



HAL
open science

Modélisation et identification de systèmes non-linéaires à l'aide de modèles de volterra à complexité réduite

Anis Khouaja

► **To cite this version:**

Anis Khouaja. Modélisation et identification de systèmes non-linéaires à l'aide de modèles de volterra à complexité réduite. Automatique / Robotique. Université Nice Sophia Antipolis, 2005. Français. NNT: . tel-00090557

HAL Id: tel-00090557

<https://theses.hal.science/tel-00090557>

Submitted on 31 Aug 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE NICE SOPHIA ANTIPOLIS - UFR SCIENCES
ÉCOLE DOCTORALE DE SCIENCES & TECHNOLOGIES DE L'INFORMATION ET
DE LA COMMUNICATION

THÈSE

POUR L'OBTENTION DU TITRE DE
DOCTEUR EN SCIENCES DE L'UNIVERSITÉ DE NICE SOPHIA
ANTIPOLIS

SPÉCIALITÉ : AUTOMATIQUE ET TRAITEMENT DU SIGNAL ET DE L'IMAGE

PRÉSENTÉE ET SOUTENUE PAR

Anis KHOUAJA

MODÉLISATION ET IDENTIFICATION DE SYSTÈMES
NON-LINÉAIRES À L'AIDE DE MODÈLES DE VOLTERRA À
COMPLEXITÉ RÉDUITE

THÈSE DIRIGÉE PAR

Gérard FAVIER et Hassani MESSAOUD

soutenue le : 1 mars 2005

Devant la commission d'examen composée de :

M. Hervé RIX	Professeur, I3S	Président
M. José AGUILAR-MARTIN	Directeur de recherche, LAAS	Rapporteur
M. Mekki KSOURI	Professeur, ENIT	Rapporteur
M. Gérard FAVIER	Directeur de recherche, I3S	Directeur de thèse
M. Hassani MESSAOUD	Maître de Conférences, ENIM	Directeur de thèse

Résumé

L'identification des systèmes dynamiques non linéaires à partir d'un ensemble de données entrée/sortie est d'une importance fondamentale pour les applications pratiques puisque beaucoup de systèmes physiques possèdent des caractéristiques non linéaires. La structure du modèle de Volterra peut être utilisée pour représenter une classe générale de systèmes non linéaires. Cependant, l'usage pratique d'une telle représentation est souvent limité à cause du grand nombre de paramètres associé à une telle structure.

Pour pallier à cet inconvénient, plusieurs solutions sont proposées dans cette thèse. La première utilise des développements en série des différents noyaux sur des bases de fonctions orthogonales. La deuxième est basée sur l'utilisation de techniques faisant appel à des décompositions d'ordre réduit des tenseurs relatifs aux noyaux d'ordre supérieur ou égal à trois.

Diverses bases de fonctions (Laguerre, Kautz et Bases Orthogonales Généralisées (BOG)) sont tout d'abord étudiées en vue de leur utilisation pour la modélisation des systèmes linéaires puis pour la représentation des noyaux de modèle de Volterra. Le problème d'identification comporte plusieurs volets : détermination des pôles caractéristiques des bases de fonctions orthogonales, de l'ordre des développements des différents noyaux, des coefficients de Fourier du développement et de l'incertitude relative à ces coefficients. Une représentation d'état associée à un développement sur une base de fonctions orthogonales généralisées est développée puis utilisée pour la construction de prédicteurs de la sortie du système ainsi modélisé.

Ensuite, plusieurs décompositions tensorielles sont étudiées. La décomposition PARAFAC est plus particulièrement considérée. Des modèles de Volterra à complexité réduite inspirés de cette technique sont proposés. En considérant le noyau quadratique

de Volterra comme une matrice et les autres noyaux comme des tenseurs d'ordres supérieurs à deux, nous utilisons une décomposition à l'aide des valeurs singulières (SVD) pour le noyau quadratique et la décomposition PARAFAC pour les noyaux d'ordres supérieurs à deux afin de construire le modèle réduit de Volterra appelé SVD-PARAFAC-Volterra. Un nouvel algorithme appelé ARLS (Alternating Recursive Least Squares) est présenté. Cet algorithme essentiellement basé sur la technique RLS appliquée d'une manière alternée estime les paramètres de tels modèles de Volterra.

Enfin, de nouvelles méthodes d'identification robuste dites à erreur bornée sont présentées. Elles sont utilisées pour l'identification de modèles linéaires issus des BOG, travail qui vise à étendre au cas des systèmes non linéaires incertains des résultats obtenus récemment pour des systèmes linéaires incertains. Parmi les techniques d'identification à erreur bornée présentées, l'approche polytopique est plus particulièrement considérée. Cette approche nous permet d'estimer les intervalles d'incertitude des coefficients de Fourier du développement sur les différentes bases orthogonales étudiées. Ces mêmes méthodes d'identification sont utilisées aussi afin d'identifier les intervalles d'incertitude des paramètres du modèle SVD-PARAFAC-Volterra. Les méthodes proposées permettent de réaliser une importante réduction de complexité numérique et un gain en temps de calcul considérables.

Summary

The identification of the non-linear dynamic systems from a set of input/output data is of a fundamental importance for the practical applications since a lot of physical systems possess non linear characteristics. The structure of the Volterra model can be used to represent a general class of non linear systems. However, the use of such a representation is often limited because of the huge number of parameters relative to such a structure.

To overcome this inconvenience, several solutions are proposed in this thesis. The first uses expansions of the different kernels on orthogonal bases functions. The second is based on the use of techniques calling on reduced-order decompositions of the tensors associated to the kernels of order superior or equal to three.

Various bases of functions (Laguerre, Kautz and Generalized Orthogonal Bases (BOG)) are first studied then to use them for the modelling the linear systems then for the representation of the kernels of the Volterra models. The problem of identification includes several parts : determination of the poles of the orthogonal bases functions, the order of the kernel developments, the Fourier coefficients of the development and the relative uncertainty to these coefficients. A state representation associated to a development on a Generalized Orthogonal Basis is developed and then used for the construction of the output predictor of the system to be modelled.

Then, several tensorial decompositions are studied. The PARAFAC decomposition is specially considered. Reduced complexity Volterra models inspired from this technique are proposed. While considering the quadratic Volterra kernel as a matrix and the other kernels as tensors of orders superior to two, we use a singular value decomposition for the quadratic kernel and the PARAFAC decomposition for the kernels of orders superior to two in order to construct a new model called SVD-PARAFAC based

Volterra model. A new algorithm called ARLS (Alternating Recursive Least Squares) is presented. This algorithm essentially based on the technical RLS applied in an alternate manner estimates the parameters of such Volterra models.

Finally, new methods of robust identification called bounded error techniques are presented. They are used for the identification of linear models based on an expansion on a GOB, this work aims to use the results found lately for uncertain linear systems in the case of the uncertain non linear systems. One of the techniques of identification, the polytopic approach is especially considered. This approach allows to estimate the uncertainty intervals of the Fourier coefficients of the expansion on the different GOBs studied. The polytopic approach is also used in order to identify the uncertainty intervals of the parameters of the SVD-PARAFAC based Volterra model. The proposed methods allows to achieve an important numeric complexity reduction and a considerable gain in time calculation.

À ma famille

et à tous mes amis.

Remerciements

J'adresse mes sincères remerciements à :

M. Gérard FAVIER pour sa constante disponibilité, l'encadrement scientifique dont il m'a fait bénéficier et pour les nombreuses discussions enrichissantes que j'ai pu avoir avec lui au cours de ma thèse.

M. Hassani MESSAOUD pour son encadrement scientifique, ses encouragements et ses remarques qui m'ont été fort précieuses. Je lui suis également reconnaissant pour m'avoir représenté auprès des autorités administratives de l'Université du Sud en Tunisie pendant mon séjour en France.

M. Hervé RIX d'avoir accepté de présider le jury de ma thèse, ainsi que M. José MARTIN-AGUILAR et M. Mekki KSOURI d'avoir accepté d'être rapporteurs de ce travail et pour leurs remarques constructives.

Tous les membres du projet ASTRE et plus particulièrement Mohamed BOULOIRD, André DE ALMEIDA, Estevaõ FERNANDES, Alain KIBANGOU, Ludwig ROTA et Vicente ZARZOSO pour leur aide précieuse et pour la bonne ambiance qu'ils sont parvenu à instaurer au sein du groupe.

Un grand Merci aux collègues ainsi qu'à tout le personnel du laboratoire I3S que j'ai eu le plaisir de côtoyer au cours de ces quatre années. Merci aussi aux collègues de l'unité de recherche ATSI de Monastir.

Mes remerciements vont enfin à ma famille et tous mes amis pour le soutien moral qu'ils m'ont témoigné.

Table des matières

Résumé	iii
Summary	v
<i>Remerciements</i>	ix
Table des matières	xi
Liste des figures	xv
Liste des tableaux	xx
Abréviations	xxii
1 Introduction Générale	1
1.1 Présentation générale	1
1.2 Démarche et objectifs scientifiques	4
1.3 Contributions	6
2 Représentation des systèmes à l'aide de la BOG	9
2.1 Introduction	9
2.2 Synthèse des fonctions de la BOG	13
2.2.1 Base de filtres FIR	16
2.2.2 Base de filtres de Laguerre	17
2.2.3 Base de filtres de Kautz	17
2.3 Réseau de filtres de la Base Orthogonale Généralisée	19
2.3.1 Réseau de filtres de la Base des fonctions de Laguerre	19

2.3.2	Réseau de filtres de la Base des fonctions de Kautz	20
2.3.3	Réseau de filtres de la Base Orthogonale Généralisée	20
2.4	Détermination des pôles de la BOG	20
2.4.1	Choix du pôle de Laguerre	21
2.4.2	Choix des pôles conjugués de Kautz	24
2.4.3	Choix des pôles de la BOG	25
2.4.4	Convergence des algorithmes d'estimation des pôles de la BOG	29
2.4.5	Résultats de simulations	32
2.5	Introduction des systèmes de Volterra	38
2.5.1	Généralités	39
2.5.2	Développement des noyaux de Volterra à l'aide des BOG	40
2.6	Représentation dans l'espace d'état	43
2.6.1	Modèle de Laguerre sous forme d'équation d'état	43
2.6.2	Modèle de Kautz sous forme d'équation d'état	44
2.6.3	Modèle de la BOG à pôles réels sous forme d'équation d'état	47
2.6.4	Modèle de la BOG à pôles réels et complexes sous forme d'équation d'état	50
2.7	Calcul de prédicteurs	50
2.7.1	Modèle et prédicteur	50
2.7.2	Prédicteur à i -pas de la sortie d'un modèle basé sur une décomposition orthogonale	51
3	Techniques de décompositions tensorielles	55
3.1	Introduction	55
3.2	Les modèles de décomposition tensorielle	56
3.2.1	Le modèle PARAFAC	57
3.2.2	Le modèle PARAFAC2	63
3.2.3	Les modèles TUCKER	65
3.2.4	Le modèle PARATUCK2	67
3.3	Les algorithmes ALS (Alternating Least Squares)	70
3.3.1	L'algorithme ALS-PARAFAC	71
3.4	Représentation des modèles de Volterra réduits	76

3.4.1	L'algorithme ALS-PARAFAC	76
3.4.2	Résultats de simulations	76
4	Identification des Systèmes de Volterra	83
4.1	Introduction	83
4.2	Le modèle SVD-PARAFAC-Volterra	84
4.2.1	La décomposition SVD (Singular Value Decomposition)	84
4.2.2	La décomposition PARAFAC généralisée	85
4.2.3	Le modèle SVD-PARAFAC-Volterra	85
4.3	Identification des modèles SVD-PARAFAC-Volterra	89
4.3.1	Première écriture de la sortie du modèle	89
4.3.2	$k^{\text{ème}}$ écriture de la sortie du modèle	90
4.3.3	L'algorithme ARLS	90
4.3.4	Identification des modèles SVD-PARAFAC-Volterra d'ordre 3 à l'aide de l'algorithme ARLS	93
4.4	Résultats de simulations	97
4.4.1	Premier système simulé	97
4.4.2	Second système simulé	101
4.5	Conclusion	103
5	Identification robuste des systèmes	105
5.1	Introduction	105
5.2	Méthodes d'identification à erreur bornée	106
5.2.1	Approche UBBE (Unknown But Bounded Error)	106
5.2.2	L'approche polytopique	108
5.2.3	L'approche orthotopique	118
5.2.4	L'approche polytopique à complexité limitée	123
5.3	Identification robuste des systèmes définis à l'aide des BOG	128
5.4	Résultats de simulations	130
5.4.1	Présentation de l'outil GBT	130
5.4.2	Critères d'évaluation	130
5.4.3	Modélisation et identification d'un système linéaire	131
5.4.4	Analyse des résultats	139

5.5	Identification robuste des systèmes de Volterra	141
5.5.1	Système 1	142
5.5.2	Système 2	144
5.5.3	Système 3	149
5.6	Identification robuste de modèle SVD-PARAFAC-Volterra	151
5.6.1	Choix de la borne d'erreur	153
5.6.2	Résultats de simulations	154
5.7	Conclusions	156
6	Conclusions et Perspectives	161
	Annexes	167
A	Représentation d'état d'un modèle de Kautz	167
B	Passage d'une F.T. à une équation d'état	173
C	Représentation d'état d'un système à l'aide des BOG	175
C.1	Réel - Réel	175
C.2	Réel - Complexe	176
C.3	Complexe - Réel	177
C.4	Complexe - Complexe	178
D	Deuxième représentation du modèle PARAFAC2	180
E	Trois formes de l'algorithme ALS-PARAFAC	183
E.1	Définition :	183
E.2	ALS-PARAFAC 1	183
E.3	ALS-PARAFAC 2	185
E.4	ALS-PARAFAC 3	187

Table des figures

1.1	Procédure d'identification expérimentale	3
2.1	Réseau de filtres de la Base des fonctions de Laguerre	19
2.2	Réseau de filtres de la Base des fonctions de Kautz	20
2.3	Réseau de filtres de la BOG	21
2.4	Détermination du pôle optimal de Laguerre	24
2.5	Détermination de b_{opt} et c_{opt} minimisant l'erreur quadratique	26
2.6	Détermination des dynamiques du système par la première méthode	27
2.7	Détermination des dynamiques du système avec la deuxième méthode	28
2.8	Détermination des dynamiques du système dans le cas général	29
2.9	Réseau de filtres de la BOG	30
2.10	Vraies dynamiques du système simulé et les pôles de la BOG estimés par les méthodes 1 et 2	34
2.11	Coefficients de Fourier $\{g_i\}_{i=0,\dots,5}$ estimés avec la méthode 1 et la méthode 2	35
2.12	EQMN entre la sortie du système 1 et celles des modèles RIF (N=50), Laguerre (N=10) et BOG (N=5)	36
2.13	Réponse impulsionnelle du système 2	37
2.14	EQMN entre la sortie du système 2 et celles des modèles RIF (N=50), Kautz (N=10) et BOG (N=7)	38
2.15	Réseau de filtres de la BOG	47
3.1	Représentation graphique d'un tenseur d'ordre 3	56
3.2	(a) Définition d'une ligne, d'une colonne et d'un tube. (b) $n_3^{ème}$ couche frontale	56
3.3	Représentation tensorielle du modèle trilineaire PARAFAC	57
3.4	Construction des matrices $H_{n_1..}$, $H_{..n_2}$, et $H_{..n_3}$	58
3.5	Construction des matrices H_{N_1} (a), H_{N_2} (b) et H_{N_3} (c)	60

3.6	Type de tenseurs modélisés à l'aide de PARAFAC2	63
3.7	Représentation graphique du modèle PARATUCK2	68
3.8	La matrice $H_{N_1 \times 5N_2}$ issue du noyau de Volterra initial	77
3.9	$EQMN_{Noyau}$ entre le noyau de Volterra non-séparable initial et le noyau reconstruit en utilisant PARAFAC	78
3.10	$EQMN_{Noyau}$ entre le noyau de Volterra original et le noyau reconstruit en utilisant le développement sur des bases de fonctions de Laguerre	79
3.11	La matrice $H_{N_1 \times 5N_2}$ issue du noyau de Volterra initial	80
3.12	$EQMN_{Noyau}$ entre le noyau de Volterra séparable initial et le noyau reconstruit en utilisant PARAFAC	81
4.1	Structure parallèle-cascade du modèle SVD-PARAFAC-Volterra d'ordre K . .	87
4.2	Réalisation d'une structure parallèle-cascade du modèle SVD-PARAFAC-Volterra d'ordre 3	88
4.3	Mise à jour des composants du modèle SVD-PARAFAC-Volterra d'ordre K à l'aide de l'algorithme ARLS	91
4.4	Mise à jour des composants du modèle SVD-PARAFAC-Volterra d'ordre 3 à l'aide de l'algorithme ARLS	97
4.5	Les noyaux linéaire, quadratique et la matrice latérale $H_{.,1}$ du noyau cubique du modèle de Volterra d'ordre 3	98
4.6	$EQMN_{sortie}$ pour trois RSBs différents	99
4.7	Les composantes estimées des noyaux linéaire et quadratique	100
4.8	Les composantes estimées du modèle PARAFAC	100
4.9	Application du modèle SVD-PARAFAC-Volterra d'ordre 3 pour l'identification d'un canal satellitaire non linéaire	101
4.10	La caractéristique entrée-sortie du dispositif non -linéaire	102
4.11	Les réponses impulsionnelles des filtres H_B et H_C	103
4.12	$EQMN_{sortie}$ pour trois RSBs différents	104
5.1	Les deux hyperplans H_{k_1} et H_{k_2}	107
5.2	Acquisition d'une nouvelle mesure : Cas où $\beta_j(i) < 0$; $i = 1, \dots, n_{k-1}$.	114
5.3	Acquisition d'une nouvelle mesure : Cas où $\beta_j(i) > 0$; $i = 1, \dots, n_{k-1}$.	114
5.4	Acquisition d'une nouvelle mesure : Cas où les $\beta_j(i)$ sont de signes différents ; $i = 1, \dots, n_{k-1}$	115

5.5	Le polytope P_k	116
5.6	Mise à jour de l'orthotope ϑ_{k-1}	119
5.7	Détermination du polytope intersection : $\vartheta_{k-1} \cap H_{k_j}^+ = \emptyset$	121
5.8	Détermination du polytope intersection : $\vartheta_{k-1} \cap H_{k_j}^+ = \vartheta_{k-1}$	121
5.9	Détermination du polytope intersection : $\vartheta_{k-1} \cap H_{k_j}^+ = P_k \neq \vartheta_{k-1}$	122
5.10	L'orthotope final ϑ_k	123
5.11	Mise à jour du polytope à complexité limitée selon Broman et Shensa	124
5.12	Mise à jour du polytope à complexité limitée selon Lahanier et Walter	125
5.13	Augmentation du volume du polytope à complexité limitée lors de sa mise à jour par la méthode de Broman et Shensa	126
5.14	Inconvénient de la méthode de Lahanier et Walter	126
5.15	Mise à jour du polytope à complexité limitée ($N_f = 4$)	129
5.16	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope ($RSB = 10dB$) (Modèle de Laguerre)	134
5.17	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope à complexité limitée ($N_f = 12$ et $RSB = 10dB$) (Modèle de Laguerre)	134
5.18	Volume du Polytope ($RSB = 10dB$) (Modèle de Laguerre)	134
5.19	Volume du polytope à complexité limitée ($N_f = 12$ et $RSB = 10dB$) (Modèle de Laguerre)	134
5.20	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope à complexité limitée ($N_f = 10$ et $RSB = 10dB$) (Modèle de Laguerre)	135
5.21	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets de l'orthotope ($RSB = 10dB$) (Modèle de Laguerre)	135
5.22	Volume du polytope à complexité limitée ($N_f = 10$ et $RSB = 10dB$) (Modèle de Laguerre)	135
5.23	Volume de l'orthotope ($RSB = 10dB$) (Modèle de Laguerre)	135
5.24	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope ($RSB = 10dB$) (modèle issu de la BOG)	137

5.25	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope à complexité limitée ($N_f = 12$ et $RSB = 10dB$) (Modèle issu de la BOG)	137
5.26	Volume du Polytope ($RSB = 10dB$) (Modèle issu de la BOG)	137
5.27	Volume du polytope à complexité limitée ($N_f = 12$ et $RSB = 10dB$) (Modèle issu de la BOG)	137
5.28	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope à complexité limitée ($N_f = 10$ et $RSB = 10dB$) (Modèle issu de la BOG)	138
5.29	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets de l'orthotope ($RSB = 10dB$) (Modèle issu de la BOG)	138
5.30	Évolution du volume du Polytope à complexité limitée ($RSB = 10dB$ et $N_f = 10$) (Modèle issu de la BOG)	138
5.31	Évolution du volume de l'orthotope ($RSB = 10dB$) (Modèle issu de la BOG) .	138
5.32	Cas des systèmes de Volterra d'ordre 3	142
5.33	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope ($RSB = 10dB$)	143
5.34	Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope ($RSB = 3dB$)	143
5.35	Paramètres du système comparés à leurs estimées par deux méthodes différentes ($RSB = 10dB$)	143
5.36	Paramètres du système comparés à leurs estimées par deux méthodes différentes ($RSB = 3dB$)	143
5.37	Sortie du système simulé et celles des deux modèles estimés par deux méthodes différentes ($RSB = 10dB$)	147
5.38	Sortie du système simulé et celles des deux modèles estimés par deux méthodes différentes ($RSB = 3dB$)	147
5.39	$EQMN_{Sortie}$ ($RSB = 10dB$)	147
5.40	$EQMN_{Sortie}$ ($RSB = 3dB$)	147
5.41	Coordonnées du centre du polytope et bornes des intervalles d'incertitude ($RSB = 10dB$)	148

5.42	Coordonnées du centre du polytope et bornes des intervalles d'incertitude ($RSB = 3dB$)	148
5.43	Paramètres du système comparés à leurs estimées par deux méthodes différentes ($RSB = 10dB$)	148
5.44	Paramètres du système comparés à leurs estimées par deux méthodes différentes ($RSB = 3dB$)	148
5.45	Système 3	149
5.46	Le Modèle BOG-Volterra	150
5.47	Mise à jour des polytopes P_1 , P_2 et P_3	152
A.1	Réseau de filtres de la Base des fonctions de Kautz	168
C.1	ξ_n réel et ξ_{n+1} réel	175
C.2	ξ_n réel, ξ_{n+1} et ξ_{n+2} complexes	176
C.3	ξ_n complexe et ξ_{n+1} réel	178
C.4	ξ_n complexe et ξ_{n+1} complexe	178

Liste des tableaux

2.1	Pôles de la BOG estimés par les méthodes 1 et 2	33
2.2	$EQ(L)$ obtenue avec les différentes BOG	33
2.3	Temps de calcul d'un pôle de la BOG	34
3.1	Complexité, TRC et $EQMN_{Noyau}$ en fonction du nombre P de facteurs du modèle PARAFAC	78
4.1	L'algorithme ARLS (Cas général)	92
4.2	L'algorithme ARLS (Cas particulier)	96
4.3	$EQMN_{sortie}$ obtenue avec le modèle SVD-PARAFAC-Volterra d'ordre 3 pour différents RSBs et différentes séquences d'entrée	99
4.4	$EQMN_{sortie}$ obtenue avec le modèle SVD-PARAFAC-Volterra d'ordre 3 pour différentes valeurs de R et P	103
5.1	Volume des régions estimées, intervalles d'incertitude (Δg_i), taux d'actualisation (TA) et temps de convergence (Tco) pour différents algorithmes UBBE (Modèle de Laguerre)	133
5.2	Coordonnées des centres des régions estimées avec les différentes approches UBBE (Modèle de Laguerre)	133
5.3	Volume des régions estimées, intervalles d'incertitude (Δg_i), taux d'actualisation (TA) et temps de convergence (Tco) pour différents algorithmes UBBE (Modèle issu de la BOG)	136
5.4	Coordonnées des centres des régions estimées avec les différentes approches UBBE (Modèle issu de la BOG)	136
5.5	Paramètres du système et du modèle estimés par deux méthodes	144
5.6	$EQMN_{sortie}$ et $EQMN_{coef}$ pour les deux méthodes d'identification et pour deux valeurs différentes du RSB	144

5.7	Coefficients du noyau de Volterra quadratique triangulaire H_2^{tri}	144
5.8	Paramètres du système et paramètres estimés par deux méthodes	146
5.9	Intervalles d'incertitude des paramètres estimés pour deux RSBs différents . .	146
5.10	$EQMN_{sortie}$ et $EQMN_{coef}$ pour les deux méthodes d'identification et pour deux différentes valeurs du RSB	147
5.11	Coefficients de Fourier du modèle BOG-Volterra estimés par deux méthodes .	150
5.12	Intervalles d'incertitude des coefficients de Fourier de la BOG pour différentes bornes de l'erreur	151
5.13	Intervalles d'incertitude des coefficients du noyau linéaire h_1 et des matrices A , B , C , D et E et volume des polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3	155
5.14	Coefficients du noyau linéaire h_1 et des matrices A , B , C , D et E à l'aide de l'approche polytopique et $EQMN_{sortie}$	156

Abréviations

ALS	Algorithme des moindres carrés alternés <i>(Alternating Least Squares (algorithm))</i>
ARLS	Algorithme des moindres carrés récursifs alternés <i>(Alternating Recursive Least Squares (algorithm))</i>
BOG	Base Orthogonale Généralisée
CANDECOMP	<i>CANonical DECOMPosition</i>
CL	Complexité Limitée
dB	Décibel
EB	Erreur Bornée
EBA	Algorithme à Erreur bornée alterné
EQ	Erreur Quadratique
EQM	Erreur Quadratique Moyenne
EQMN	Erreur Quadratique Moyenne Normalisée
EVD	Décomposition en valeurs propres <i>(Eigen Value Decomposition)</i>
i.i.d	Distribué de manière identique et indépendante <i>(Independent and Identically Distributed)</i>
LMS	Algorithme des moindres carrés moyens (<i>Least Mean Squares</i>)
PAM	Modulation à amplitude pulsée <i>(Pulse Amplitude Modulation)</i>
PARAFAC	<i>PARallel FACTor (analysis)</i>
RIF	Réponse Impulsionnelle Finie

(Liste d'abréviations suite)

RLS	Algorithme des moindres carrés récursifs <i>(Recursive Least Squares (algorithm))</i>
RSB	Rapport Signal sur Bruit
SVD	Décomposition en valeurs singulières <i>(Singular Value Decomposition)</i>
VBFO	Modèle de Volterra représenté à l'aide d'une Base de Fonctions Orthogonales
TA	Taux d'Actualisation de l'algorithme
TC	Temps de Convergence de l'algorithme

Chapitre 1

Introduction Générale

1.1 Présentation générale

Identifier un système revient à trouver un modèle mathématique capable de décrire ses caractéristiques temporelles ou/et fréquentielles. Ce modèle peut être linéaire ou non linéaire. Depuis longtemps, l'hypothèse de linéarité a été privilégiée en supposant que la partie non linéaire du système, quand elle existe, est négligeable. Ceci explique le fait que les modèles linéaires ont beaucoup été étudiés et qu'une littérature complète sur le sujet est désormais disponible [60], [51]. Depuis quelques dizaines d'années et grâce au développement de nouveaux calculateurs très puissants, les structures non linéaires ont été de plus en plus utilisées pour modéliser les systèmes physiques. Grâce à cette connaissance croissante de plusieurs nouveaux types de modèles mathématiques non linéaires, l'identification a connu durant la dernière décennie un fort engouement dans des domaines aussi variés que la physique, les statistiques ou la médecine.

L'identification a pour but d'estimer les paramètres d'un modèle mathématique qui reproduit le comportement entrée/sortie d'un système donné. Ce modèle mathématique décrit le système à l'aide d'une expression mathématique reliant les différentes variables du système. Plusieurs caractéristiques peuvent être attribuées à un modèle mathématique : Il est dit à temps continu ou à temps discret, déterministe ou stochastique, linéaire ou non linéaire. Ces modèles sont beaucoup utilisés dans le domaine de la physique et de l'ingénierie.

La problématique de l'identification d'un modèle mathématique peut être définie

comme suit. Étant donné une séquence d'entrée/sortie, nous cherchons à modéliser le système (représenté par une boîte noire) à l'aide d'un modèle mathématique qui soit représentatif de la réalité dans un certain domaine. Ce modèle est sensé décrire le comportement temporel ou fréquentiel du système.

Nous pouvons maintenant poser le problème de l'estimation sous la forme suivante : trouver un vecteur de paramètres $\hat{\theta}$, solution d'un problème d'optimisation déterministe [59] :

$$\hat{\theta} = \arg \min \{J(\theta); \theta \in S\} \quad (1.1)$$

$J(\theta)$ étant une fonction de coût définie sur $S \subset \mathbb{R}^d$. De nombreux critères sont possibles. Mais, parmi eux, le critère du minimum de l'erreur quadratique moyenne est le plus couramment utilisé. Ce critère signifie que nous souhaitons trouver le modèle optimal de telle manière que l'erreur quadratique soit minimale. Présentons certains arguments en faveur de ce critère. Comme il s'agit d'une quantité quadratique, il possède une interprétation physique et représente l'énergie de l'erreur. D'un point de vue mathématique, il peut être interprété comme un critère de distance [56]. Dans le cas où le système est supposé linéaire le calcul de l'erreur quadratique peut se faire uniquement à partir de la connaissance des propriétés du second ordre des signaux. Malheureusement, ce n'est pas le cas pour un système non linéaire, qui nécessite la connaissance des propriétés statistiques des signaux d'ordres supérieurs. Ce problème d'estimation paramétrique peut être résolu à l'aide d'algorithmes d'estimation.

Deux catégories d'algorithmes d'estimation paramétrique sont utilisées :

- Les algorithmes non-récurrents vis-à-vis de temps qui découlent de l'algorithme du maximum de vraisemblance ou de l'algorithme des moindres carrés non-récurrents. Ce dernier est basé sur la minimisation de l'Erreur Quadratique Moyenne (EQM) ou sur son équivalent déterministe à savoir le critère des Moindres Carrés (MC) [43].
- Les algorithmes récurrents vis-à-vis du temps. Ce sont des algorithmes en ligne traitant directement les mesures d'entrée/sortie. Ils sont appelés adaptatifs puisqu'ils permettent l'ajustement du modèle aux données. Parmi ces méthodes, on trouve l'algorithme des moindres carrés récurrents, l'algorithme LMS, ...

En général, l'identification expérimentale d'un système consiste à choisir d'abord

un modèle paramétrique supposé approcher le comportement du système à identifier. Dans une deuxième étape, des méthodes d'estimation sont utilisées pour estimer les paramètres du modèle. Ainsi, une structure expérimentale est choisie et les paramètres correspondants sont estimés. Le modèle (parfaitement connu) est alors testé pour voir s'il reproduit bien la sortie du système. Si le modèle n'arrive pas à suivre les variations du système, une autre famille de modèles plus complexes est testée, les paramètres sont estimés et le modèle est validé [59], [60]. La procédure est illustrée dans la figure 1.1.

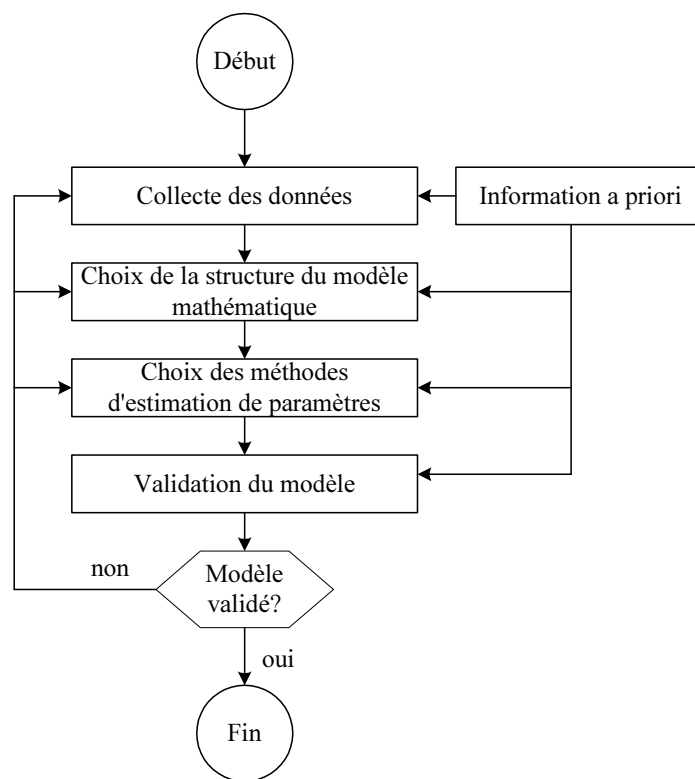


FIG. 1.1 – Procédure d'identification expérimentale

L'identification des systèmes dynamiques non linéaires à partir d'un ensemble de données entrée/sortie est d'une importance fondamentale pour les applications pratiques puisque beaucoup de systèmes physiques possèdent des caractéristiques non linéaires. La structure du modèle de Volterra peut être utilisée pour représenter une classe générale de systèmes non linéaires. Cependant, l'usage pratique d'une telle représentation est limité souvent à cause du grand nombre de paramètres associés à une telle structure, à savoir les coefficients des noyaux.

1.2 Démarche et objectifs scientifiques

Cette thèse a pour objectif d'étudier de nouvelles méthodes de modélisation et d'identification de systèmes non linéaires. La classe des modèles de type Volterra est plus particulièrement considérée. Leur utilisation pour l'analyse des systèmes non linéaires a été, pour la première fois, effectuée par N. Wiener [70] dans le but de caractériser la réponse d'un système non linéaire. Plusieurs autres travaux dans cet axe de recherche (Billings [3] et Schetzen [57]) ont suivi. L'étude des systèmes non linéaires discrets à l'aide des séries de Volterra a commencé avec le travail de Alper en 1965 [1].

