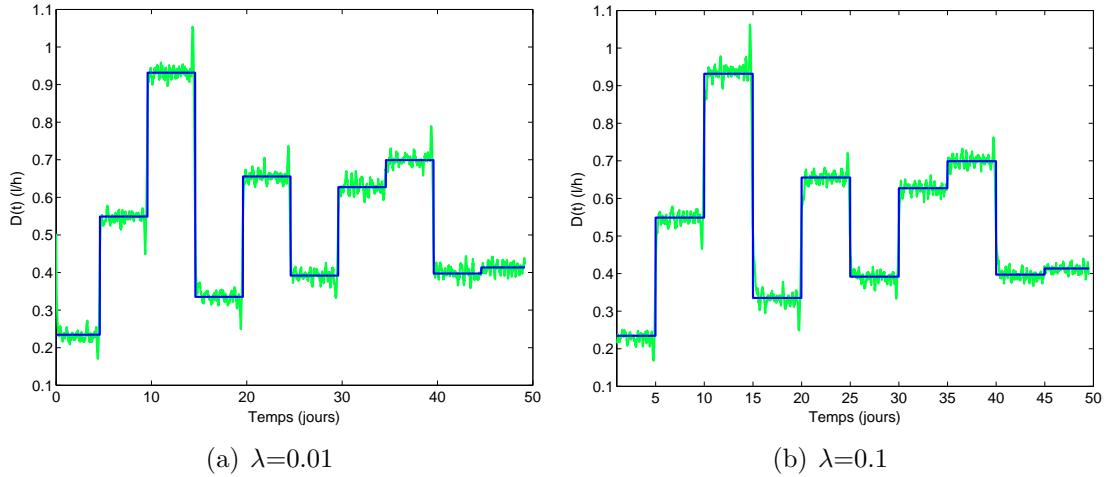


Fig. 3.29 - Comparaison entre $D(t)$ (courbe foncée) et sa reconstruction (courbe claire) pour différentes valeurs de λ

L'utilisation de la matrice des dérivées secondes permet une bonne reconstruction du débit d'entrée sans qu'il y ait pour autant une perte d'information sur le signal reconstruit.

Ces différents essais en présence de bruit de mesure permettent de valider la méthode d'inversion par régularisation dans le cas du modèle du digesteur anaérobio. On utilise maintenant le débit d'entrée $D(t)$ et la DCO en sortie mesurés expérimentalement (figure 3.30).

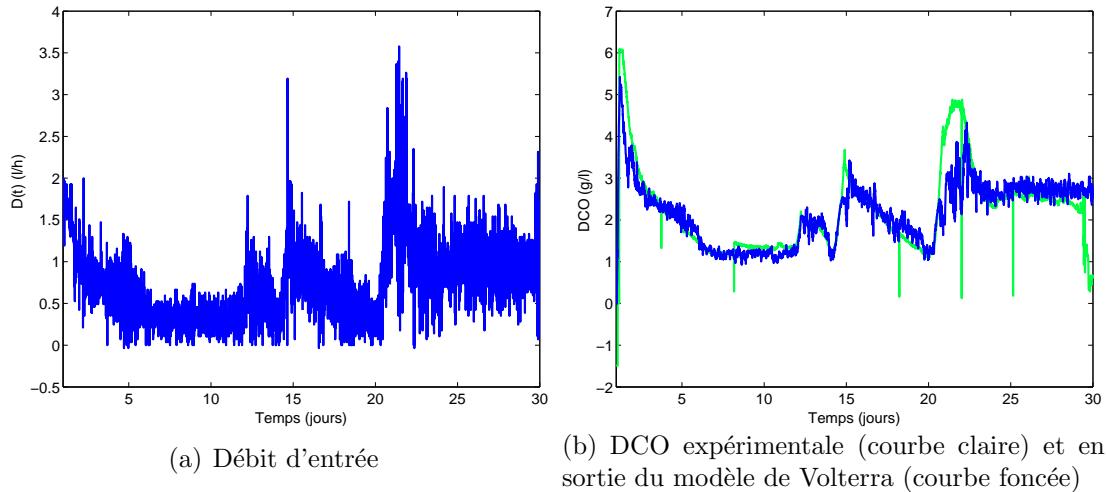


Fig. 3.30 - Mesures expérimentales du débit d'entrée et de la DCO en sortie

Le modèle de Volterra utilisé reste le même que précédemment (troncature aux deux premiers noyaux). Malgré une différence notable entre la DCO mesurée et la sortie du modèle de Volterra entre 20 et 25 jours (figure 3.30(b)), le modèle de Volterra est suffisamment précis pour envisager la reconstruction du débit d'entrée.

L'application de la méthode d'inversion conduit aux résultats présentés sur les courbes de la figure 3.31.

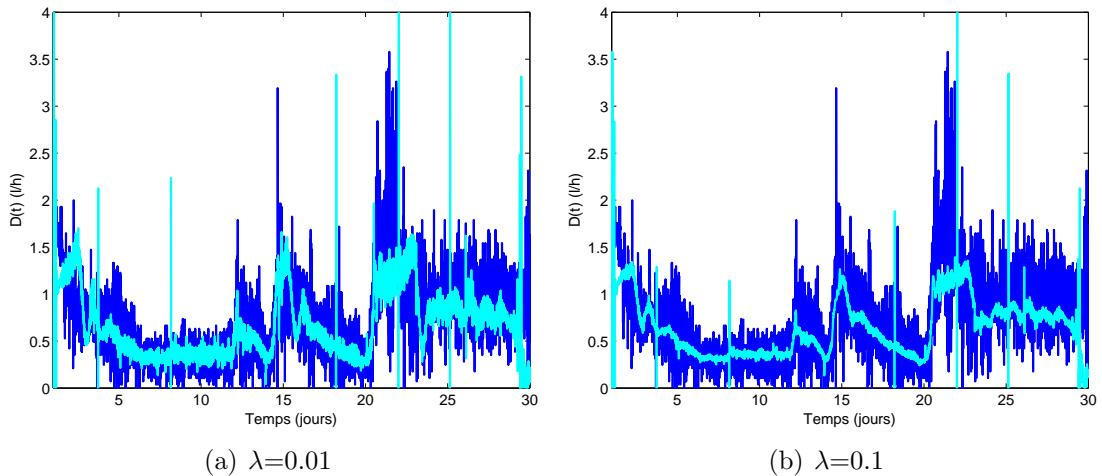


Fig. 3.31 - Débit d'entrée expérimental (courbe foncée) et sa reconstruction (courbe claire)

Si l'on compare les deux courbes, l'influence du paramètre de régularisation est nettement visible. La valeur optimale de λ se situe à nouveau autour de 0.1.

La méthode d'inversion du modèle de Volterra par régularisation de Tikhonov peut donc s'appliquer à des fichiers de points expérimentaux. Les résultats de reconstruction du débit d'entrée $D(t)$ sont de bonne qualité malgré là aussi une différence notable entre les deux courbes de la figure 3.31(b) entre 20 et 25 jours.

La deuxième étape consiste maintenant à reconstruire une grandeur d'entrée dont le protocole de mesure est difficile, voire impossible à implémenter : la DCO en entrée du réacteur biologique. L'accès à la mesure de la DCO en entrée est important dans le cas du digesteur anaérobiose étudié car cette grandeur permet de quantifier le taux de pollution présent dans le liquide en entrée du réacteur.

La figure 3.32(a) représente la DCO en entrée DCO_{in} du modèle dynamique et la figure 3.32(b) la DCO obtenue en sortie comparée à la sortie du modèle de Volterra. Dans ce cas, le modèle de Volterra est tronqué à ses 3 premiers noyaux, chacun développé respectivement sur une, une et deux fonctions de la base orthonormée généralisée. Les pôles des fonctions sont fixés à 5. Le nombre total de paramètres est égal à 6.

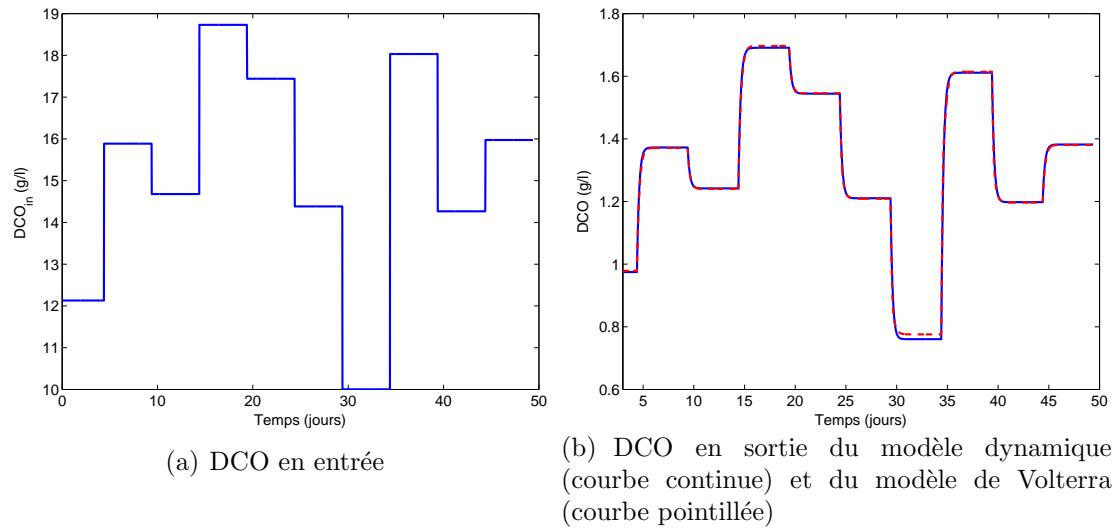


Fig. 3.32 - Modélisation du lien entre DCO en entrée et DCO en sortie

La reconstruction de la DCO en entrée peut être effectuée sans régularisation ($\lambda=0$). Le résultat est représenté sur la figure 3.33.

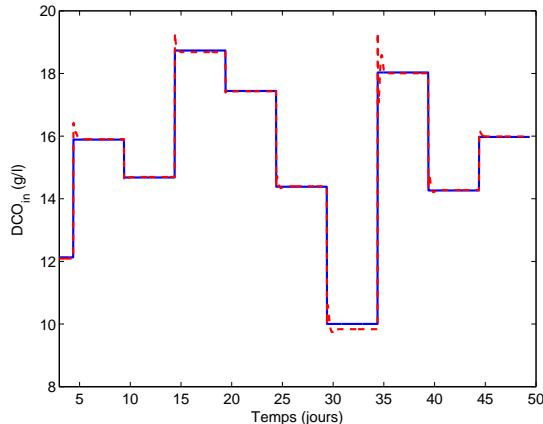


Fig. 3.33 - Comparaison entre la DCO en entrée (courbe continue) et sa reconstruction (courbe pointillée)

On constate également dans ce cas la présence de pics de reconstruction mais la qualité de reconstruction de la DCO en entrée reste tout à fait correcte.