Ces modèles qui sont non linéaires vis-à-vis du signal d'entrée, présentent l'avantage d'être linéaires vis-à-vis de leurs paramètres. Toutefois, ils sont caractérisés par une complexité beaucoup plus grande que les modèles linéaires, en raison du nombre élevé des coefficients qui caractérisent leurs différents noyaux, ce qui rend difficile leur identification, tout particulièrement dans le cadre d'applications en temps réel nécessitant un ajustement en ligne de ces paramètres. En vue de réduire la complexité paramétrique des modèles de Volterra, deux approches sont envisagées :

- L'une consiste à utiliser des développements en série des différents noyaux sur des bases de fonctions orthogonales. Les représentations simplifiées des modèles de Volterra ainsi développés sont désignées dans la suite par VBFO.
- L'autre fait appel à des décompositions d'ordre réduit des tenseurs relatifs aux noyaux d'ordre supérieur ou égal à trois.

Dans le deuxième chapitre, diverses bases de fonctions (Laguerre, Kautz et Bases Orthogonales Généralisées (BOG)) sont tout d'abord étudiées en vue de leur utilisation pour la modélisation des systèmes linéaires. Une représentation d'état associée à un développement sur une base de fonctions orthogonales généralisées est développée puis utilisée pour la construction de prédicteurs de la sortie du système ainsi modélisé. Ces bases de fonctions sont ensuite utilisées pour la représentation des noyaux d'un modèle de Volterra. Le problème d'identification comporte plusieurs volets : détermination des paramètres caractéristiques des bases de fonctions orthogonales (pôle réel unique pour les bases de fonctions de Laguerre, deux pôles complexes conjugués pour les bases de fonctions de Kautz et plusieurs pôles réels et complexes conjugués pour les Bases Orthogonales Généralisées), de l'ordre des développements des différents noyaux appelé ordre

de troncature, des coefficients de ces développements appelés coefficients de Fourier et de l'incertitude relative à ces coefficients estimés. Une étude comparative des différentes bases est effectuée afin de savoir quels types de bases utiliser pour la représentation d'un système donné.

Dans le troisième chapitre, on s'intéresse aux décompositions tensorielles (PARAFAC, PARAFAC2, TUCKER1, TUCKER2, TUCKER3, PARATUCK). Une étude comparative de ces différentes techniques de décomposition est effectuée afin de dégager les avantages et les inconvénients de chacune d'elle.

Dans le quatrième chapitre, on propose d'autres types de modèles de Volterra à complexité réduite. En considérant le noyau quadratique comme une matrice et les autres noyaux comme des tenseurs d'ordres supérieurs à deux, nous utilisons une décomposition à l'aide des valeurs singulières (SVD) pour le noyau quadratique et la décomposition tensorielle appelée PARAFAC pour les noyaux d'ordres supérieurs à deux.

Le nouveau modèle présenté est appelé SVD-PARAFAC-Volterra. Un nouvel algorithme appelé ARLS (Alternating Recursive Least Squares algorithm) est présenté pour l'estimation paramétrique de ce type de modèles. Cet algorithme est essentiellement basé sur la technique RLS appliquée d'une manière alternée. Un canal satellitaire simplifié est modélisé en utilisant les modèles non linéaires proposés puis identifié à l'aide du nouvel algorithme ARLS.

Enfin dans le cinquième chapitre, des méthodes d'identification robuste dites à erreur bornée sont présentées. Elles sont d'abord utilisées pour l'identification des modèles VBFO, travail qui vise à étendre au cas des systèmes non linéaires incertains des résultats obtenus récemment pour des systèmes linéaires incertains. Parmi ces techniques d'identification on considère : l'approche polytopique classique, l'approche orthotopique et les approches polytopiques à complexité limitée. Ces méthodes nous permettent d'estimer les intervalles d'incertitude des coefficients de Fourier du développement sur les différentes bases orthogonales étudiées. Cet axe de recherche a fait l'objet d'un projet de coopération CNRS/DGRST entre le laboratoire I3S et l'École Nationale d'Ingénieurs de Monastir (ENIM), projet auquel j'ai participé. Ces mêmes méthodes d'identification sont utilisées aussi afin d'identifier les intervalles d'incertitude des paramètres du modèle SVD-PARAFAC-Volterra.

1.3 Contributions

Les principales contributions de cette thèse sont :

- Développement de nouvelles méthodes d'estimation des pôles d'une base de fonctions orthonormales (Laguerre, Kautz et BOG).
- Représentation des modèles de Volterra à l'aide des BOG.
- Représentation d'état associée à un développement sur une base de fonction orthogonale généralisée (Laguerre, Kautz et BOG).
- Construction de prédicteur de la sortie de système à l'aide des BOG.
- Unification et étude comparative des méthodes de décomposition tensorielle.
- Développement d'un nouveau modèle de Volterra à complexité réduite SVD-PARAFAC-Volterra.
- Identification de modèles de Volterra à complexité réduite à l'aide d'un nouvel algorithme ARLS.
- Utilisation des méthodes d'identification à erreur bornée dans le cas non linéaire.
- Identification alternée à erreur bornée dans le cas du modèle SVD-PARAFAC-Volterra.

Ces contributions ont donné lieu aux publications suivantes :

– **Articles soumis à des revues internationales**

1. A. Khouaja, G. Favier and H. Messaoud, "Pole estimation for constructing generalized orthonormal bases in the context of system identification", submitted to *Systems, Analysis, Modeling and Simulation (SAMS) Special Issue* 2004.
2. A. Khouaja and G. Favier, "Third-order Volterra systems identification using a SVD-PARAFAC approach", submitted to *IEEE Transactions on Signal Processing*, 2004.

– **Communications parues dans des actes de conférences internationales**

1. A. Khouaja, G. Favier and H. Messaoud, "System identification using Generalized Orthonormal Bases", *In Proc. IEEE Conference on Signal, Systems, Decision and information theory (SSD)*, Sousse, Tunisia, 26-28 March, 2003.

2. A. Khouaja, A. Kibangou and G. Favier, "Third-order Volterra kernels complexity reduction using PARAFAC", *In Proc IEEE International Symposium on Control, Communications and Signal Processing (ISCCSP)*, Hammamet, Tunisia, pp. 857-860, March, 2004.
 3. A. Khouaja and G. Favier, "Identification of PARAFAC-Volterra cubic models using an alternating recursive least squares algorithm", *In Proc. EURASIP European Signal Processing Conference (EUSIPCO)*, Vienna, Austria, pp. 1903-1906, September, 2004.
 4. G. Favier, A. Kibangou and A. Khouaja, "Nonlinear system modelling by means of Volterra models. Approaches for the parameter complexity reduction", *Invited paper in Symposium Techniques Avancées et Stratégies Innovantes en Modélisation et Commandes Robustes des processus Industriels*, Martigues, France, September, 2004.
- **Communications soumises à des conférences internationales**
1. A. Khouaja and G. Favier, "Identification of SVD-PARAFAC based Third-order Volterra models using an ARLS algorithm", *Accepted for publication at IFAC World Congress*, Prague, Czech Republic, July, 2005

Chapitre 2

Représentation des systèmes à l'aide de la Base Orthogonale Généralisée

2.1 Introduction

La représentation d'un système sur une base de fonctions orthogonales présente le grand avantage de s'exprimer de façon linéaire par rapport aux coefficients de Fourier, une fois les paramètres de la base (pôles) fixés ; ceci confère à ce mode de représentation des propriétés intéressantes pour l'identification des systèmes.

Avant d'entamer une démarche d'identification, il est toutefois important d'effectuer une étude que nous qualifions de "mathématique" sur la représentation et l'approximation de systèmes à l'aide des bases de fonctions orthogonales.

Les systèmes traités dans ce chapitre sont invariants dans le temps, causaux et asymptotiquement stables. Ils sont décrits par une équation aux différences, une réponse impulsionnelle discrète ou une fonction de transfert.

Les fonctions delta de Dirac, de Laguerre ainsi que les fonctions issues de la base orthogonale généralisée forment une base orthogonale complète dans $l_2[0, \infty[$ (Espace des fonctions de carré sommable). De même leur transformée en z forme une base orthogonale complète dans l'espace de Hardy $H^2(\mathcal{D}^c)$ (Espace des fonctions complexes analytiques sur le cercle unité \mathcal{C} et continues à l'extérieur du disque unité \mathcal{D} et sur le cercle unité \mathcal{C} , \mathcal{D}^c désigne le complément du disque unité dans H^2). Ces fonctions feront donc l'objet de cette étude.

Rappelons la notion d'orthogonalité qui vient d'être évoquée pour ces catégories de fonctions.

Définition 1 : Une séquence de fonctions $\{b_n(k)\}$, $k \in \mathbb{N}$, forme un ensemble de fonctions orthogonales dans l'espace $l_2[0, \infty[$, si et seulement si leur produit scalaire est tel que :

$$\langle b_n(k), b_m(k) \rangle = \sum_{k=0}^{\infty} b_n(k)b_m^*(k) = C_{nm}\delta_{nm} \quad ; \quad \forall n, m \in \mathbb{N}, C_{nm} \in \mathbb{R} \quad (2.1)$$

$b_m^*(k)$ représente le conjugué de $b_m(k)$ et δ_{nm} représente le symbole de Kronecker qui vaut 1 si $n = m$ et zéro autrement.

Définition 2 : Une séquence de fonctions $\{B_n(e^{j\omega})\}$, $\omega \in]-\pi, \pi[$, forme un ensemble de fonctions orthogonales dans l'espace de Hardy $H^2(\mathcal{C})$, si et seulement si leur produit scalaire est tel que :

$$\langle B_n(e^{j\omega}), B_m(e^{j\omega}) \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} B_n(e^{j\omega})B_m^*(e^{-j\omega})d\omega = C_{nm}\delta_{nm} \quad ; \quad \forall n, m \in \mathbb{N}, C_{nm} \in \mathbb{R} \quad (2.2)$$

Définition 3 : Une séquence de fonctions $\{B_n(z)\}$, $z \in \mathbb{C}$, forme un ensemble de fonctions orthogonales dans l'espace de Hardy $H^2(\mathcal{D}^c)$, si et seulement si leur produit scalaire est tel que :

$$\langle B_n(z), B_m(z) \rangle = \frac{1}{2\pi j} \oint_{\mathcal{C}} B_n(z)B_m^*(z^{-1})\frac{dz}{z} = C_{nm}\delta_{nm} \quad ; \quad \forall n, m \in \mathbb{N}, C_{nm} \in \mathbb{R} \quad (2.3)$$

Remarque : Rappelons que pour un polynôme $B(z) = \sum_{i=0}^N a_i z^i$ en z , son conjugué est un polynôme en z^{-1} qui s'écrit $B^*(z) = \sum_{i=0}^N a_i^* z^{-i}$.

La base orthogonale complète la plus utilisée en automatique est incontestablement la base des fonctions delta de Dirac. La transformée en z de ces dernières engendre les filtres à Réponse Impulsionnelle Finie (RIF). Ces filtres présentent l'avantage d'une implémentation simple. Une approximation de toute fonction de transfert d'un système stable peut être réalisée à l'aide d'un modèle RIF. Cependant, lorsqu'ils modélisent des systèmes à longue réponse impulsionnelle, le nombre de paramètres nécessaires à

la modélisation du système devient élevée. D'où l'intérêt de trouver des modèles qui gèrent des approximations précises avec moins de paramètres.

Pour réduire la longueur des filtres RIF, l'utilisation des filtres de Laguerre et de Kautz est envisagée. Les filtres de Laguerre $\{L_k\}_{k=1,\dots}$ sont caractérisés par un pôle unique, stable et réel. Ils sont adaptés à la modélisation des systèmes à dynamique dominante du premier ordre (Gunnarsson et Wahlberg [25], Fu et Dumont [24], Oliveira e Silva [19] et Masnadi-Shirazi et Ghasemi [48]). Les filtres de Kautz $\{\Psi_k^{(1)}, \Psi_k^{(2)}\}_{k=1,\dots}$, quant à eux, sont caractérisés par une paire de pôles complexes conjugués. De ce fait, ils sont fortement indiqués pour la représentation de systèmes ayant une dynamique dominante du second ordre (Wahlberg [67], Oliveira e Silva [20] et Wahlberg et Mäkilä [68]).

Bien que la complétude de l'espace engendré par les fonctions de Laguerre ou de Kautz permette de représenter tout système Linéaire Invariant dans le Temps (LIT) stable par une série infinie de fonctions et que la convergence de chaque série permette sa troncature à un ordre fini, l'ordre de troncature doit être élevé pour approcher au mieux le comportement de systèmes réels possédant plusieurs dynamiques éloignées les unes des autres.

L'idée de Linskog [42] est d'utiliser plusieurs réseaux de filtres en parallèles, afin que chacun puisse approcher au mieux le comportement résultant d'une dynamique et qu'il permette ainsi la troncature de la série associée à chaque réseau à un ordre "peu élevé". Ainsi s'il s'agit de modéliser le comportement dû à une dynamique amortie ou oscillante. Le réseau de filtres de Laguerre ou de Kautz peut respectivement être utilisé. La connaissance a priori du système est alors très utile pour fixer les pôles de chaque réseau. Linskog propose d'écrire toute fonction de transfert appartenant à $H^2(\mathcal{D}^c)$ sous la forme :

$$G(z) = \sum_{l=1}^{n^L} \sum_{k=0}^{N_l^L} g_{l,k}^L L_{l,k}(z, \xi_l) + \sum_{l=1}^{n^\Psi} \sum_{k=0}^{N_l^\Psi} g_{l,k}^{\Psi_1} \Psi_{l,k}^{(1)}(z, p_l) + g_{l,k}^{\Psi_2} \Psi_{l,k}^{(2)}(z, p_l^*) \quad (2.4)$$

où :

- $L_{l,k}$ est la $k^{\text{ème}}$ fonction de la $l^{\text{ème}}$ base de Laguerre de pôle ξ_l et $g_{l,k}^L$ le coefficient de Fourier correspondant,
- $\Psi_{l,k}^{(1)}$ et $\Psi_{l,k}^{(2)}$ la $k^{\text{ème}}$ paire de fonctions de la $l^{\text{ème}}$ base de Kautz de pôles complexes conjugués p_l et p_l^* et $g_{l,k}^{\Psi_1}$ et $g_{l,k}^{\Psi_2}$ leurs coefficients de Fourier correspondants.

- $n^L + n^\Psi$ est le nombre total de réseaux de Laguerre et de Kautz utilisés, N_l^L et $2N_l^\Psi$ le nombre de filtres dans chaque réseau.

Cette configuration a l'avantage principal des filtres de Laguerre et de Kautz, à savoir une représentation linéaire par rapport aux coefficients, une fois que tous les pôles sont fixés. Cependant son inconvénient majeur est que l'orthogonalité entre fonctions des deux réseaux différents n'est pas assurée. On peut alors s'attendre à des problèmes numériques, lors du calcul des différents coefficients de Fourier $g_{l,k}^L$, $g_{l,k}^{\Psi_1}$ et $g_{l,k}^{\Psi_2}$, quand les pôles de deux réseaux différents sont proches l'un de l'autre. De plus, l'écriture (2.4), introduit de nombreux paramètres à savoir, les nombres n^L et n^Ψ , les pôles et les ordres de troncature N_l^L et N_l^Ψ de chaque réseau ainsi que le nombre total ($n^L N_l^L + 2n^\Psi N_l^\Psi$) des coefficients de Fourier. Alors, plusieurs questions interviennent concernant le choix de tous ces paramètres :

- Combien de réseaux de filtres utiliser ?
- Quel ordre de troncature et quel pôle choisir pour chaque réseau ?
- Si l'erreur quadratique n'est pas suffisamment petite, quelle stratégie adopter pour la réduire.
- Ajouter un nouveau réseau avec de nouvelles dynamiques.
- Optimiser le placement des pôles.
- Ajouter des filtres supplémentaires dans les réseaux existants. Lesquels ?

Toutefois il est sûr qu'il faut recalculer tous les coefficients à chaque modification du nombre de filtres, puisque la propriété d'orthogonalité des fonctions constituant tous les réseaux n'est pas préservée.

Pour assurer l'orthogonalité tout en introduisant plusieurs dynamiques dans le modèle, Ninness et Gustafsson [53], proposent d'utiliser la Base Orthogonale Généralisée (BOG) qu'ils synthétisent à partir de choix arbitraires de pôles. Cette base est également connue sous le nom de base des filtres de Kautz généralisée, que lui a donné den Brinker [15]. A son origine, on trouve les travaux de Broome [8]. Cependant, elle a été délaissée au profit des bases de Laguerre et de Kautz à deux paramètres. Heuberger [30] et Ninness et Gustafsson [53] ont le mérite d'avoir attiré l'attention de la communauté automatique sur son utilité pour la modélisation de systèmes ayant plusieurs dynamiques. Ils l'ont synthétisée en fixant les pôles des fonctions de transfert.

Dans ce chapitre, nous commencerons notre étude par présenter les méthodes de

construction de la Base Orthogonale Généralisée. Nous examinerons par la suite les différentes structures d'implémentation de ces bases. Enfin, nous traitons le choix de la BOG optimale.

2.2 Synthèse des fonctions de la base orthogonale généralisée

Ninness et Gustafsson [53] ont montré que les fonctions définies par :

$$B_0(z) = \frac{\sqrt{1 - |\xi_0|^2}}{z - \xi_0} \quad (2.5)$$

$$B_n(z) = \frac{\sqrt{1 - |\xi_n|^2}}{z - \xi_n} \prod_{k=0}^{n-1} \left(\frac{1 - \xi_k^* z}{z - \xi_k} \right) ; n = 1, 2, \dots \quad (2.6)$$

avec $\xi_k ; k = 0, 1, \dots$, leurs pôles caractéristiques respectifs, vérifient l'orthonormalité d'une part et la complétude de l'espace qu'elles engendrent d'autre part. Elles permettent donc d'écrire toute fonction de transfert appartenant à $H^2(D^c)$ sous la forme :

$$G(z) = \sum_{n=0}^{\infty} g_n B_n(z) \quad (2.7)$$

La procédure d'orthonormalisation qu'ils utilisent est détaillée ci après.

En supposant que le pôle dominant de la fonction de transfert originelle $G(z)$ est à proximité de ξ_0 avec $|\xi_0| < 1$, ils fixent alors le pôle de B_0 à ξ_0 et introduisent la constante de normalisation A :

$$B_0(z) = A \frac{z}{z - \xi_0} \quad (2.8)$$

Le calcul de A se fait alors ainsi :

$$\|B_0\|^2 = 1 \Leftrightarrow \frac{1}{2\pi j} \oint_{\mathcal{C}} B_0(z) B_0^*(z^{-1}) \frac{dz}{z} = 1 \quad (2.9)$$

où \mathcal{C} représente le cercle unité.

$$\Leftrightarrow A = \sqrt{1 - |\xi_0|^2} \quad (2.10)$$

A présent, en supposant que la fonction de transfert $G(z)$ possède un second pôle qui se trouve à proximité de ξ_1 , alors la deuxième fonction de la base sera de la forme :

$$B_1(z) = A' \frac{(1 - \xi_0^* z)}{(z - \xi_0)(z - \xi_1)} \quad (2.11)$$

Afin de mieux appréhender le choix de B_1 dans (2.11), on note que $B_0(z)$ possède un pôle en ξ_0 dans le disque unité $\mathcal{D} = \{z \in \mathbb{C}; |z| < 1\}$. Pour assurer l'orthogonalité de B_0 et B_1 , il est nécessaire d'inclure un zéro dans $B_1^*(z)$ au point ξ_0 pour que la fonction $B_0(z)B_1^*(z^{-1})$ soit analytique sur le cercle unité \mathcal{C} , ce qui assure la nullité du produit scalaire de B_0 et B_1 .

$$\langle B_0, B_1 \rangle = \frac{\sqrt{1 - |\xi_0|^2}}{2\pi j} \oint_{\mathcal{C}} \frac{A'(z - \xi_0)}{(z - \xi_0)(1 - \xi_0^*z)(1 - \xi_1^*z)} dz = 0 \quad (2.12)$$

où A' est choisi de façon que B_1 soit normé :

$$\|B_1\|^2 = 1 \Leftrightarrow \frac{A'^2}{2\pi j} \oint_{\mathcal{C}} \frac{(z - \xi_0)(1 - \xi_0^*z) dz}{(z - \xi_0)(z - \xi_1)(1 - \xi_0^*z)(1 - \xi_1^*z)} = 1 \quad (2.13)$$

$$\Leftrightarrow A' = \sqrt{1 - |\xi_1|^2} \quad (2.14)$$

Notons que lors du choix de B_1 , le pôle ξ_1 peut être pris n'importe où dans le cercle unité. En continuant ce raisonnement avec des pôles fixés arbitrairement, les auteurs arrivent à la formulation générale (2.6).

Remarque : Si l'on se restreint au choix d'un pôle réel et unique $\xi_n = \xi \in]-1, 1[$, quel que soit n , on retrouve la base particulière des fonctions de Laguerre. De même si ce pôle ξ est nul, la base des filtres RIF est générée.

Cette procédure d'orthonormalisation est applicable même si l'on souhaite fixer des pôles complexes. Puisque la réponse temporelle de toute fonction de transfert doit appartenir à l'ensemble des nombres réels, Ninness et Gustafsson proposent de modifier légèrement la procédure de sélection des pôles de façon à choisir, par paire, des pôles complexes conjugués. De plus, ils suggèrent de substituer la paire de fonctions (B_n, B_{n+1}) , générée à partir de l'équation (2.6) par la paire (B'_n, B''_n) issue d'une combinaison linéaire des fonctions originelles comme le montre l'équation (2.15), ce qui garantit l'orthogonalité de B'_n ainsi que celle de B''_n avec les autres fonctions de la base.

$$\begin{pmatrix} B'_n \\ B''_n \end{pmatrix} = \begin{pmatrix} c_0 & c_1 \\ c'_0 & c'_1 \end{pmatrix} \begin{pmatrix} B_n \\ B_{n+1} \end{pmatrix} \quad c_0, c_1, c'_0 \text{ et } c'_1 \in \mathbb{C} \quad (2.15)$$

Il est intéressant de souligner que les fonctions B'_n et B''_n engendrent un sous espace dans l'espace fonctionnel de dimension infinie. Ce sous espace est orthogonal au sous

espace engendré par B_0, B_1, \dots, B_{n-1} . Les conditions que posent Ninness et Gustafsson concernant le choix des coefficients complexes, sont alors plus simples à comprendre. Les deux fonctions B'_n et B''_n doivent être orthonormées. Pour cela, il faut vérifier les trois conditions suivantes :

- La normalité de B'_n qui impose $|c_0|^2 + |c_1|^2 = 1$
- La normalité de B''_n qui impose $|c'_0|^2 + |c'_1|^2 = 1$
- L'orthogonalité de B'_n et B''_n qui impose la condition $c_0 c'_0{}^* + c_1 c'_1{}^* = 0$

Afin de pouvoir retrouver les fonction B_n et B_{n+1} à partir de B'_n et B''_n , on rajoute la condition d'inversibilité suivante : $c_0 c'_1 - c_1 c'_0 \neq 0$.

Puisque sur tout plan vectoriel, il existe une infinité de paires de vecteurs ortho-normés, on obtient alors une infinité de solutions. Pour déterminer l'ensemble de ces solutions, les auteurs proposent le changement de paramètres suivant :

$$c_0 = \frac{\beta + \xi_n^* \mu}{1 - \xi_n^{*2}}; \quad c_1 = \frac{\xi_n^* \beta + \mu}{1 - \xi_n^{*2}}; \quad c'_0 = \frac{\beta' + \xi_n^* \mu'}{1 - \xi_n^{*2}} \quad \text{et} \quad c'_1 = \frac{\xi_n^* \beta' + \mu'}{1 - \xi_n^{*2}} \quad (2.16)$$

ce qui donne :

$$B'_n(z) = \sqrt{\frac{1 - |\xi_n|^2}{1 - |\xi_{n-1}|^2}} \left\{ \frac{(1 - \xi_{n-1}^* z)(\beta z + \mu)}{z^2 - (\xi_n + \xi_n^*)z + |\xi_n|^2} \right\} B_{n-1}(z) \quad (2.17)$$

$$B''_n(z) = \sqrt{\frac{1 - |\xi_n|^2}{1 - |\xi_{n-1}|^2}} \left\{ \frac{(1 - \xi_{n-1}^* z)(\beta' z + \mu')}{z^2 - (\xi_n + \xi_n^*)z + |\xi_n|^2} \right\} B_{n-1}(z) \quad (2.18)$$

Ninness et Gustafsson montrent que les paramètres β, μ, β' et μ' qui s'écrivent en fonction de c_0, c_1, c'_0 et c'_1 et qui respectent les trois conditions d'orthonormalité précédentes, doivent satisfaire les trois équations suivantes :

$$\begin{aligned} (i) \quad & x^T M x = |1 - \xi_n^2|^2 \\ (ii) \quad & y^T M y = |1 - \xi_n^2|^2 \end{aligned} \quad (2.19)$$

où

$$\begin{aligned} x &= (\beta, \mu)^T \\ y &= (\beta', \mu')^T \end{aligned} \quad \text{et} \quad M = \begin{pmatrix} 1 + |\xi_n|^2 & 2\text{Re}(\xi_n) \\ 2\text{Re}(\xi_n) & 1 + |\xi_n|^2 \end{pmatrix} \quad (2.20)$$

$$(iii) \quad x^T M y = 0$$

La résolution de ce système d'équations utilise trois degrés de liberté. Il en reste donc un. D'où l'existence de plusieurs solutions x et y aux équations (2.19) et (2.20).

Un choix possible selon Malti [44] est :

$$x = (\beta, \mu)^T = \sqrt{\frac{(1 + \xi_n^*)(1 + \xi_n)}{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad (2.21)$$

et

$$y = (\beta', \mu')^T = \sqrt{\frac{(1 - \xi_n^*)(1 - \xi_n)}{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (2.22)$$

En remplaçant β , μ , β' et μ' dans les équations (2.17) et (2.18), Malti retrouve les deux vecteurs B'_n et B''_n obtenus par combinaison linéaire des vecteurs B_n et B_{n+1} issus à leur tour de la structure (2.6). Les équations (2.23) et (2.24) rejoignent la définition des fonctions de Kautz selon Oliveira e Silva [20].

$$B'_n(z) = \sqrt{\frac{(1 + \xi_n^*)(1 + \xi_n)(1 - |\xi_n|^2)}{2}} \frac{z - 1}{z^2 - (\xi_n + \xi_n^*)z + |\xi_n|^2} \prod_{k=0}^{n-1} \left(\frac{1 - \xi_k^* z}{z - \xi_k} \right) \quad (2.23)$$

$$B''_n(z) = \sqrt{\frac{(1 - \xi_n^*)(1 - \xi_n)(1 - |\xi_n|^2)}{2}} \frac{z + 1}{z^2 - (\xi_n + \xi_n^*)z + |\xi_n|^2} \prod_{k=0}^{n-1} \left(\frac{1 - \xi_k^* z}{z - \xi_k} \right) \quad (2.24)$$

• **Résumé de la méthode utilisée pour inclure des pôles complexes dans les fonctions issues de la BOG**

Tous les pôles réels des fonctions issues de la BOG (2.6) sont fixés sans aucune contrainte. Par contre si l'on souhaite introduire un pôle complexe ξ_n dans la fonction B_n , alors il faut aussi introduire son conjugué dans la fonction B_{n+1} , (i.e. $\xi_{n+1} = \xi_n^*$). Ensuite, il faut substituer la paire de fonctions (B_n, B_{n+1}) issue de (2.6) par la nouvelle paire de fonctions (B'_n, B''_n) de définitions (2.23) et (2.24).

$$G(z) = \sum_{i=0}^{n-1} g_i B_i(z) + g'_n B'_n(z) + g''_n B''_n(z) + \dots \quad (2.25)$$

La BOG ainsi construite selon Ninness et Gustafsson est la généralisation de bases orthogonales bien connues comme celles des filtres RIF, de Laguerre et de Kautz.

2.2.1 Base de filtres FIR

En prenant $\xi_n = 0$ dans l'équation (2.6), on a :

$$B_n(z) = z^{-(n+1)} \quad (2.26)$$

ceci constitue la base des filtres FIR.

2.2.2 Base de filtres de Laguerre

En prenant $\xi_n = \xi$; $\forall n = 1, \dots, N$ dans l'équation (2.6), on a :

$$B_n(z) = \frac{\sqrt{1 - \xi^2}}{z - \xi} \left(\frac{1 - \xi z}{z - \xi} \right)^n \quad (2.27)$$

Ceci constitue la base de filtres de Laguerre caractérisée par un pôle réel unique $\xi \in] - 1, 1[$.

2.2.3 Base de filtres de Kautz

On trouve dans la littérature deux définitions des fonctions de Kautz à deux paramètres. Elles sont toutes les deux issues de la forme générale proposée par Broome [8] et sont équivalentes.

Première définition

Elle est couramment utilisée par Oliveira e Silva [20] et par den Brinker et al [16]. Elle consiste à expliciter directement les filtres de Kautz à partir du pôle complexe de Kautz ξ et de son conjugué ξ^* .

Reprenons le cas de la construction d'une BOG à pôles complexes. Considérons l'introduction d'un pôle complexe ξ . Les équations (2.23) et (2.24) donnent :

$$B'_n(z) = \sqrt{\frac{(1 + \xi^*)(1 + \xi)(1 - |\xi|^2)}{2}} \frac{z - 1}{z^2 - (\xi + \xi^*)z + |\xi|^2} \prod_{k=0}^{n-1} \left(\frac{1 - \xi_k^* z}{z - \xi_k} \right) \quad (2.28)$$

$$B''_n(z) = \sqrt{\frac{(1 - \xi^*)(1 - \xi)(1 - |\xi|^2)}{2}} \frac{z + 1}{z^2 - (\xi + \xi^*)z + |\xi|^2} \prod_{k=0}^{n-1} \left(\frac{1 - \xi_k^* z}{z - \xi_k} \right) \quad (2.29)$$

On regroupe les termes des produits précédents deux par deux. En sachant que selon la propriété des réseaux de filtres de Kautz si $\xi_k = \xi$ alors $\xi_{k+1} = \xi^*$ et en posant $\Psi_{2p}(z, \xi) = B'_n(z)$, $n = 0, 1, 2, \dots$ les fonction de Kautz d'ordres pairs et $\Psi_{2p+1}(z, \xi) = B''_n(z)$, $n = 0, 1, 2, \dots$ les fonctions de Kautz d'ordres impairs, on a :

$$\Psi_{2p}(z, \xi) = K_1 \frac{z - 1}{z^2 - (\xi + \xi^*)z + |\xi|^2} \left(\frac{(1 - \xi^* z)(1 - \xi z)}{z^2 - (\xi + \xi^*)z + |\xi|^2} \right)^p \quad (2.30)$$

$$\Psi_{2p+1}(z, \xi) = K_2 \frac{z + 1}{z^2 - (\xi + \xi^*)z + |\xi|^2} \left(\frac{(1 - \xi^* z)(1 - \xi z)}{z^2 - (\xi + \xi^*)z + |\xi|^2} \right)^p \quad (2.31)$$

avec

$$K_1 = \sqrt{\frac{(1 + \xi^*)(1 + \xi)(1 - |\xi|^2)}{2}} \quad (2.32)$$

et

$$K_2 = \sqrt{\frac{(1 - \xi^*)(1 - \xi)(1 - |\xi|^2)}{2}} \quad (2.33)$$

Deuxième définition

Elle est utilisée par Wahlberg [67] et par Lindskog [42]. Elle exprime le dénominateur ainsi qu'une partie du numérateur de ces fonctions comme une multiplication de polynômes d'ordre 2, dont les racines, imposées par le choix des paramètres b et c , sont complexes conjuguées. Elle se présente ainsi :

$$\overline{\overline{\Psi}}_{2p}(z, b, c) = \frac{\sqrt{1 - c^2}(z - b)}{z^2 + b(c - 1)z - c} \left[\frac{-cz^2 + b(c - 1)z + 1}{z^2 + b(c - 1)z - c} \right]^p \quad (2.34)$$

$$\overline{\overline{\Psi}}_{2p+1}(z, b, c) = \frac{\sqrt{(1 - c^2)(1 - b^2)}}{z^2 + b(c - 1)z - c} \left[\frac{-cz^2 + b(c - 1)z + 1}{z^2 + b(c - 1)z - c} \right]^p \quad (2.35)$$

avec $p = \frac{n}{2} = 1, 2, \dots, ; -1 < b < 1$ et $-1 < c < 1$.

Ces deux dernières contraintes assurent la stabilité asymptotique des filtres de Kautz discrets.

Équivalence entre les deux définitions

Pour que la paire de fonctions $(\Psi_{2p}(z, \xi), \Psi_{2p+1}(z, \xi))$, issue de la première définition, engendre le même espace que la paire de fonctions $(\overline{\overline{\Psi}}_{2p}(z, b, c), \overline{\overline{\Psi}}_{2p+1}(z, b, c))$, issue de la deuxième définition, il est évident que l'emplacement du pôle ξ de Kautz soit identique pour les deux paires de fonctions. Ceci est évidemment valable pour son conjugué ξ^* . En conséquence, on déduit les relations suivantes entre leurs paramètres respectifs :

$$b = \frac{\xi + \xi^*}{1 + \xi\xi^*} \quad (2.36)$$

$$c = -\xi\xi^* \quad (2.37)$$

Pourtant, en remplaçant les valeurs b et c précédentes dans les équations (2.34) et (2.35), on ne retrouve pas les fonctions définies par (2.30) et (2.31). On conclut donc que les fonctions de Kautz issues de la première et deuxième définition ne sont pas colinéaires, bien qu'elles engendrent le même espace. Malti [44] a donc calculé l'angle

Θ entre les fonctions paires. Du fait de l'orthogonalité, Θ doit également définir l'angle entre les fonctions impaires. Il a montré que pour passer de la première définition des fonctions de Kautz à la deuxième, il suffit d'effectuer une rotation d'un angle Θ qui dépend de l'emplacement des pôles de Kautz.