On ajoute à nouveau un bruit de mesure à la DCO en sortie (rapport signal/bruit égal à 100). Le modèle de Volterra est à nouveau identifié et l'on conserve la même DCO en entrée (figure 3.32(a)). La comparaison entre les deux courbes de la figure 3.34 montre que le modèle de Volterra est suffisamment précis pour modéliser correctement la DCO en sortie.

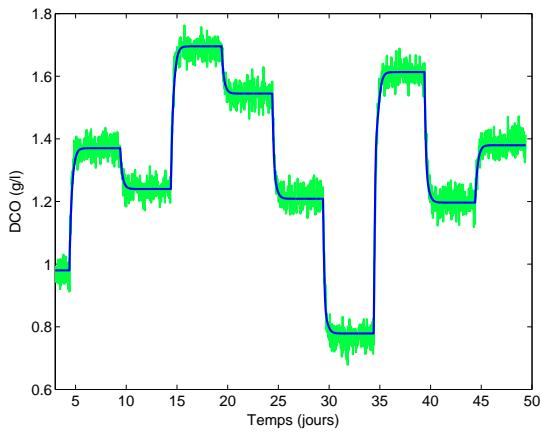


Fig. 3.34 - Comparaison entre DCO bruitée (courbe claire) et DCO en sortie du modèle de Volterra (courbe foncée)

La figure 3.35 représente la DCO en entrée reconstruite pour deux valeurs du paramètre de régulation ($\lambda = 10^{-5}$ et $\lambda = 10^{-4}$). Dans chacun des cas, la matrice de régulation utilisée est la matrice des dérivées secondes.

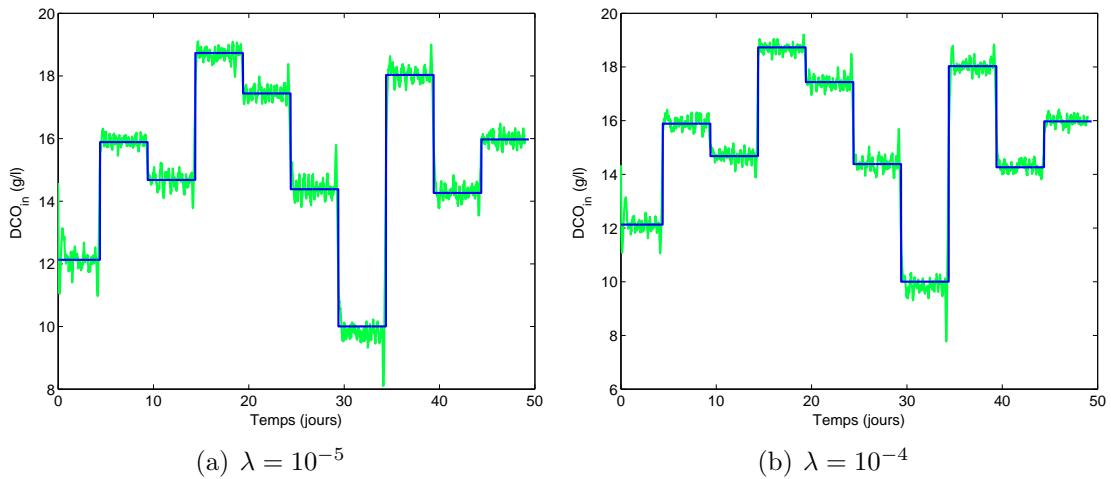


Fig. 3.35 - Comparaison entre la DCO en entrée (courbe foncée) et sa reconstruction (courbe claire) pour différentes valeurs de λ

La valeur optimale du coefficient de régularisation se situe autour de 10^{-4} . On obtient pour cette valeur une bonne reconstruction de la DCO sans dégradation.

Cette méthode d'inversion du modèle de Volterra par régularisation de Tikhonov peut donc être appliquée à la reconstruction de grandeurs non mesurables directement.

La concentration en polluant étant beaucoup plus importante dans le liquide à l'entrée du réacteur, cette méthode pourrait éventuellement être utilisée en lieu et place d'un

capteur. Nous avons constaté que le temps de calcul nécessaire à l'inversion du modèle de Volterra était au mieux égal à quelques minutes, pour une durée globale du procédé qui s'étend sur plusieurs jours. Ceci permet d'envisager une reconstruction de la DCO d'entrée en temps réel.

b) Inversion par retour d'entrée reconstruite

Les résultats de reconstruction par régularisation de Tikhonov étant probants, nous allons à présent les comparer à ceux obtenus par les deux méthodes d'inversion par retour d'entrée reconstruite décrites dans le deuxième chapitre du mémoire.

On utilise le débit d'entrée $D(t)$ de la figure 3.36(a). La DCO en sortie du modèle dynamique à laquelle s'ajoute un bruit de mesure (rapport signal sur bruit égal à 100) (3.23) est représentée sur la figure 3.36(b).

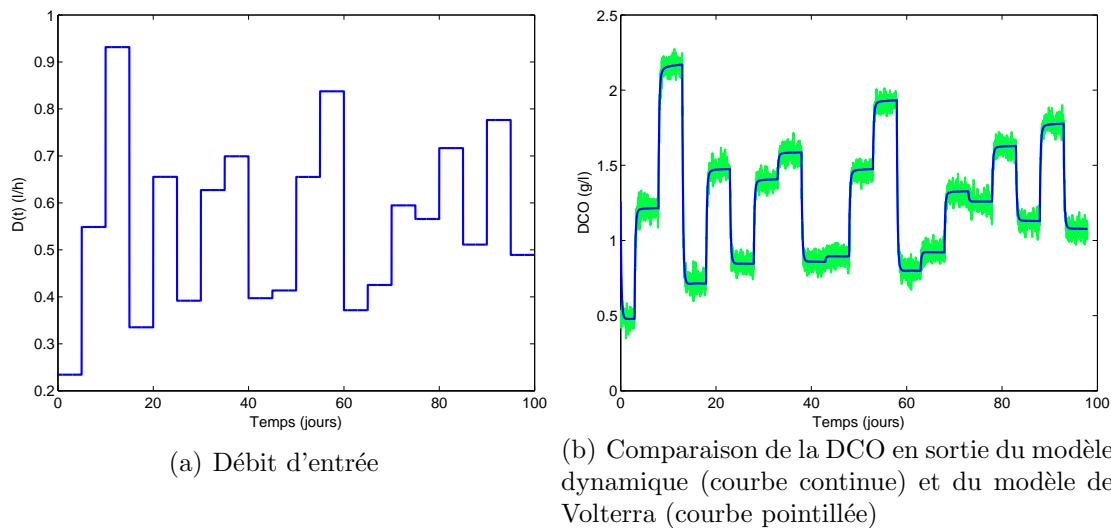


Fig. 3.36 - Signaux d'entrée et de sortie

On cherche ensuite à représenter le lien entre le débit d'entrée $D(t)$ et la DCO en sortie par un modèle de Volterra développé sur fonctions de transfert. On utilise dans toute cette partie un modèle de Volterra tronqué à ses deux premiers termes, le premier noyau est modélisé par une fonction de transfert tandis que le deuxième noyau est modélisé par un développement sur deux fonctions de transfert. Le nombre total de paramètres du modèle est égal à 8. La figure 3.36(b) compare la sortie du modèle de Volterra avec la DCO en sortie du modèle dynamique. On constate à nouveau que la modèle de Volterra est suffisamment précis pour envisager son inversion et ainsi reconstruire le débit d'entrée $D(t)$ à partir de la DCO en sortie.

On applique alors les deux méthodes de reconstruction d'un signal par retour d'entrée reconstruite décrites dans le premier chapitre. Le résultat des deux reconstructions est

représenté sur la figure 3.37.

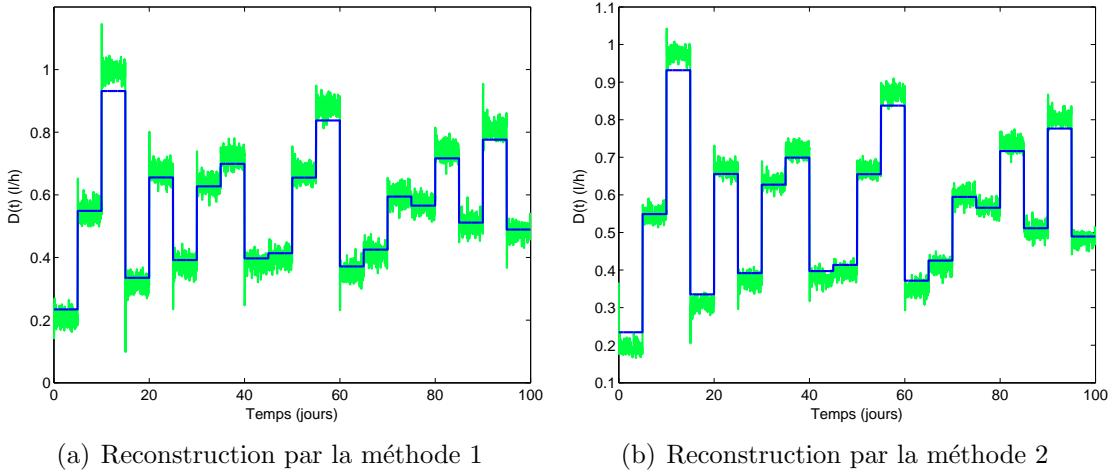


Fig. 3.37 - Comparaison entre débit d'entrée reconstruit (courbe claire) et mesuré (courbe foncée) pour les 2 méthodes de reconstruction

Malgré une qualité de reconstruction qui peut sembler moindre par rapport à celle obtenue par régularisation (figure 3.29(a)), les résultats obtenus restent satisfaisants et cela sans que nous ayons eu à tenir compte d'une éventuelle amplification du bruit de mesure ajouté à la DCO en sortie. L'amplitude du bruit (ou erreur de reconstruction) que l'on retrouve sur le signal $D(t)$ reconstruit est du même ordre de grandeur que l'amplitude du bruit de mesure ajouté à la DCO utilisée pour reconstruire $D(t)$.

Afin d'appliquer ces deux méthodes sur les données expérimentales, on utilise maintenant le débit d'entrée $D(t)$ et la DCO en sortie mesurés expérimentalement (figure 3.38).

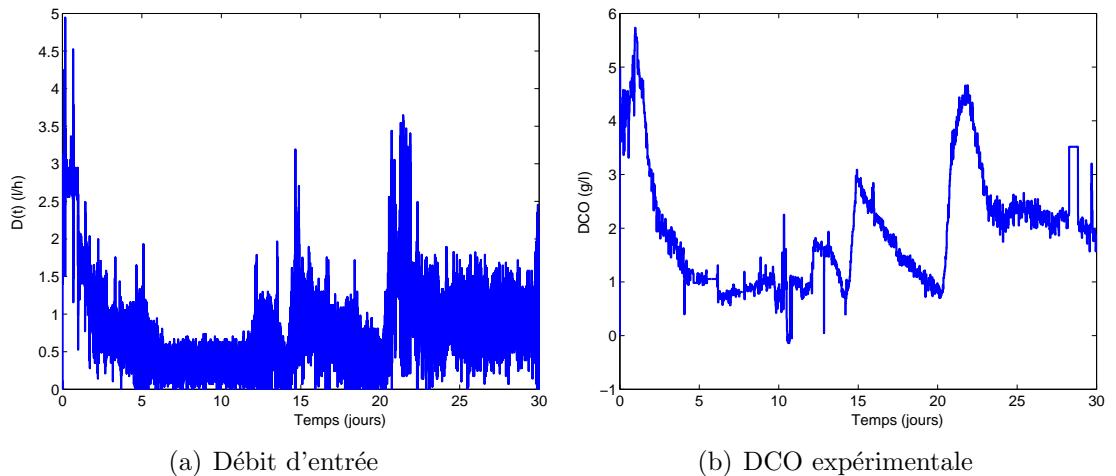


Fig. 3.38 - Mesures expérimentales du débit d'entrée et de la DCO en sortie

Le modèle de Volterra utilisé reste le même que précédemment (troncature aux deux premiers noyaux).

L'application des deux méthodes d'inversion conduit aux résultats présentés sur les courbes de la figure 3.39.

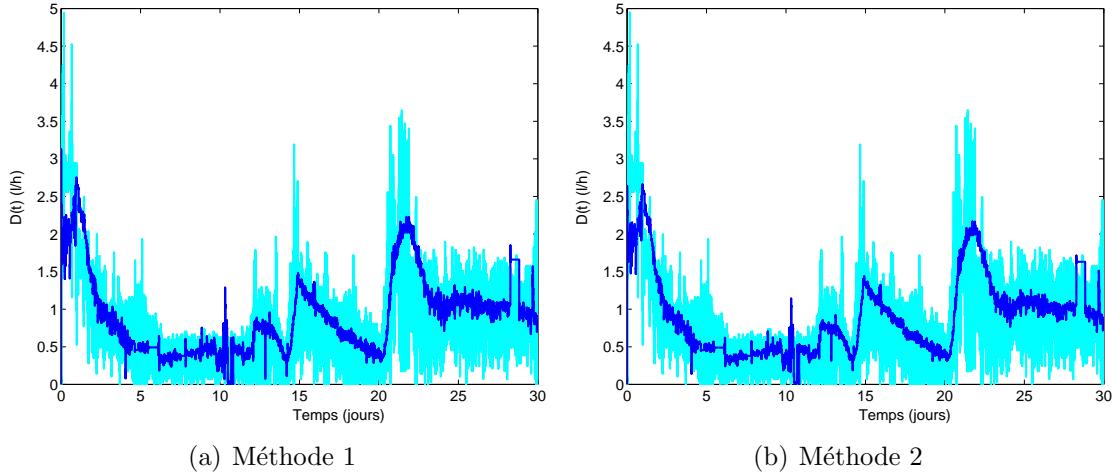


Fig. 3.39 - Débit d'entrée expérimental (courbe claire) et sa reconstruction (courbe foncée)

De manière plus précise, la figure 3.40 compare la reconstruction du débit d'entrée par les deux méthodes avec le débit d'entrée mesuré expérimentalement et filtré.

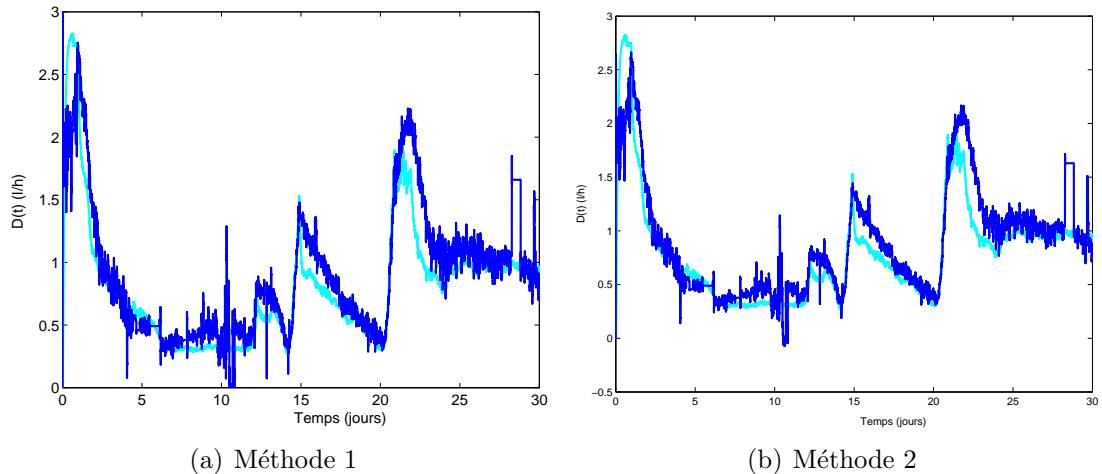


Fig. 3.40 - Débit d'entrée expérimental filtré (courbe claire) et sa reconstruction (courbe foncée)

Les résultats obtenus sur des fichiers de points expérimentaux sont de bonne qualité et restent tout à fait comparables à ceux obtenus par la méthode d'inversion par régularisation (figure 3.31(b)). En comparaison avec cette dernière méthode, on note également une

nette amélioration de la rapidité des calculs (moins d'une minute) lors de l'application de ces deux méthodes.

La deuxième étape consiste maintenant à appliquer ces deux méthodes afin de reconstruire la DCO en entrée du réacteur.

La figure 3.41(a) représente la DCO en entrée (DCO_{in}) du modèle dynamique et la figure 3.41(b) la DCO bruitée obtenue en sortie comparée à la sortie du modèle de Volterra. Le modèle de Volterra qui relie ces deux grandeurs est tronqué à ses 2 premiers noyaux, chacun développé respectivement sur une et deux fonctions de transfert. Le nombre total de paramètres est égal à 8.

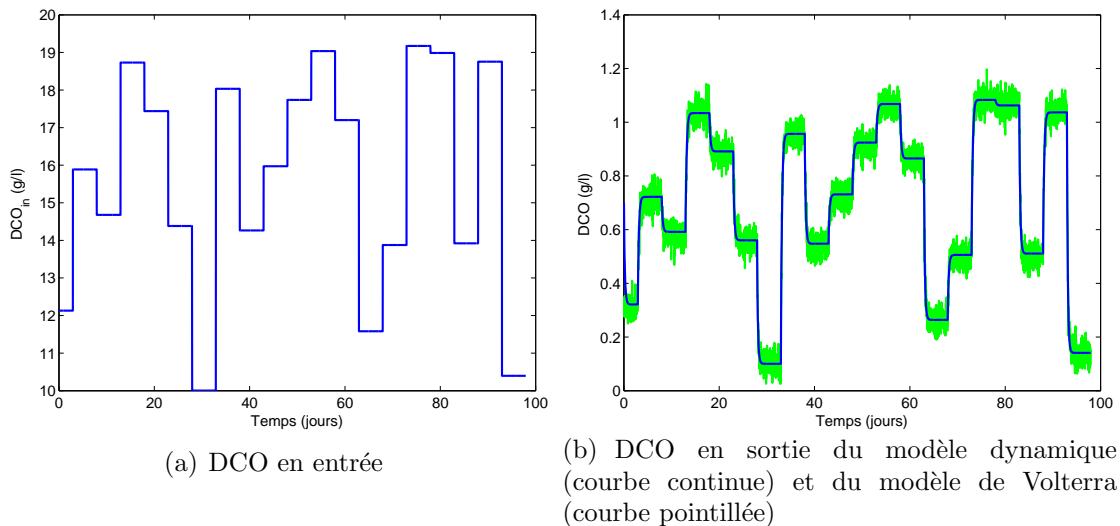


Fig. 3.41 - Modélisation du lien entre DCO en entrée et DCO en sortie

Le résultat de la reconstruction de la DCO en entrée par chacune des deux méthodes est représenté sur la figure 3.42.

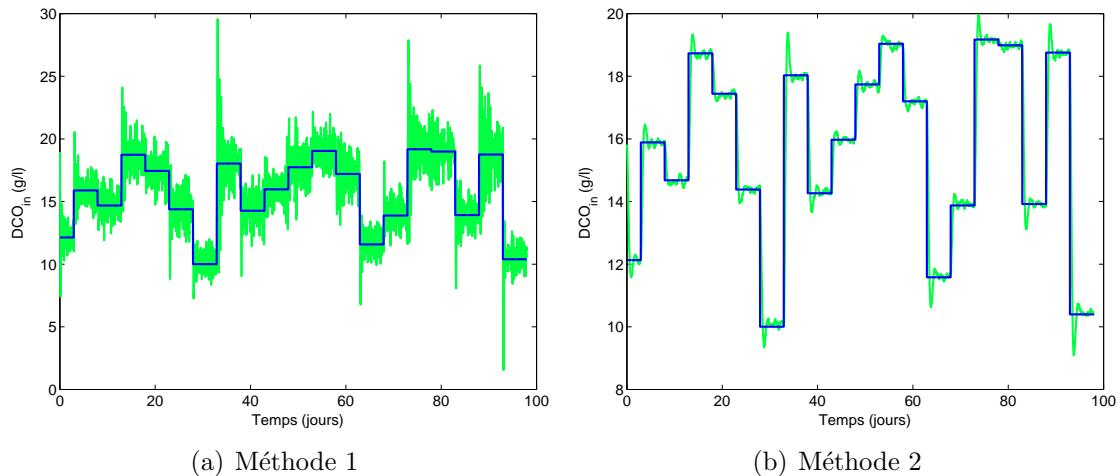


Fig. 3.42 - DCO en entrée (courbe foncée) et DCO reconstruite (courbe claire)

Dans le cas de la reconstruction de la DCO en entrée à partir de la DCO en sortie bruitée, il apparaît en plus de l'erreur de reconstruction due au bruit de mesure, des pics d'amplitude importante qui nuisent à la qualité de la DCO en entrée reconstruite. Ce phénomène est atténué lorsque l'on utilise la deuxième méthode de reconstruction.