2.3 Réseau de filtres de la Base Orthogonale Généralisée

2.3.1 Réseau de filtres de la Base des fonctions de Laguerre

Les fonctions de Laguerre présentées par l'équation (2.27), possèdent des transformées en z permettant d'établir la relation de récurrence suivante :

$$L_0(z, \xi) = \frac{\sqrt{1 - \xi^2}}{z - \xi} \quad (2.38)$$

$$L_n(z, \xi) = \frac{1 - \xi z}{z - \xi} L_{n-1}(z, \xi) \quad (2.39)$$

Les filtres $L_n(z, \xi)$, $n = 0, 1, 2, \dots$ sont communément appelés filtres de Laguerre discrets. Il est facile de constater qu'ils sont stables si le pôle ξ , appelé pôle de Laguerre, est choisi dans $] - 1, 1[$. S'il est nul, on retrouve les filtres RIF. En conséquence, les filtres de Laguerre discrets sont considérés comme une extension des filtres RIF. Ils ont l'avantage de pouvoir être implantés en réseau comme le montre la figure 2.1.

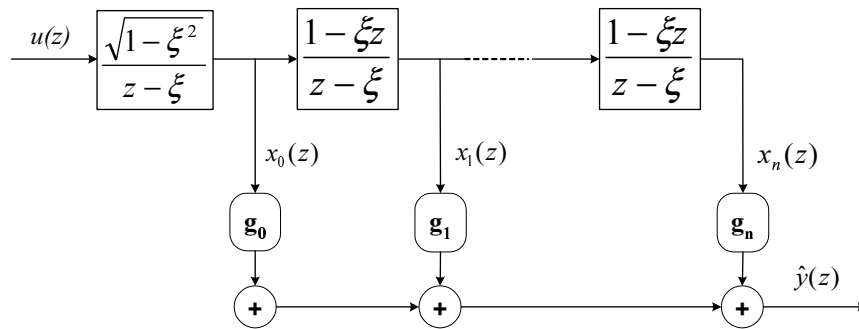


FIG. 2.1 – Réseau de filtres de la Base des fonctions de Laguerre

2.3.2 Réseau de filtres de la Base des fonctions de Kautz

Les filtres de Kautz discrets définis dans (2.34) et (2.35) et utilisés par Wahlberg [67] permettent de construire une représentation en réseau analogue à la représentation des filtres de Laguerre comme le montre la figure 2.2.

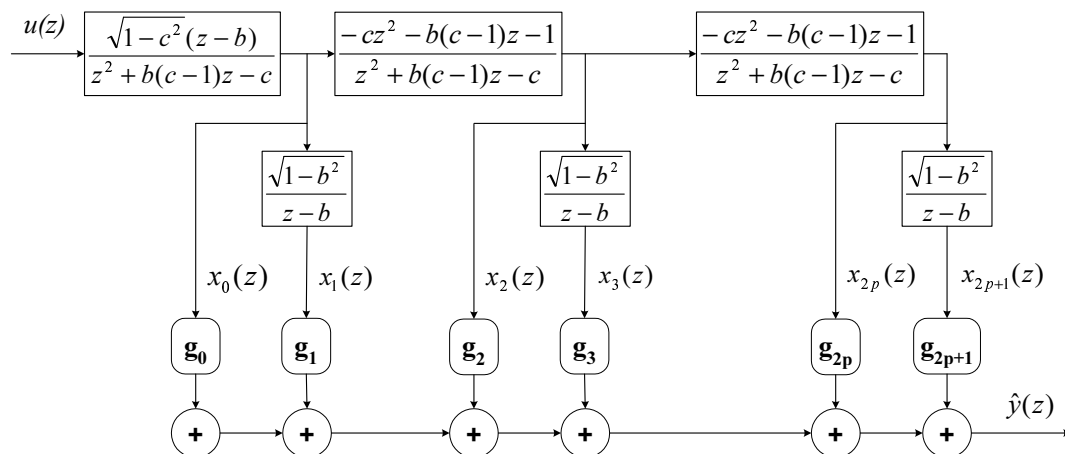


FIG. 2.2 – Réseau de filtres de la Base des fonctions de Kautz

2.3.3 Réseau de filtres de la Base Orthogonale Généralisée

A partir des équations (2.5) et (2.6), on peut formuler la relation de récurrence suivante entre les fonctions de la base généralisée :

$$B_0(z) = \frac{\sqrt{1-|\xi_0|^2}}{z-\xi_0} \quad (2.40)$$

$$B_n(z) = \frac{\sqrt{1-|\xi_n|^2}}{\sqrt{1-|\xi_{n-1}|^2}} \left(\frac{1-\xi_{n-1}^*z}{z-\xi_n} \right) B_{n-1}(z) \quad (2.41)$$

On retrouve dans l'article de Ninness et Gustafsson [53], le schéma bloc de la figure 2.3, qui montre que les filtres de la BOG sont constitués de plusieurs blocs passe-bas ayant différentes fréquences de coupure, reliés entre eux par des blocs passe-tout.

2.4 Détermination des pôles de la BOG

En théorie, tout système stable peut être exactement représenté par une série infinie de fonctions de la BOG. En pratique, on ne peut utiliser que des séries de la BOG

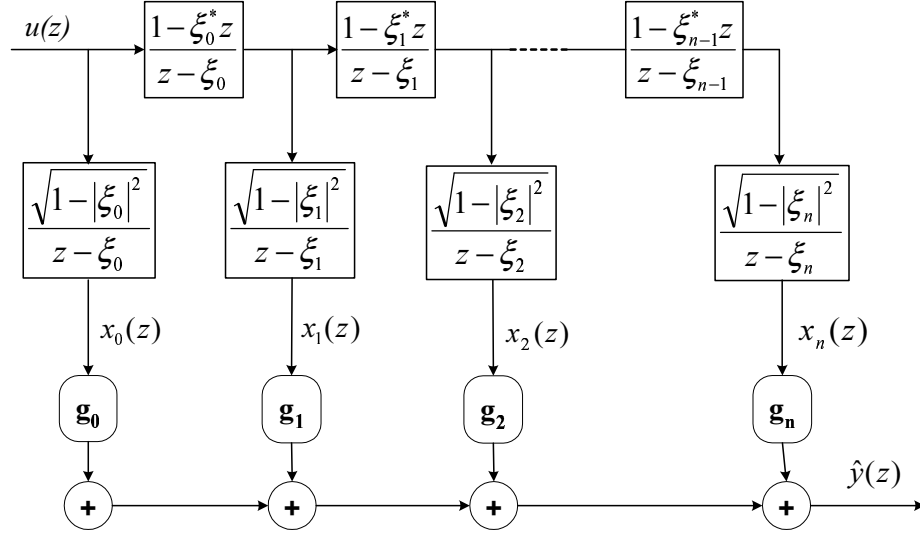


FIG. 2.3 – Réseau de filtres de la BOG

tronquées. Pour un système donné, l'erreur de troncature est fonction du nombre de filtres de la BOG utilisés et de leurs pôles correspondants. Pour un nombre fixé de filtres, il existe un ensemble de pôles de la BOG qui minimise l'erreur de troncature. Le problème est donc de trouver cet ensemble de pôles.

Nous commencerons par traiter le cas du pôle optimal de Laguerre, puis celui des deux pôles conjugués de Kautz pour terminer par le cas général de la BOG.

2.4.1 Choix du pôle de Laguerre

On rappelle que les fonctions discrètes de Laguerre dans le domaine Z sont définies par :

$$L_n(z, \xi) = \frac{\sqrt{1 - \xi^2}}{z - \xi} \left(\frac{1 - \xi z}{z - \xi} \right)^n \quad (2.42)$$

Toute fonction de transfert $H(z)$ d'un système linéaire échantillonné peut être exprimée à l'aide des fonctions de Laguerre en z comme suit :

$$H(z) = \sum_{n=0}^{+\infty} g_n L_n(z, \xi) \quad (2.43)$$

où les g_n , ($n = 0, 1, 2, \dots$) sont appelés les coefficients de Fourier et ξ ($|\xi| < 1$) est appelé le pôle de Laguerre. Soit $l_n(k, \xi)$ la transformée en z inverse de $L_n(z, \xi)$, *i.e.* le signal

temporel original associé à $L_n(z, \xi)$. Donc si $h(k)$ représente la réponse impulsionnelle d'un système stable, alors elle s'écrit comme suit :

$$h(k) = \sum_{n=0}^{+\infty} g_n l_n(k, \xi) \quad (2.44)$$

Algorithme de Fu et Dumont, 1993 [24] - Méthode 1

Dans cet algorithme, le pôle optimal de Laguerre est choisi en minimisant le critère suivant :

$$J = \sum_{n=0}^{+\infty} n g_n^2 \quad (2.45)$$

Ce critère croit d'une manière quadratique en fonction des coefficients de Fourier, ce qui accélère la convergence de la série de Laguerre.

Définissons :

$$\|h\|^2 = \sum_{k=0}^{\infty} h^2(k) = \sum_{n=1}^{\infty} g_n^2 \quad (2.46)$$

$$M_1 = \frac{1}{\|h\|^2} \sum_{k=0}^{\infty} k h^2(k) \quad (2.47)$$

$$M_2 = \frac{1}{\|h\|^2} \sum_{k=0}^{\infty} k [h(k) - h(k-1)]^2 \quad (2.48)$$

M_1 , M_2 et $\|h\|$ dépendent seulement du système discret à modéliser.

Les auteurs ont montré que :

$$J = \sum_{n=0}^{\infty} n g_n^2 = \frac{\xi M_2 + (1 - \xi)^2 M_1 + \xi - \xi^2}{1 - \xi^2} \|h\|^2 \quad (2.49)$$

Le pôle optimal de Laguerre ξ est choisi tel que le critère de performance :

$$J' = \frac{\sum_{n=0}^{+\infty} n g_n^2}{\|h\|^2} = \frac{\xi M_2 + (1 - \xi)^2 M_1 + \xi - \xi^2}{1 - \xi^2} \quad (2.50)$$

soit minimal.

Remarque : On note que les critères de performance J et J' sont équivalents.

On trouve :

$$\xi_{opt} = \frac{2M_1 - 1 - M_2}{2M_1 - 1 + \sqrt{4M_1 M_2 - M_2^2 - 2M_2}} \quad (2.51)$$

et

$$J_{min} = \frac{1 + \sqrt{4M_1 M_2 - M_2^2 - 2M_2}}{2} \quad (2.52)$$

Algorithme de Tanguy et al, 1995 [62] - Méthode 2

Cet algorithme est utilisé pour des systèmes non strictement causaux. Dans ce cas les fonctions de Laguerre sont définies par :

$$L_n(z, \xi) = \frac{\sqrt{1 - \xi^2} z}{z - \xi} \left(\frac{1 - \xi z}{z - \xi} \right)^n \quad (2.53)$$

Les auteurs prennent comme fonction de coût à minimiser :

$$J_N = \frac{\sum_{n=0}^{+\infty} n g_n^2}{N \|h\|^2} \quad (2.54)$$

J_N représente une borne supérieure de la fonction de coût J' utilisée par Fu et Dumont [24].

En définissant :

$$M_3 = \frac{1}{\|h\|^2} \sum_{k=0}^{\infty} k [h(k)h(k-1)] \quad (2.55)$$

ils démontrent que :

$$J_N = \frac{(1 + M_1)\xi^2 - 2M_3\xi + M_1}{N(1 - \xi^2)} \quad (2.56)$$

$\|h\|^2$ et M_1 ont été définis dans (2.46) et (2.47) et N est l'ordre de troncature du réseau de filtres de Laguerre.

En posant :

$$\rho \hat{=} \frac{2M_1 + 1}{2M_3} \quad (2.57)$$

les auteurs montrent que le pôle optimal au sens du critère (2.54), est une racine de l'équation $\xi^2 - 2\rho\xi + 1 = 0$. D'où :

$$\xi_{opt} = \begin{cases} \rho - \sqrt{\rho^2 - 1}, & \text{si } \rho \geq 1 \\ \rho + \sqrt{\rho^2 - 1}, & \text{si } \rho \leq -1 \end{cases} \quad (2.58)$$

Remarque : La fonction de coût minimisée aussi bien dans la méthode de Fu et Dumont que dans celle de Tanguy et al est une borne supérieure de l'erreur quadratique du système.

Méthode expérimentale pour déterminer le pôle de Laguerre - Méthode 3

Cette méthode consiste à discrétiser l'espace paramétrique $]-1, 1[$ associé au pôle de Laguerre en P valeurs possibles p_j ($j = 1, \dots, P$). Le pôle optimal ξ est choisi comme

la valeur p_j qui minimise l'erreur quadratique (EQ) entre la sortie du système et celle du modèle de Laguerre. Le critère EQ est défini comme suit :

$$EQ(p_j) = \frac{1}{L} \sum_{k=1}^L (y(k) - \hat{y}_j(k, p_j))^2 \quad (2.59)$$

où $y(k)$ représente la sortie du système, $\hat{y}_j(k, p_j)$ la sortie du modèle de Laguerre ayant p_j pour pôle et L le nombre de mesures utilisées. On a alors :

$$\xi_{opt} = \text{Arg min}_{p_j} (EQ(p_j)) \quad ; \quad j = 1, \dots, P \quad (2.60)$$

Sur la figure 2.4, le système étudié est du premier ordre de pôle égal à 0.8.

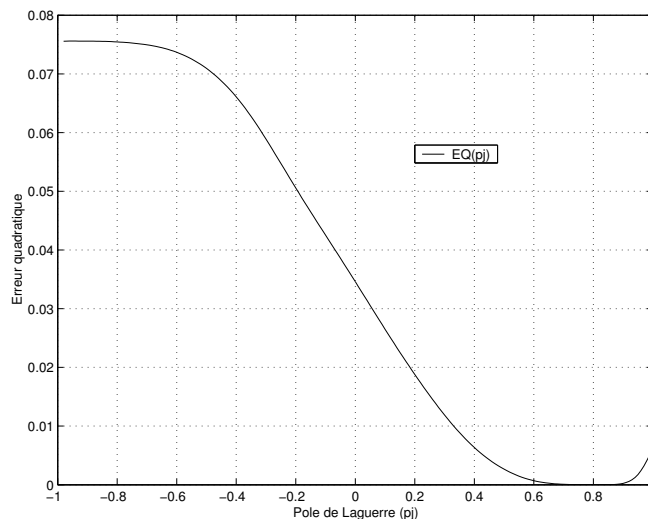


FIG. 2.4 – Détermination du pôle optimal de Laguerre

2.4.2 Choix des pôles conjugués de Kautz

Pour la détermination des pôles optimaux de Kautz, on utilise la deuxième définition qui exprime les fonctions de Kautz à partir des paramètres b et c qui dépendent exclusivement des pôles complexes conjugués de Kautz.

On discrétise l'espace des paramètres $(-1, 1)^2$ associé à la paire des paramètres (p_j, q_m) , $j, m = 1, \dots, P$ qui caractérisent selon la deuxième définition des fonctions de

Kautz, la paire de pôles complexes conjugués ξ_l et ξ_l^* comme suit :

$$p_j = \frac{\xi_l + \xi_l^*}{1 + \xi_l \xi_l^*} \quad (2.61)$$

$$q_m = -\xi_l \xi_l^* \quad (2.62)$$

avec $j, m = 1, \dots, P$ et $l = 1, \dots, P^2$

Les fonctions de Kautz sont définies comme suit :

$$\bar{\Psi}_{2p}(z, p_j, q_m) = \frac{\sqrt{1 - q_m^2}(z - p_j)}{z^2 + p_j(q_m - 1)z - q_m} \left[\frac{-q_m z^2 + p_j(q_m - 1)z + 1}{z^2 + p_j(q_m - 1)z - q_m} \right]^{p-1} \quad (2.63)$$

$$\bar{\Psi}_{2p+1}(z, p_j, q_m) = \frac{\sqrt{(1 - p_j^2)(1 - q_m^2)}}{z^2 + p_j(q_m - 1)z - q_m} \left[\frac{-q_m z^2 + p_j(q_m - 1)z + 1}{z^2 + p_j(q_m - 1)z - q_m} \right]^{p-1} \quad (2.64)$$

Les paramètres optimaux b_{opt} et c_{opt} sont choisis comme les valeurs de p_j et q_m qui minimisent l'erreur quadratique (EQ) entre la sortie du système et celle des modèles de Kautz caractérisés par les paramètres p_j et q_m comme il est illustré dans la figure 2.5.

$$EQ(p_j, q_m) = \frac{1}{L} \sum_{k=1}^L (y(k) - \hat{y}_{j,m}(k, p_j, q_m))^2 \quad (2.65)$$

où $y(k)$ représente la sortie du système, $\hat{y}_{j,m}(k, p_j, q_m)$ la sortie du modèle de Kautz pour les paramètres p_j et q_m et L le nombre de mesures utilisées. On a alors :

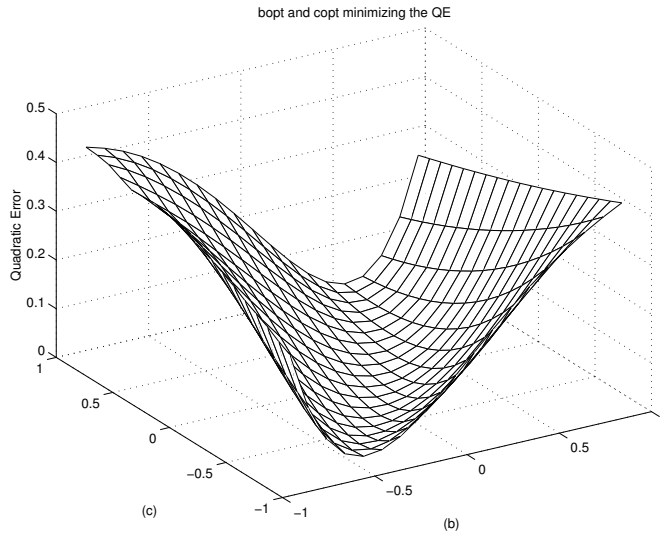
$$(b_{opt}, c_{opt}) = \text{Arg} \min_{(p_j, q_m)} (EQ(p_j, q_m)) ; \quad j, m = 1, \dots, P \quad (2.66)$$

Sur la figure 2.5, le système étudié est du second ordre défini par la fonction de transfert $H(z)$ suivante :

$$H(z) = \frac{0.7}{z^2 - 0.5z + 0.2} \quad (2.67)$$

2.4.3 Choix des pôles de la BOG

Contrairement aux bases de Laguerre et de Kautz, en utilisant les BOGs, on a plusieurs pôles à optimiser. Den Brinker et al [15] ont développé une méthode récursive pour déterminer les pôles d'une BOG. Malti et al [45] ont utilisé l'algorithme du gradient et l'algorithme de Gauss-Newton pour minimiser l'erreur quadratique de la sortie. Ces algorithmes garantissent seulement la convergence vers des minima locaux et si le pas d'adaptation de l'algorithme du gradient est mal choisi, alors la convergence est lente oscillante ou non garantie.

FIG. 2.5 – Détermination de b_{opt} et c_{opt} minimisant l'erreur quadratique

Dans ce qui suit, on présente différentes méthodes pour déterminer d'une manière itérative les pôles d'une BOG.

BOG à pôles réels - Méthodes 1 et 2

Rappelons que l'algorithme de Fu et Dumont [24] permet de trouver analytiquement le pôle optimal de Laguerre en minimisant une fonction de coût qui est une borne supérieure de l'erreur quadratique. Si le système est du premier ordre, le pôle de Laguerre coïncide avec le pôle du système. Si le système est d'ordre supérieur à un, alors le pôle de Laguerre approche la dynamique dominante du système.

Les deux méthodes proposées pour déterminer les pôles d'une BOG sont itératives. A chaque étape ($i = 1, \dots, N$), la procédure comporte deux sous étapes :

1. Déterminer la dynamique dominante $\xi_{i,opt}$ du signal résiduel de sortie $\nu_{i-1}(k)$ obtenu à l'itération $(i - 1)$. ν_0 étant :
 - La réponse impulsionnelle du système à modéliser, pour la première méthode.
 - La sortie du système pour une entrée quelconque, pour la deuxième méthode.
2. Filtrer le signal résiduel de sortie $\nu_{i-1}(k)$ avec le filtre suivant $F_i(q^{-1}) = 1 - \xi_{i,opt}q^{-1}$, qui délivre le signal résiduel $\nu_i(k)$.

Cette procédure est répétée jusqu'à ce que le signal résiduel devienne négligeable.

Une fois les pôles de la BOG déterminés, les coefficients de Fourier du développement du système sur cette BOG sont estimés en appliquant la méthode des moindres carrés.

Les deux méthodes diffèrent l'une de l'autre par le moyen de déterminer le pôle dominant estimé à chaque étape. Dans la méthode 1, le pôle optimal est déterminé à l'aide de l'algorithme de Fu et Dumont. Cette méthode est illustrée dans la figure 2.6. Dans la méthode 2, le pôle optimal est déterminé à l'aide d'une procédure expérimentale détaillée dans ce qui suit.

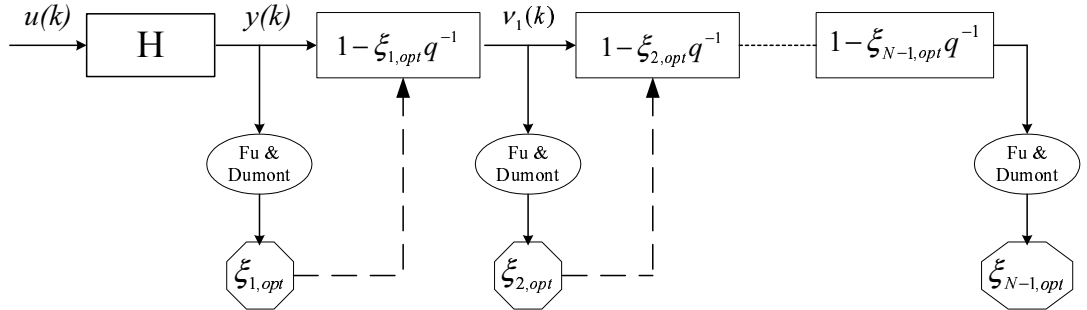


FIG. 2.6 – Détermination des dynamiques du système par la première méthode

Dans la méthode 2,

- D'abord, on initialise à N l'ordre du modèle en utilisant les informations a priori sur le système simulé.
- Ensuite, on discrétise l'intervalle $]-1, 1[$ associé au pôle de Laguerre en P valeurs possibles p_j ($j = 1, \dots, P$). Cette opération est illustrée dans la figure 2.4.
- A la $i^{\text{ème}}$ étape de la méthode d'estimation, on décrémente de 1 l'ordre du modèle de Laguerre. Le pôle optimal $\xi_{i,opt}$ est choisi comme la valeur p_j qui minimise l'erreur quadratique $EQ(p_j)$ entre le signal résiduel $\nu_{i-1}(k)$ et la sortie $\hat{\nu}_{i-1}(k, p_j) = \sum_{\tau=1}^L F_{i-1}(\tau, p_j) \hat{\nu}_{i-2}(k - \tau)$ des P modèles de Laguerre différents d'ordre $N - i + 1$, chacun caractérisé par son pôle $\xi_i = p_j$ ($j = 1, \dots, P$).

Ainsi, on a :

$$\xi_{i,opt} = \underset{p_j}{\text{Arg min}}(EQ(p_j)) \quad ; \quad j = 1, \dots, P \quad (2.68)$$

où $EQ(p_j)$ est l'erreur quadratique associée au modèle de Laguerre caractérisé par le pôle $\xi_i = p_j$. Cette méthode est illustrée dans la figure 2.7.

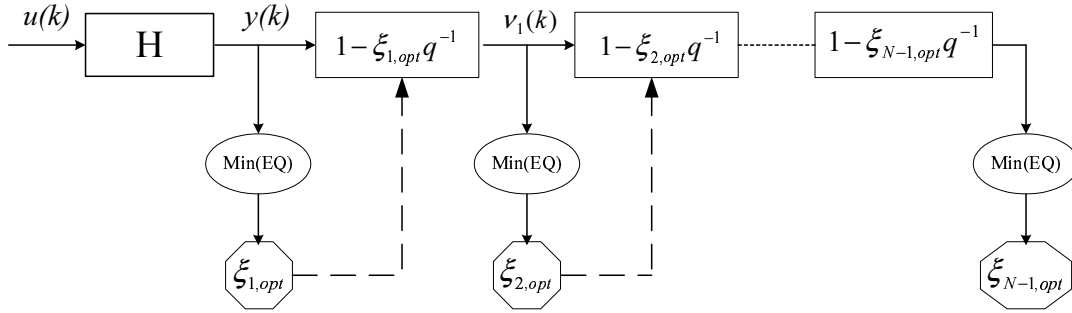


FIG. 2.7 – Détermination des dynamiques du système avec la deuxième méthode

BOG à pôles réels et complexes - Méthode 3

On peut étendre cette méthode pour déterminer des pôles complexes pour la BOG en utilisant la deuxième définition des fonctions de Kautz donnée par les équations (2.34) et (2.35). A chaque étape de l'algorithme, on détermine d'abord le pôle optimal $\xi_{i,opt}$ en utilisant la méthode 2. Ensuite, on détermine les paramètres $b_{i,opt}$ et $c_{i,opt}$ relatifs aux pôles complexes conjugués optimaux de Kautz. On procède comme suit : On discrétise l'espace des paramètres $(-1, 1]^2$ associé à la paire des paramètres (p_j, q_m) , $j, m = 1 \dots, P$ qui caractérisent la paire de pôles conjugués de Kautz ξ et ξ^* comme le montre la figure 2.5.

A la $i^{\text{ème}}$ étape, le signal résiduel de sortie $v_{i-1}(k)$ est filtré par :

- un filtre du premier ordre donné par :

$$F_i(\xi_{i,opt}, q^{-1}) = 1 - \xi_{i,opt} q^{-1}$$

dans le cas d'un pôle réel **ou**

- un filtre du second ordre donné par :

$$H_i(b_{i,opt}, c_{i,opt}, q^{-1}) = 1 + b_{i,opt}(c_{i,opt} - 1)q^{-1} - c_{i,opt}q^{-2}$$

dans le cas de pôles complexes conjugués, où $b_{i,opt}$ et $c_{i,opt}$ sont définis par l'équation (2.69).

Pour obtenir le signal résiduel $v_i(k)$, le filtre sélectionné est celui qui donne la plus petite erreur quadratique.

A la $(i + 1)^{\text{ème}}$ étape :

- D'abord, on décrémente de 1 ou de 2 l'ordre du modèle (1 si le filtre utilisé dans la $i^{\text{ème}}$ itération est du premier ordre et 2 s'il est du second ordre).
- Ensuite, on procède comme pour la méthode 2 utilisée pour les BOGs à pôles réels, pour décider de choisir un pôle réel $\xi_{(i+1),opt}$ ou deux pôles complexes conjugués caractérisés par le couple $(b_{(i+1),opt}, c_{(i+1),opt})$ pour définir le nouveau modèle de l'étape $(i + 1)$. Les paramètres $b_{(i+1),opt}$ et $c_{(i+1),opt}$ sont choisis comme les valeurs de p_j et q_m qui minimisent l'Erreur Quadratique $EQ(p_j, q_m)$ entre le signal résiduel $v_i(k)$ et la sortie $\hat{v}_i(k, p_j, q_m) = \sum_{\tau=1}^L H_i(\tau, p_j, q_m) \hat{v}_{i-1}(k - \tau)$ des P^2 modèles de Kautz caractérisés chacun par $(b_{i+1}, c_{i+1}) = (p_j, q_m)$, $(j, m = 1, \dots, P)$. On obtient :

$$(b_{i+1,opt}, c_{i+1,opt}) = \text{Arg} \min_{(p_j, q_m)} (EQ(p_j, q_m)) ; \quad j, m = 1, \dots, P \quad (2.69)$$

Cette méthode est illustrée dans la figure 2.8.

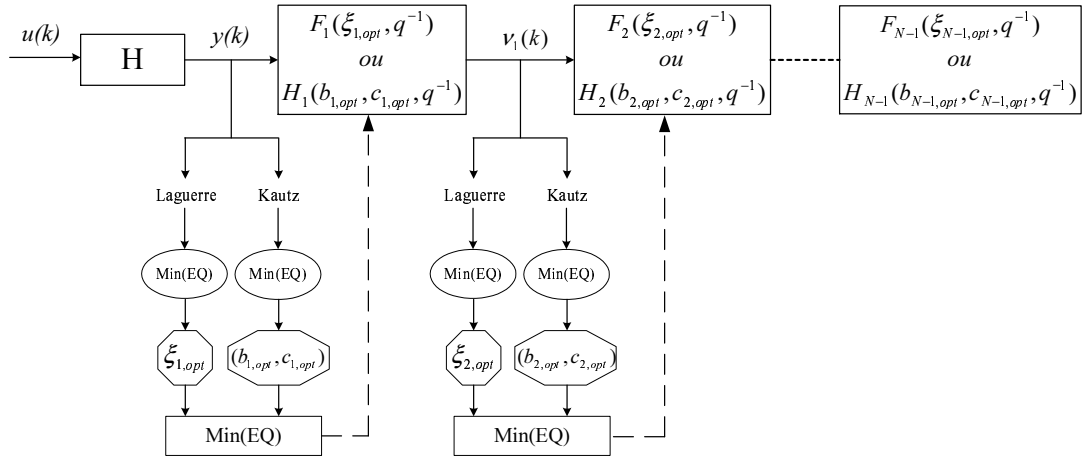


FIG. 2.8 – Détermination des dynamiques du système dans le cas général

2.4.4 Convergence des algorithmes d'estimation des pôles de la BOG

Les méthodes présentées dans le paragraphe précédent sont itératives. Nous apportons ici la preuve de leur convergence (A. Khouaja et al [34]).

La sortie d'un modèle issu d'une BOG caractérisée par i pôles comme le montre la figure 2.9 s'écrit comme suit :

$$\hat{Y} = \mathcal{X}_i \mathcal{G}_i \quad (2.70)$$

où

$$\mathcal{X}_i = \begin{bmatrix} x_0(1) & \cdots & x_{i-1}(1) & x_i(1) \\ \vdots & & \vdots & \vdots \\ x_0(L) & \cdots & x_{i-1}(L) & x_i(L) \end{bmatrix} = \begin{bmatrix} \mathcal{X}_{i-1} & \mathbf{x}_i \end{bmatrix}; \quad (2.71)$$

$$\hat{Y} = \begin{bmatrix} \hat{y}(1) \\ \vdots \\ \hat{y}(L) \end{bmatrix} \text{ et } \mathcal{G}_i = \begin{bmatrix} g_0 \\ \vdots \\ g_i \end{bmatrix} \quad (2.72)$$

Les $\{x_l\}_{l=0,\dots,i}$ représentent les sorties des fonctions de la BOG, $\{g_l\}_{l=0,\dots,i}$ les coefficients de Fourier du développement et L le nombre des sorties mesurées.