3.4.4 Conclusion

L'utilisation du modèle dynamique du procédé et des données expérimentales permettent dans ce cas d'étude de valider en simulation les méthodes de modélisation et d'inversion non linéaire présentées dans les deux premiers chapitres du mémoire.

La modélisation des liens non linéaires entre la variation de débit de liquide pollué en entrée du procédé et différentes grandeurs en sortie (DCO, AGV, pH, débits de gaz) est réalisée à l'aide de modèles de Volterra dont on s'est attaché à réduire la complexité et le nombre de paramètres. Cette simplification des modèles permet ensuite d'envisager leur inversion en un temps relativement réduit.

Le deuxième objectif de l'étude était celui de la reconstruction de grandeurs non mesurables par l'inversion du modèle de Volterra et l'utilisation de grandeurs plus facilement mesurables. La DCO en entrée du réacteur a été reconstruite à partir de la mesure de la DCO en sortie et l'inversion par deux approches différentes du modèle de Volterra. L'approche d'inversion par régularisation de Tikhonov offre des résultats de reconstruction satisfaisants au détriment d'une charge de calcul importante. Cependant, le procédé étudié possède une dynamique globale lente par rapport au temps de calcul nécessaire. On peut donc envisager l'implémentation du procédé de reconstruction. En outre, l'approche d'inversion par retour d'entrée reconstruite ne repose pas sur une optimisation itérative et permet donc de limiter davantage la charge de calcul : une implémentation en temps réel est donc tout à fait concevable.

3.5 Conclusion du chapitre

Les deux procédés de traitement des eaux étudiés dans ce chapitre font appel à des principes différents : un processus chimique d’oxydation dans le premier cas et un processus biologique de digestion dans le deuxième cas. Cependant, l’objectif final reste le même, la neutralisation d’un composé organique potentiellement dangereux pour l’environnement. Ces deux procédés comportent d’autres points communs, mis en évidence lors de leur étude. Il s’agit avant tout de deux procédés continus : la mise en oeuvre de la neutralisation du composé organique est envisagée dans chacun des cas sur une longue durée, pouvant aller jusqu’à plusieurs mois. La modélisation de chacun des deux procédés tient compte de ce fait et les structures des deux modèles cinétiques obtenus sont donc très proches : on aboutit dans chacun des cas à un ensemble d’équations différentielles dans lequel des non-linéarités sont mises en évidence.

Les résultats fournis par le modèle de Volterra sont suffisamment proches de ceux obtenus en utilisant les modèles cinétiques des deux procédés pour justifier son utilisation et valider la méthode de modélisation proposée. Ceci montre s’il en est besoin qu’un modèle de comportement peut-être utilisé dans les domaines de la chimie et de la biochimie afin de prévoir le comportement d’un procédé continu, assimilable à un système mono-entrée mono-sortie.

La reconstruction de grandeurs chimiques non directement mesurables a, dans les deux cas d’étude, un objectif semblable (l’accès à une grandeur) et des motivations d’ordre économique : l’optimisation de la quantité de réactif dans le cas du procédé chimique d’oxydation et la possibilité d’éviter la mise en place onéreuse d’un protocole de mesure directe en entrée du réacteur dans le cas du procédé biologique de digestion. Les méthodes présentées permettent d’obtenir des grandeurs reconstruites tout à fait satisfaisantes. On a montré que la méthode de régularisation de Tikhonov, le plus souvent employée dans le cadre de systèmes linéaires, peut voir son utilisation élargie au domaine plus vaste des systèmes non linéaires si ceux-ci sont modélisables par une série de Volterra. Les deux méthodes d’inversion par retour d’entrée reconstruite permettent de s’affranchir d’un certain nombre de calculs et d’obtenir une rapidité satisfaisante si l’on désire ensuite utiliser ces méthodes sur une application en temps réel.

Conclusion générale

Le double objectif de ce travail de thèse s'inscrit dans la mise en place actuelle d'un pôle innovant de traitement des eaux, ceci dans le cadre du 12^{ème} Contrat de Plan État-Région de l'Université de Poitiers.

Le premier objectif du travail de recherche consistait en la modélisation de procédés chimiques et biologiques de traitement des eaux. Dans le domaine plus général du génie des procédés, de telles tentatives de modélisation sont depuis longtemps l'objet de recherches. La modélisation des deux procédés de traitement des eaux par les séries de Volterra présentée dans ce mémoire est dans chaque cas la première étape d'un projet global : le modèle du procédé d'oxydation de composé organique polluant doit ensuite permettre d'envisager la commande de ce procédé en agissant sur certaines grandeurs ; le modèle du procédé de digestion anaérobie permet, en faisant le lien entre différentes grandeurs, d'envisager la reconstruction d'une de ces grandeurs par inversion de modèle. Il s'agit là du deuxième objectif de cette thèse. Un des objectifs à terme du projet global de recherche dans lequel figure ce travail est la création d'un capteur intelligent capable de reconstruire une grandeur non directement mesurable. Ce type de capteur pourra être ensuite utilisé lors de la synthèse et de la mise en place d'un dispositif de commande du procédé de traitement des eaux.

Le premier chapitre répond à l'objectif de modélisation de systèmes non linéaires. Les différentes formes de modélisation et techniques présentées contribuent à augmenter le plus possible la parcimonie paramétrique du modèle de Volterra. C'est la principale contribution de ce chapitre. L'utilisation des séries de Volterra pour représenter un système non linéaire fournit un modèle souple et abordable car il généralise la notion de convolution, très utilisée pour la représentation de systèmes linéaires. Nous avons tenté de répondre à l'un des inconvénients du modèle de Volterra développé sur bases de fonctions généralisées : la forte augmentation du nombre de paramètres en l'absence de connaissance a priori sur le système étudié. La première solution proposée consiste à développer les noyaux de Volterra sur des fonctions de transfert : on obtient ainsi un modèle plus souple dont les paramètres sont tous identifiés à partir de signaux d'entrée et de sortie du système. Le problème de l'absence de connaissance a priori sur le système est éludé et la parcimonie globale du modèle de Volterra est augmentée. La deuxième solution proposée, que l'on peut coupler à la première, consiste à séparer la partie statique et la partie dynamique du signal d'entrée considéré. Le modèle de Volterra ne représente alors que la partie dynamique du système, ce qui permet de réduire le nombre total de noyaux et donc

Conclusion générale

de paramètres.

Le deuxième chapitre s'attache à l'objectif de reconstruction de grandeurs non directement mesurables. De nombreuses méthodes d'inversion de modèles linéaires et non linéaires existent, dont la méthode de régularisation de Tikhonov. Une des contributions de ce chapitre consiste en l'application de cette méthode de régularisation au modèle de Volterra défini dans le premier chapitre. Des difficultés similaires à celles du cas linéaire se posent (choix du paramètre et de la matrice de régularisation) et sont solutionnées de manière satisfaisante. Un problème supplémentaire se pose dans le cas du modèle de Volterra : l'augmentation du temps de calcul dû à la procédure d'optimisation non linéaire des paramètres. Une solution classique que nous proposons consiste en la discréétisation du modèle de Volterra afin de limiter le volume de calcul nécessaire à la simulation des sorties. Une solution moins classique consiste à coupler cette discréétisation à l'optimisation paramétrique et à la simulation de modèle sur un horizon fuyant, réduisant encore davantage le temps de calcul global. Cependant, l'optimisation et la simulation restent des étapes nécessaires de la procédure d'inversion par régularisation de Tikhonov. Nous avons alors proposé deux méthodes calculatoires d'inversion au principe différent de celui de la régularisation afin de contourner l'optimisation paramétrique. Ces deux méthodes d'inversion par retour d'entrée reconstruite sont équivalentes dans le cas d'un système linéaire mais donnent des résultats sensiblement différents dans le cas d'un système non linéaire. Dans les deux cas, le gain en temps de calcul est relativement important mais au prix de certaines difficultés numériques qui n'apparaissent pas dans le cas de la régularisation de Tikhonov.

Le troisième chapitre est consacré à l'étude de deux procédés de traitement des eaux aux objectifs semblables (la neutralisation de composés organiques) mais dont les techniques mises en oeuvre diffèrent (utilisation de réactifs chimiques dans un cas et de populations de bactéries dans l'autre). Nous nous sommes efforcés de montrer qu'un modèle de Volterra (modèle de comportement) est capable de représenter correctement le fonctionnement de ces deux procédés, dont deux modèles de connaissance avaient déjà été développés sous forme d'un ensemble d'équations différentielles. La validation du modèle de Volterra montre qu'un procédé chimique ou biologique peut être considéré comme un système comportant une ou plusieurs entrées et sorties sur lesquelles on peut agir afin de commander (et d'optimiser) son fonctionnement. Cependant, il n'est possible d'agir sur ces entrées et sorties qu'à la condition de pouvoir les mesurer à n'importe quel instant du processus. L'étude des deux procédés a montré que la mesure directe de certaines grandeurs était trop complexe et/ou trop coûteuse à mettre en oeuvre. Nous avons donc proposé une méthode indirecte de mesure de ces grandeurs par l'utilisation des méthodes d'inversion exposées dans le deuxième chapitre. Ces méthodes font appel à la mesure directe de grandeurs accessibles à l'aide de capteurs et à la reconstruction des grandeurs non directement mesurables par des moyens informatiques. Les temps de mesure et de calcul sont suffisamment faibles pour envisager d'implémenter ces méthodes de mesure indirecte en temps réel.

On peut envisager différents prolongements à ce travail de thèse.

Tout d'abord, l'étude effectuée dans le troisième chapitre pourrait aboutir à la mise au point d'un capteur logiciel dédié à la mesure de grandeurs physico-chimiques par une méthode indirecte comme nous l'avons vu précédemment. Un tel capteur présenterait l'avantage de pouvoir être implanté directement sur un pilote expérimental et cela à moindre coût. Une autre possibilité consisterait à ajouter à un capteur déjà existant une composante informatique afin de reconstruire une ou plusieurs grandeurs à partir des mesures de ce capteur et d'un modèle préétabli reliant les grandeurs entre elles. Une telle démarche réduirait encore davantage le coût de mise en place, celui-ci se limitant au matériel informatique et d'interfaçage nécessaire.

La suite logique de la mise en place d'un tel capteur pourra être la synthèse de lois de commandes visant à optimiser le fonctionnement des procédés de traitement des eaux, particulièrement dans le cas de l'oxydation de composés organiques par le procédé Fenton, pour lequel aucune méthode de dosage optimal de réactif en entrée du processus n'a été développée. Le dosage optimal conduirait également à une diminution du coût global du procédé, tout en assurant un taux de neutralisation de polluant conforme aux normes préétablies.

Nous nous sommes limités dans notre étude au cas de systèmes mono-entrée mono-sortie. Les modèles de Volterra n'ont donc été développés que dans ce cas. Or, les deux études effectuées au troisième chapitre montrent qu'il serait souhaitable d'un point de vue pratique de pouvoir agir sur plusieurs entrées des procédés. Dans le cas de l'oxydation par le procédé Fenton, une action simultanée sur le dosage des deux réactifs pourrait être envisagée. Elle implique donc la représentation du procédé par un modèle multi-entrées mono-sortie. Le modèle de Volterra est généralisable à plusieurs entrées il n'existe pas de moyen certain de quantification de l'influence de chacune des entrées sur la sortie, ceci à cause de nombreux termes de couplage présents dans le développement en série de Volterra. De plus, le nombre de paramètres augmente proportionnellement au nombre d'entrées du modèle, sacrifiant ainsi à l'objectif de parcimonie du modèle global. Un étude de pertinence des paramètres du modèle de Volterra multi-entrées mono-sortie serait donc nécessaire afin d'en réduire le nombre.

Enfin, un problème beaucoup plus délicat serait l'application de techniques d'inversion au modèle de Volterra multi-entrées mono-sortie. La reconstruction de plusieurs signaux d'entrée à partir d'un seul signal de sortie nécessiterait de connaître exactement l'influence de chacune des entrées du système. Cette information pourrait être déduite de connaissances a priori sur le système étudié. Les recherches dans ce domaine particulier ne se limitent pas au cadre du traitement des eaux mais à l'ensemble des problèmes inverses.

Conclusion générale

Annexe

1 Preuve du théorème 1.3

La première étape du calcul consiste à réécrire l'intégrale multiple $y_n(t)$ (1.99) en remplaçant les fonctions $f_a^p(t)$ par leur expression (1.100).

$$\int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} \left(\sum_{j=0}^{p_0-1} \frac{C_{p_0-1}^j}{j!} a_0^j (t - \tau_n)^j \right) e^{a_0(t-\tau_n)} \dots \\ \dots \left(\sum_{j=0}^{p_{n-1}-1} \frac{C_{p_{n-1}-1}^j}{j!} a_{n-1}^j (\tau_2 - \tau_1)^j \right) e^{a_{n-1}(\tau_2-\tau_1)} \left(\sum_{j=0}^{p_n-1} \frac{C_{p_n-1}^j}{j!} a_n^j \tau_1^j \right) e^{a_n \tau_1} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (1)$$

Afin de simplifier les calculs, nous allons poser :

$$\gamma_{i,j} = \gamma_{i,j}(a_i^j) = \frac{C_{p_i-1}^j}{j!} a_i^j \quad \text{pour } 0 \leq i \leq n \text{ et } 0 \leq j \leq p_i - 1 \quad (2)$$

L'équation (1) devient :

$$\int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} \left(\sum_{j=0}^{p_0-1} \gamma_{0,j}(t - \tau_n)^j \right) e^{a_0(t-\tau_n)} \dots \\ \dots \left(\sum_{j=0}^{p_{n-1}-1} \gamma_{n-1,j}(\tau_2 - \tau_1)^j \right) e^{a_{n-1}(\tau_2-\tau_1)} \left(\sum_{j=0}^{p_n-1} \gamma_{n,j} \tau_1^j \right) e^{a_n \tau_1} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (3)$$

Posons maintenant

$$D_i = \begin{cases} \sum_{j=0}^{p_0-1} \gamma_{0,j}(t - \tau_n)^j & \text{pour } i = 0 \\ \sum_{j=0}^{p_i-1} \gamma_{i,j}(\tau_{n-i+1} - \tau_{n-i})^j & \text{pour } 1 \leq i \leq n-1 \\ \sum_{j=0}^{p_n-1} \gamma_{n,j} \tau_1^j & \text{pour } i = n \end{cases} \quad (4)$$

On obtient donc comme nouvelle expression de $y_n(t)$:

$$\int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} D_0 D_1 \dots D_n e^{a_0(t-\tau_n)} \dots e^{a_{n-1}(\tau_2-\tau_1)} e^{a_n \tau_1} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (5)$$

que l'on peut également écrire :

$$\int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} D_0 D_1 \dots D_n e^{(a_0-a_1)(t-\tau_n) + \dots + (a_{n-1}-a_n)(t-\tau_1) + a_n t} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (6)$$

A ce stade, il faut faire la remarque que l'intégrale multiple (1.99) comporte $n+1$ fonctions (1.100) et seulement n termes $u(\tau_i)$. Si l'on parvient à décomposer l'intégrale multiple en un produit d'intégrales simples, nous aboutirons à un produit de $n+1$ intégrales, dont les n premières seront des intégrales de convolution simples alors que la dernière sera indépendante des variables τ_i et pourra s'écrire sous la forme $e^{a_n t}$. Dans le cas général, cette exponentielle peut être considérée comme un facteur de pondération ou un coefficient d'amortissement.

Afin de simplifier les calculs, nous allons poser $p_n = 0$. Il est important de noter que cette simplification n'aura pas d'importance dans la suite des calculs, la séparation de l'intégrale multiple ou l'identification paramétrique.

Nous obtenons bien dans l'intégrale multiple (6) un terme $e^{a_n t}$ indépendant des variables τ_i que l'on peut donc considérer comme un facteur constant et "sortir" de l'intégrale multiple :

$$e^{a_n t} \int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} D_0 D_1 \dots D_n e^{(a_0-a_1)(t-\tau_n) + \dots + (a_{n-1}-a_n)(t-\tau_1)} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (7)$$

Posons $p_n = 0$ pour obtenir :

$$\left(\sum_{j=0}^{p_n-1} \frac{C_{p_n-1}^j}{j!} a_n^j \tau_1^j \right) e^{a_n \tau_1} = 1 \quad (8)$$

et ainsi simplifier le calcul sans affecter la suite des manipulations mathématiques. L'intégrale multiple $y_n(t)$ s'écrira donc au final :

$$\int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} D_0 \dots D_{n-1} e^{(a_0-a_1)(t-\tau_n)} \dots e^{a_{n-1}(t-\tau_1)} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (9)$$

Le but étant de séparer l'intégrale multiple (9), il reste à déterminer si le produit

$$D_0 D_1 \dots D_{n-1} = \prod_{i=0}^{n-1} D_i \quad (10)$$

peut être séparé en un produit de termes qui ne dépendent que des différences $t - \tau_i$. La séparation de l'intégrale multiple nécessite donc la connaissance du produit (10) quelle que soit la valeur du coefficient p_i de chaque développement D_i défini par (4).

Les termes D_i s'écrivent de manière générale :

$$D_i = \begin{cases} \sum_{j=0}^{p_0-1} \gamma_{0,j} (t - \tau_n)^j & \text{pour } i = 0 \\ \sum_{j=0}^{p_i-1} \gamma_{i,j} (\tau_{n-i+1} - \tau_{n-i})^j & \text{pour } 1 \leq i \leq n-1 \end{cases} \quad (11)$$

Afin de simplifier les formules théoriques, on a posé

$$\gamma_{i,j} = \gamma_{i,j}(a_i^j) = \frac{C_{p_i-1}^j}{j!} a_i^j \text{ pour } 0 \leq i \leq n-1 \text{ et } 0 \leq j \leq p_i - 1 \quad (12)$$

En ajoutant et en retranchant la variable t , l'expression (11), pour $1 \leq i \leq n-1$, peut s'écrire

$$D_i = \sum_{j=0}^{p_i-1} \gamma_{ij} ((t - \tau_{n-i}) - (t - \tau_{n-i+1}))^j \quad (13)$$

Afin de développer cette expression, on considère la loi du binôme de Newton :

$$(a - b)^j = \sum_{k=0}^j (-1)^k C_j^k a^{j-k} b^k \quad (14)$$

On obtient ainsi

$$D_i = \sum_{j=0}^{p_i-1} \gamma_{ij} \sum_{k=0}^j (-1)^k C_j^k (t - \tau_{n-i})^{j-k} (t - \tau_{n-i+1})^k \quad (15)$$

On utilise également la simplification suivante pour alléger les notations :

$$\begin{cases} \eta_{ijk} = \gamma_{ij} (-1)^k C_j^k & \text{pour } i = 1, \dots, n-1 \\ \eta_{ij} = \gamma_{ij} & \text{pour } i = 0 \end{cases} \quad (16)$$

L'expression (15) devient donc

$$D_i = \sum_{j=0}^{p_i-1} \sum_{k=0}^j \eta_{ijk} (t - \tau_{n-i})^{j-k} (t - \tau_{n-i+1})^k \quad (17)$$

Le produit $\prod_{i=0}^{n-1} D_i$ (généralisé à l'ordre n et pour des valeurs quelconques mais entières des termes p_i) s'exprime alors de la manière suivante

$$\prod_{i=0}^{n-1} D_i = \left(\sum_{j=0}^{p_0-1} \eta_{0j} (t - \tau_n)^j \right) \left(\prod_{i=1}^{n-1} \sum_{j=0}^{p_i-1} \sum_{k=0}^j \eta_{ijk} (t - \tau_{n-i})^{j-k} (t - \tau_{n-i+1})^k \right) \quad (18)$$

Le produit $\prod_{i=0}^{n-1} D_i$ (avec p_i quelconque, $i = 0, \dots, n-1$) peut donc s'écrire sous la forme d'une somme de termes de la forme $\alpha_{ijk} (t - \tau_1)^{q_{n,1,r}} (t - \tau_2)^{q_{n,2,r}} \dots (t - \tau_n)^{q_{n,n,r}}$, où α_{ijk} est une constante qui ne dépend que des constantes η_{ijk} et où les termes $q_{n,i,r}$, $i = 1, \dots, n$ sont des entiers positifs ou nuls qui dépendent des valeurs des termes p_i , $i = 0, \dots, n-1$.

$$\prod_{i=0}^{n-1} D_i = \sum_{r=1}^{P_n} \alpha_{ijk} (t - \tau_1)^{q_{n,1,r}} (t - \tau_2)^{q_{n,2,r}} \dots (t - \tau_n)^{q_{n,n,r}} \quad (19)$$

On notera chacun des termes du produit de la manière suivante :

$$K_r(t, \tau_1, \dots, \tau_n) = \alpha_{ijk} \prod_{i=1}^n (t - \tau_i)^{q_{n,i,r}} \quad (20)$$

avec

$$\begin{cases} 0 \leq q_{n,1,r} \leq p_{n-1} - 1 \\ 0 \leq q_{n,i,r} \leq (p_{n-i} - 1)(p_{n-i+1} - 1) \text{ pour } i = 2, \dots, n \end{cases} \quad (21)$$

On a donc transformé un produit en somme :

$$\prod_{i=0}^{n-1} D_i = \sum_{r=1}^{P_n} K_r(t, \tau_1, \dots, \tau_n) \quad (22)$$