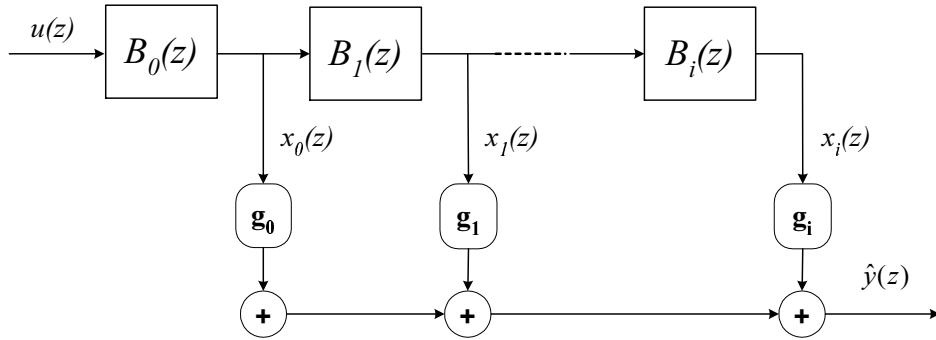


FIG. 2.9 – Réseau de filtres de la BOG

L'erreur quadratique entre la sortie du système et celle du modèle issu de la BOG caractérisée par i pôles s'écrit comme suit :

$$E_{min,i}^2 = (Y - \hat{Y})^T (Y - \hat{Y}) \quad (2.73)$$

$$= (Y - \mathcal{X}_i \mathcal{G}_{i,opt})^T (Y - \mathcal{X}_i \mathcal{G}_{i,opt}) \quad (2.74)$$

La solution de $Y = \mathcal{X} \mathcal{G}_{opt}$ au sens des moindres carrés s'écrit :

$$\mathcal{G}_{opt} = (\mathcal{X}^T \mathcal{X})^{-1} \mathcal{X}^T Y \quad (2.75)$$

L'équation (2.74) devient alors :

$$E_{min,i}^2 = (Y - \mathcal{X}_i (\mathcal{X}_i^T \mathcal{X}_i)^{-1} \mathcal{X}_i^T Y)^T (Y - \mathcal{X}_i (\mathcal{X}_i^T \mathcal{X}_i)^{-1} \mathcal{X}_i^T Y) \quad (2.76)$$

$$\begin{aligned} &= Y^T Y - 2Y^T \mathcal{X}_i (\mathcal{X}_i^T \mathcal{X}_i)^{-1} \mathcal{X}_i^T Y \\ &+ Y^T \underbrace{\mathcal{X}_i (\mathcal{X}_i^T \mathcal{X}_i)^{-1} \mathcal{X}_i^T \mathcal{X}_i (\mathcal{X}_i^T \mathcal{X}_i)^{-1} \mathcal{X}_i^T}_{I_i} Y \end{aligned} \quad (2.77)$$

$$= Y^T Y - Y^T \mathcal{X}_i (\mathcal{X}_i^T \mathcal{X}_i)^{-1} \mathcal{X}_i^T Y \quad (2.78)$$

Pour vérifier la convergence de l'algorithme de sélection des pôles de la BOG, on calcule :

$$E_{min,i}^2 - E_{min,i+1}^2 = Y^T (\mathcal{X}_{i+1}(\mathcal{X}_{i+1}^T \mathcal{X}_{i+1})^{-1} \mathcal{X}_{i+1}^T - \mathcal{X}_i(\mathcal{X}_i^T \mathcal{X}_i)^{-1} \mathcal{X}_i^T) Y \quad (2.79)$$

En définissant

$$\Xi_{i+1} = (\mathcal{X}_{i+1}^T \mathcal{X}_{i+1})^{-1} \quad (2.80)$$

et la matrice de projection P_i^\perp

$$P_i^\perp = I - \mathcal{X}_i \Xi_i \mathcal{X}_i^T \quad (2.81)$$

et en utilisant la formule d'inversion d'une matrice partitionnée [21], on peut écrire

$$\Xi_{i+1} = \begin{bmatrix} \Xi_i + \frac{\Xi_i \mathcal{X}_i^T \mathbf{x}_{i+1} \mathbf{x}_{i+1}^T \mathcal{X}_i \Xi_i}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} & \frac{\Xi_i \mathcal{X}_i^T \mathbf{x}_{i+1}}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} \\ \frac{\mathbf{x}_{i+1}^T \mathcal{X}_i \Xi_i}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} & \frac{1}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} \end{bmatrix} \quad (2.82)$$

et

$$\begin{aligned} \mathcal{X}_{i+1}(\mathcal{X}_{i+1}^T \mathcal{X}_{i+1})^{-1} \mathcal{X}_{i+1}^T &= \mathcal{X}_{i+1} \begin{bmatrix} \Xi_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathcal{X}_{i+1}^T \\ &+ \mathcal{X}_{i+1} \begin{bmatrix} \frac{\Xi_i \mathcal{X}_i^T \mathbf{x}_{i+1} \mathbf{x}_{i+1}^T \mathcal{X}_i \Xi_i}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} & \frac{\Xi_i \mathcal{X}_i^T \mathbf{x}_{i+1}}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} \\ \frac{\mathbf{x}_{i+1}^T \mathcal{X}_i \Xi_i}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} & \frac{1}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} \end{bmatrix} \mathcal{X}_{i+1}^T \quad (2.83) \end{aligned}$$

$$\begin{aligned} &= \mathcal{X}_i(\mathcal{X}_i^T \mathcal{X}_i)^{-1} \mathcal{X}_i^T \\ &+ \mathcal{X}_{i+1} \begin{bmatrix} \frac{\Xi_i \mathcal{X}_i^T \mathbf{x}_{i+1} \mathbf{x}_{i+1}^T \mathcal{X}_i \Xi_i}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} & \frac{\Xi_i \mathcal{X}_i^T \mathbf{x}_{i+1}}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} \\ \frac{\mathbf{x}_{i+1}^T \mathcal{X}_i \Xi_i}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} & \frac{1}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} \end{bmatrix} \mathcal{X}_{i+1}^T \quad (2.84) \end{aligned}$$

L'équation (2.79) devient alors :

$$E_{min,i}^2 - E_{min,i+1}^2 = \frac{1}{\mathbf{x}_{i+1}^T P_i^\perp \mathbf{x}_{i+1}} Y^T \left(\mathcal{X}_{i+1} \begin{bmatrix} \Xi_i \mathcal{X}_i^T \mathbf{x}_{i+1} \\ 1 \end{bmatrix} \right) \left(\begin{bmatrix} \Xi_i \mathcal{X}_i^T \mathbf{x}_{i+1} \\ 1 \end{bmatrix}^T \mathcal{X}_{i+1}^T \right) Y \quad (2.85)$$

$$\geq 0 \quad (2.86)$$

Donc l'Erreur Quadratique pour un modèle issu de la BOG d'ordre $i+1$ est inférieure à celle d'un modèle d'ordre i et défini par les i premiers pôles du premier modèle. On peut alors conclure que les méthodes itératives d'estimation des pôles de la BOG sont convergentes.

2.4.5 Résultats de simulations

• Exemple 1

• Détermination des pôles de la BOG

Dans ce paragraphe, on va comparer les différentes méthodes de détermination des pôles réels d'une BOG. Le système simulé est d'ordre 3. Il est défini par sa fonction de transfert $G(z)$.

$$G_1(z) = 0.0017 \frac{z^{-1}(1 + 0.673z^{-1})}{(1 - 0.368z^{-1})(1 - 0.819z^{-1})(1 - 0.995z^{-1})} \quad (2.87)$$

Pour déterminer les pôles, l'algorithme de Fu et Dumont utilise la réponse impulsionnelle du système. Tandis que la méthode 2 expérimentale utilise un signal quelconque.

Le signal d'entrée est une séquence de bruit blanc gaussien de moyenne nulle et de variance l'unité. Le Rapport Signal sur Bruit (RSB) est fixé à 20dB.

L'ordre de troncature du développement en série de fonctions issues de la BOG optimisée est fixé à $N=6$. Les résultats de simulations sont obtenus en utilisant la méthode de Monte Carlo avec 50 séquences différentes de bruit additif. Le pas de discrétisation de la deuxième méthode est de 10^{-3} .

Les performances des méthodes 1 et 2 d'identification sont comparées en termes de pôles identifiés. Dans le tableau 2.1, on donne pour le système simulé, les vraies dynamiques du système ainsi que les pôles identifiés par les deux méthodes définies dans le paragraphe précédent. Les pôles identifiés sont donnés avec leur écart-type σ correspondant. Ils sont représentés dans la figure 2.10.

En analysant les résultats du tableau 2.1 et de la figure 2.10, on peut conclure que les deux méthodes estiment des valeurs proches des vraies dynamiques du système simulé avec des écart-types relativement petits.

La performance des méthodes d'identification est aussi évaluée en termes d'erreur

	Vrais pôles	Méthode1	Méthode2
P 1	0.995	0.922	0.992
$\sigma(\text{P1})$		$9.9762 \cdot 10^{-4}$	$4.0406 \cdot 10^{-3}$
P 2	0.819	0.844	0.746
$\sigma(\text{P2})$		$3.038 \cdot 10^{-2}$	$1.1294 \cdot 10^{-2}$
P 3	0.368	0.308	0.416
$\sigma(\text{P3})$		$3.0432 \cdot 10^{-2}$	$1.3235 \cdot 10^{-1}$
P 4	0	0.190	0.217
$\sigma(\text{P4})$		$3.0041 \cdot 10^{-2}$	$8.2503 \cdot 10^{-2}$
P 5	0	0.117	0.112
$\sigma(\text{P5})$		$3.0122 \cdot 10^{-2}$	$8.7366 \cdot 10^{-2}$
P 6	0	0.075	0.052
$\sigma(\text{P6})$		$3.7851 \cdot 10^{-2}$	$8.3033 \cdot 10^{-2}$

TAB. 2.1 – Pôles de la BOG estimés par les méthodes 1 et 2

quadratique calculée comme suit :

$$EQ(L) = \frac{1}{50} \sum_{l=1}^{50} \left(\frac{1}{L} \sum_{k=1}^L (y_l(k) - \hat{y}_l(k))^2 \right) \quad (2.88)$$

où $y_l(k)$ représente la sortie du système à la $l^{\text{ème}}$ simulation de Monte Carlo, $\hat{y}_l(k)$ est la sortie correspondante du modèle basé sur un développement sur une BOG et L représente le nombre des sorties mesurées utilisées pour l'identification. Dans le tableau 2.2, on donne les erreurs quadratiques entre la sortie du système et les différentes sorties des modèles de BOGs caractérisés par les différents pôles identifiés. Ces valeurs d'erreurs quadratiques sont relativement proches ce qui peut s'expliquer par le surdimensionnement des modèles.

	$EQ(L)$
Vrais pôles	$7.630 \cdot 10^{-5}$
Pôles estimés par la méthode 1	$1.640 \cdot 10^{-4}$
Pôles estimés par la méthode 2	$1.693 \cdot 10^{-4}$

TAB. 2.2 – $EQ(L)$ obtenue avec les différentes BOG

Dans la figure 2.11, sont représentés les coefficients de Fourier estimés $\{g_i\}_{i=0,\dots,5}$. Ils décroissent en valeur absolue jusqu'à ce qu'il deviennent relativement négligeables.

Dans le tableau 2.3, on présente le temps de calcul d'un pôle de la BOG par les deux méthodes utilisées.

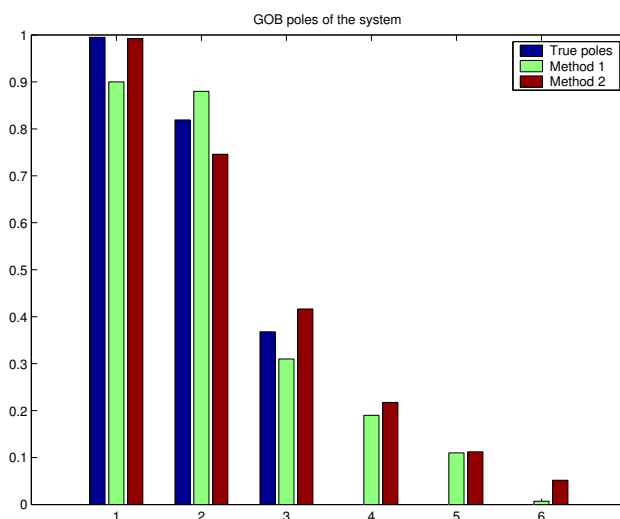


FIG. 2.10 – Vraies dynamiques du système simulé et les pôles de la BOG estimés par les méthodes 1 et 2

	Temps de calcul en secondes
Algorithme de Fu et Dumont	$6.19 \cdot 10^{-3}$
Algorithme par discrétisation	8.22

TAB. 2.3 – Temps de calcul d'un pôle de la BOG

Remarques

- La méthode 1 basée sur des formules analytiques, donne des résultats satisfaisants pour des systèmes à dynamiques rapprochées. Cependant, plus les dynamiques du système sont éloignées les unes des autres ou lorsque le système devient plus oscillant, moins les résultats de l'identification sont bons et la méthode 2 devient plus adaptée et plus précise pour l'identification des pôles.
- La méthode 1 est plus rapide que la méthode 2 comme le montre le tableau 2.3. Le temps de calcul pour déterminer un seul pôle de la BOG en utilisant l'algorithme de Fu et Dumont est nettement plus court que le temps de calcul nécessaire pour déterminer le même pôle en utilisant un algorithme itératif (à P pas) nécessitant à chaque pas une modélisation de Laguerre du système. (P représente aussi le pas de discrétisation de l'intervalle paramétrique $] - 1, 1[$). Cependant la méthode 1 utilise la réponse impulsionnelle du système qui n'est pas toujours accessible.

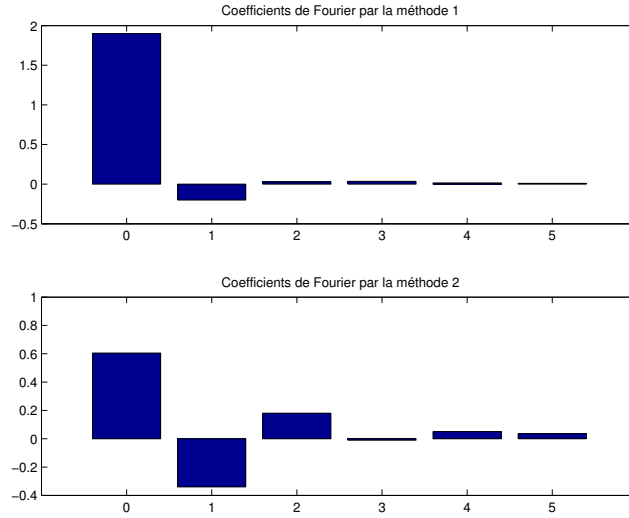


FIG. 2.11 – Coefficients de Fourier $\{g_i\}_{i=0,\dots,5}$ estimés avec la méthode 1 et la méthode 2

• Apport des BOG

Dans cette partie, on va montrer l'apport des BOG à pôles réels du point de vue réduction de la complexité numérique. Pour cela, nous comparons la représentation du système à l'aide de la BOG avec sa représentation à l'aide de la base de fonctions de Laguerre et de la base de filtres FIR.

Le critère de performance utilisé est l'Erreur Quadratique Moyenne Normalisée $EQMN$ entre la sortie du système et celle du modèle utilisé. Elle s'écrit comme suit :

$$EQMN(L) = \frac{1}{L} \frac{\sum_{k=1}^L (y(k) - \hat{y}(k))^2}{\sum_{k=1}^L (y(k))^2} \quad (2.89)$$

où $y(k)$ représente la sortie du système, $\hat{y}(k)$ celle du modèle et L représente le nombre des sorties mesurées utilisées pour l'identification.

La figure 2.12 représente l'évolution des erreurs quadratiques moyennes normalisées entre la sortie du système et celles des modèles RIF, Laguerre et BOG à pôles réels. Le pôle de Laguerre est égal au pôle optimal $\xi_{opt} = 0.982$, calculé à l'aide de l'algorithme de Fu et Dumont [24]. Les pôles de la BOG sont réels et ont été déterminés à l'aide de l'algorithme qu'on a mis au point [33] basé sur le filtrage ($P_0 = 0.992$, $P_1 = 0.811$,

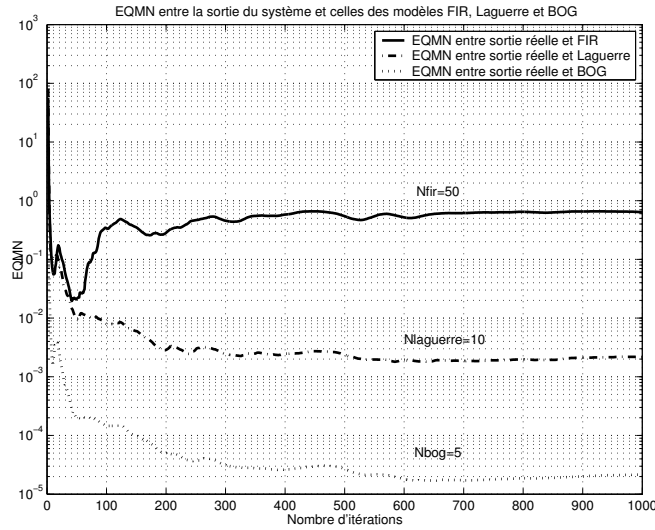


FIG. 2.12 – EQMN entre la sortie du système 1 et celles des modèles RIF ($N=50$), Laguerre ($N=10$) et BOG ($N=5$)

$P_2 = 0.470$, $P_3 = 0.244$ et $P_4 = 0.131$). Le signal d'entrée est une séquence de bruit blanc gaussien de moyenne nulle et de variance l'unité. Le rapport signal sur bruit est fixé à $RSB = 30dB$.

L'utilisation d'un modèle optimal basé sur les BOG avec un ordre de troncature $N_{BOG} = 5$ donne une valeur de l'erreur quadratique moyenne normalisée ($EQMN_{BOG} = 2.25 \cdot 10^{-5}$) nettement inférieure à celle donnée par les modèles basés sur les bases de fonctions de Laguerre avec un ordre de troncature $N_{Laguerre} = 10$ ($EQMN_{Laguerre} = 2.18 \cdot 10^{-3}$) ou les modèles RIF avec un ordre de troncature $N_{FIR} = 50$ ($EQMN_{FIR} = 6.38 \cdot 10^{-1}$).

⇒ L'utilisation des BOG permet une meilleure réduction de complexité du modèle ainsi qu'une meilleure modélisation des systèmes par rapport aux modèles basés sur les bases de fonctions de Laguerre et surtout par rapport aux modèles RIF.

• Exemple 2

Dans ce paragraphe, le système simulé dont la réponse impulsionnelle est représentée dans la figure 2.13, est d'ordre 5. Il est défini par sa fonction de transfert $G_2(z)$.

$$G_2(z) = \frac{b_1(z)b_2(z)}{(z - 0.5)[a(z)]^2} \quad (2.90)$$

avec

$$a(z) = z^2 - 2r \cos(\varphi)z + r^2 \quad (2.91)$$

$$b_1(z) = z^2 - 2r \cos(\varphi + \Delta\varphi)z + r^2 \quad (2.92)$$

$$b_2(z) = z^2 - 2r \cos(\varphi - \Delta\varphi)z + r^2 \quad (2.93)$$

$$\varphi = 1.3\pi/4 \quad (2.94)$$

$$\Delta\varphi = 0.2\pi/4 \quad (2.95)$$

$$r = 0.9 \quad (2.96)$$

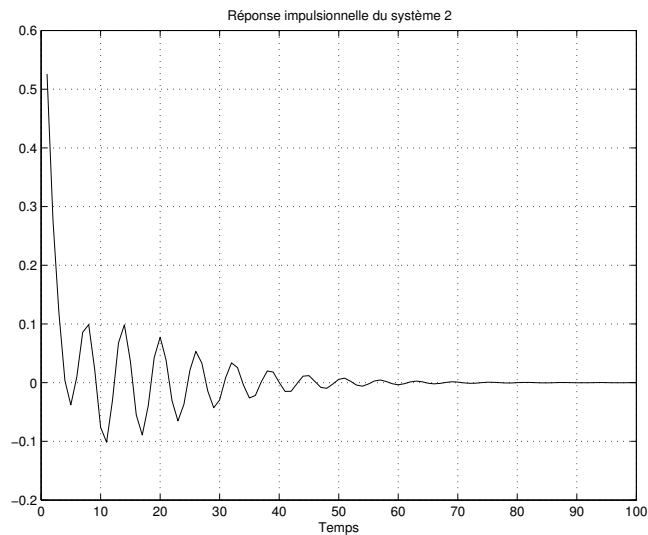


FIG. 2.13 – Réponse impulsionnelle du système 2

Apport des BOG

Pour ce système qui a des dynamiques complexes, on va comparer sa modélisation utilisant les BOG à pôles complexes et réels avec celles utilisant les bases de filtres Kautz et RIF.

Les pôles complexes conjugués de Kautz sont calculés à l'aide de l'algorithme qu'on a mis au point [33] basé sur la minimisation de l'erreur quadratique entre la sortie du système et celle du modèle de Kautz. Les pôles de Kautz trouvés sont $\xi, \xi^* = 0.48 \pm j 0.61$. Les pôles optimaux de la BOG sont réels et complexes et ont été déterminés à l'aide de l'algorithme 3 détaillé précédemment.

La figure 2.14 représente l'évolution des erreurs quadratiques moyennes normalisées entre la sortie du système et celles des modèles RIF, Kautz et BOG à pôles réels et complexes.

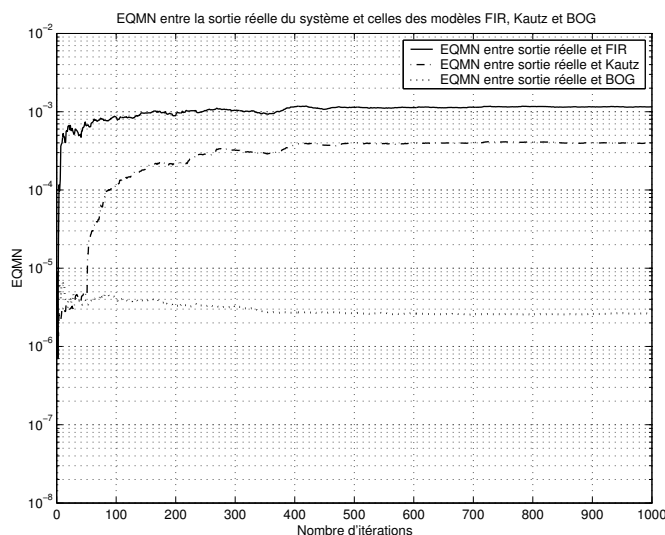


FIG. 2.14 – EQMN entre la sortie du système 2 et celles des modèles RIF (N=50), Kautz (N=10) et BOG (N=7)

L'analyse de la figure 2.14 nous permet de conclure une fois de plus que l'utilisation des BOG permet une meilleure réduction de la complexité du modèle ainsi qu'une meilleure modélisation des systèmes par rapport aux modèles basés sur les bases de fonctions de Kautz et surtout par rapport aux modèles RIF.

2.5 Introduction des systèmes de Volterra

Le modèle de Volterra est un modèle non-linéaire par rapport à l'entrée du système qui présente l'avantage d'être linéaire par rapport à ses paramètres. Ceci permet l'utilisation de plusieurs résultats obtenus pour l'estimation paramétrique de modèles linéaires (Billings [3], Schetzen [57]). Cependant la représentation paramétrique du modèle de Volterra nécessite un nombre de paramètres élevés, ce qui limite l'utilisation de ce modèle dans des applications pratiques.

Dans cette section, nous présentons le modèle de Volterra classique pour étudier ensuite un modèle particulier de Volterra dit à complexité réduite. Ce dernier est basé

sur le développement des différents noyaux de Volterra sur des bases de fonctions orthonormales.

2.5.1 Généralités

Le modèle de Volterra est un modèle polynômial, non récursif dont la relation entrée/sortie à temps discret est donnée par :

$$y(k) = h_0 + \sum_{i=1}^{\infty} H_i [u(k)] \quad (2.97)$$

où $u(k)$ et $y(k)$ sont respectivement l'entrée et la sortie du système et $H_i [u(k)]$ le terme homogène d'ordre i défini par :

$$H_i [u(k)] = \sum_{m_1=-\infty}^{+\infty} \sum_{m_2=-\infty}^{+\infty} \cdots \sum_{m_i=-\infty}^{+\infty} h_i(m_1, \dots, m_i) \prod_{p=1}^i u(k - m_p) \quad (2.98)$$

Le terme $h_i(m_1, \dots, m_i)$ est le noyau d'ordre i du système.

Boyd et Chua [4] ont montré que tout système non-linéaire, causal, invariant dans le temps et de mémoire finie peut être représenté par un développement en série de Volterra. Le système causal d'ordre P et de mémoire M s'écrit :

$$y(k) = h_0 + \sum_{i=1}^P \sum_{m_1=0}^{M-1} \cdots \sum_{m_i=0}^{M-1} h_i(m_1, \dots, m_i) \prod_{p=1}^i u(k - m_p) \quad (2.99)$$

Ce modèle devant représenter un système physique, il doit donc vérifier la condition de causalité suivante :

$$h_i(m_1, \dots, m_i) = 0 \text{ si } \exists m_j; j = 1, \dots, i \text{ tel que } m_j < 0 \quad (2.100)$$

Il est dit stable au sens entrée bornée - sortie bornée (Bounded Input Bounded Output : BIBO) si et seulement si ses noyaux sont absolument sommables. Il vient donc :

$$\sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} \cdots \sum_{m_i=0}^{M-1} |h_i(m_1, m_2, \dots, m_i)| < \infty; \forall i = 1, \dots, P \quad (2.101)$$

Les modèles de Volterra sont très utiles dans la représentation des systèmes grâce à leur structure non-linéaire et leur propriété de linéarité par rapport aux coefficients du noyau . Cependant, quand un modèle de Volterra d'ordre élevé est nécessaire pour une

application, même avec mémoire courte, le nombre de coefficients du noyau reste élevé. Un filtre de Volterra de mémoire M et d'ordre P a un nombre de paramètres égal à :

$$\sum_{i=1}^P M^i \quad (2.102)$$

Toutefois, on peut constater que certains paramètres différents utilisent les mêmes produits des entrées $u(k)$. On peut donc par construction se ramener à un modèle non redondant en regroupant les termes produits identiques. On obtient alors une forme triangulaire. La réduction du nombre de paramètres est très significative puisqu'il est réduit à :

$$\sum_{i=1}^P \frac{(M-1+i)!}{(M-1)!i!} \quad (2.103)$$

Ainsi, un filtre de Volterra triangulaire a ses termes homogènes donnés par :

$$H_i [u(k)] = \sum_{m_1=0}^{M-1} \sum_{m_2=m_1}^{M-1} \cdots \sum_{m_i=m_{i-1}}^{M-1} h_i^{tri}(m_1, m_2, \dots, m_i) \prod_{p=1}^i u(k - m_p) \quad (2.104)$$

D'autres solutions sont proposées dans la littérature pour réduire la complexité paramétrique des modèles de Volterra. Parmi elles on cite :

- Le réarrangement des coefficients du noyau dans des matrices qui sont décomposées en appliquant une décomposition en valeurs singulières (SVD) qui donne une réalisation parallèle-cascade à complexité réduite du filtre de Volterra (Panicker et Mathews [55]).
- Le développement du noyau sur une base orthonormale comme les bases de fonctions de Laguerre (Campello et al [11], Dumont et Fu [18]) ou les Bases Orthogonales Généralisées (BOG) (Favier et al [22], Kibangou et al [36]).

2.5.2 Développement des noyaux de Volterra à l'aide des BOG

Afin de réduire la complexité paramétrique d'un modèle de Volterra, une approche intéressante a été présentée par Wiener [70]. Elle consiste à développer les noyaux de Volterra en utilisant des bases de fonctions orthonormales comme celles de Laguerre ou de Kautz (Broome [8], Wahlberg [66], Wahlberg [67], Wahlberg et Mäkilä [68]) ou en utilisant des BOGs (Heuberger et al [30], Ninness et Gustafsson [53]).

Les bases de fonctions de Laguerre ont la propriété de dépendre exclusivement d'un seul paramètre, le pôle de Laguerre. Grâce à cette propriété, les modèles de Laguerre ont

été beaucoup utilisés dans le contexte de l'identification des systèmes et de la commande (Marmarelis [47], Dumont et Fu [18], Costa-Oliveira et al [14]). La sélection du pôle optimal de Laguerre a été d'abord proposée par Clowes [13] dans le cas des systèmes linéaires à temps continu. Ensuite, Fu et Dumont [24] ont traité le problème dans le cas discret. Dans le contexte des systèmes non linéaires, Campello et al [10] ont généralisé l'approche de Fu et Dumont pour les modèles de Volterra quadratiques. Ils ont proposé l'utilisation de deux bases de Laguerre indépendantes pour la représentation des noyaux, linéaire et quadratique. Ils ont ensuite généralisé leur résultat [9] au cas des systèmes de Volterra d'ordre quelconque.

• **Cas linéaire quadratique (Fu et Dumont [24])**

Considérons le modèle de Volterra discret du second ordre dont la sortie est la suivante :

$$y(n) = \sum_{n_1=0}^{\infty} h_1(n_1)u(n-n_1) + \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} h_2(n_1, n_2)u(n-n_1)u(n-n_2) \quad (2.105)$$

Supposons que le modèle soit stable, ce qui implique que ses noyaux soient sommables sur $[0; +\infty[$. Ce modèle peut être développé sur deux BOGs différentes. Chacune pour un noyau de Volterra différent d'où :

$$h_1(n_1) = \sum_{k_1=0}^{\infty} g_{k_1} l_{k_1}(n_1) \quad (2.106)$$

$$h_2(n_1, n_2) = \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \alpha_{k_1, k_2} \phi_{k_1}(n_1) \phi_{k_2}(n_2) \quad (2.107)$$

avec les coefficients de Fourier donnés par :

$$g_{k_1} = \sum_{n_1=0}^{\infty} h_1(n_1) l_{k_1}(n_1) \quad (2.108)$$

$$\alpha_{k_1, k_2} = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} h_2(n_1, n_2) \phi_{k_1}(n_1) \phi_{k_2}(n_2) \quad (2.109)$$

En utilisant les équations (2.106) et (2.107), le modèle (2.105) donne :

$$y(n) = \underbrace{\sum_{k_1=0}^{\infty} g_{k_1} \bar{l}_{k_1}(n)}_{y_1(n)} + \underbrace{\sum_{k_1=0}^{\infty} \sum_{k_2=1}^{\infty} \alpha_{k_1, k_2} \bar{\phi}_{k_1}(n) \bar{\phi}_{k_2}(n)}_{y_2(n)} \quad (2.110)$$

- $y_1(n)$ est la partie linéaire de la sortie du système de Volterra et $y_2(n)$ la partie quadratique.
- \bar{l}_i et $\bar{\phi}_j$ sont respectivement la sortie du filtre d'ordre i de la première BOG et la sortie du filtre d'ordre j de la deuxième BOG. Elles sont données par :

$$\bar{l}_i(n) = \sum_{\tau=0}^{\infty} l_i(\tau)u(n-\tau) \quad (2.111)$$

$$\bar{\phi}_j(n) = \sum_{\tau=0}^{\infty} \phi_j(\tau)u(n-\tau) \quad (2.112)$$

• Cas général

Considérons le modèle de Volterra d'ordre P et de mémoire M dont la relation entrée/sortie est la suivante :

$$y(n) = \sum_{p=1}^P \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} \cdots \sum_{n_p=0}^{M-1} h_p(n_1, \dots, n_p) \prod_{j=1}^p u(n-n_j) \quad (2.113)$$

Afin de réduire la complexité paramétrique de ce modèle, nous développons chacun des noyaux sur une BOG.

$$h_p(n_1, \dots, n_p) = \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} \cdots \sum_{k_p=0}^{K-1} g_{k_1, \dots, k_p} b_{k_1}(n_1) \cdots b_{k_p}(n_p) \quad (2.114)$$

où K est l'ordre de troncature du développement du modèle sur la BOG et b_k une fonction de la BOG dont la transformée en z s'écrit comme suit :

$$B_k(z) = \frac{\sqrt{1-|\xi_k|^2}}{z-\xi_k} \prod_{i=0}^{k-1} \frac{1-\xi_i^* z}{z-\xi_i} \quad (2.115)$$

D'après l'équation (2.114), le modèle résultant devient :

$$y(n) = \sum_{p=1}^P \sum_{k_1=0}^{K-1} \cdots \sum_{k_p=0}^{K-1} g_{k_1, \dots, k_p} \prod_{j=1}^p x_{k_j}(n) \quad (2.116)$$

où les x_k sont les entrées filtrées telles que :

$$x_k(n) = \sum_{i=0}^{M-1} b_k(i)u(n-i) \quad (2.117)$$

La sortie du modèle peut s'écrire sous la forme suivante :

$$y(n) = \mathcal{G}^T \mathcal{X}(n) \quad (2.118)$$

avec

$$\mathcal{G} = [g_0 \cdots g_{K-1} \ g_{0,0} \ g_{0,1} \cdots g_{K-1,K-1} \ \cdots \cdots] ^T \quad (2.119)$$

$$\mathcal{X}(n) = \left[\underbrace{x_0(n) \cdots x_{K-1}(n)}_{\text{termes linéaires}} \underbrace{x_0(n)^2 \ x_0(n)x_1(n) \cdots x_{K-1}(n)^2}_{\text{termes quadratiques}} \underbrace{\cdots \cdots \cdots}_{\text{termes d'ordres } > 2} \right]^T \quad (2.120)$$

$g_0, \dots, g_{K-1}, g_{0,0}, \dots, g_{K-1,K-1}, \dots$ sont les coefficients de Fourier issus du développement sur la BOG.

2.6 Représentation dans l'espace d'état

2.6.1 Modèle de Laguerre sous forme d'équation d'état

La modélisation des systèmes linéaires sur la base des filtres de Laguerre admet une représentation dans l'espace d'état utilisée fréquemment dans la commande des systèmes.

La représentation d'état, que l'on retrouve fréquemment dans la littérature, correspond à un réseau de filtres de Laguerre strictement causal. Un retard pur peut être rajouté en amont du réseau de la figure 2.1. Ceci n'altère pas la propriété d'orthogonalité des filtres de Laguerre. On peut établir ensuite les équations d'état discrètes suivantes, en prenant les sorties des filtres comme variables d'état.

$$\begin{cases} x_0(k+1) = \xi x_0(k) + \sqrt{1-\xi^2}u(k) \\ x_1(k+1) + \xi x_0(k+1) = x_0(k) + \xi x_1(k) \\ x_2(k+1) + \xi x_1(k+1) = x_1(k) + \xi x_2(k) \\ \vdots \\ x_n(k+1) + \xi x_{n-1}(k+1) = x_{n-1}(k) + \xi x_n(k) \end{cases} \quad (2.121)$$

A partir de la figure 2.1, il est facile de voir que la sortie du réseau global \hat{y} , s'exprime en fonction des différentes sorties des filtres x_i multipliées par les coefficients de Fourier respectifs :

$$\hat{y}(k) = g_0 x_0(k) + g_1 x_1(k) + \cdots + g_n x_n(k) \quad (2.122)$$

Après quelques manipulations sur le système (2.121), on montre que l'état évolue selon la relation :

$$\begin{cases} X(k+1) = AX(k) + Bu(k) \\ \hat{y}(k) = CX(k) \end{cases} \quad (2.123)$$

où le vecteur d'état $X(k)$, de dimension $n+1$, contient l'ensemble des sorties du filtre de Laguerre. Les éléments de la matrice carrée A , de dimension $(n+1, n+1)$, et du vecteur B de dimension $n+1$, dépendent uniquement du pôle de Laguerre ξ . Les éléments de C sont les coefficients de Fourier de la décomposition de la fonction de transfert originelle sur la base des fonctions de Laguerre.

$$A = \begin{bmatrix} \xi & 0 & 0 & \cdots & 0 \\ 1 - \xi^2 & \xi & 0 & \cdots & 0 \\ -\xi(1 - \xi^2) & 1 - \xi^2 & \xi & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-\xi)^{n-1}(1 - \xi^2) & (-\xi)^{n-2}(1 - \xi^2) & \cdots & & \xi \end{bmatrix} ;$$

$$X(k) = \begin{bmatrix} x_0(k) \\ x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} ; B = \sqrt{1 - \xi^2} \begin{bmatrix} 1 \\ -\xi \\ (-\xi)^2 \\ \vdots \\ (-\xi)^n \end{bmatrix}$$

et

$$C = [g_0 \ g_1 \ g_2 \ \cdots \ g_n] \quad (2.124)$$

2.6.2 Modèle de Kautz sous forme d'équation d'état

Soit le vecteur d'état :

$$XX(k+1) = \begin{pmatrix} XX_0(k+1) \\ XX_1(k+1) \\ \vdots \\ XX_{2p}(k+1) \\ XX_{2p+1}(k+1) \end{pmatrix} \quad (2.125)$$

Remarque : Les termes $XX_i(k+1)$ de ce vecteur d'indices i pairs sont de dimension 2 et ceux d'indices i impairs de dimension 1. Donc le vecteur d'état XX est de dimension $3(p+1)$. ($n = 2(p+1)$) est l'ordre de troncature du réseau des filtres de Kautz illustré dans la figure 2.2).