Le nombre P_n de termes de la somme peut s'exprimer de manière générale en fonction des termes $p_i, i = 1, \dots, n$:

$$P_n = p_0 \prod_{i=1}^{n-1} \frac{p_i(p_i + 1)}{2} \quad (23)$$

L'intégrale multiple $y_n(t)$ (1.99)

$$\int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} D_0 D_1 \dots D_{n-1} e^{(a_0-a_1)(t-\tau_n)} e^{(a_1-a_2)(t-\tau_{n-1})} \dots e^{a_{n-1}(t-\tau_1)} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (24)$$

se décompose alors en une somme de P_n intégrales multiples :

$$y_n(t) = \sum_{r=1}^{P_n} y_{n,r}(t) \quad (25)$$

que l'on peut expliciter davantage :

$$y_n(t) = \sum_{r=1}^{P_n} \int_0^t \int_0^{\tau_2} \dots \int_0^{\tau_n} \alpha_{ijk} \prod_{i=1}^n (t - \tau_i)^{q_{n,i,r}} e^{(a_0-a_1)(t-\tau_n) + (a_1-a_2)(t-\tau_{n-1}) + \dots + a_{n-1}(t-\tau_1)} \prod_{i=1}^n u(\tau_i) d\tau_i \quad (26)$$

Chacune de ces P_n intégrales multiples $y_{n,r}$ peut ensuite être exprimée sous la forme d'un produit de n intégrales simples :

$$y_{n,r}(t) = \int_0^t \alpha_{n,0}(t - \tau_n)^{q_{n,n,r}} e^{(a_0-a_1)(t-\tau_n)} u(\tau_n) d\tau_n \int_0^t \alpha_{n,1}(t - \tau_{n-1})^{q_{n,n-1,r}} e^{(a_1-a_2)(t-\tau_{n-1})} u(\tau_{n-1}) d\tau_{n-1} \dots \int_0^t \alpha_{n,n-1}(t - \tau_1)^{q_{n,1,r}} e^{a_{n-1}(t-\tau_1)} u(\tau_1) d\tau_1 \quad (27)$$

avec $\alpha_{ijk} = \alpha_{n,0}\alpha_{n,1}\dots\alpha_{n,n-1}$.

De manière générale, chacune des intégrales simples du produit précédent aura la forme suivante :

$$y_{n,r,i}(t) = \begin{cases} \int_0^t \alpha_{n,i}(t - \tau)^{q_{n,n-i,r}} e^{(a_i-a_{i+1})(t-\tau)} u(\tau) d\tau, & \text{si } 0 \leq i \leq n-2 \\ \int_0^t \alpha_{n,i}(t - \tau)^{q_{n,n-i,r}} e^{a_i(t-\tau)} u(\tau) d\tau, & \text{si } i = n-1 \end{cases} \quad (28)$$

ce que nous voulions montrer.

2 Cas particulier du théorème 1.3

On reprend le calcul pour un cas particulier : $n = 3$.

La première étape du calcul consiste à réécrire l'intégrale multiple $y_3(t)$ (1.99) en remplaçant les fonctions $f_a^p(t)$ par leur expression (1.100).

$$\int_0^t \int_0^{\tau_2} \int_0^{\tau_3} \left(\sum_{j=0}^{p_0-1} \frac{C_{p_0-1}^j}{j!} a_0^j (t - \tau_3)^j \right) e^{a_0(t-\tau_3)} \left(\sum_{j=0}^{p_1-1} \frac{C_{p_1-1}^j}{j!} a_1^j (\tau_3 - \tau_2)^j \right) e^{a_1(\tau_3-\tau_2)} \\ \left(\sum_{j=0}^{p_2-1} \frac{C_{p_2-1}^j}{j!} a_2^j (\tau_2 - \tau_1)^j \right) e^{a_2(\tau_2-\tau_1)} \left(\sum_{j=0}^{p_3-1} \frac{C_{p_3-1}^j}{j!} a_3^j \tau_1^j \right) e^{a_3\tau_1} \prod_{i=1}^3 u(\tau_i) d\tau_i \quad (29)$$

Afin de simplifier les calculs, on pose :

$$\gamma_{i,j} = \gamma_{i,j}(a_i^j) = \frac{C_{p_i-1}^j}{j!} a_i^j \quad \text{pour } 0 \leq i \leq 3 \text{ et } 0 \leq j \leq p_i - 1 \quad (30)$$

L'équation (29) devient :

$$\int_0^t \int_0^{\tau_2} \int_0^{\tau_3} \left(\sum_{j=0}^{p_0-1} \gamma_{0,j} (t - \tau_3)^j \right) e^{a_0(t-\tau_3)} \left(\sum_{j=0}^{p_1-1} \gamma_{1,j} (\tau_3 - \tau_2)^j \right) e^{a_1(\tau_3-\tau_2)} \\ \left(\sum_{j=0}^{p_2-1} \gamma_{2,j} (\tau_2 - \tau_1)^j \right) e^{a_2(\tau_2-\tau_1)} \left(\sum_{j=0}^{p_3-1} \gamma_{3,j} \tau_1^j \right) e^{a_3\tau_1} \prod_{i=1}^3 u(\tau_i) d\tau_i \quad (31)$$

Posons maintenant

$$D_i = \begin{cases} \sum_{j=0}^{p_0-1} \gamma_{0,j} (t - \tau_3)^j & \text{pour } i = 0 \\ \sum_{j=0}^{p_i-1} \gamma_{i,j} (\tau_{3-i+1} - \tau_{3-i})^j & \text{pour } 1 \leq i \leq 2 \\ \sum_{j=0}^{p_3-1} \gamma_{3,j} \tau_1^j & \text{pour } i = 3 \end{cases} \quad (32)$$

On obtient donc comme nouvelle expression de $y_3(t)$:

$$\int_0^t \int_0^{\tau_2} \int_0^{\tau_3} D_0 D_1 D_2 D_3 e^{a_0(t-\tau_3)} e^{a_1(\tau_3-\tau_2)} e^{a_2(\tau_2-\tau_1)} e^{a_3\tau_1} \prod_{i=1}^3 u(\tau_i) d\tau_i \quad (33)$$

que l'on peut également écrire :

$$\int_0^t \int_0^{\tau_2} \int_0^{\tau_3} D_0 D_1 D_2 D_3 e^{(a_0-a_1)(t-\tau_3)+(a_1-a_2)(t-\tau_2)+(a_2-a_3)(t-\tau_1)+a_3 t} \prod_{i=1}^3 u(\tau_i) d\tau_i \quad (34)$$

A ce stade, il faut faire la remarque que l'intégrale multiple (1.99) comporte 4 fonctions (1.100) et seulement 3 termes $u(\tau_i)$. Si l'on parvient à décomposer l'intégrale multiple en un produit d'intégrales simples, nous aboutirons à un produit de 4 intégrales, dont les 3 premières seront des intégrales de convolution simples alors que la dernière sera indépendante des variables τ_i et pourra s'écrire sous la forme $e^{a_n t}$. Dans le cas général, cette exponentielle peut être considérée comme un facteur de pondération ou un coefficient

d'amortissement.

Afin de simplifier les calculs, nous allons poser $p_3 = 0$. Il est important de noter que cette simplification n'aura pas d'importance dans la suite des calculs, la séparation de l'intégrale multiple ou l'identification paramétrique.

Nous obtenons bien dans l'intégrale multiple (34) un terme $e^{a_3 t}$ indépendant des variables τ_i que l'on peut donc considérer comme un facteur constant et "sortir" de l'intégrale multiple :

$$e^{a_3 t} \int_0^t \int_0^{\tau_2} \int_0^{\tau_3} D_0 D_1 D_2 D_3 e^{(a_0-a_1)(t-\tau_3)+(a_1-a_2)(t-\tau_2)+(a_2-a_3)(t-\tau_1)} \prod_{i=1}^3 u(\tau_i) d\tau_i \quad (35)$$

Posons $p_3 = 0$ pour obtenir :

$$\left(\sum_{j=0}^{p_3-1} \frac{C_{p_3-1}^j}{j!} a_3^j \tau_1^j \right) e^{a_3 \tau_1} = 1 \quad (36)$$

et ainsi simplifier le calcul sans affecter la suite des manipulations mathématiques. L'intégrale multiple $y_3(t)$ s'écrira donc au final :

$$\int_0^t \int_0^{\tau_2} \int_0^{\tau_3} D_0 D_1 D_2 e^{(a_0-a_1)(t-\tau_3)} e^{(a_1-a_2)(t-\tau_2)} e^{a_2(t-\tau_1)} \prod_{i=1}^3 u(\tau_i) d\tau_i \quad (37)$$

Le but étant de séparer l'intégrale multiple (37), il reste à déterminer si le produit

$$D_0 D_1 D_2 = \prod_{i=0}^2 D_i \quad (38)$$

peut être séparé en un produit de termes qui ne dépendent que des différences $t - \tau_i$. La séparation de l'intégrale multiple nécessite donc la connaissance du produit (38) quelle que soit la valeur du coefficient p_i de chaque développement D_i défini par (32).

Les termes D_i s'écrivent dans ce cas :

$$\begin{aligned} D_0 &= \sum_{j=0}^{p_0-1} \gamma_{0j} (t - \tau_3)^j \\ D_1 &= \sum_{j=0}^{p_1-1} \gamma_{1j} (\tau_3 - \tau_2)^j \\ D_2 &= \sum_{j=0}^{p_2-1} \gamma_{2j} (\tau_2 - \tau_1)^j \end{aligned} \quad (39)$$

Afin de simplifier les formules théoriques, on a posé

$$\gamma_{i,j} = \gamma_{i,j}(a_i^j) = \frac{C_{p_i-1}^j}{j!} a_i^j \quad \text{pour } 0 \leq i \leq 2 \text{ et } 0 \leq j \leq p_i - 1 \quad (40)$$

En ajoutant et en retranchant la variable t , les expressions (39), pour $1 \leq i \leq 2$, peut s'écrire

$$\begin{aligned} D_0 &= \sum_{j=0}^{p_0-1} \gamma_{0j} (t - \tau_3)^j \\ D_1 &= \sum_{j=0}^{p_1-1} \gamma_{1j} ((t - \tau_2) - (t - \tau_3))^j \\ D_2 &= \sum_{j=0}^{p_2-1} \gamma_{2j} ((t - \tau_1) - (t - \tau_2))^j \end{aligned} \quad (41)$$

Afin de développer cette expression, on considère la loi du binôme de Newton :

$$(a - b)^j = \sum_{k=0}^j (-1)^k C_j^k a^{j-k} b^k \quad (42)$$

On obtient ainsi

$$\begin{aligned} D_0 &= \sum_{j=0}^{p_0-1} \gamma_{0j} (t - \tau_3)^j \\ D_1 &= \sum_{j=0}^{p_1-1} \gamma_{1j} \sum_{k=0}^j (-1)^k C_j^k (t - \tau_2)^{j-k} (t - \tau_3)^k \\ D_2 &= \sum_{j=0}^{p_2-1} \gamma_{2j} \sum_{k=0}^j (-1)^k C_j^k (t - \tau_1)^{j-k} (t - \tau_2)^k \end{aligned} \quad (43)$$

On utilise également la simplification suivante pour alléger les notations :

$$\begin{cases} \eta_{ijk} = \gamma_{ij} (-1)^k C_j^k & \text{pour } i = 1, \dots, 2 \\ \eta_{ij} = \gamma_{ij} & \text{pour } i = 0 \end{cases} \quad (44)$$

On peut réécrire les expressions (43) sous la forme :

$$\begin{aligned} D_0 &= \sum_{j=0}^{p_0-1} \eta_{0j} (t - \tau_3)^j \\ D_1 &= \sum_{j=0}^{p_1-1} \sum_{k=0}^j \eta_{1jk} (t - \tau_2)^{j-k} (t - \tau_3)^k \\ D_2 &= \sum_{j=0}^{p_2-1} \sum_{k=0}^j \eta_{2jk} (t - \tau_1)^{j-k} (t - \tau_2)^k \end{aligned} \quad (45)$$

Dans le cas particulier où l'on pose $p_0 = p_1 = p_2 = 2$, l'expression (43) devient

$$\begin{aligned} D_0 &= \eta_{00} + \eta_{01} (t - \tau_3) \\ D_1 &= \eta_{100} + \eta_{110} (t - \tau_2) + \eta_{111} (t - \tau_3) \\ D_2 &= \eta_{200} + \eta_{210} (t - \tau_1) + \eta_{211} (t - \tau_2) \end{aligned} \quad (46)$$

et le produit $D_0 D_1 D_2$ s'exprime alors de la manière suivante

$$D_0 D_1 D_2 = (\eta_{00} + \eta_{01} (t - \tau_3)) (\eta_{100} + \eta_{110} (t - \tau_2) + \eta_{111} (t - \tau_3)) (\eta_{200} + \eta_{210} (t - \tau_1) + \eta_{211} (t - \tau_2)) \quad (47)$$

Une fois ce produit développé, on obtient une expression du type

$$\prod_{i=0}^2 D_i = \left(\sum_{j=0}^{p_0-1} \eta_{0j} (t - \tau_3)^j \right) \left(\prod_{i=1}^2 \sum_{j=0}^{p_i-1} \sum_{k=0}^j \eta_{ijk} (t - \tau_{3-i})^{j-k} (t - \tau_{3-i+1})^k \right) \quad (48)$$

Le produit $D_0 D_1 D_2$ peut donc s'écrire sous la forme d'une somme de 18 termes de la forme $\alpha_{ijk} (t - \tau_1)^{q_{3,1,r}} (t - \tau_2)^{q_{3,2,r}} (t - \tau_3)^{q_{3,3,r}}$, où α_{ijk} est une constante qui ne dépend que des constantes η_{ijk} et où les termes $q_{3,i,r}$, $i = 1, \dots, 3$ sont des entiers positifs ou nuls qui dépendent des valeurs des termes p_i , $i = 0, \dots, 2$.

L'intégrale multiple $y_3(t)$ peut alors être développée en une somme de P_3 intégrales multiples :

$$y_3(t) = \sum_{r=1}^{P_3} y_{3,r}(t) \quad (49)$$

que l'on peut expliciter davantage :

$$y_3(t) = \sum_{r=1}^{P_3} \int_0^t \int_0^{\tau_2} \int_0^{\tau_3} \alpha_{ijk} \prod_{i=1}^3 (t - \tau_i)^{q_{3,i,r}} e^{(a_0-a_1)(t-\tau_3)+(a_1-a_2)(t-\tau_2)+a_2(t-\tau_1)} \prod_{i=1}^3 u(\tau_i) d\tau_i \quad (50)$$

Le nombre P_3 d'intégrales multiples est défini en fonction des constantes p_0 , p_1 , p_2 (toutes trois égales à 2) de la manière suivante :

$$P_3 = p_0 \frac{p_1(p_1+1)}{2} \frac{p_2(p_2+1)}{2} \quad (51)$$

Chacune des $P_3 = 18$ intégrales multiples $y_{3,r}$ peut ensuite être exprimée sous la forme d'un produit de 3 intégrales simples :

$$y_{3,r}(t) = \int_0^t \alpha_{3,0}(t - \tau_3)^{q_{3,3,r}} e^{(a_0-a_1)(t-\tau_3)} u(\tau_3) d\tau_3 \int_0^t \alpha_{3,1}(t - \tau_2)^{q_{3,2,r}} e^{(a_1-a_2)(t-\tau_2)} u(\tau_2) d\tau_2 \\ \int_0^t \alpha_{3,2}(t - \tau_1)^{q_{3,1,r}} e^{a_2(t-\tau_1)} u(\tau_1) d\tau_1 \quad (52)$$

avec $\alpha_{ijk} = \alpha_{3,0}\alpha_{3,1}\alpha_{3,2}$.

On voit bien maintenant que chacune des intégrales simples du produit précédent aura la forme suivante :

$$y_{3,r,i}(t) = \begin{cases} \int_0^t \alpha_{3,i}(t - \tau)^{q_{3,3-i,r}} e^{(a_i-a_{i+1})(t-\tau)} u(\tau) d\tau, & \text{si } 0 \leq i \leq 1 \\ \int_0^t \alpha_{3,i}(t - \tau)^{q_{3,3-i,r}} e^{a_i(t-\tau)} u(\tau) d\tau, & \text{si } i = 2 \end{cases} \quad (53)$$

ce que nous voulions montrer pour le cas particulier où $n = 3$.

Bibliographie

- [Albert, 1972] A. Albert. *Regression and the Moore-Penrose Pseudoinverse*. Academic Press, 1972.
- [Akçay et Ninness, 1999] H. Akçay et B. Ninness. Orthonormal basis functions for modelling continuous-time systems. *Signal Processing*, 77(3) :261–274, 1999.
- [Audebrand *et al.*, 2002] A. Audebrand, H. Gallard, et J. DeLaat. Evolution du potentiel redox et de la concentration en H₂O₂ dans un réacteur d’oxydation par le réactif de Fenton. Technical report, Laboratoire de Chimie de l’Eau et de l’Environnement - Ecole Supérieure d’Ingénieurs de Poitiers, 2002.
- [Barb *et al.*, 1951] W.G. Barb, J.H. Baxendale, P. George, et K.R. Hargrave. Reactions of ferrous and ferric ions with hydrogen peroxide. *Transactions of Faraday Society*, 47 :462–500, 591–616, 1951.
- [Barbeni *et al.*, 1987] M. Barbeni, C. Minero, E. Pelizzetti, E. Borgarello, et N. Serpone. Chemical degradation of chlorophenols with Fenton’s reagent. *Chemosphere*, 16(10-12) :2225–2237, 1987.
- [Ben-Israel et Greville, 2003] A. Ben-Israel et T.N.E. Greville. *Generalized Inverses : Theory and Applications, 2nd edition*. Springer Verlag New York, 2003.
- [Bernard *et al.*, 2001] O. Bernard, Z. Hadj-Sadok, D. Dochain, A. Genovesi, et J-Ph. Steyer. Dynamical model development and parameter identification for an anaerobic wastewater treatment process. *Biotechnology and Bioengineering*, 75(4) :424–438, 2001.
- [Bibes *et al.*, 2003a] G. Bibes, P. Coirault, et R. Ouvrard. Atrazine oxidation modeling using Volterra series expanded on transfer functions. *American Control Conference*, 2003.
- [Bibes *et al.*, 2003b] G. Bibes, P. Coirault, R. Ouvrard, et J.P. Steyer. A methodology for nonlinear system identification using Volterra series. application to an anaerobic digestor. *Symposium on System Identification*, 2003.
- [Bibes *et al.*, 2003c] G. Bibes, R. Ouvrard, et P. Coirault. A regularization method for nonlinear inverse problems by using a Volterra model. *European Control Conference*, 2003.
- [Bibes *et al.*, 2004] G. Bibes, R. Ouvrard, P. Coirault, et J.C. Trigeassou. Méthode inverse non linéaire basée sur la régularisation et un modèle de Volterra. *Conférence Internationale Francophone d’Automatique*, 2004.

Bibliographie

- [Billings et Fakhouri, 1979] S.A. Billings et S.Y. Fakhouri. Nonlinear system identification using the Hammerstein model. *Int. Journal Systems Sci.*, 10 :567–578, 1979.
- [Boeglin, 1997] J.-C. Boeglin. Inventaire des traitements d'eaux résiduaires. Technical report, Techniques de l'ingénieur, 1997.
- [Boukhris, 1998] A. Boukhris *Identification de systèmes non linéaires par une approche multi-modèle*. Thèse de Doctorat, Institut National Polytechnique de Lorraine, 1998.
- [Boukhris *et al.*, 1999] A. Boukhris, G. Mourot, et J. Ragot. Non-linear dynamic system identification : A multi-model approach. *International Journal of Control*, 72(7/8) :591–604, 1999.
- [Chavent et Kunisch, 1994] G. Chavent et K. Kunisch. Convergence of Tikhonov regularization for constrained ill-posed problems. *Inverse Problems*, 10 :63–76, 1994.
- [Chen, 1989] S. Chen. Representation of nonlinear systems : The NARMAX model. *Int. J. Control*, 49(3) :1013–1032, 1989.
- [Corriou, 2001] J.P. Corriou. *Automatique et Procédés Chimiques*. Editions Hermès, 2001.
- [DeLaat et Gallard, 1999] J. DeLaat et H. Gallard. Catalytic decomposition of hydrogen peroxide by Fe(III) in homogeneous aqueous solutions : Mechanism and kinetic modeling. *Environ. Sci. Technol.*, 33(16) :2726–2732, 1999.
- [Demoment, 1987] G. Demoment. Déconvolution des signaux. Cours de l'Ecole Supérieure d'Electricité, 1987.
- [Demoment *et al.*] G. Demoment, J. Idier, J.-F. Giovannelli, et A. Mohammad-Djafari. Problèmes inverses en traitement du signal et de l'image. Techniques de l'ingénieur.
- [Doyle *et al.*, 2002] F.J. Doyle, R.K. Pearson, et B.A. Ogunnaike. *Identification and Control Using Volterra Models*. Springer, 2002.
- [ESIP, 2000] Ecole Supérieure d'Ingénieurs de Poitiers. *Journées Information Eaux*, 2000.
- [Flake, 1963] R.H. Flake. Volterra series representation of time-varying nonlinear systems. *IEEE Trans. Appl. Ind.*, 81 :330–335, septembre 1963.
- [Fliess *et al.*, 1983] M. Fliess, M. Lamnabhi, et F. Lamnabhi-Lagarrigue. Algebraic approach to nonlinear functional expansions. *IEEE CS*, 30 :550–570, 1983.
- [Gallard, 1998] H. Gallard. *Modélisation Cinétique de la Décomposition de H₂O₂ et de Composés Organiques En Milieu Aqueux Par Les Procédés D'oxydation H₂O₂/Fe(II) et H₂O₂/Fe(III)*. Thèse de Doctorat, Université de Poitiers, 1998.
- [Gallard et DeLaat, 2000] H. Gallard et J. DeLaat. Kinetic modelling of Fe(III)/H₂O₂ oxidation reactions in dilute aqueous solution : Using atrazine as a model organic compound. *Water Research*, 34 :3107–3116, 2000.
- [George, 1959] D.A. George. Continuous Non-Linear Systems. Tech. Rep. 355, Research Laboratory of Electronics, M.I.T., 1959.