L'équation d'état s'écrit comme suit :

$$\begin{cases} XX(k) = AXX(k-1) + Bu(k-1) \\ \hat{y}(k) = CXX(k) \end{cases} \quad (2.126)$$

A partir de la figure 2.2, en développant les fonctions de transfert en z liant les différents $x_i(z)$; $i = 0, \dots, 2p+1$, on détermine les équations d'état partielles qui lient entre eux les vecteurs x_i . Toutes les équations d'état sont des formes compagnes observables. La démonstration est détaillée dans l'annexe A.

Equation d'état relative au $(2p+1)^{\text{ème}}$ bloc

$$\begin{aligned} XX_{2p}(k+1) &= \begin{pmatrix} 0 & c \\ 1 & -b(c-1) \end{pmatrix} XX_{2p}(k) \\ &+ (-c)^0(1-c^2) \begin{pmatrix} 0 & 1 \\ 0 & -b \end{pmatrix} XX_{2p-2}(k) \\ &+ \dots \\ &+ (-c)^{p-2}(1-c^2) \begin{pmatrix} 0 & 1 \\ 0 & -b \end{pmatrix} XX_0(k) \end{aligned}$$

$$\begin{aligned} x_{2p}(k) &= (-c)^0 \begin{bmatrix} 0 & 1 \end{bmatrix} XX_{2p}(k) \\ &+ (-c)^1 \begin{bmatrix} 0 & 1 \end{bmatrix} XX_{2p-2}(k) \\ &+ \dots \\ &+ (-c)^{p-1} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k) \end{aligned}$$

Equation d'état relative au $(2p+2)^{\text{ème}}$ bloc

$$\begin{aligned} XX_{2p+1}(k+1) &= b.XX_{2p+1}(k) + \\ &+ (-c)^0\sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_{2p}(k) \\ &+ \dots \\ &+ (-c)^{p-1}\sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k) \end{aligned}$$

$$x_{2p+1}(k) = XX_{2p+1}(k)$$

Le choix des paramètres b et c conditionne le nombre de coefficients g_i ayant une valeur significative dans le développement en série sur des bases de fonctions de Kautz. Afin de réduire la complexité (i.e. l'ordre n) du modèle, ces paramètres doivent être choisis tels que l'un des pôles complexes conjugués de Kautz soit proche de la dynamique dominante du système.

Une fois l'ordre n et les pôles ξ et ξ^* du modèle de Kautz fixés, il reste à déterminer les coefficients g_i . Ces coefficients sont obtenus à travers une étape d'estimation paramétrique. Les équations d'état (2.127) sont alors simulées avec une séquence d'entrée $\{u(k)\}$ quelconque, puis le vecteur C de Kautz est estimé en appliquant la méthode des moindres carrés.

2.6.3 Modèle de la BOG à pôles réels sous forme d'équation d'état

Le schéma bloc correspondant au réseau de la BOG utilisé dans ce qui suit est défini dans la figure 2.15. Il est équivalent au schéma déjà défini dans la figure 2.3. Il servira pour synthétiser les équations d'état des filtres issus de la BOG.

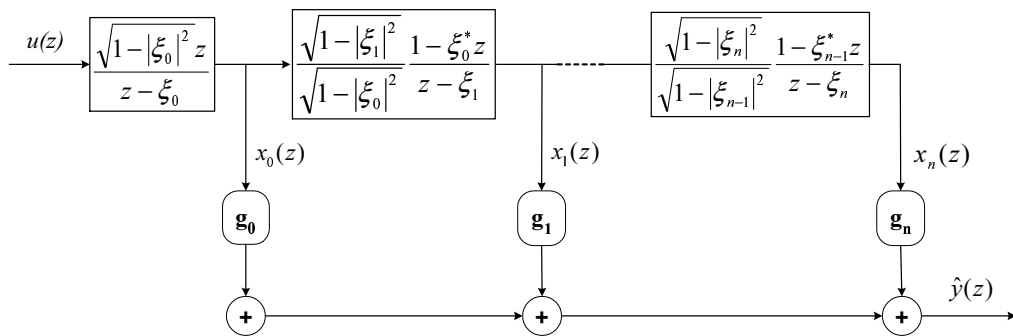


FIG. 2.15 – Réseau de filtres de la BOG

On se propose de générer les équations d'état des filtres strictement causaux issus de la BOG. A partir de la relation de récurrence (2.41) et de la figure 2.15, on peut écrire :

$$\begin{cases} x_0(k) = \frac{\sqrt{1-|\xi_0|^2}}{q-\xi_0} u(k) \\ x_1(k) = A_1 \frac{1-\xi_0^* q}{q-\xi_1} x_0(k) \\ \vdots \\ x_n(k) = A_n \frac{1-\xi_{n-1}^* q}{q-\xi_n} x_{n-1}(k), \quad n \geq 1 \end{cases} \quad (2.128)$$

$$\text{où } A_n = \frac{\sqrt{1-|\xi_n|^2}}{\sqrt{1-|\xi_{n-1}|^2}}$$

$$\Rightarrow \begin{cases} x_0(k+1) = \xi_0 x_0(k) + \sqrt{1-|\xi_0|^2} u(k) \\ x_1(k+1) + A_1 \xi_0^* x_0(k+1) = A_1 x_0(k) + \xi_1 x_1(k) \\ \vdots \\ x_n(k+1) + A_n \xi_{n-1}^* x_{n-1}(k+1) = A_n x_{n-1}(k) + \xi_n x_n(k), \quad n \geq 1 \end{cases} \quad (2.129)$$

La sortie globale s'écrit :

$$\hat{y}(k) = g_0 x_0(k) + g_1 x_1(k) + g_2 x_2(k) + \dots + g_n x_n(k) \quad (2.130)$$

Définissons le changement de variables :

$$\begin{cases} xx_0(k) = x_0(k) \\ xx_1(k) = x_1(k) + A_1 \xi_0^* x_0(k) \\ xx_2(k) = x_2(k) + A_2 \xi_1^* x_1(k) \\ \vdots \\ xx_n(k) = x_n(k) + A_n \xi_{n-1}^* x_{n-1}(k) \end{cases} \quad (2.131)$$

dont le changement réciproque est :

$$\begin{cases} x_0(k) = xx_0(k) \\ x_1(k) = xx_1(k) - A_1 \xi_0^* xx_0(k) \\ x_2(k) = xx_2(k) - A_2 \xi_1^* xx_1(k) + A_2 A_1 \xi_0^* \xi_1^* xx_0(k) \\ \vdots \\ x_n(k) = xx_n(k) - A_n \xi_{n-1}^* xx_{n-1}(k) + A_n A_{n-1} \xi_{n-1}^* \xi_{n-2}^* xx_{n-2}(k) \\ \quad + \dots + (-1)^n \prod_{k=1}^n A_k \xi_{k-1}^* xx_0(k) \end{cases} \quad (2.132)$$

En remplaçant (2.131) et (2.132) dans (2.129), on obtient :

$$\left\{ \begin{array}{l} xx_0(k+1) = \xi_0 xx_0(k) + \sqrt{1 - |\xi_0|^2} u(k) \\ xx_1(k+1) = \xi_1 xx_1(k) + A_1(1 - \xi_0^* \xi_1) xx_0(k) \\ \vdots \\ xx_n(k+1) = \xi_n xx_n(k) \\ \quad + A_n(1 - \xi_{n-1}^* \xi_n) xx_{n-1}(k) \\ \quad - A_n A_{n-1} \xi_{n-2}^* (1 - \xi_{n-1}^* \xi_n) xx_{n-2}(k) \\ \quad + \dots \\ \quad + (-1)^{n-1} A_n \prod_{k=0}^{n-2} A_{k+1} \xi_k^* (1 - \xi_{n-1}^* \xi_n) xx_0(k) \end{array} \right. \quad (2.133)$$

et en remplaçant (2.131) dans (2.130), on obtient :

$$\begin{aligned} \hat{y}(k) &= g_n xx_n(k) \\ &+ (g_{n-1} - g_n A_n \xi_{n-1}^*) xx_{n-1} \\ &+ \dots \\ &+ \left(g_0 + \sum_{k=1}^n (-1)^k g_k \prod_{j=1}^k A_j \xi_{j-1}^* \right) xx_0(k) \end{aligned} \quad (2.134)$$

D'où la représentation d'état du réseau de la BOG :

$$\left\{ \begin{array}{l} XX(k+1) = A XX(k) + Bu(k) \\ \hat{y}(k) = C XX(k) \end{array} \right. \quad (2.135)$$

avec :

$$XX(k) = \begin{bmatrix} xx_0(k) \\ xx_1(k) \\ xx_2(k) \\ \vdots \\ xx_n(k) \end{bmatrix}; A = \begin{bmatrix} \xi_0 & 0 & 0 & 0 & 0 \\ A_1(1 - \xi_0^* \xi_1) & \xi_1 & 0 & 0 & 0 \\ -A_2 A_1 \xi_0^* (1 - \xi_1^* \xi_2) & \xi_2 & 0 & 0 & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_0 & \alpha_1 & \alpha_2 & \dots & \xi_n \end{bmatrix}$$

$$B = \begin{bmatrix} \sqrt{1 - |\xi_0|^2} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{et} \quad C = [c_0, c_1, c_2, \dots, c_n].$$

où

$$\alpha_i = (-1)^{n+1-i} A_n \prod_{k=i}^{n-2} A_{k+1} \xi_k^* (1 - \xi_{n-1}^* \xi_n) \quad (2.136)$$

$$c_i = g_i + \sum_{j=i+1}^n (-1)^j g_j \prod_{k=i+1}^j A_k \xi_{k-1}^* \quad (2.137)$$

et

$$A_k = \frac{\sqrt{1 - |\xi_k|^2}}{\sqrt{1 - |\xi_{k-1}|^2}}, \quad k = 1, \dots, n \quad (2.138)$$

2.6.4 Modèle de la BOG à pôles réels et complexes sous forme d'équation d'état

La nature des pôles (réels ou complexes) d'une BOG ainsi que leur ordre de succession sont importants lors de la construction de la BOG. C'est pour cela que le modèle d'état de la BOG dépend des facteurs précédemment cités. Les différents cas de succession des pôles de natures différentes (Réels ou complexes) sont détaillés dans l'annexe C.

Dans ce qui suit, on présente le calcul de prédicteurs comme l'une des illustrations de l'utilité des modèles d'état calculés précédemment.

2.7 Calcul de prédicteurs

2.7.1 Modèle et prédicteur

Nous présentons le modèle de type Réponse Impulsionnelle Finie (RIF) et montrons par la suite comment la prédiction $y(k + i/k)$ à i pas de la sortie peut être décomposée en deux parties :

$$y(k + i/k) = y_l(k + i/k) + y_f(k + i/k) \quad (2.139)$$

Le premier terme correspond à la réponse libre du système, c'est à dire la valeur de la sortie à l'instant $(k + i)$ prédite à l'aide des sorties mesurées jusqu'à l'instant kT , tandis que le deuxième terme correspond à la réponse forcée, c'est à dire la composante de la prédiction résultant de l'action des commandes futures.

2.7.2 Prédicteur à i -pas de la sortie d'un modèle basé sur une décomposition orthogonale

Dans cette partie, nous allons déterminer le prédicteur à i pas de la sortie, construit sur un modèle issu de la BOG mis sous la forme incrémentale. Le calcul de prédicteurs basé sur des modèles de Laguerre a été développé par Oliveira et al [54].

En supposant l'erreur de modélisation négligeable et après multiplication par le polynôme $\Delta(q) = 1 - q^{-1}$, l'équation d'état du modèle issu de la BOG s'écrit sous la forme incrémentale suivante :

$$\delta XX(k) = A\delta XX(k-1) + B\delta u(k-1) \quad (2.140)$$

$$\hat{y}(k) = \hat{y}(k-1) + C\delta XX(k) \quad (2.141)$$

où les incréments du signal de commande et du vecteur d'état sont définis par :

$$\delta u(k) = u(k) - u(k-1) ; \delta XX(k) = XX(k) - XX(k-1) \quad (2.142)$$

La prédiction à i pas de la sortie se déduit de l'équation de mesure (2.141) comme suit :

$$\hat{y}(k+i/k) = \hat{y}(k+i-1/k) + C\delta XX(k+i) ; i \geq 1 \quad (2.143)$$

où $y(k+i/k)$ signifie que la prédiction à l'instant $(k+i)T$ est calculée à partir de mesures de la sortie effectuées jusqu'à l'instant kT .

D'après l'équation (2.140), on a :

$$\delta XX(k+i) = A^i \delta XX(k) + \sum_{j=1}^i A^{i-j} B \delta u(k+j-1) \quad (2.144)$$

La substitution de (2.144) dans (2.143) donne :

$$\hat{y}(k+i/k) = \hat{y}(k+i-1/k) + CA^i \delta XX(k) + C \sum_{j=1}^i A^{i-j} B \delta u(k+j-1) \quad (2.145)$$

ou encore

$$\begin{aligned} \hat{y}(k+i/k) &= \hat{y}(k+i-2/k) + C [A^i + A^{i-1}] \delta XX(k) \\ &+ C \left[\sum_{j=1}^i A^{i-j} + \sum_{j=1}^{i-1} A^{i-1-j} \right] B \delta u(k+j-1) \end{aligned} \quad (2.146)$$

Par substitution successive et en tenant compte que $\hat{y}(k/k) = \hat{y}(k)$, on obtient :

$$\begin{aligned} \hat{y}(k+i/k) &= \hat{y}(k) + C [A^i + A^{i-1} + \dots + A] \delta X X(k) \\ &+ C \left[\sum_{j=1}^i A^{i-j} + \sum_{j=1}^{i-1} A^{i-1-j} + \dots + \sum_{j=1}^1 A^{1-j} \right] B \delta u(k+j-1) \end{aligned} \quad (2.147)$$

ou encore, après une réorganisation des sommes

$$\hat{y}(k+i/k) = \hat{y}(k) + C \sum_{j=1}^i A^j \delta X X(k) + C \sum_{j=1}^i \left(\sum_{l=0}^{i-j} A^l \right) B \delta u(k+j-1) \quad (2.148)$$

Définissant

$$K_i = \sum_{j=0}^i A^j ; \text{ pour } i \geq 0 \text{ et } K_i = 0 \text{ pour } i < 0 \quad (2.149)$$

nous obtenons

$$\hat{y}(k+i/k) = \hat{y}(k) + C [K_i - I] \delta X X(k) + C \sum_{j=1}^i K_{i-j} B \delta u(k+j-1) \quad (2.150)$$

Par suite, la prédiction à i -pas de la sortie se décompose sous la forme (2.139) avec :

$$y_l(k+i/k) = \hat{y}(k) + C [K_i - I] \delta X X(k) \quad (2.151)$$

$$y_f(k+i/k) = C \sum_{j=1}^i K_{i-j} B \delta u(k+j-1) \quad (2.152)$$

Le vecteur $Y(k)$ des prédictions, sur l'horizon de prédiction $[k+1, k+N_p]$, se décompose alors en

$$Y(k) = Y_f(k) + Y_l(k) = M \delta U(k) + Y_l(k) \quad (2.153)$$

où, le vecteur $Y_l(k)$ peut être calculé en utilisant l'expression (2.151) pour $i = 1, \dots, N_p$, $\delta U(k)$ est le vecteur des incréments de commande, de dimension N_p

$$\delta U(k) = \left[\begin{array}{cccc} \delta u(k) & \delta u(k-1) & \dots & \delta u(k+N_p-1) \end{array} \right]^T \quad (2.154)$$

et M est la matrice de dimension (N_p, N_p) , donnée par :

$$M = \left[\begin{array}{cccc} m_0 & 0 & \dots & 0 \\ m_1 & m_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ m_{N_p} & m_{N_p-1} & \dots & m_0 \end{array} \right] \quad (2.155)$$

où

$$m_i = CK_{i-1}B \quad (2.156)$$

En utilisant la définition (2.149) de K_i et en notant que $h_j = CA^{j-1}B$, on a

$$m_i = \sum_{j=0}^{i-1} CA^j B = \sum_{j=1}^i h_j \quad (2.157)$$

Les éléments $\{h_j\}_{j=1,\dots,i}$ représentent les coefficients de la réponse indicielle du modèle de la BOG.

Chapitre 3

Techniques de décompositions tensorielles

3.1 Introduction

Les tenseurs généralisent les notions de vecteur et de matrice. On définit un tenseur \mathbb{H} comme un ensemble dont les éléments peuvent s'écrire de la manière suivante :

$$\mathbb{H}(n_1, n_2, \dots, n_N) \left\{ \begin{array}{l} n_1 = 1, \dots, N_1 \\ n_2 = 1, \dots, N_2 \\ \vdots \\ n_N = 1, \dots, N_N \end{array} \right. \quad (3.1)$$

Seuls les tenseurs à éléments réels seront considérés dans ce chapitre.

Un tenseur à un seul indice $n_1 = 1, \dots, N_1$ est un vecteur, avec deux indices n_1 et n_2 , c'est une matrice. Un tenseur avec trois indices ou d'ordre trois peut géométriquement être arrangé dans un parallélépipède (Voir figure 3.1). Comme pour les tenseurs d'ordre 1 (les vecteurs) et les tenseurs d'ordre 2 (les matrices), les termes "lignes" et "colonnes" sont aussi utilisés pour les tenseurs d'ordre 3. Les éléments de la troisième dimension s'appelleront "tubes" (voir figure 3.2). Il est possible de définir des sous-matrices à partir de tenseurs d'ordre 3. Dans la figure 3.2.b, la sous-matrice en gris a été obtenue en fixant le troisième indice du tenseur. Ces matrices sont généralement appelées couches ou tranches du tenseur. Dans le cas de la matrice grise de la figure

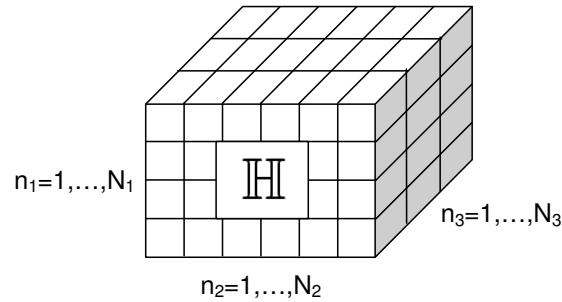
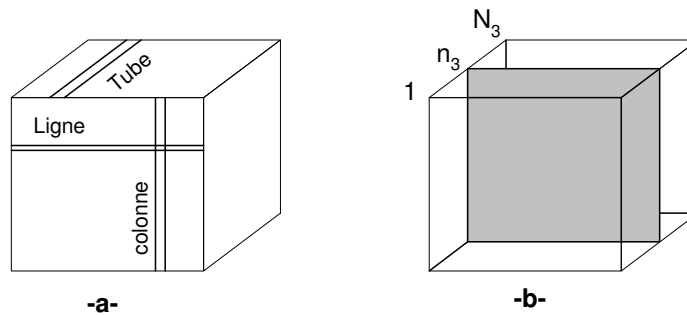


FIG. 3.1 – Représentation graphique d'un tenseur d'ordre 3

3.2.b, il s'agit d'une couche frontale. Les deux autres catégories de couches s'appelleront les couches horizontales et les couches verticales (Kroonenberg [37], Harshman et Lundy [28]). Analogiquement aux matrices, chaque direction du tenseur d'ordre 3 s'appelle "mode" et le nombre d'éléments de chaque mode s'appelle "dimension".

FIG. 3.2 – (a) Définition d'une ligne, d'une colonne et d'un tube. (b) $n_3^{\text{ème}}$ couche frontale

Dans ce chapitre, Les scalaires sont désignés par des caractères minuscules (h). Les vecteurs sont pris sous forme de colonnes par défaut et sont désignés ainsi que les matrices par des caractères majuscules (H). Les tenseurs de dimensions supérieures à deux sont désignés par des lettres à caractère spécial (III).

3.2 Les modèles de décomposition tensorielle

Dans cette section, on s'intéresse à l'écriture des tenseurs d'ordre supérieur à 2 sous forme de différents modèles permettant la réduction de leur complexité. Ces modèles

conservent l'intégralité de l'information contenue dans les tenseurs initiaux.

Plusieurs modèles de decomposition tensorielle sont décrits dans cette section. Parmi ces modèles, le modèle PARAFAC et le modèle Tucker3 sont les plus utilisés.

3.2.1 Le modèle PARAFAC

Ce modèle trilinéaire a été proposé par Harshman [26] et par Carroll et Chang [12] qui l'ont appelé respectivement PARAFAC (PARAllel FACtor) et CANDECOMP (CANonical DECOMPosition).

Le modèle PARAFAC d'un tenseur \mathbb{H} d'ordre 3 est défini, comme le montre la figure 3.3, par trois matrices A , B , C et un tenseur \mathbb{I} cubique de dimension $(P \times P \times P)$ dont tous les éléments sont nuls sauf la super-diagonale dont les éléments sont égaux à l'unité ($\mathbb{I}_{ppp} = 1 ; p = 1, \dots, P$).

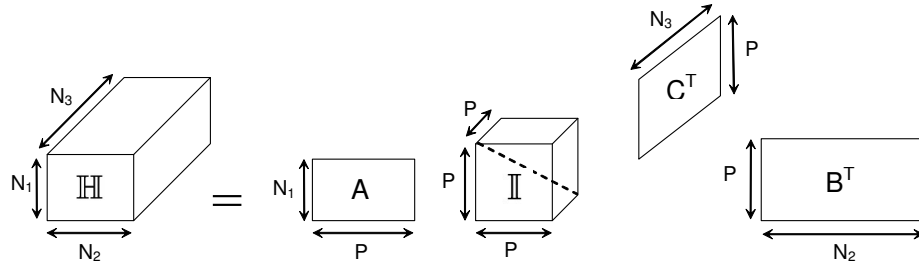


FIG. 3.3 – Représentation tensorielle du modèle trilinéaire PARAFAC

La représentation scalaire de ce modèle s'écrit comme suit :

$$h_{n_1 n_2 n_3} = \sum_{p=1}^P a_{n_1 p} b_{n_2 p} c_{n_3 p} ; \quad \begin{array}{l} n_1 = 1, \dots, N_1 \\ n_2 = 1, \dots, N_2 \\ n_3 = 1, \dots, N_3 \end{array} \quad (3.2)$$

où $h_{n_1 n_2 n_3}$ est l'élément (n_1, n_2, n_3) du tenseur \mathbb{H} , $a_{n_1 p}$ est l'élément (n_1, p) de la matrice A de dimension $(N_1 \times P)$, $b_{n_2 p}$ est l'élément (n_2, p) de la matrice B de dimension $(N_2 \times P)$, $c_{n_3 p}$ est l'élément (n_3, p) de la matrice C de dimension $(N_3 \times P)$ et P est le nombre de facteurs de ce modèle. P sera défini par la suite.

Dans le cas général, la représentation scalaire du modèle PARAFAC d'un tenseur \mathbb{X}

d'ordre N s'écrit comme suit :

$$x_{i_1 i_2 \dots i_N} = \sum_{p=1}^P \left(\prod_{n=1}^N a_{i_n p, n} \right) \quad (3.3)$$

où $a_{i_n p, n}$ désigne l'élément (i_n, p) de la matrice A_n .

Dans la littérature, la décomposition PARAFAC a été présentée sous plusieurs formes. Dans ce qui suit, on détaille trois représentations différentes du modèle PARAFAC.

Première représentation du modèle PARAFAC

Définition : On définit les matrices suivantes :

- $H_{n_1..}$ ($n_1 = 1, \dots, N_1$) sont des matrices de dimension $N_2 \times N_3$ telles que

$$H_{n_1..}(n_2, n_3) = \mathbb{H}(n_1, n_2, n_3).$$

- $H_{.n_2.}$ ($n_2 = 1, \dots, N_2$) sont des matrices de dimension $N_3 \times N_1$ telles que

$$H_{.n_2.}(n_3, n_1) = \mathbb{H}(n_1, n_2, n_3).$$

- $H_{..n_3}$ ($n_3 = 1, \dots, N_3$) sont des matrices de dimension $N_1 \times N_2$ telles que

$$H_{..n_3}(n_1, n_2) = \mathbb{H}(n_1, n_2, n_3).$$

La construction des matrices $H_{n_1..}$, $H_{.n_2.}$ et $H_{..n_3}$ est détaillée dans la figure 3.4.

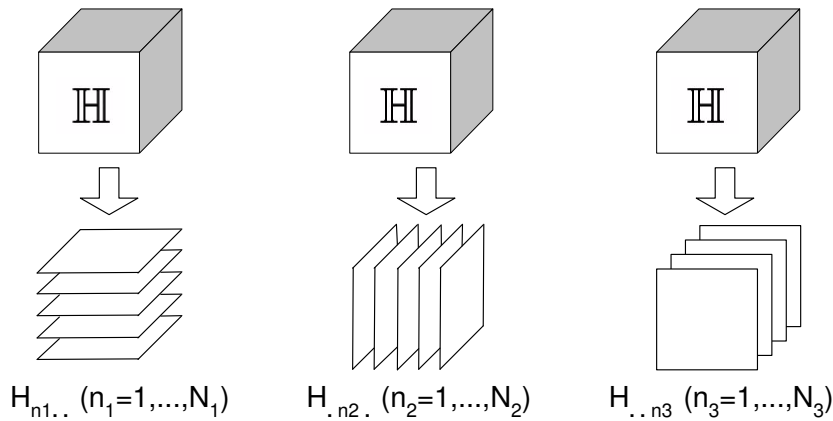


FIG. 3.4 – Construction des matrices $H_{n_1..}$, $H_{.n_2.}$ et $H_{..n_3}$

A partir de l'équation (3.2), on peut écrire les matrices $H_{n_1..}$, $H_{..n_2}$ et $H_{..n_3}$ de la manière suivante :

$$H_{n_1..} = BD_{n_1}^A C^T \quad (3.4)$$

$$H_{..n_2} = CD_{n_2}^B A^T \quad (3.5)$$

$$H_{..n_3} = AD_{n_3}^C B^T \quad (3.6)$$

avec

$$D_{n_1}^A = \mathbf{diag}(A_{n_1.}) \quad (3.7)$$

$$D_{n_2}^B = \mathbf{diag}(B_{n_2.}) \quad (3.8)$$

$$D_{n_3}^C = \mathbf{diag}(C_{n_3.}) \quad (3.9)$$

$(A_{n_1.})$, $(B_{n_2.})$, et $(C_{n_3.})$ désignent respectivement la $n_1^{\text{ème}}$, la $n_2^{\text{ème}}$ et la $n_3^{\text{ème}}$ ligne des matrices A , B et C . $\mathbf{diag}(C_{n_3.}) = D_{n_3}^C$ désigne la matrice carrée diagonale de dimension $(P \times P)$ dont les éléments diagonaux sont les éléments du vecteur $(C_{n_3.})$.

Deuxième représentation du modèle PARAFAC

Définition : Une deuxième représentation matricielle du modèle PARAFAC triliénaire est définie par les matrices H_{N_1} , H_{N_2} et H_{N_3} de dimensions respectives $(N_1 \times N_2 N_3)$, $(N_2 \times N_1 N_3)$ et $(N_3 \times N_1 N_2)$ représentées dans la figure 3.5.

$$H_{N_1} = \begin{bmatrix} H_{..1} & H_{..2} & \cdots & H_{..N_3} \end{bmatrix} \quad (3.10)$$

$$H_{N_2} = \begin{bmatrix} H_{1..} & H_{2..} & \cdots & H_{N_1..} \end{bmatrix} \quad (3.11)$$

$$H_{N_3} = \begin{bmatrix} H_{.1.} & H_{.2.} & \cdots & H_{.N_2.} \end{bmatrix} \quad (3.12)$$

La construction d'une représentation matricielle de tenseurs d'ordre N se fait suivant la même technique que pour l'ordre 3. Par exemple, l'une des représentations matricielles d'un tenseur \mathbb{X} d'ordre N est une matrice X_{N_1} de dimension $(N_1 \times N_2 \dots N_N)$.

Pour mieux comprendre la construction de la matrice X_{N_1} relative à un tenseur d'ordre supérieur à 3, on construit la matrice X_{N_1} relative à un tenseur d'ordre 4 comme suit :

$$X_{N_1} = \left[\underbrace{\begin{array}{ccc} \overbrace{(X_{...1})_{..1}}^{(N_1 \times N_2)} & \cdots & (X_{...1})_{..N_3} \\ \hline \underbrace{\hspace{10em}}_{N_3 \text{ fois}} & | \cdots | & \underbrace{\hspace{10em}}_{N_3 \text{ fois}} \end{array}}_{N_4 \text{ fois}} \right] \quad (3.13)$$

$X_{\dots n_4}$ représente un tenseur d'ordre 3 et de dimension $(N_1 \times N_2 \times N_3)$ obtenu en fixant la quatrième coordonnée n_4 du tenseur \mathbb{X} .

$(X_{\dots n_4})_{..n_3}$ représente une matrice de dimension $(N_1 \times N_2)$ obtenue en fixant la troisième coordonnée n_3 du tenseur $X_{\dots n_4}$.

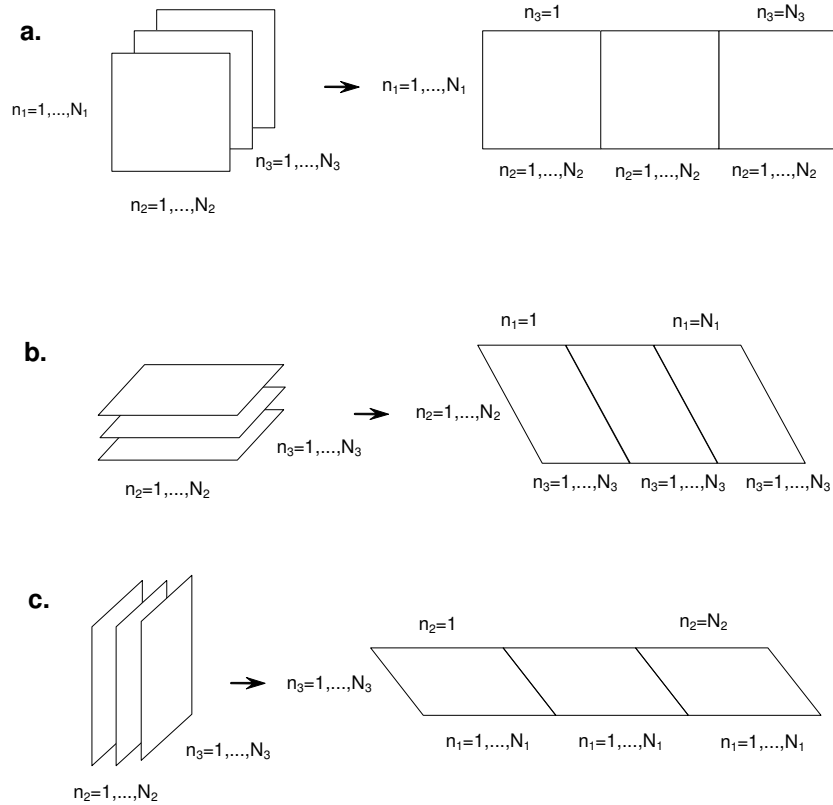


FIG. 3.5 – Construction des matrices H_{N_1} (a), H_{N_2} (b) et H_{N_3} (c)

Définition : Le produit de Kronecker de deux matrices X (de dimension $(I \times J)$) et Y est défini par :

$$X \otimes Y = \begin{bmatrix} x_{11}Y & \cdots & x_{1J}Y \\ \vdots & & \vdots \\ x_{I1}Y & \cdots & x_{IJ}Y \end{bmatrix} \quad (3.14)$$

Définition : Le produit de Khatri-Rao de deux matrices X et Y ayant le même nombre de colonnes N .

$$X| \otimes |Y = \left[X_{.1} \otimes Y_{.1} \quad X_{.2} \otimes Y_{.2} \quad \cdots \quad X_{.N} \otimes Y_{.N} \right] \quad (3.15)$$

$(X_{.n})$ et $(Y_{.n})$ représentent les $n^{\text{èmes}}$ colonnes respectives des matrices X et Y .