-
- [Ghannam, 2000] F. Ghannam. *Reconstruction de Signal Par Convolution Inverse. Application À un Problème Thermique*. Thèse de Doctorat, Université de Poitiers, 2000.
- [Haber et Weiss, 1934] F. Haber et J. Weiss. The catalytic decomposition of hydrogen peroxide by iron salts. *Proceedings of the Royal Society A*, 134 :332–351, 1934.
- [Hadamard, 1923] J. Hadamard. *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. Yale University Press, 1923.
- [Hansen, 2001] P.C. Hansen. The l-curve and its use in the numerical treatment of inverse problems, 2001.
- [Hassouna, 2001] S. Hassouna. *Identification Des Systèmes Continus Non Linéaires Par Développement sur Les Séries de Volterra*. Thèse de Doctorat, Université de Poitiers, 2001.
- [Hassouna *et al.*, 2001] S. Hassouna, R. Ouvrard, P. Coirault et G. Bibes. Estimating of Volterra Kernels Expanded on Orthonormal Bases. *European Control Conference*, 2001.
- [Hérault et Jutten, 1994] J. Hérault et C. Jutten. *Réseaux neuronaux (et traitement du signal)*. Editions Hermès, 1994.
- [Inglada, 2000] J. Inglada. *Etude Des Signatures Radar de la Topographie Sous-Marine à la Surface de l'Océan*. Thèse de Doctorat, Université de Rennes I, 2000.
- [Jansson, 1984] P.A. Jansson. *Deconvolution with Application in Spectroscopy*. Academic Press, London, 1984.
- [Johansen, 1997] T.A. Johansen. On Tikhonov regularization, bias and variance in non-linear system identification. *Automatica*, 33 :441–446, 1997.
- [Lamnabhi-Lagarrigue, 1995] F. Lamnabhi-Lagarrigue. Volterra and Fliess series expansions for nonlinear systems. In IEEE Press, editor, *The Control Handbook*, pages 879–888. CRC Press, 1995.
- [Lee, 1932] Y.W. Lee. Synthesis of electric networks by means of the fourier transforms of laguerre functions. *J. Maths ans Phys. II*, pages 83–113, juin 1932.
- [Lindskog et Wahlberg, 1993] P. Lindskog et B. Wahlberg. Application of Kautz models in system identification. *Proceedings of 12th Triennial World Congress IFAC*, pages 41–44, 1993.
- [Ljung, 1987] L. Ljung. *System Identification : Theory for the User*. Prentice Hall, 1987.
- [Ljung *et al.*, 1992] L. Ljung, J. Sjöberg, et T. McKelvey. On the use of regularization in system identification. Technical report, Department of Electrical Engineering, Linköping University, Sweden, 1992.
- [Louis, 1992] A.K. Louis. Medical imaging : State of the art and future development. *Inverse Problems*, 8 :709–738, 1992.
- [Louis, 1997] A.K. Louis. Application of the approximate inverse to 3d x-ray CT and ultrasound tomography. *Inverse Problems in Medical Imaging and Nondestructive Testing*, pages 120–133, 1997.

Bibliographie

- [Marquardt, 1963] D.W. Marquardt. An algorithm for least squares estimation of non-linear parameters. *SIAM, Journal on Applied Mathematics*, 11(2) :431–441, 1963.
- [Mohammad-Djafari, 1998] A. Mohammad-Djafari. Bayesian inference for inverse problems. In *Proceedings of SPIE Volume 3459*, 1998.
- [Mohammad-Djafari, 1999a] A. Mohammad-Djafari. Entropie en traitement du signal. *Traitement du signal*, 15(6) :545–551, 1999.
- [Mohammad-Djafari, 1999b] A. Mohammad-Djafari. Problèmes inverses. Technical report, 1999.
- [Monsion, 1976] M. Monsion *Identification Par Analyse de Fourier Généralisée Des Systèmes Non-Linéaires Caractérisés Par Séries de Volterra. Application à un Système Physiologique*. Thèse d'Etat, Université de Bordeaux I
- [Murray-Smith et Johansen, 1997] R. Murray-Smith et T. A. Johansen. *Multiple Model Approaches to Modelling and Control*. Taylor and Francis, 1997.
- [Ninness et Gustafsson, 1994] B. Ninness et F. Gustafsson. An unifying construction of orthonormal bases for system identification. *Proceedings of the CDC 94*, pages 3388–3393, 1994.
- [Ralston et Zoubir, 1995] J.C. Ralston et A.M. Zoubir. Identification of time-varying hammerstein models. *Proceedings of ICASSP'95*, pages 1685–1688, 1995.
- [Rolland, 1995] R. Rolland. *Le Développement de l'Analyse Fonctionnelle Au Début Du XXe Siècle*. Institut de Mathématiques de Luminy, 1995.
- [Rugh, 1981] W.J. Rugh. *Nonlinear System Theory - The Volterra/Wiener Approach*. The Johns Hopkins University Press, 1981.
- [Scales et Smith, 1996] J.A. Scales et M.L. Smith. *Introductory Geophysical Inverse Theory*. Samizdat Press, 1996.
- [Schetzen, 1980] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*. John Wiley and Sons, 1980.
- [Sedlak et Andren, 1991] D.L. Sedlak et A.W. Andren. Oxidation of chlorobenzene with Fenton's reagent. *Environ. Sci. Technol.*, 25(4) :777–782, 1991.
- [Sjöberg *et al.*, 1995] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P.Y. Glorennec, H. Hjalmarsson, et A. Juditsky. Nonlinear black-box modeling in system identification : An unified overview. *Automatica*, 31(3) :1691–1724, mars 1995.
- [Steyer *et al.*, 2003] J.P. Steyer, L. Lardon, J.C. Bouvier, T. Conte, P. Neveu, L. Menut, B. Simon, et C. Hacquart. Instrumentation et gestion de l'information en traitement des eaux. AutoAgro 2003 Colloque Automatique et Agronomie Montpellier, 22-24 janvier 2003.
- [Sullivan, 1990] *Ill-conditionned signal restoration and the conjugate gradient method*, 1990.
- [Tikhonov et Arsénine, 1976] A. Tikhonov et V. Arsénine. *Méthodes de Résolution de Problèmes Mal Posés*. Editions Mir - Moscou, 1976.

-
- [Tricomi, 1985] F.G. Tricomi. *Integral Equations*. Dover Publications Inc., 1985.
- [Volterra, 1959] V. Volterra. *Theory of Functionnals and of Integral and Integro-Differential Equations*. Dover Publications, 1959.
- [Wahlberg, 1991] B. Wahlberg. System identification using Laguerre models. *IEEE Trans. Automat. Contr.*, 36(5) :551–562, mai 1991.
- [Wahlberg, 1994] B. Wahlberg. System identification using Kautz models. *IEEE Trans. Automat. Contr.*, 39(6) :1276–1282, 1994.
- [Walling, 1975] C. Walling. Fenton’s reagent revisited. *Accounts of chemical research*, 8 :125–131, 1975.
- [Wiener, 1943] M. Wiener. Nonlinear problems in random theory. *MIT Press*, 1943.

Bibliographie

Résumé

Ce mémoire de thèse est consacré à la modélisation de systèmes non linéaires par les séries de Volterra ainsi qu'à la reconstruction de grandeurs non directement mesurables par inversion du modèle de Volterra. Le premier chapitre aborde la modélisation parcimonieuse de systèmes non linéaires. Les noyaux de la série de Volterra sont d'abord développés sur une base de fonctions généralisée. On se propose de limiter le nombre total de paramètres en développant les noyaux de Volterra sur des fonctions de transfert d'une part, puis en séparant la partie statique de la partie dynamique du modèle de Volterra. Le deuxième chapitre traite de l'inversion du modèle de Volterra. On utilise d'abord la méthode de régularisation de Tikhonov puis deux méthodes d'inversion par retour d'entrée reconstruite. Afin de limiter le volume de calcul nécessaire à la régularisation de Tikhonov, nous proposons de discréteriser le modèle de Volterra et d'effectuer les procédures d'optimisation et de simulation sur un horizon fuyant. Dans le troisième chapitre, les méthodes de modélisation et d'inversion sont appliquées à un procédé chimique, l'oxydation d'un composé organique par le réactif de Fenton, puis à un procédé biologique, la digestion anaérobie d'un composé carboné.

Mots-clés: système non linéaire, modélisation, identification, série de Volterra, problème inverse, régularisation de Tikhonov, oxydation composé organique, réactif de Fenton, digestion anaérobie.

Abstract

This PhD thesis is dedicated to nonlinear system modelling using Volterra series and to Volterra model inversion in order to restore quantities which are not directly measurable. First chapter is about parsimonious modelling for nonlinear systems. Volterra series kernels are first expanded on a generalized function basis. In order to limit the total number of parameters we propose to expand Volterra kernels on transfer functions on the one hand, and to separate Volterra model static part from its dynamic part on the other hand. Second chapter deals with Volterra model inversion. First, we use Tikhonov's regularization method, then we propose two inversion methods using restored input feedback. In order to reduce the necessary computational burden for Tikhonov regularization, we suggest a Volterra model discretization combined with optimization and simulation procedures performed on a sliding window. In third chapter, modelling and inversion methods are put into practice with a chemical process, organic compound oxidation with Fenton reagent, and a biological process, anaerobic digestion of a carbonaceous compound.

Keywords: nonlinear system, modelling, identification, Volterra series, inverse problem, Tikhonov regularization, organic compound oxidation, Fenton's reagent, anaerobic digestion.