PROPOSITION : En utilisant le produit de Khatri-Rao ($|\otimes|$), on écrit la matrice H_{N_1} comme suit :

$$H_{N_1} = A(C| \otimes |B)^T \quad (3.16)$$

DÉMONSTRATION :

$$A(C| \otimes |B)^T = A \left[C_{.1} \otimes B_{.1} \quad \cdots \quad C_{.P} \otimes B_{.P} \right]^T \quad (3.17)$$

$$= A \begin{bmatrix} (C_{.1} \otimes B_{.1})^T \\ \vdots \\ (C_{.P} \otimes B_{.P})^T \end{bmatrix} \quad (3.18)$$

$$= A \begin{bmatrix} c_{11}B_{.1}^T & \cdots & c_{N_3 1}B_{.1}^T \\ \vdots & & \vdots \\ c_{1P}B_{.P}^T & \cdots & c_{N_3 P}B_{.P}^T \end{bmatrix} \quad (3.19)$$

$$= A \left[D_1^C B^T \quad \cdots \quad D_{N_3}^C B^T \right] \quad (3.20)$$

$$= \left[AD_1^C B^T \quad \cdots \quad AD_{N_3}^C B^T \right] \quad (3.21)$$

D'après (3.6) et (3.10), on obtient :

$$A(C| \otimes |B)^T = \left[H_{.1} \quad \cdots \quad H_{.N_3} \right] \quad (3.22)$$

$$= H_{N_1} \quad \blacksquare \quad (3.23)$$

De la même manière, on peut écrire H_{N_2} et H_{N_3} comme :

$$H_{N_2} = B(A| \otimes |C)^T \quad (3.24)$$

et

$$H_{N_3} = C(B| \otimes |A)^T \quad (3.25)$$

Nombre de facteurs du modèle PARAFAC

Le nombre P déjà défini comme étant le nombre de facteurs du modèle PARAFAC, représente le rang du tenseur \mathbb{H} . Kruskal [38] a montré que P satisfait l'inégalité suivante :

$$P = \mathbf{rang}(\mathbb{H}) \geq \max \left\{ \mathbf{rank}(H_{N_1}), \mathbf{rank}(H_{N_2}), \mathbf{rank}(H_{N_3}) \right\} \quad (3.26)$$

où les matrices H_{N_1} , H_{N_2} et H_{N_3} ont été définies dans (3.10-3.12)

Unicité

L'un des avantages du modèle PARAFAC est son unicité. Cette unicité se traduit par la perte d'informations d'une représentation PARAFAC $\mathcal{R}_1(\mathbb{H})$ d'un tenseur \mathbb{H} , fonction des matrices A , B et C à une autre représentation $\mathcal{R}_2(\mathbb{H})$ fonction des matrices A' , B' et C' si la transformation \mathcal{T} telle que $A' = \mathcal{T}(A)$, $B' = \mathcal{T}(B)$ et $C' = \mathcal{T}(C)$, est autre qu'une matrice diagonale ou une permutation non-singulière. Cette unicité a été prouvée par Harshman [27].

Théorème

Soient $H_{..1}$ et $H_{..2}$, deux matrices composantes du tenseur \mathbb{H} .

$$H_{..1} = AD_1^C B^T \quad (3.27)$$

$$H_{..2} = AD_2^C B^T \quad (3.28)$$

où A et B sont deux matrices de rang plein de dimensions respectives $(N_1 \times P)$ et $(N_2 \times P)$ et D_1^C et D_2^C sont deux matrices diagonales de dimensions $(P \times P)$ telles que :

$$D_i^C = \mathbf{diag}(C_i.) ; i = 1, \dots, N_3 \quad (3.29)$$

S'il existe une autre représentation possible des matrices $H_{..1}$ et $H_{..2}$ qui s'écrit comme suit :

$$H_{..1} = \tilde{A} \tilde{D}_1^C \tilde{B}^T \quad (3.30)$$

$$H_{..2} = \tilde{A} \tilde{D}_2^C \tilde{B}^T \quad (3.31)$$

avec

$$\tilde{A} = AT_A \quad (3.32)$$

$$\tilde{B} = BT_B \quad (3.33)$$

alors, T_A et T_B sont des matrices diagonales et/ou de permutation non-singulières ayant un seul élément non nul sur chacune de leurs colonnes. \square

3.2.2 Le modèle PARAFAC2

Dans certains cas, les modèles réels ne sont pas conformes au modèle PARAFAC comme par exemple pour des problèmes rencontrés en chimie ou en psychologie quand des données manquent sur un patient (même nombre de mesures sur N_2 et N_3 , mais des mesures qui varient d'un patient à un autre sur N_1), comme le montre la figure 3.6. Dans ce cas de figures, au lieu de forcer l'information dans un modèle PARAFAC, le problème est résolu en utilisant le modèle PARAFAC2.

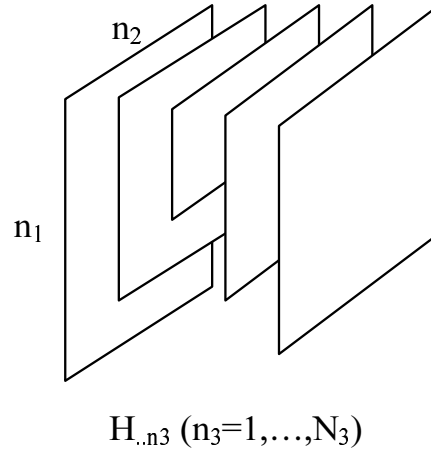


FIG. 3.6 – Type de tenseurs modélisés à l'aide de PARAFAC2

Considérons le modèle PARAFAC classique suivant :

$$H_{..n_3} = AD_{n_3}^C B^T \quad (3.34)$$

Le modèle de l'équation (3.34) n'est pas valable dans ce cas puisque les matrices $\{H_{..n_3}\}_{n_3=1,\dots,N_3}$ ne sont pas de mêmes dimensions (N_1 varie d'une matrice à une autre). On les modélise alors en les écrivant sous la forme scalaire suivante :

$$H_{..n_3} = A_{n_3} D_{n_3}^C B^T = P_{n_3} A D_{n_3}^C B^T ; n_3 = 1, \dots, N_3 \quad (3.35)$$

La différence avec le modèle PARAFAC est que pour le modèle PARAFAC2 chacune des matrices ($H_{..n_3}$) a sa propre matrice caractéristique (A_{n_3}) de dimension ($N_1(n_3) \times P$). La matrice A étant de dimension ($P \times P$), la matrice P_{n_3} est carrée de dimension ($N_1(n_3) \times N_1(n_3)$) et les matrices B et $D_{n_3}^C$ sont définies de la même manière que dans le modèle PARAFAC.

De l'équation (3.35), on peut déduire une représentation scalaire du tenseur \mathbb{H} comme suit :

$$h(n_1(n_3), n_2, n_3) = \sum_{p=1}^P a_{n_1(n_3)p} b_{n_2p} c_{n_3p} ; \quad \begin{array}{l} n_2 = 1, \dots, N_2 \\ n_3 = 1, \dots, N_3 \end{array} \quad (3.36)$$

Pour obtenir une représentation matricielle du modèle PARAFAC2, il est plus facile de modéliser les matrices $Y_{..n_3} = H_{..n_3}^T H_{..n_3}$ qui forment le tenseur \mathbb{Y} et qui sont de même dimension égale à $(N_2 \times N_2)$ contrairement aux matrices $\{H_{..n_3}\}_{n_3=1, \dots, N_3}$ qui ont des dimensions différentes. L'écriture de cette nouvelle forme du modèle nécessite la formulation de la contrainte suivante (3.37) proposée par Harshman.

$$\Phi = A_{n_3}^T A_{n_3} \quad ; \quad n_3 = 1, \dots, N_3 \quad (3.37)$$

où Φ est une matrice de dimension $(P \times P)$.

La représentation matricielle du modèle utilisant les matrices $Y_{..n_3}$ de dimension $(N_2 \times N_2)$ s'écrit alors :

$$Y_{..n_3} = H_{..n_3}^T H_{..n_3} \quad (3.38)$$

$$= (A_{n_3} D_{n_3}^C B^T)^T (A_{n_3} D_{n_3}^C B^T) \quad (3.39)$$

$$= B D_{n_3}^C A_{n_3}^T A_{n_3} D_{n_3}^C B^T \quad (3.40)$$

$$= B D_{n_3}^C \Phi D_{n_3}^C B^T \quad (3.41)$$

Une représentation scalaire du modèle PARAFAC2 du tenseur \mathbb{Y} défini à partir des matrices $Y_{..n_3}$ s'écrit comme suit :

$$y_{n_2 n'_2 n_3} = \sum_p^P \sum_{p'}^{P'} b_{n_2 p} c_{n_3 p} \phi_{pp'} c_{n_3 p'} b_{n'_2 p'} \quad (3.42)$$

Pour une autre représentation matricielle du modèle PARAFAC2, on définit la matrice Y_{N_2} de dimension $(N_2 \times N_2 N_3)$ et la matrice \mathcal{Y}_{N_2} de dimension $(N_3 \times N_2 N_2)$ qui contient la même information que Y_{N_2} distribuée différemment.

$$Y_{N_2} = \left[\begin{array}{cccc} Y_{..1} & Y_{..2} & \cdots & Y_{..N_3} \end{array} \right]_{N_2 \times N_2 N_3} \quad (3.43)$$

$$= \left[\begin{array}{ccc} H_{..1}^T H_{..1} & H_{..2}^T H_{..2} & \cdots & H_{..N_3}^T H_{..N_3} \end{array} \right] \quad (3.44)$$

et

$$\mathcal{Y}_{N_2} = \begin{bmatrix} (\mathbf{vec} Y_{..1})^T \\ (\mathbf{vec} Y_{..2})^T \\ \vdots \\ (\mathbf{vec} Y_{..N_3})^T \end{bmatrix}_{N_3 \times N_2 N_2} \quad (3.45)$$

Une deuxième représentation matricielle du modèle PARAFAC2 s'écrit alors :

$$\mathcal{Y}_{N_2} = (C^T | \otimes | C^T)^T \mathbf{diag}(\mathbf{vec} \Phi) (B \otimes B)^T \quad (3.46)$$

Démonstration : Voir annexe D.

Unicité

Le modèle PARAFAC2 est unique seulement dans certains cas. Son unicité a été beaucoup moins étudiée que celle du modèle PARAFAC. Harshman et Lundy en 1996 [29] ainsi que ten Berge et Kiers en 1996 [2] donnent quelques cas particuliers de modèle PARAFAC2 où l'unicité est assurée. Par exemple, ten Berge et Kiers montrent que dans le cas d'un tenseur de dimension $(I \times I \times 3)$, l'unicité est assurée si la matrice C de la décomposition est non négative et si les matrices A et B sont de rang plein.

3.2.3 Les modèles TUCKER

Dans les années soixante, L. Tucker [63] a mis au point des modèles qui s'appellent Tucker1, Tucker2 et Tucker3. Les deux derniers sont des cas particuliers du premier. Plusieurs applications dans plusieurs domaines comme la chromatographie (De Ligny et al [41]) et la psychologie (Van der Kloot et Kroonenberg [17]), utilisant ces modèles ont été établies avec succès.

Une représentation scalaire du modèle TUCKER3 du tenseur \mathbb{H} de dimension $(N_1 \times N_2 \times N_3)$ est :

$$h_{n_1 n_2 n_3} = \sum_{p_1=1}^{P_1} \sum_{p_2=1}^{P_2} \sum_{p_3=1}^{P_3} g_{p_1 p_2 p_3} a_{n_1 p_1} b_{n_2 p_2} c_{n_3 p_3} ; \quad \begin{array}{l} n_1 = 1, \dots, N_1 \\ n_2 = 1, \dots, N_2 \\ n_3 = 1, \dots, N_3 \end{array} \quad (3.47)$$

où $g_{p_1 p_2 p_3}$ est l'élément (p_1, p_2, p_3) du tenseur noyau \mathbb{G} de dimension $(P_1 \times P_2 \times P_3)$.

Dans le cas général des tenseurs d'ordre N , on utilise une autre notation des matrices impliquées dans la décomposition. On utilisera les matrices A_1, A_2, \dots, A_N . Une représentation scalaire du modèle TUCKER3 du tenseur \mathbb{X} d'ordre N est :

$$x_{i_1 i_2 \dots i_N} = \sum_{p_1=1}^{P_1} \sum_{p_2=1}^{P_2} \cdots \sum_{p_N=1}^{P_N} g_{p_1 \dots p_N} \left(\prod_{n=1}^N a_{i_n p_n, n} \right) \quad (3.48)$$

où $(a_{i_n p_n, n})$ désigne l'élément (i_n, p_n) de la matrice A_n .

Remarque : Comme on peut le voir dans l'équation (3.47) pour les tenseurs d'ordre 3 et dans l'équation (3.48) pour les tenseurs d'ordre N , le modèle TUCKER3 est une généralisation du modèle PARAFAC puisque ce dernier est un modèle TUCKER3 particulier avec un tenseur noyau \mathbb{G} de dimension $P \times P \times P$ et dont tous les éléments sont nuls sauf ceux de la super-diagonale $(g_{nnn}; n = 1, \dots, N)$ qui sont égaux à $\mathbf{1}$. \square

En utilisant le produit de Kronecker, une représentation matricielle du tenseur \mathbb{H} défini dans (3.47) s'écrit sous la forme suivante :

$$H_{N_1} = AG_{P_1}(C \otimes B)^T \quad (3.49)$$

où G_{P_1} est une matrice de dimension $(P_1 \times P_2 P_3)$ définie à partir du noyau \mathbb{G} comme ont été définies les matrices H_{N_1} , H_{N_2} et H_{N_3} à partir du tenseur \mathbb{H} dans la figure 3.5. A , B et C sont trois matrices de dimensions respectives $(N_1 \times P_1)$, $(N_2 \times P_2)$ et $(N_3 \times P_3)$.

De la même manière on pourra écrire H_{N_2} et H_{N_3} comme suit :

$$H_{N_2} = BG_{P_2}(A \otimes C)^T \quad (3.50)$$

$$H_{N_3} = CG_{P_3}(B \otimes A)^T \quad (3.51)$$

Dans le cas général, une représentation matricielle du modèle scalaire présenté par l'équation (3.48) s'écrit comme suit :

$$X_{N_1} = A_1 G_{P_1}(A_N \otimes \cdots \otimes A_3 \otimes A_2)^T \quad (3.52)$$

où (X_{N_1}) , déjà défini dans l'équation (3.13), est une forme matricielle de dimension $(N_1 \times N_2 N_3 \dots N_N)$ du tenseur \mathbb{X} . Les matrices A_1, \dots, A_N sont de dimensions respectives $(N_1 \times P_1)$, $(N_2 \times P_2), \dots, (N_N \times P_N)$.

G_{P_1} dans l'équation (3.52) est une matrice de dimension $(P_1 \times P_2 P_3 \dots P_N)$ définie à partir du noyau \mathbb{G} comme a été définie la matrice X_{N_1} à partir du tenseur \mathbb{X} .

On donne les représentations scalaires et matricielles des modèles TUCKER2 et TUCKER1 pour le tenseur \mathbb{H} d'ordre 3 et de dimension $(N_1 \times N_2 \times N_3)$.

– Modèle TUCKER2

$$h_{n_1 n_2 n_3} = \sum_{p_1=1}^{P_1} \sum_{p_2=1}^{P_2} g_{p_1 p_2 n_3} a_{n_1 p_1} b_{n_2 p_2} \quad (3.53)$$

$$H_{N_1} = AG_{P_1}(I_{N_3} \otimes B^T) \quad (3.54)$$

– Modèle TUCKER1

$$h_{n_1 n_2 n_3} = \sum_{p_1=1}^{P_1} g_{p_1 n_2 n_3} a_{n_1 p_1} \quad (3.55)$$

$$H_{N_1} = AG_{P_1}(I_{N_3} \otimes I_{N_2}) \quad (3.56)$$

Unicité

Contrairement au modèle PARAFAC, les modèles TUCKER ne sont pas uniques. Ceci peut être facilement montré en remplaçant par exemple la représentation matricielle (3.49) du modèle TUCKER3 par une autre représentation dans laquelle la matrice A de la décomposition subit une rotation (S) (S est une matrice non-singulière). En appliquant la rotation inverse (S^{-1}) à la matrice noyau G_{P_1} de la décomposition, on obtient

$$H_{N_1} = (AS)(S^{-1}G_{P_1})(C \otimes B)^T \quad (3.57)$$

$$= AG_{P_1}(C \otimes B)^T \quad (3.58)$$

Donc, plusieurs représentations matricielles du modèle TUCKER3 sont possibles. De même, les modèles TUCKER1 et TUCKER2 ne sont pas uniques.

3.2.4 Le modèle PARATUCK2

Le modèle PARATUCK2 a été proposé par Harshman et Lundy [28] dans la littérature psychométrique. Ce modèle convient dans les applications où il y a des interactions entre les différents facteurs de la décomposition du modèle.

La figure 3.7 est une représentation tensorielle du modèle PARATUCK2. Une représentation matricielle du modèle PARATUCK2 d'un tenseur \mathbb{H} d'ordre 3 est définie comme suit :

$$H_{..n_3} = AD_{n_3}^{C^A} GD_{n_3}^{C^B} B^T \quad (3.59)$$

ce qui correspond à une représentation scalaire de la forme :

$$h_{n_1 n_2 n_3} = \sum_{p_1}^{P_1} \sum_{p_2}^{P_2} a_{n_1 p_1} c_{n_3 p_1}^A g_{p_1 p_2} c_{n_3 p_2}^B b_{n_2 p_2} \quad (3.60)$$

- A et B sont deux matrices de dimensions respectives $(N_1 \times P_1)$ et $(N_2 \times P_2)$. Le nombre de colonnes des deux matrices est différent ce qui nous rappelle l'une des caractéristiques du modèle TUCKER.
- $D_{n_3}^{C^A}$ et $D_{n_3}^{C^B}$ sont deux matrices diagonales de dimensions respectives $(P_1 \times P_1)$ et $(P_2 \times P_2)$.
- G est une matrice de dimension $(P_1 \times P_2)$, appelée matrice noyau du modèle.

Deux matrices C^A et C^B , de dimensions respectives $(N_3 \times P_1)$ et $(N_3 \times P_2)$, sont définies telles que leurs $n_3^{\text{ièmes}}$ lignes forment respectivement les diagonales des matrices $D_{n_3}^{C^A}$ et $D_{n_3}^{C^B}$.

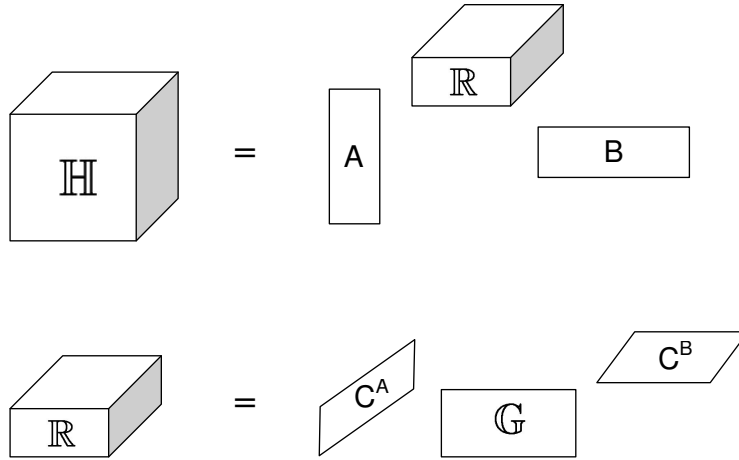


FIG. 3.7 – Représentation graphique du modèle PARATUCK2

PROPOSITION :

En définissant la matrice H_{N_1} comme suit :

$$H_{N_1} = \left[\begin{array}{cccc} H_{..1} & H_{..2} & \cdots & H_{..N_3} \end{array} \right]_{N_1 \times N_2 N_3} \quad (3.61)$$

et en la réarrangeant dans une matrice \mathcal{H}_{N_1} comme on l'a fait pour Y_{N_1} dans le modèle

PARAFAC2,

$$\mathcal{H}_{N_1} = \begin{bmatrix} (\mathbf{vec} H_{..1})^T \\ (\mathbf{vec} H_{..2})^T \\ \vdots \\ (\mathbf{vec} H_{..N_3})^T \end{bmatrix}_{N_3 \times N_1 N_2} \quad (3.62)$$

il est possible de donner une deuxième représentation matricielle au modèle PARATUCK2 en utilisant le produit matriciel de Khatri-Rao comme suit :

$$\mathcal{H}_{N_1} = ((C^B)^T | \otimes | (C^A)^T) \mathbf{diag}(\mathbf{vec}(G))(B \otimes A)^T \quad (3.63)$$

DÉMONSTRATION :

En utilisant la première représentation matricielle du modèle PARATUCK2 (3.59), on peut écrire :

$$(\mathbf{vec}(H_{..n_3}))^T = \left(\mathbf{vec}(AD_{n_3}^{C^A} GD_{n_3}^{C^B} B^T) \right)^T \quad (3.64)$$

D'après le théorème 1 de l'annexe D on peut écrire :

$$(\mathbf{vec}(H_{..n_3}))^T = \left((B \otimes A) \mathbf{vec}(D_{n_3}^{C^A} GD_{n_3}^{C^B}) \right)^T \quad (3.65)$$

$$= \left((B \otimes A)(D_{n_3}^{C^B} \otimes D_{n_3}^{C^A}) \mathbf{vec}(G) \right)^T \quad (3.66)$$

$$= (\mathbf{vec}(G))^T (D_{n_3}^{C^B} \otimes D_{n_3}^{C^A})(B \otimes A)^T \quad (3.67)$$

$$= \left(D_{n_3}^{C^B} \otimes D_{n_3}^{C^A} \right) \mathbf{vec}(G)^T (B \otimes A)^T \quad (3.68)$$

$$= (C_{n_3.}^{C^B} \otimes C_{n_3.}^{C^A}) \mathbf{diag}(\mathbf{vec}(G))(B \otimes A)^T \quad (3.69)$$

En assemblant les lignes de la matrice \mathcal{H}_{N_1} on trouve :

$$\mathcal{H}_{N_1} = \begin{bmatrix} (\mathbf{vec} H_{..1})^T \\ (\mathbf{vec} H_{..2})^T \\ \vdots \\ (\mathbf{vec} H_{..N_3})^T \end{bmatrix} \quad (3.70)$$

$$= \begin{bmatrix} (C_{1.}^{C^B} \otimes C_{1.}^{C^A}) \mathbf{diag}(\mathbf{vec}(G))(B \otimes A)^T \\ (C_{2.}^{C^B} \otimes C_{2.}^{C^A}) \mathbf{diag}(\mathbf{vec}(G))(B \otimes A)^T \\ \vdots \\ (C_{N_3.}^{C^B} \otimes C_{N_3.}^{C^A}) \mathbf{diag}(\mathbf{vec}(G))(B \otimes A)^T \end{bmatrix} \quad (3.71)$$

$$= ((C^B)^T | \otimes | (C^A)^T)^T \mathbf{diag}(\mathbf{vec}(G))(B \otimes A)^T \quad \blacksquare \quad (3.72)$$

Unicité

Harshman et Lundy [29] ont travaillé sur les propriétés d'unicité du modèle PARATUCK2. Ils ont traité seulement le cas où $P_1 = P_2$ dans lequel l'unicité est assurée si toutes les matrices de la décomposition A , B , C^A , C^B et G sont de rang plein.

3.3 Les algorithmes ALS (Alternating Least Squares)

Dans cette section, on définit d'abord les fonctions de coût à minimiser pour chacun des modèles présentés dans la section précédente. Ensuite, on présente les algorithmes ALS de calcul des matrices de la décomposition des modèles présentés.

Le principe de cet algorithme est ancien (Yates, 1933) et consiste à estimer une par une les composantes d'un modèle ou d'une représentation, dans le sens des moindres carrés et d'une manière alternée, en supposant les autres composantes fixes. Cet algorithme est néanmoins sous-optimal. L'estimation des paramètres est itérative jusqu'à la satisfaction d'un critère d'arrêt.

Définition : Soit \mathcal{H} une représentation matricielle d'un tenseur d'ordre 3 : \mathbb{H} . Considérons le modèle général suivant qui définit \mathcal{H} en fonction de trois matrices A , B et C :

$$\mathcal{H} = f(A, B, C) \tag{3.73}$$

L'algorithme ALS est alors le suivant :

1. Initialiser les paramètres A , B et C
2. A est la solution de :

$$\min_A \|\mathcal{H} - f(A, B, C)\|_F^2$$

3. B est la solution de :

$$\min_B \|\mathcal{H} - f(A, B, C)\|_F^2$$

4. C est la solution de :

$$\min_C \|\mathcal{H} - f(A, B, C)\|_F^2$$

5. Retour à l'étape 2 jusqu'à satisfaction du critère d'arrêt.

$\|\cdot\|_F$ représente la norme de Frobenius.

Dans ce qui suit, on va adapter l'algorithme ALS à chacun des modèles déjà présentés dans la section précédente.

3.3.1 L'algorithme ALS-PARAFAC

Pour la décomposition de tenseurs de dimension 3, plusieurs algorithmes utilisant le modèle PARAFAC précédent et se basant sur des procédures itératives pour mettre à jour les matrices A , B et C sont établis. Ils minimisent des fonctions de coût différentes mais issues d'une même fonction de coût généralisée η .

$$\eta = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \sum_{n_3=1}^{N_3} \left(h_{n_1 n_2 n_3} - \sum_{p=1}^P a_{n_1 p} b_{n_2 p} c_{n_3 p} \right)^2 \quad (3.74)$$

où $h_{n_1 n_2 n_3}$ est l'élément (n_1, n_2, n_3) du tenseur \mathbb{H} , $a_{n_1 p}$ est l'élément (n_1, p) de la matrice A de dimension $(N_1 \times P)$, $b_{n_2 p}$ est l'élément (n_2, p) de la matrice B de dimension $(N_2 \times P)$, $c_{n_3 p}$ est l'élément (n_3, p) de la matrice C de dimension $(N_3 \times P)$ et P est le nombre de facteurs de ce modèle.

Dans ce qui suit, trois formes différentes de l'algorithme ALS-PARAFAC sont présentés. Leurs démonstrations sont détaillées dans l'annexe E. Définissons d'abord les trois fonctions de coût $\eta_1^{N_1}$, $\eta_1^{N_2}$ et $\eta_1^{N_3}$ suivantes :

$$\eta_1^{N_1} = \sum_{n_1=1}^{N_1} \|H_{n_1..} - BD_{n_1}^A C^T\|_F^2 \quad (3.75)$$

$$\eta_1^{N_2} = \sum_{n_2=1}^{N_2} \|H_{..n_2} - CD_{n_2}^B A^T\|_F^2 \quad (3.76)$$

$$\eta_1^{N_3} = \sum_{n_3=1}^{N_3} \|H_{..n_3} - AD_{n_3}^C B^T\|_F^2 \quad (3.77)$$

Algorithme 1

L'algorithme 1 met à jour les lignes des matrices A , B et C en minimisant au sens des moindres carrés l'une des fonctions de coût $\eta_1^{N_1}$, $\eta_1^{N_2}$ et $\eta_1^{N_3}$ respectivement. On obtient :

$$D_{n_1}^A = (B^T B)^{-1} B^T (H_{n_1..}) C (C^T C)^{-1} \quad n_1 = 1, \dots, N_1 \quad (3.78)$$

$$D_{n_2}^B = (C^T C)^{-1} C^T (H_{..n_2}) A (A^T A)^{-1} \quad n_2 = 1, \dots, N_2 \quad (3.79)$$

$$D_{n_3}^C = (A^T A)^{-1} A^T (H_{..n_3}) B (B^T B)^{-1} \quad n_3 = 1, \dots, N_3 \quad (3.80)$$

avec

$$D_{n_1}^A = \mathbf{diag}(A_{n_1..}) \quad (3.81)$$

$$D_{n_2}^B = \mathbf{diag}(B_{..n_2}) \quad (3.82)$$

$$D_{n_3}^C = \mathbf{diag}(C_{..n_3}) \quad (3.83)$$

Algorithme 2

Cet algorithme, mis au point par Bro et Sidiropoulos [6], utilise l'expression de l'information du tenseur à l'aide de la famille des matrices $\{H_{..n_3}\}_{n_3=1..N_3}$ définie dans la figure 3.4. Il met à jour les matrices A , B et les lignes de la matrice C en minimisant au sens des moindres carrés la même fonction de coût $\eta_1^{N_3}$. On obtient :

$$A = \left(\sum_{n_3=1}^{N_3} H_{..n_3} B D_{n_3}^C \right) \left(\sum_{n_3=1}^{N_3} D_{n_3}^C B^T B D_{n_3}^C \right)^{-1} \quad (3.84)$$

$$B = \left(\sum_{n_3=1}^{N_3} H_{..n_3}^T A D_{n_3}^C \right) \left(\sum_{n_3=1}^{N_3} D_{n_3}^C A^T A D_{n_3}^C \right)^{-1} \quad (3.85)$$

et

$$D_{n_3}^C = (A^T A)^{-1} A^T (H_{..n_3}) B (B^T B)^{-1}; \quad n_3 = 1, \dots, N_3 \quad (3.86)$$

Algorithme 3

Cet algorithme a été mis au point par Harshman [26] et Carroll et Chang [12]. Il utilise la deuxième représentation matricielle du modèle PARAFAC présenté dans la section précédente ainsi que le produit de Khatri-Rao ($| \otimes |$). Il met à jour les matrices A , B et C en minimisant respectivement au sens des moindres carrés les fonctions de coût $\eta_2^{N_1}$, $\eta_2^{N_2}$ et $\eta_2^{N_3}$ suivantes :

$$\eta_2^{N_1} = \|H_{N_1} - A(C| \otimes |B)^T\|_F^2 \quad (3.87)$$

$$\eta_2^{N_2} = \|H_{N_2} - B(A| \otimes |C)^T\|_F^2 \quad (3.88)$$

$$\eta_2^{N_3} = \|H_{N_3} - C(B| \otimes |A)^T\|_F^2 \quad (3.89)$$

On obtient :

$$\Psi_{N_1} = (C| \otimes |B) \implies A = H_{N_1} \Psi_{N_1} (\Psi_{N_1}^T \Psi_{N_1})^{-1} \quad (3.90)$$

$$\Psi_{N_2} = (A| \otimes |C) \implies B = H_{N_2} \Psi_{N_2} (\Psi_{N_2}^T \Psi_{N_2})^{-1} \quad (3.91)$$

$$\Psi_{N_3} = (B| \otimes |A) \implies C = H_{N_3} \Psi_{N_3} (\Psi_{N_3}^T \Psi_{N_3})^{-1} \quad (3.92)$$

Liens entre les trois algorithmes

• Liens entre l'algorithme1 et l'algorithme2

Les algorithmes 1 et 2 déterminent la matrice C de la même manière, c'est à dire en minimisant au sens des moindres carrés la même fonction de coût $\eta_1^{N_3}$ par rapport à $D_{n_3}^C$ qui contient l'information de la $n_3^{\text{ème}}$ ligne de la matrice C .

Contrairement à la matrice C , la mise à jour des matrices A et B diffère entre l'algorithme 1 et l'algorithme 2. Dans l'algorithme 1, la matrice A est déterminée ligne par ligne en minimisant la fonction de coût $\eta_1^{N_1}$ par rapport à chacune des lignes de A . Par contre dans l'algorithme 2, elle est déterminée en minimisant la fonction de coût $\eta_1^{N_3}$ par rapport à la matrice A .

\Rightarrow Utilisant des fonctions de coût différentes pour mettre à jour les matrices A et B , les algorithmes 1 et 2 sont différents même s'ils mettent à jour la matrice C de la même manière.

• Liens entre l'algorithme1 et l'algorithme3

Dans ce qui suit, on montre l'équivalence entre les deux fonctions $\eta_1^{N_3}$ et $\eta_2^{N_1}$. L'équivalence entre $\eta_1^{N_1}$ et $\eta_2^{N_2}$ et entre $\eta_1^{N_2}$ et $\eta_2^{N_3}$ se démontre de la même manière.

$$\eta_2^{N_1} = \|H_{N_1} - A(C| \otimes |B)^T\|_F^2 \quad (3.93)$$

$$= \sum_{n_3=1}^{N_3} \|H_{..n_3} - AD_{n_3}^C B^T\|_F^2 \quad (3.94)$$

$$= \eta_1^{N_3} \quad (3.95)$$

\Rightarrow Les algorithmes 1 et 3 minimisent donc au sens des moindres carrés, des fonctions de coût équivalentes. La mise à jour des matrices A , B et C du modèle PARAFAC reste néanmoins différente pour les deux algorithmes.

Prenons par exemple, la mise à jour de la matrice A . Elle est optimisée dans l'algorithme 1, en minimisant la fonction de coût $\eta_1^{N_1}$ et dans l'algorithme 3, en minimisant la

fonction de coût $\eta_2^{N_1}$ qui est différente de $\eta_1^{N_1}$ puisque qu'on a montré que cette dernière est équivalente à $\eta_2^{N_2}$. Pareil pour la mise à jour des matrices B et C qui est différente dans les algorithmes 1 et 3.

Ceci nous amène à conclure que les algorithmes 1 et 3 sont différents même s'ils utilisent des fonctions de coût équivalentes.

• Liens entre l'algorithme2 et l'algorithme3

La fonction de coût dans l'algorithme 2 est :

$$\eta_1^{N_3} = \sum_{n_3=1}^{N_3} \|H_{..n_3} - AD_{n_3}^C B^T\|_F^2 \quad (3.96)$$

En minimisant $\eta_1^{N_3}$ au sens des moindres carrés par rapport à A on obtient :

$$A = \left(\sum_{n_3=1}^{N_3} H_{..n_3} B D_{n_3}^C \right) \left(\sum_{n_3=1}^{N_3} D_{n_3}^C B^T B D_{n_3}^C \right)^{-1} \quad (3.97)$$

Dans l'algorithme 3, la fonction de coût utilisée pour déterminer A est :

$$\eta_2^{N_1} = \|H_{N_1} - A(C| \otimes |B)^T\|_F^2 \quad (3.98)$$

En minimisant $\eta_2^{N_1}$ au sens des moindres carrés par rapport à A on obtient :

$$A = H_{N_1} \Psi_{N_1} (\Psi_{N_1}^T \Psi_{N_1})^{-1} \quad \text{avec} \quad \Psi_{N_1} = (C| \otimes |B) \quad (3.99)$$

PROPOSITION : Equivalence entre les équations (3.97) et (3.99)

Dans ce qui suit, on montre que les formules (3.97) et (3.99) donnant la matrice A sont identiques.

DÉMONSTRATION :

Montrons que :

$$\left(\sum_{n_3=1}^{N_3} H_{..n_3} B D_{n_3}^C \right) \left(\sum_{n_3=1}^{N_3} D_{n_3}^C B^T B D_{n_3}^C \right)^{-1} = H_{N_1} \Psi_{N_1} (\Psi_{N_1}^T \Psi_{N_1})^{-1} \quad (3.100)$$

d'après l'équation (E.60), on a :

$$H_{N_1} \Psi_{N_1} = \sum_{n_3=1}^{N_3} H_{..n_3} B D_{n_3}^C \quad (3.101)$$

Il suffit donc de montrer que :

$$\sum_{n_3=1}^{N_3} D_{n_3}^C B^T B D_{n_3}^C = \Psi_{N_1}^T \Psi_{N_1} \quad (3.102)$$

Rappelons que $D_{n_3}^C$ est une matrice carrée dont les éléments diagonaux sont les éléments de la $n_3^{\text{ème}}$ ligne de la matrice C et posons $\Theta = B^T B$. on a alors :

$$\sum_{n_3=1}^{N_3} D_{n_3}^C B^T B D_{n_3}^C = \sum_{n_3=1}^{N_3} D_{n_3}^C \Theta D_{n_3}^C \quad (3.103)$$

$$= \begin{pmatrix} \left(\sum_{n_3=1}^{N_3} c_{n_3 1}^2 \right) \theta_{11} & \cdots & \left(\sum_{n_3=1}^{N_3} c_{n_3 1} c_{n_3 P} \right) \theta_{1P} \\ \vdots & & \vdots \\ \left(\sum_{n_3=1}^{N_3} c_{n_3 P} c_{n_3 1} \right) \theta_{P1} & \cdots & \left(\sum_{n_3=1}^{N_3} c_{n_3 P}^2 \right) \theta_{PP} \end{pmatrix} \quad (3.104)$$

$$= \Theta \circ \begin{pmatrix} \sum_{n_3=1}^{N_3} c_{n_3 1}^2 & \cdots & \sum_{n_3=1}^{N_3} c_{n_3 1} c_{n_3 P} \\ \vdots & & \vdots \\ \sum_{n_3=1}^{N_3} c_{n_3 P} c_{n_3 1} & \cdots & \sum_{n_3=1}^{N_3} c_{n_3 P}^2 \end{pmatrix} \quad (3.105)$$

$$= (B^T B) \circ (C^T C) \quad (3.106)$$

Or d'après l'équation (E.66), on a trouvé que : $(B^T B) \circ (C^T C) = \Psi_{N_1}^T \Psi_{N_1}$ d'où :

$$\sum_{n_3=1}^{N_3} D_{n_3}^C B^T B D_{n_3}^C = \Psi_{N_1}^T \Psi_{N_1} \quad \blacksquare \quad (3.107)$$

\Rightarrow La mise à jour de la matrice A est donc équivalente pour les algorithmes 2 et 3. Ce qui n'est pas le cas pour la mise à jour des matrices B et C .

Conclusions :

Les trois algorithmes utilisant la technique des moindres carrés alternés (Alternating Least Square **ALS**) sont différents. Néanmoins leurs complexités de calcul sont équivalentes vu que dans ces trois algorithmes, les matrices à inverser au cours de la mise à jour sont toutes de dimension $(P \times P)$.

Plusieurs autres algorithmes ALS adaptés aux différentes décompositions tensorielles PARAFAC2, TUCKER3 et PARATUCK2 ont été mis au point dans la littérature. Tous estiment dans le sens des moindres carrés alternés les éléments de ces décompositions tensorielles.

3.4 Représentation des modèles de Volterra réduits à l'aide de l'algorithme ALS-PARAFAC

Dans ce paragraphe on met en valeur l'efficacité de la modélisation PARAFAC-Volterra pour réduire la complexité des modèles de Volterra.

La décomposition PARAFAC est unique [5] et applicable à n'importe quel tenseur. Soit \mathbb{H} un noyau de Volterra cubique. Il peut être considéré comme un tenseur d'ordre 3 et de dimension $(M \times M \times M)$ avec une complexité en terme de nombre de coefficients $C_{standard} = M^3$.

Le modèle PARAFAC-Volterra a une complexité $C_{parafac} = 3MP$ (matrices A , B et C de dimension $M \times P$ chacune). D'où, le Taux de Réduction de Complexité (TRC) en utilisant PARAFAC par rapport au modèle standard de Volterra cubique $TRC = \frac{3MP}{M^3} = \frac{3P}{M^2}$. Quand $P \ll M$, on obtient une importante réduction de complexité.

A partir du TRC, on peut conclure que la réduction de complexité est possible quand $P < \frac{M^2}{3}$.

3.4.1 L'algorithme ALS-PARAFAC

Cet algorithme utilise la représentation matricielle du modèle PARAFAC donnée par les équations (3.4), (3.5) et (3.6) et basé sur une procédure alternée. Il met à jour les matrices A , B et C en minimisant les fonctions de coût η_{N_1} , η_{N_2} et η_{N_3} définies dans (3.75), (3.76) et (3.77).

En alternant la minimisation des trois fonctions de coût η_{N_1} , η_{N_2} et η_{N_3} précédentes, on obtient les équations (3.78), (3.79) et (3.80) afin de mettre à jour les matrices A , B et C .

3.4.2 Résultats de simulations

Le noyau de Volterra cubique simulé est défini comme suit :

$$h_{n_1 n_2 n_3} = \begin{cases} 0.2(n_2 + n_3) \exp(-0.07n_1 n_2 n_3) & \text{si } n_1 \neq 1, n_2 \neq 1 \text{ ou } n_3 \neq 1 \\ 0 & \text{sinon} \end{cases} \quad (3.108)$$

où $n_1, n_2, n_3 = 1, \dots, M$; $M = 20$.

La matrice $H_{N_1 \times 5N_2} = \begin{bmatrix} H_{..1} & \dots & H_{..5} \end{bmatrix}$ est représentée dans la figure 3.8.

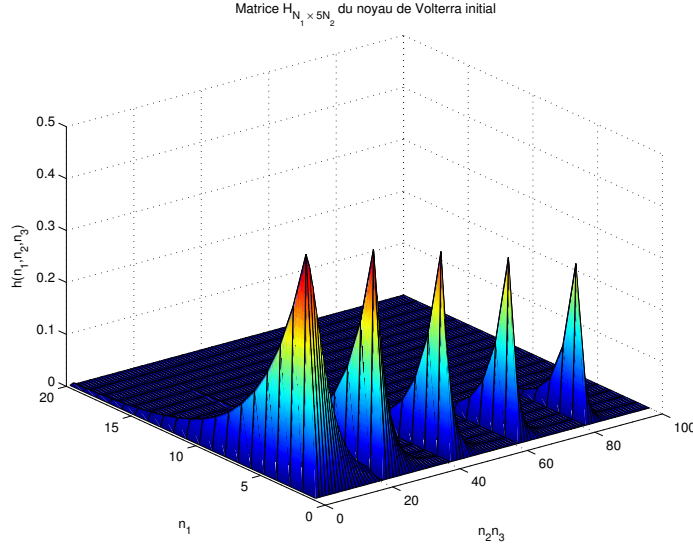


FIG. 3.8 – La matrice $H_{N_1 \times 5N_2}$ issue du noyau de Volterra initial

L'Erreur Quadratique Moyenne Normalisée ($EQMN_{Noyau}$) entre le noyau de Volterra initial \mathbb{H} et le noyau reconstitué est calculée comme suit :

$$EQMN_{Noyau} = \frac{\sum_{n_1=1}^M \sum_{n_2=1}^M \sum_{n_3=1}^M \left(h_{n_1 n_2 n_3} - \sum_{p=1}^P a_{n_1 p} b_{n_2 p} c_{n_3 p} \right)^2}{\sum_{n_1}^M \sum_{n_2}^M \sum_{n_3}^M h_{n_1 n_2 n_3}^2} \quad (3.109)$$

Comme on peut le voir dans le tableau 3.1, La méthode ALS-PARAFAC permet de reconstruire le noyau de Volterra initial avec une $EQMN_{Noyau}$ satisfaisante. Elle permet aussi de réduire de manière significative la complexité du modèle original de $C_{standard} = M^3 = 8000$ à $C_{parafac} = 3MP = 60P$.

La figure 3.9 représente $EQMN_{Noyau}$ en fonction du nombre d'itérations de l'algorithme ALS-PARAFAC. On peut conclure que pour un nombre de facteurs du modèle PARAFAC P fixé, l'algorithme ALS-PARAFAC converge rapidement en deux ou trois itérations. En augmentant davantage le nombre de facteurs P , on peut faire décroître $EQMN_{Noyau}$ d'une manière significative.

P	$C_{parafac}$	TRC	$EQMN_{Noyau}$
1	60	0.0075	$2.12 \cdot 10^{-2}$
2	120	0.015	$3.73 \cdot 10^{-4}$
3	180	0.0225	$4.71 \cdot 10^{-6}$
4	240	0.03	$4.01 \cdot 10^{-8}$
5	300	0.0375	$2.39 \cdot 10^{-10}$
6	360	0.045	$9.76 \cdot 10^{-13}$

TAB. 3.1 – Complexité, TRC et $EQMN_{Noyau}$ en fonction du nombre P de facteurs du modèle PARAFAC

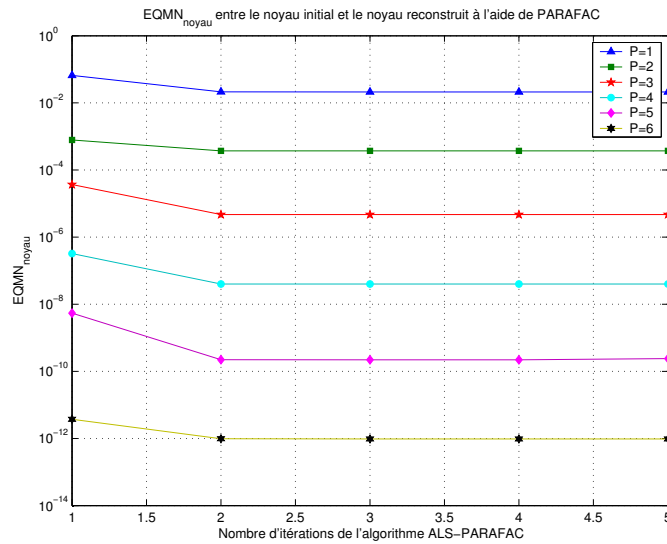


FIG. 3.9 – $EQMN_{Noyau}$ entre le noyau de Volterra non-séparable initial et le noyau reconstruit en utilisant PARAFAC

Comparons maintenant la réduction de complexité obtenue en utilisant le modèle PARAFAC avec celle obtenue en utilisant le développement du noyau de Volterra sur une base de fonctions de Laguerre. Le pôle optimal de Laguerre est donné par l'algorithme de Campello et al [11], $\xi_{opt} = 0.665$. La figure 3.10 présente $EQMN_{Noyau}$ entre le noyau de Volterra initial et celui reconstruit à l'aide des bases de fonctions de Laguerre, pour différents ordres de troncature.

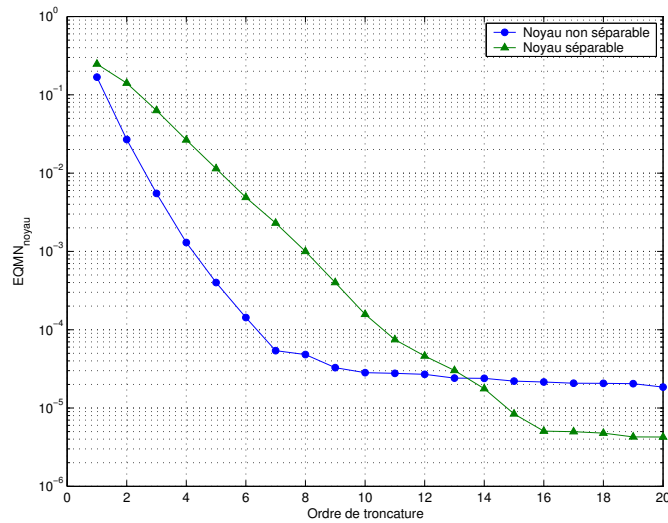


FIG. 3.10 – $EQMN_{Noyau}$ entre le noyau de Volterra original et le noyau reconstruit en utilisant le développement sur des bases de fonctions de Laguerre

Des résultats de la figure 3.10, on peut noter qu'en utilisant l'approche basée sur les bases de fonctions de Laguerre, $EQMN_{Noyau}$ converge à partir d'un ordre de troncature $K = 9$ relatif à $EQMN_{Noyau} = 3.27 \cdot 10^{-5}$, ce qui est équivalent à une complexité $C_{Laguerre} = \frac{K(K+1)(K+2)}{6} = 165$ coefficients. Pour obtenir des valeurs de $EQMN_{Noyau}$ plus petites et à partir des résultats du tableau 3.1, l'approche PARAFAC utilise un nombre de facteurs $P = 3$, ce qui équivalent à une complexité $C_{parafac} = 180$ paramètres. Ceci montre que pour ce noyau non séparable, les performances des approches PARAFAC et celle basée sur les bases de fonctions de Laguerre sont proches.

Dans un deuxième exemple, on a simulé un système de Volterra de mémoire $M = 20$ (donc de complexité égale à celle du noyau précédent), à noyau séparable défini comme

suit :

$$h_{n_1 n_2 n_3} = g(n_1)g(n_2)g(n_3) \quad (3.110)$$

où $g(\cdot)$ est la réponse impulsionnelle du système du second ordre défini par sa fonction de transfert $G(z)$.

$$G(z) = \frac{z + 0.5}{(z - 0.75)(z - 0.2)} \quad (3.111)$$

La matrice $H_{N_1 \times 5N_2} = \begin{bmatrix} H_{..1} & \cdots & H_{..5} \end{bmatrix}$ est représentée dans la figure 3.11.

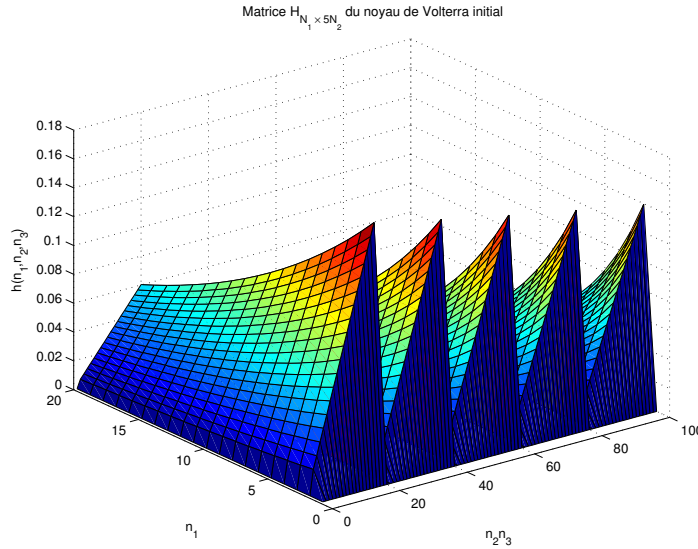


FIG. 3.11 – La matrice $H_{N_1 \times 5N_2}$ issue du noyau de Volterra initial

La figure 3.12 représente $EQMN_{Noyau}$ en fonction du nombre d'itérations de l'algorithme ALS-PARAFAC. On remarque que quelque soit le nombre de facteurs P du modèle PARAFAC, l'algorithme ALS-PARAFAC converge à peu près vers la même valeur $EQMN_{Noyau} = 1.93 \cdot 10^{-31}$. Cette reconstitution remarquable du noyau initial en utilisant la décomposition PARAFAC s'explique par la séparabilité du noyau de Volterra. En fait, les colonnes de la matrice $H_{N_1 \times N_2 N_3}$, comme celles de $H_{N_2 \times N_1 N_3}$ et $H_{N_3 \times N_1 N_2}$, sont linéairement dépendantes, donc le rang du noyau tenseur est égal à 1, selon la définition du rang de tenseur de Kruskal [38]. Ceci implique qu'on peut choisir le nombre de facteur du modèle PARAFAC $P = 1$.

En utilisant le développement sur des bases de fonctions de Laguerre avec un pôle optimal de Laguerre $\xi_{opt} = 0.765$, on atteint une $EQMN_{Noyau} = 4.27 \cdot 10^{-6}$

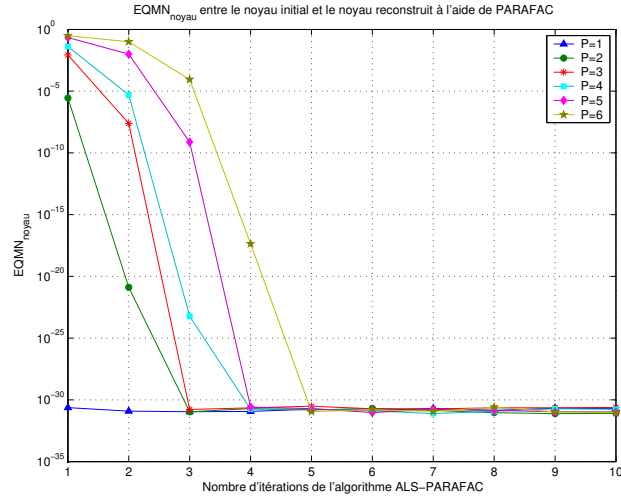


FIG. 3.12 – $EQMN_{Noyau}$ entre le noyau de Volterra séparable initial et le noyau reconstruit en utilisant PARAFAC

pour un ordre de troncature $K = 16$, ce qui correspond à un Taux de Réduction de Complexité $TRC = 0.204$. Tandis que l'algorithme ALS-PARAFAC converge vers $EQMN_{Noyau} = 1.93 \cdot 10^{-31}$ avec un nombre de facteur de PARAFAC égal à $P = 1$ ce qui correspond à $TRC = 7.5 \cdot 10^{-3}$. En raison de la séparabilité du noyau du système de Volterra simulé, les performances de l'approche PARAFAC sont nettement supérieures à celles de l'approche basée sur les bases de fonctions de Laguerre. D'où l'intérêt d'utiliser la décomposition PARAFAC pour des noyaux séparables puisque leur rang selon Kruskal [38] est égal à l'unité.

Chapitre 4

Modélisation et Identification des systèmes de Volterra à l'aide de l'approche tensorielle

4.1 Introduction

La relation Entrée/Sortie d'un système de Volterra d'ordre K discret, invariant dans le temps est donnée par :

$$y(n) = \sum_{k=1}^K \sum_{n_1=1}^{M_k} \cdots \sum_{n_k=1}^{M_k} h_m(n_1, \dots, n_k) \prod_{j=1}^k u(n - n_j) \quad (4.1)$$

où $\{h_m(n_1, \dots, n_k)\}$ représente les coefficients du noyau de Volterra d'ordre k . Ce dernier peut être considéré comme un tenseur d'ordre k et de dimension $\underbrace{M \times \dots \times M}_{k \text{ fois}}$ de rang P_k . $\{M_k\}_{k=1}^K$ sont les mémoires des noyaux de Volterra.

On propose des modèles de Volterra à complexité réduite. En considérant le noyau quadratique comme une matrice et les autres noyaux d'ordres supérieurs à 2 comme des tenseurs, nous utilisons une décomposition en valeurs singulières (SVD) pour le noyau quadratique et la décomposition tensorielle PARAFAC définie dans le chapitre précédent pour les autres noyaux.

Un nouveau modèle est présenté : SVD-PARAFAC-Volterra relatif aux systèmes de Volterra non homogènes. L'algorithme ARLS (Alternating Recursive Least Squares

algorithm) est utilisé pour estimer les paramètres de tels modèles de Volterra.

4.2 Le modèle SVD-PARAFAC-Volterra

4.2.1 La décomposition SVD (Singular Value Decomposition)

La décomposition SVD est utilisée dans le domaine du traitement du signal depuis les années 1970. Elle est décrite comme la représentation générale la plus informative d'une matrice. En fait, elle est liée à beaucoup de problèmes fondamentaux d'algèbre linéaire et est souvent utilisée dans l'analyse et le calcul matriciel.

La décomposition SVD d'une matrice X de dimension $(m \times n)$ de rang R est une factorisation de la forme :

$$X = U\Lambda V^T \quad (4.2)$$

où U et V sont des matrices unitaires de dimensions respectives $(m \times m)$ et $(n \times n)$ dont les colonnes représentent respectivement les vecteurs singuliers de gauche, les vecteurs singuliers de droite de la matrice X et Λ est une matrice de dimension $(m \times n)$ ayant R entrées diagonales positives $\{\lambda_r\}$ et $\min(m, n) - R$ entrées diagonales nulles, où $\{\lambda_r/\lambda_r \geq \lambda_{r+1}\}_{r=1, \dots, \min(m, n)}$ sont les valeurs singulières de X . On peut aussi écrire la décomposition SVD de la matrice X comme suit :

$$X = \sum_{r=1}^R \lambda_r U_{.r} V_{.r}^T \quad (4.3)$$

où $U_{.r}$ et $V_{.r}$ sont respectivement la $r^{\text{ème}}$ colonne de U et de V .

On peut aussi écrire X comme suit :

$$X = \sum_{r=1}^R A_{.r} B_{.r}^T \quad (4.4)$$

où

$$A_{.r} = \lambda_r U_{.r} \quad (4.5)$$

$$B_{.r} = V_{.r} \quad (4.6)$$

avec les contraintes suivantes :

$$A^T A = \Lambda_R^2 \quad (4.7)$$

$$B^T B = I_R \quad (4.8)$$

où $\Lambda_R = \text{diag}(\lambda_1, \dots, \lambda_R)$ et I_R la matrice identité d'ordre R .

La représentation scalaire de l'équation (4.4) s'écrit :

$$X(n_1, n_2) = \sum_{r=1}^R a_{n_1 r} b_{n_2 r} \quad (4.9)$$

où $a_{n_1 r}$ représente l'élément (n_1, r) de la matrice A , $b_{n_2 r}$ l'élément (n_2, r) de la matrice B .

4.2.2 La décomposition PARAFAC généralisée

Rappelons que d'après (Sidiropoulos et R. Bro [58]), tout tenseur \mathbb{H} d'ordre k et de dimension $(M \times M \times \dots \times M)$ peut s'écrire à l'aide de k matrices $\{X^{(i)}\}_{i=1, \dots, k}$, sous la forme d'un modèle PARAFAC multilinéaire. La représentation scalaire du modèle PARAFAC est la suivante :

$$h_k(n_1, n_2, \dots, n_k) = \sum_{p_k=1}^{P_k} x_{n_1 p_k}^{(1,k)} x_{n_2 p_k}^{(2,k)} \dots x_{n_k p_k}^{(k,k)} ; \quad \begin{array}{l} n_1 = 1, \dots, M \\ n_2 = 1, \dots, M \\ \vdots \\ n_k = 1, \dots, M \end{array} \quad (4.10)$$

où $h_k(n_1, n_2, \dots, n_k)$ est l'élément (n_1, n_2, \dots, n_k) du tenseur \mathbb{H} , $x_{n_i p_k}^{(i,k)}$ l'élément (n_i, p_k) de la matrice $X^{(i,k)}$ et P_k est le nombre de facteurs de la décomposition PARAFAC du tenseur \mathbb{H} .

4.2.3 Le modèle SVD-PARAFAC-Volterra

En utilisant la représentation scalaire (4.9) de la décomposition SVD et celle du modèle PARAFAC (4.10), l'équation (4.1) devient :

$$\begin{aligned}
\widehat{y}(n) &= \sum_{n_1=1}^{M_1} h_1(n_1)u(n-n_1) + \sum_{n_1=1}^{M_2} \sum_{n_2=1}^{M_2} \left(\sum_{p_2=1}^{P_2} x_{n_1 p_2}^{(1,2)} x_{n_2 p_2}^{(2,2)} \right) u(n-n_1)u(n-n_2) \\
&+ \sum_{n_1=1}^{M_3} \sum_{n_2=1}^{M_3} \sum_{n_3=1}^{M_3} \left(\sum_{p_3=1}^{P_3} x_{n_1 p_3}^{(1,3)} x_{n_2 p_3}^{(2,3)} x_{n_3 p_3}^{(3,3)} \right) u(n-n_1)u(n-n_2)u(n-n_3) \\
&+ \dots \\
&+ \sum_{n_1=1}^{M_K} \sum_{n_2=1}^{M_K} \dots \sum_{n_K=1}^{M_K} \left(\sum_{p_K=1}^{P_K} x_{n_1 p_K}^{(1,K)} x_{n_2 p_K}^{(2,K)} \dots x_{n_K p_K}^{(K,K)} \right) \prod_{i=1}^M u(n-n_i) \tag{4.11}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{n_1=1}^{M_1} h_1(n_1)u(n-n_1) \\
&+ \sum_{p_2=1}^{P_2} \left(\sum_{n_1=1}^{M_2} x_{n_1 p_2}^{(1,2)} u(n-n_1) \right) \left(\sum_{n_2=1}^{M_2} x_{n_2 p_2}^{(2,2)} u(n-n_2) \right) \\
&+ \sum_{p_3=1}^{P_3} \left(\sum_{n_1=1}^{M_3} x_{n_1 p_3}^{(1,3)} u(n-n_1) \right) \left(\sum_{n_2=1}^{M_3} x_{n_2 p_3}^{(2,3)} u(n-n_2) \right) \left(\sum_{n_3=1}^{M_3} x_{n_3 p_3}^{(3,3)} u(n-n_3) \right) \\
&+ \dots \\
&+ \sum_{p_K=1}^{P_K} \left(\sum_{n_1=1}^{M_K} x_{n_1 p_K}^{(1,K)} u(n-n_1) \right) \dots \left(\sum_{n_K=1}^{M_K} x_{n_K p_K}^{(K,K)} u(n-n_K) \right) \tag{4.12}
\end{aligned}$$

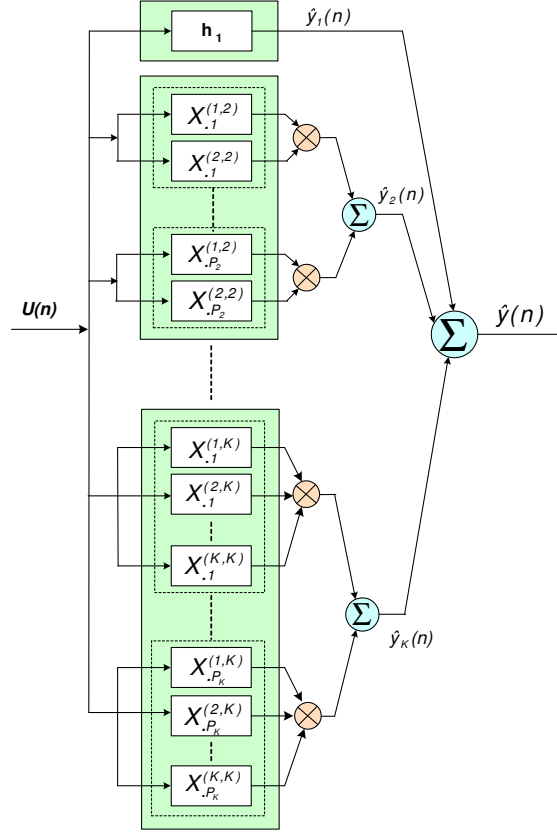
Dans ce qui suit, on utilise des modèles de Volterra dont les noyaux ont tous une mémoire égale à M .

La relation entrée/sortie (4.12) peut être implémentée en utilisant une structure parallèle-cascade, comme le montre la figure 4.1, où $U(n) = \begin{bmatrix} u(n-1) & \dots & u(n-M) \end{bmatrix}^T$, h_1 est le vecteur contenant les coefficients du noyau linéaire de Volterra. $X_i^{(1,2)}$ et $X_j^{(2,2)}$ représentent respectivement la $i^{\text{ème}}$ colonne de la matrice $X^{(1,2)}$ et la $j^{\text{ème}}$ colonne de la matrice $X^{(2,2)}$ et les boîtes d'ordre p_2 correspondent aux opérations de convolution $U^T(n)X_{p_2}^{(1,2)}$ et $U^T(n)X_{p_2}^{(2,2)}$. Pour le noyau d'ordre K , les boîtes d'ordre p_K correspondent aux opérations de convolution $U^T(n)X_{p_K}^{(1,K)}$, $U^T(n)X_{p_K}^{(2,K)}$ et $U^T(n)X_{p_K}^{(K,K)}$.

Le noyau de Volterra d'ordre K peut être considéré comme un tenseur d'ordre K , de dimension $\underbrace{(M \times M \times \dots \times M)}_{K \text{ fois}}$, avec une complexité numérique $C_K = M^K$ en termes de nombre de ses coefficients. Par conséquent, la complexité d'un modèle de Volterra standard d'ordre K est $C_{\text{Standard}}^{(K)} = \sum_{i=1}^K M^i$.

La complexité du modèle PARAFAC homogène d'ordre k est $C_{\text{parafac}}^{(k)} = kMP_k$ et celle du modèle SVD quadratique est $C_{\text{SVD}} = 2MP_2$. Donc, le modèle SVD-PARAFAC-Volterra d'ordre K a une complexité $C_{\text{SVD-PARAFAC}} = M(1 + \sum_{i=2}^K iP_i)$.

En utilisant la décomposition SVD-PARAFAC, le Taux de Réduction de Complexité

FIG. 4.1 – Structure parallèle-cascade du modèle SVD-PARAFAC-Volterra d'ordre K

(TRC) par rapport au modèle de Volterra d'ordre K standard est $TRC = \frac{1 + \sum_{i=1}^K i P_i}{\sum_{i=1}^K M^{i-1}}$. D'où, quand $P_i \ll M$, la complexité du modèle de Volterra est considérablement réduite.

Cas particulier : Le Modèle SVD-PARAFAC-Volterra d'ordre 3

En se basant sur l'écriture de la sortie du modèle SVD-PARAFAC-Volterra (4.12), la sortie d'un modèle SVD-PARAFAC-Volterra d'ordre 3 s'écrit :

$$\begin{aligned} \hat{y}(n) &= \sum_{n_1=1}^M h_1(n_1)u(n - n_1) \\ &+ \sum_{r=1}^R \left(\sum_{n_1=1}^M a_{n_1 r} u(n - n_1) \right) \left(\sum_{n_2=1}^M b_{n_2 r} u(n - n_2) \right) \\ &+ \sum_{p=1}^P \left(\sum_{n_1=1}^M c_{n_1 p} u(n - n_1) \right) \left(\sum_{n_2=1}^M d_{n_2 p} u(n - n_2) \right) \left(\sum_{n_3=1}^M e_{n_3 p} u(n - n_3) \right) \end{aligned} \quad (4.13)$$

où h_1 est le noyau linéaire de Volterra, les matrices A et B sont les éléments de la décomposition SVD du noyau quadratique et C , D et E sont les éléments de la décomposition PARAFAC du noyau cubique.

La relation entrée/sortie (4.13) peut être implémentée en utilisant une structure

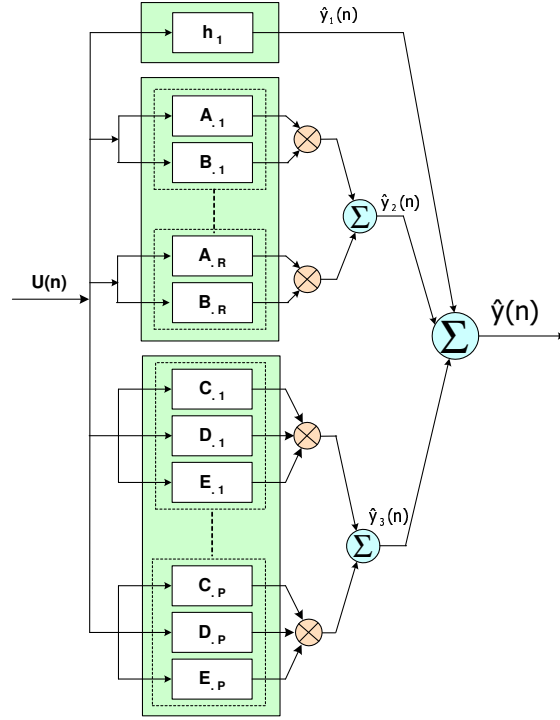


FIG. 4.2 – Réalisation d'une structure parallèle-cascade du modèle SVD-PARAFAC-Volterra d'ordre 3

parallèle-cascade, comme le montre la figure 4.2, où $U(n) = \begin{bmatrix} u(n-1) & \dots & u(n-M) \end{bmatrix}^T$, h_1 est le vecteur contenant les coefficients du noyau linéaire de Volterra. A_i et B_j représentent respectivement la $i^{\text{ème}}$ colonne de A et la $j^{\text{ème}}$ colonne de B et les boîtes d'ordre r correspondent aux opérations de convolutions $U^T(n)A_r$ et $U^T(n)B_r$. Pour le noyau cubique, les boîtes d'ordre p correspondent aux opérations de convolution $U^T(n)C_p$, $U^T(n)D_p$ et $U^T(n)E_p$.

Le noyau cubique peut être considéré comme un tenseur d'ordre 3, de dimension $(M \times M \times M)$, avec une complexité numérique $C_3 = M^3$ en termes de nombre de ses coefficients. Par conséquent, la complexité d'un modèle de Volterra standard d'ordre 3 est $C_{Standard} = M + M^2 + M^3$.

La complexité du modèle PARAFAC cubique est $C_{parafac} = 3MP$ et celle du modèle SVD quadratique est $C_{SVD} = 2MR$. Donc, Le modèle SVD-PARAFAC-Volterra d'ordre 3 a une complexité $C_{SVD-PARAFAC} = M(1 + 2R + 3P)$.

Le Taux de Réduction de Complexité (TRC) par rapport au modèle de Volterra d'ordre 3 standard est $TRC = \frac{1+2R+3P}{1+M+M^2}$. Quand $P \ll M$ et $R \ll M$, la complexité du

modèle de Volterra est considérablement réduite.

4.3 Identification des modèles SVD-PARAFAC-Volterra à l'aide de l'algorithme ARLS

En utilisant la représentation scalaire (4.9) de la décomposition SVD et celle de PARAFAC donnée dans (4.10), on obtient K écritures différentes de la sortie du modèle SVD-PARAFAC-Volterra d'ordre K . On définit pour $k = 2, \dots, K$:

$$\begin{aligned} \phi_{p_k}^{(i,k)}(n) &= \left(\sum_{n_1=1}^M x_{n_1 p_k}^{(1,k)} u(n - n_1) \right) \cdots \\ &\times \left(\sum_{n_{i-1}=1}^M x_{n_{i-1} p_k}^{(i-1,k)} u(n - n_{i-1}) \right) \left(\sum_{n_{i+1}=1}^M x_{n_{i+1} p_k}^{(i+1,k)} u(n - n_{i+1}) \right) \\ &\times \cdots \left(\sum_{n_K=1}^M x_{n_K p_k}^{(k,k)} u(n - n_K) \right) \end{aligned} \quad (4.14)$$

et

$$\Phi_{(i,k)}(n) = \left[\phi_1^{(i,k)}(n), \phi_2^{(i,k)}(n), \dots, \phi_{P_k}^{(i,k)}(n) \right]^T \quad (4.15)$$

4.3.1 Première écriture de la sortie du modèle

L'équation (4.12) s'écrit :

$$\hat{y}(n) = \sum_{n_1=1}^M \left[h_1(n_1) + \sum_{p_2=1}^{P_2} x_{n_1 p_2}^{(1,2)} \phi_{p_2}^{(1,2)}(n) + \cdots + \sum_{p_K=1}^{P_K} x_{n_1 p_K}^{(1,K)} \phi_{p_K}^{(1,K)}(n) \right] u(n - n_1) \quad (4.16)$$

$$= \underbrace{\begin{bmatrix} h_1 \\ \text{vec}(X^{(1,2)}) \\ \vdots \\ \text{vec}(X^{(1,K)}) \end{bmatrix}}_{\Theta_1^T} \underbrace{\left(\begin{bmatrix} 1 \\ \Phi_{(1,2)}(n) \\ \vdots \\ \Phi_{(1,K)}(n) \end{bmatrix} \otimes U(n) \right)}_{P_1(n)} \quad (4.17)$$

$$= \Theta_1^T P_1(n) \quad (4.18)$$

4.3.2 $k^{\text{ème}}$ écriture de la sortie du modèle

L'équation (4.12) s'écrit aussi :

$$\begin{aligned} \hat{y}(n) &= \sum_{i=1}^{k-1} \hat{y}_i(n) \\ &+ \sum_{n_k=1}^M \left[\sum_{p_k=1}^{P_k} x_{n_k p_k}^{(k,k)} \phi_{p_k}^{(k,2)}(n) + \cdots + \sum_{p_K=1}^{P_K} x_{n_k p_K}^{(k,K)} \phi_{p_K}^{(k,K)}(n) \right] u(n - n_k) \end{aligned} \quad (4.19)$$

$$= \sum_{i=1}^{k-1} \hat{y}_i(n) + \underbrace{\begin{bmatrix} \text{vec}(X^{(k,k)}) \\ \vdots \\ \text{vec}(X^{(k,K)}) \end{bmatrix}}_{\Theta_k^T} \underbrace{\left(\begin{bmatrix} \Phi_{(k,k)}(n) \\ \vdots \\ \Phi_{(k,K)}(n) \end{bmatrix} \otimes U(n) \right)}_{P_k(n)} \quad (4.20)$$

$$= \sum_{i=1}^{k-1} \hat{y}_i(n) + \Theta_k^T P_k(n) \quad (4.21)$$

où

$$\hat{y}_1(n) = \sum_{n_1=1}^M h_1(n_1) u(n - n_1) \quad (4.22)$$

et

$$\hat{y}_i(n) = \sum_{p_i=1}^{P_i} \phi_{p_i}^{(1,i)}(n) \cdots \phi_{p_i}^{(i,i)}(n); \text{ pour } i \geq 2 \quad (4.23)$$

4.3.3 L'algorithme ARLS

L'algorithme ARLS utilise les K écritures de la sortie du modèle SVD-PARAFAC-Volterra régies par l'équation (4.21), ($k = 1, \dots, K$). Il met à jour le noyau linéaire h_1 , les matrices $X^{(1,2)}$ et $X^{(2,2)}$ de la décomposition SVD du noyau quadratique et les matrices $X^{(i,j)}$, $i \leq j = 1, \dots, K$ des différentes décompositions PARAFAC des noyaux, en minimisant dans le sens des moindres carrés récursifs alternés la fonction de coût η suivante :

$$\eta(N) = \sum_{n=1}^N (y(n) - \hat{y}(n))^2 \quad (4.24)$$

où $y(n)$ représente la sortie du système à modéliser et $\hat{y}(n)$ la sortie du modèle SVD-PARAFAC-Volterra d'ordre K donnée par l'équation (4.12) et N le nombre de mesures.

En remplaçant la sortie du modèle $\hat{y}(n)$ par les différentes écritures (4.21), ($k = 1, \dots, K$), la fonction de coût $\eta(N)$ peut s'écrire comme suit :

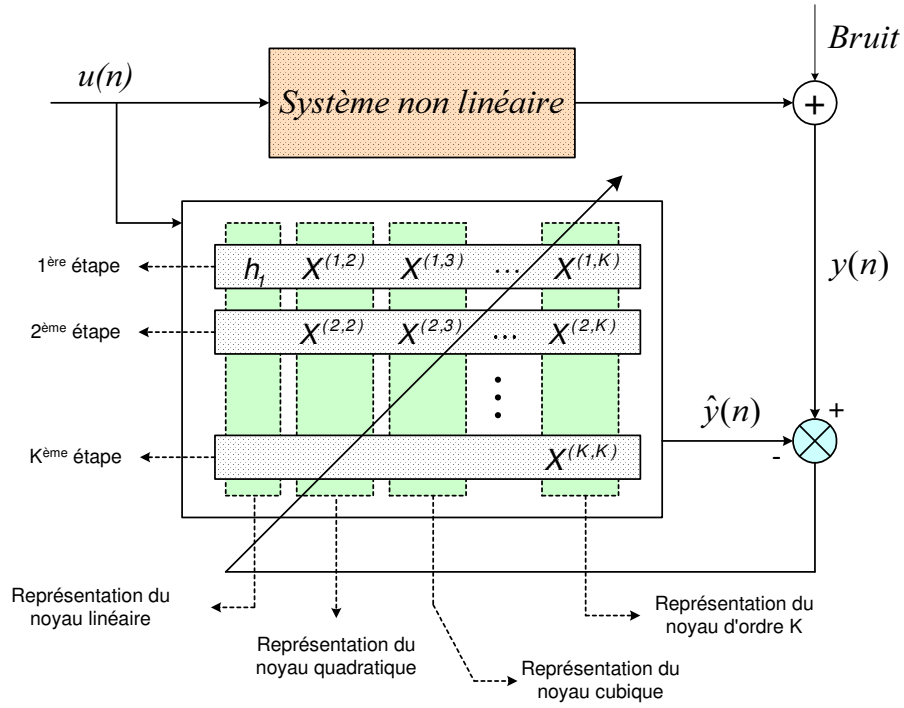


FIG. 4.3 – Mise à jour des composants du modèle SVD-PARAFAC-Volterra d'ordre K à l'aide de l'algorithme ARLS

$$\eta_1(N) = \sum_{n=1}^N (y(n) - \Theta_1^T P_1(n))^2 \quad (4.25)$$

$$\eta_2(N) = \sum_{n=1}^N (y(n) - \hat{y}_1(n) - \Theta_2^T P_2(n))^2 \quad (4.26)$$

$$\vdots \quad (4.27)$$

$$\eta_K(N) = \sum_{n=1}^N \left(y(n) - \sum_{i=1}^{K-1} \hat{y}_i(n) - \Theta_K^T P_K(n) \right)^2 \quad (4.28)$$

En minimisant alternativement les fonctions de coût $\eta_k(N)$ respectivement par rapport à Θ_k , ($k = 1, \dots, K$), on met à jour les matrices estimées $X^{(i,j)}$, $i, j = 1, \dots, K$, ainsi que le noyau linéaire h_1 . L'algorithme ARLS est illustré par la figure 4.3 et ses équations sont présentées dans le tableau 4.1.

1. Initialisation

- $\hat{\Theta}_i(0), \forall i = 1, \dots, K$
- $Q_i(0) = I_{M(1+\sum_{k=2}^i P_k)} \forall i = 1, \dots, K$

2. Mise à jour des composants du modèle SVD-PARAFAC-Volterra d'ordre K

- $U(n) = [u(n-1) \quad \dots \quad u(n-M)]^T$
- $\left\{ \begin{array}{l} \phi_{p_2}^{(1,2)}(n) \dots \phi_{p_K}^{(1,K)}(n) \\ \text{Construire } \Phi_{(1,2)}(n) \dots \Phi_{(1,K)}(n) \\ P_1(n) = [1 \quad \Phi_{(1,2)}^T(n) \quad \dots \quad \Phi_{(1,K)}^T(n)]^T \otimes U(n) \\ \varepsilon_1(n) = y(n) - P_1^T(n) \hat{\Theta}_1(n-1) \\ K_1(n) = \frac{Q_1(n-1)P_1(n)}{1+P_1^T(n)Q_1(n-1)P_1(n)} \\ Q_1(n) = [I - K_1(n)P_1^T(n)]Q_1(n-1) \\ \hat{\Theta}_1(n) = \hat{\Theta}_1(n-1) + K_1(n)\varepsilon_1(n) \\ \hat{\Theta}_1(n) = [\hat{h}_1^T(n) \quad (\text{vec } \hat{X}^{(1,2)}(n))^T \quad \dots \quad (\text{vec } \hat{X}^{(1,K)}(n))^T]^T \end{array} \right.$
- $\left\{ \begin{array}{l} \vdots \\ \vdots \end{array} \right.$
- $\left\{ \begin{array}{l} \text{Calculer } \hat{y}_i(n) \forall i = 1, \dots, k-1 \\ \text{Calculer } \phi_{p_2}^{(k,2)}(n) \dots \phi_{p_K}^{(k,K)}(n) \\ \text{Construire } \Phi_{(k,2)}(n) \dots \Phi_{(k,K)}(n) \\ P_k(n) = [1 \quad \Phi_{(k,2)}^T(n) \quad \dots \quad \Phi_{(k,K)}^T(n)]^T \otimes U(n) \\ \varepsilon_k(n) = y(n) - \sum_{i=1}^{k-1} \hat{y}_i(n) - P_k^T(n) \hat{\Theta}_k(n-1) \\ K_k(n) = \frac{Q_k(n-1)P_k(n)}{1+P_k^T(n)Q_k(n-1)P_k(n)} \\ Q_k(n) = [I - K_k(n)P_k^T(n)]Q_k(n-1) \\ \hat{\Theta}_k(n) = \hat{\Theta}_k(n-1) + K_k(n)\varepsilon_k(n) \\ \hat{\Theta}_k(n) = [(\text{vec } \hat{X}^{(k,k)}(n))^T \quad \dots \quad (\text{vec } \hat{X}^{(k,K)}(n))^T]^T \end{array} \right.$

3. Reconstruction des noyaux du modèle de Volterra

- noyau linéaire \hat{h}_1 .
- noyau d'ordre k :

$$h_k(n_1, n_2, \dots, n_k) = \sum_{p_k=1}^{P_k} x_{n_1 p_k}^{(1,k)} x_{n_2 p_k}^{(2,k)} \dots x_{n_k p_k}^{(k,k)} ; \quad \begin{array}{l} n_1 = 1, \dots, M \\ n_2 = 1, \dots, M \\ \vdots \\ n_k = 1, \dots, M \end{array}$$

4. Retour à l'étape 2 jusqu'à convergence de l'algorithme

TAB. 4.1 – L'algorithme ARLS (Cas général)

4.3.4 Identification des modèles SVD-PARAFAC-Volterra d'ordre 3 à l'aide de l'algorithme ARLS

En utilisant la représentation scalaire de la decomposition SVD dans (4.9) et celle de PARAFAC donnée dans (3.2), on obtient trois écritures différentes de la sortie du modèle SVD-PARAFAC-Volterra d'ordre 3. On définit,

$$\psi_r^A(n) = \sum_{n_2=1}^M b_{n_2 r} u(n - n_2) \quad (4.29)$$

$$\psi_r^B(n) = \sum_{n_1=1}^M a_{n_1 r} u(n - n_1) \quad (4.30)$$

$$\phi_p^C(n) = \sum_{n_2=1}^M d_{n_2 p} u(n - n_2) \sum_{n_3=1}^M e_{n_3 p} u(n - n_3) \quad (4.31)$$

$$\phi_p^D(n) = \sum_{n_1=1}^M c_{n_1 p} u(n - n_1) \sum_{n_3=1}^M e_{n_3 p} u(n - n_3) \quad (4.32)$$

$$\phi_p^E(n) = \sum_{n_1=1}^M c_{n_1 p} u(n - n_1) \sum_{n_2=1}^M d_{n_2 p} u(n - n_2) \quad (4.33)$$

Première écriture de la sortie du modèle

L'équation (4.13) s'écrit :

$$\hat{y}(n) = \sum_{n_1=1}^M \left[h_1(n_1) + \sum_{r=1}^R a_{n_1 r} \psi_r^A(n) + \sum_{p=1}^P c_{n_1 p} \phi_p^C(n) \right] \times u(n - n_1) \quad (4.34)$$

$$= \underbrace{\begin{bmatrix} h_1 \\ \text{vec}(A) \\ \text{vec}(C) \end{bmatrix}}_{\Theta_1^T} \underbrace{\left(\begin{bmatrix} 1 \\ \Psi_A(n) \\ \Phi_C(n) \end{bmatrix} \otimes U(n) \right)}_{P_1(n)} \quad (4.35)$$

$$= \Theta_1^T P_1(n) \quad (4.36)$$

où

$$\Psi_A(n) = \begin{bmatrix} \psi_1^A(n) & \psi_2^A(n) & \cdots & \psi_R^A(n) \end{bmatrix}^T \quad (4.37)$$

$$\Phi_C(n) = \begin{bmatrix} \phi_1^C(n) & \phi_2^C(n) & \cdots & \phi_P^C(n) \end{bmatrix}^T \quad (4.38)$$

Deuxième écriture de la sortie du modèle

L'équation (4.13) s'écrit aussi :

$$\widehat{y}(n) = y_L(n) + \sum_{n_2=1}^M \left[\sum_{r=1}^R b_{n_2 r} \psi_r^B(n) + \sum_{p=1}^P d_{n_2 p} \phi_p^D(n) \right] u(n - n_2) \quad (4.39)$$

$$= y_L(n) + \underbrace{\begin{bmatrix} \text{vec}(B) \\ \text{vec}(D) \end{bmatrix}}_{\Theta_2^T} \underbrace{\left(\begin{bmatrix} \Psi_B(n) \\ \Phi_D(n) \end{bmatrix} \otimes U(n) \right)}_{P_2(n)} \quad (4.40)$$

$$= y_L(n) + \Theta_2^T P_2(n) \quad (4.41)$$

où

$$y_L(n) = \sum_{n_1=1}^M h_1(n_1) u(n - n_1) \quad (4.42)$$

$$\Psi_B(n) = \begin{bmatrix} \psi_1^B(n) & \psi_2^B(n) & \cdots & \psi_R^B(n) \end{bmatrix}^T \quad (4.43)$$

$$\Phi_D(n) = \begin{bmatrix} \phi_1^D(n) & \phi_2^D(n) & \cdots & \phi_P^D(n) \end{bmatrix}^T \quad (4.44)$$

Troisième écriture de la sortie du modèle

La troisième écriture de l'équation (4.13) est :

$$\widehat{y}(n) = y_L(n) + y_Q(n) + \sum_{n_3=1}^M \left(\sum_{p=1}^P e_{n_3 p} \phi_p^E(n) \right) u(n - n_3) \quad (4.45)$$

$$= y_L(n) + y_Q(n) + \underbrace{[\text{vec}(E)]^T}_{\Theta_3^T} \underbrace{(\Phi_E(n) \otimes U(n))}_{P_3(n)} \quad (4.46)$$

$$= y_L(n) + y_Q(n) + \Theta_3^T P_3(n) \quad (4.47)$$

où

$$y_Q(n) = \sum_{r=1}^R \psi_r^A(n) \psi_r^B(n) \quad (4.48)$$

$$\Phi_E(n) = \begin{bmatrix} \phi_1^E(n) & \phi_2^E(n) & \cdots & \phi_P^E(n) \end{bmatrix}^T \quad (4.49)$$

L'algorithme ARLS

L'algorithme ARLS utilise les trois écritures de la sortie du modèle SVD-PARAFAC-Volterra d'ordre 3 définies par les équations (4.35), (4.40) et (4.46). Il met à jour le noyau linéaire h_1 , les matrices A et B de la décomposition SVD du noyau quadratique et les

matrices C , D et E de la décomposition PARAFAC du noyau cubique, en minimisant dans le sens des moindres carrés récursifs alternés la fonction de coût η suivante :

$$\eta(N) = \sum_{n=1}^N (y(n) - \hat{y}(n))^2 \quad (4.50)$$

où $y(n)$ représente la sortie du système à modéliser et $\hat{y}(n)$ la sortie du modèle SVD-PARAFAC-Volterra d'ordre 3 donnée par (4.13).

En remplaçant la sortie du modèle $\hat{y}(n)$ par ses trois différentes écritures (4.35), (4.40) et (4.46), la fonction de coût $\eta(N)$ peut s'écrire de trois manières différentes comme suit :

$$\eta_1(N) = \sum_{n=1}^N (y(n) - \Theta_1^T P_1(n))^2 \quad (4.51)$$

$$\eta_2(N) = \sum_{n=1}^N (y(n) - y_L(n) - \Theta_2^T P_2(n))^2 \quad (4.52)$$

$$\eta_3(N) = \sum_{n=1}^N (y(n) - y_L(n) - y_Q(n) - \Theta_3^T P_3(n))^2 \quad (4.53)$$

En minimisant alternativement les fonctions de coût $\eta_1(N)$, $\eta_2(N)$ et $\eta_3(N)$ respectivement par rapport à Θ_1 , Θ_2 et Θ_3 , on met à jour le noyau linéaire h_1 ainsi que les matrices estimées A , B , C , D et E éléments des décompositions SVD et PARAFAC des noyaux quadratique et cubique. L'algorithme ARLS est illustré dans la figure 4.4 et ses équations sont présentées dans le tableau 4.2.

1. Initialisation

- $\hat{\Theta}_1(0)$, $\hat{\Theta}_2(0)$ et $\hat{\Theta}_3(0)$
- $Q_1(0) = I_{M(P+R+1)}$, $Q_2(0) = I_{M(P+R)}$ et $Q_3(0) = I_{MP}$

2. Mise à jour des composants du modèle SVD-PARAFAC-Volterra d'ordre 3

- $U(n) = [u(n-1) \quad \dots \quad u(n-M)]^T$

$$\bullet \left\{ \begin{array}{l} \text{Calculer } \psi_r^A(n) \text{ et } \phi_p^C(n) \\ \text{Construire } \Psi_A(n) \text{ et } \Phi_C(n) \\ P_1(n) = [1 \quad \Psi_A^T(n) \quad \Phi_C^T(n)]^T \otimes U(n) \\ \varepsilon_1(n) = y(n) - P_1^T(n) \hat{\Theta}_1(n-1) \\ K_1(n) = \frac{Q_1(n-1)P_1(n)}{1+P_1^T(n)Q_1(n-1)P_1(n)} \\ Q_1(n) = [I - K_1(n)P_1^T(n)]Q_1(n-1) \\ \hat{\Theta}_1(n) = \hat{\Theta}_1(n-1) + K_1(n)\varepsilon_1(n) \\ \hat{\Theta}_1(n) = [\hat{h}_1^T(n) \quad (\text{vec } \hat{A}(n))^T \quad (\text{vec } \hat{C}(n))^T]^T \end{array} \right.$$

$$\bullet \left\{ \begin{array}{l} \text{Calculer } y_L(n), \psi_r^B(n) \text{ et } \phi_p^D(n) \\ \text{Construire } \Psi_B(n) \text{ et } \Phi_D(n) \\ P_2(n) = [\Psi_B^T(n) \quad \Phi_D^T(n)]^T \otimes U(n) \\ \varepsilon_2(n) = y(n) - y_L(n) - P_2^T(n) \hat{\Theta}_2(n-1) \\ K_2(n) = \frac{Q_2(n-1)P_2(n)}{1+P_2^T(n)Q_2(n-1)P_2(n)} \\ Q_2(n) = [I - K_2(n)P_2^T(n)]Q_2(n-1) \\ \hat{\Theta}_2(n) = \hat{\Theta}_2(n-1) + K_2(n)\varepsilon_2(n) \\ \hat{\Theta}_2(n) = [(\text{vec } \hat{B}(n))^T \quad (\text{vec } \hat{D}(n))^T]^T \end{array} \right.$$

$$\bullet \left\{ \begin{array}{l} \text{Calculer } y_Q(n) \text{ et } \phi_p^E(n) \\ \text{Construire } \Phi_E(n) \\ P_3(n) = \Phi_E(n) \otimes U(n) \\ \varepsilon_3(n) = y(n) - y_L(n) - y_Q(n) - P_3^T(n) \hat{\Theta}_3(n-1) \\ K_3(n) = \frac{Q_3(n-1)P_3(n)}{1+P_3^T(n)Q_3(n-1)P_3(n)} \\ Q_3(n) = [I - K_3(n)P_3^T(n)]Q_3(n-1) \\ \hat{\Theta}_3(n) = \hat{\Theta}_3(n-1) + K_3(n)\varepsilon_3(n) \\ \hat{\Theta}_3(n) = \text{vec}(\hat{E}(n)) \end{array} \right.$$

3. Reconstruction des noyaux du modèle de Volterra

- noyau linéaire \hat{h}_1 .
- noyau quadratique \hat{H}_2 :

$$\hat{h}_2(n_1, n_2) = \sum_{r=1}^R \hat{a}_{n_1 r} \hat{b}_{n_2 r} ; \quad \begin{array}{l} n_1 = 1, \dots, M \\ n_2 = 1, \dots, M \end{array}$$

- noyau cubique \hat{H}_3 :

$$\hat{h}_3(n_1, n_2, n_3) = \sum_{p=1}^P \hat{c}_{n_1 p} \hat{d}_{n_2 p} \hat{e}_{n_3 p} ; \quad \begin{array}{l} n_1 = 1, \dots, M \\ n_2 = 1, \dots, M \\ n_3 = 1, \dots, M \end{array}$$

4. Retour à l'étape 2 jusqu'à convergence de l'algorithme

TAB. 4.2 – L'algorithme ARLS (Cas particulier)

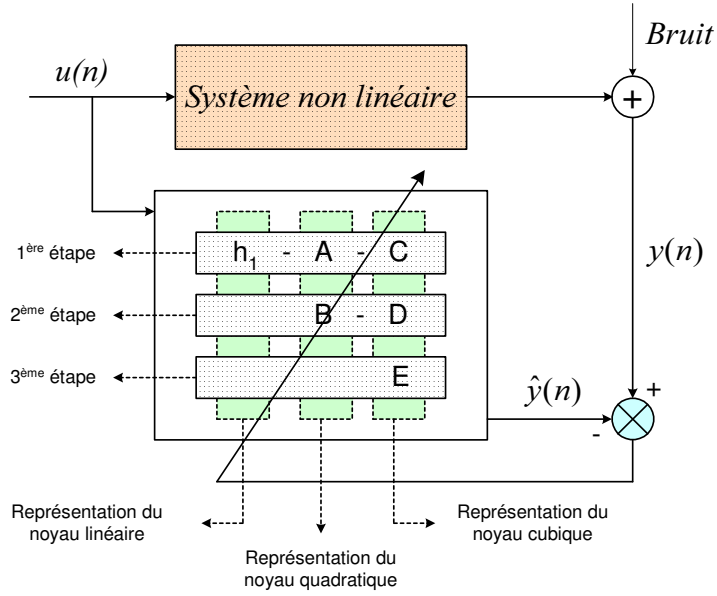


FIG. 4.4 – Mise à jour des composants du modèle SVD-PARAFAC-Volterra d'ordre 3 à l'aide de l'algorithme ARLS

4.4 Résultats de simulations

Deux systèmes sont simulés :

4.4.1 Premier système simulé

C'est un système de Volterra d'ordre 3 discret, invariant dans le temps. Ses noyaux sont définis comme suit :

$$h_j(n_1, \dots, n_j) = \prod_{i=1}^j g_{j,i}(n_i) \quad j = 1, 2, 3 \quad (4.54)$$

où $g_{j,i}(\cdot)$ est la réponse impulsionnelle du système défini par sa fonction de transfert $G_{j,i}(z)$.

$$\begin{aligned} G_{1,1}(z) &= \frac{z - 0.4}{(z + 0.8)(z - 0.5)} & ; & \quad G_{2,1}(z) = \frac{z - 0.3}{(z + 0.7)(z - 0.4)} ; \\ G_{2,2}(z) &= \frac{z - 0.8}{(z + 0.6)(z - 0.3)} & ; & \quad G_{3,1}(z) = \frac{z - 0.1}{(z + 0.7)(z - 0.5)} ; \\ G_{3,2}(z) &= \frac{z - 0.4}{(z + 0.8)(z - 0.3)} & \text{et} & \quad G_{3,3}(z) = \frac{z - 0.3}{(z + 0.3)(z - 0.5)} \end{aligned} \quad (4.55)$$

Chaque fonction de transfert est simulée avec une réponse impulsionnelle tronquée à l'ordre 5, ce qui veut dire que la mémoire M du modèle de Volterra est choisie égale à 5.

Le noyau linéaire h_1 , le noyau quadratique H_2 et la matrice latérale $H_{..1}$ du noyau cubique H_3 sont représentés dans la figure 4.5.

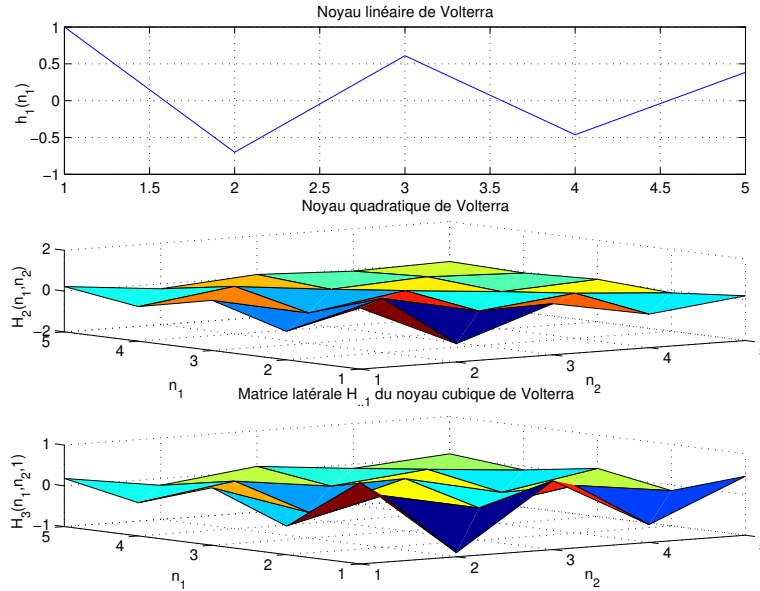


FIG. 4.5 – Les noyaux linéaire, quadratique et la matrice latérale $H_{..1}$ du noyau cubique du modèle de Volterra d'ordre 3

Comme les noyaux H_2 et H_3 sont séparables, donc $\text{rang}(H_2) = \text{rang}(H_3) = 1$. Par conséquent, le nombre de facteurs du modèle PARAFAC est fixé à $P = 1$ et le nombre de facteur de la décomposition SVD est $R = 1$ dans toutes les simulations pour ce premier système. La complexité numérique du modèle SVD-PARAFAC-Volterra est $C_{SVD-PARAFAC} = M(1 + 2P + 3R) = 30$ paramètres, ce qui correspond à un taux de réduction de complexité $TRC = 0.194$.

La performance de la méthode d'identification proposée est évaluée en utilisant l'Erreur Quadratique Moyenne Normalisée ($EQMN_{sortie}$) entre la sortie du système simulé $y(n)$ et celle du modèle SVD-PARAFAC-Volterra $\hat{\hat{y}}(n)$ reconstruit à l'aide des paramètres obtenus à la convergence de l'algorithme ARLS. Elle est calculée comme suit :

$$EQMN_{sortie}(N) = \frac{\sum_{n=1}^N \left(y(n) - \hat{\hat{y}}(n) \right)^2}{\sum_{n=1}^N y^2(n)} \quad (4.56)$$

La figure 4.6 montre les variations de $EQMN_{sortie}$ en fonction du nombre d'itérations pour trois RSBs différents. Le signal d'entrée est une séquence de bruit blanc

	RSB = 60dB	RSB = 20dB	RSB = 5dB
B.B.G	$5.399 \cdot 10^{-3}$	$7.511 \cdot 10^{-3}$	$2.015 \cdot 10^{-2}$
4PAM	$2.953 \cdot 10^{-2}$	$3.705 \cdot 10^{-2}$	$9.161 \cdot 10^{-2}$
6PAM	$1.842 \cdot 10^{-2}$	$2.897 \cdot 10^{-2}$	$5.723 \cdot 10^{-2}$

TAB. 4.3 – $EQMN_{sortie}$ obtenue avec le modèle SVD-PARAFAC-Volterra d'ordre 3 pour différents RSBs et différentes séquences d'entrée

gaussien de longueur $N = 5000$, de moyenne nulle et de variance l'unité. Les résultats des simulations sont obtenus en utilisant la méthode de Monte Carlo avec 20 séquences différentes de bruits gaussiens additifs.

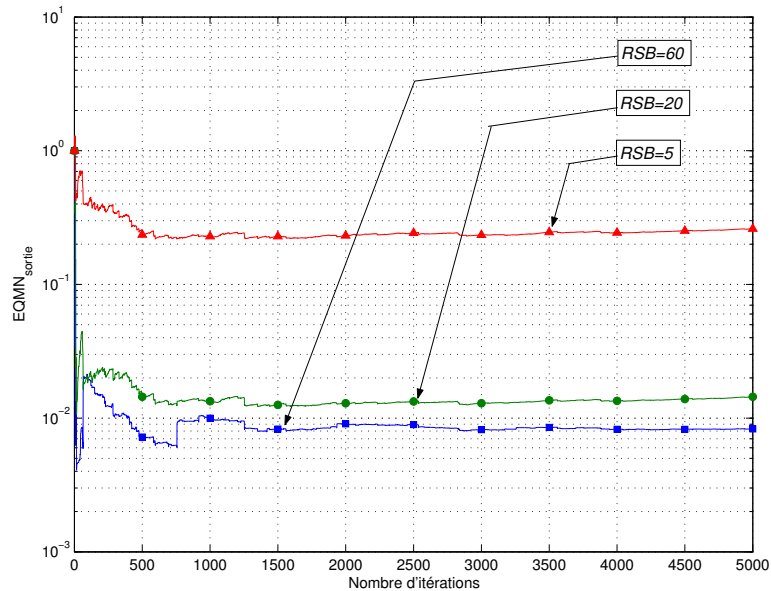


FIG. 4.6 – $EQMN_{sortie}$ pour trois RSBs différents

Les figures 4.7 montrent les variations des coefficients estimés du noyau linéaire h_1 et des composantes de la SVD A et B , en fonction du nombre d'itérations pour un $RSB = 20dB$. Pour le même RSB , les figures 4.8 montrent les variations des coefficients estimés des composantes du modèle PARAFAC C , D et E , en fonction du nombre d'itérations.

Le tableau 4.3 contient les valeurs de $EQMN_{sortie}$ pour différentes valeurs de RSB et différentes séquences d'entrées : un Bruit Blanc Gaussien (B.B.G), une séquence 4-PAM et une séquence 6-PAM. On note après lecture du tableau que le choix de la séquence d'entrée n'influence pas beaucoup la performance de l'identification du modèle.

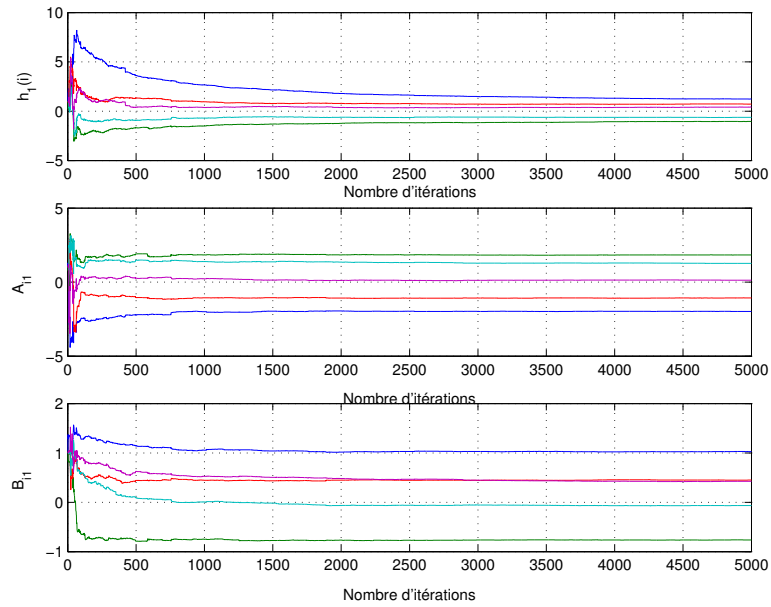


FIG. 4.7 – Les composantes estimées des noyaux linéaire et quadratique

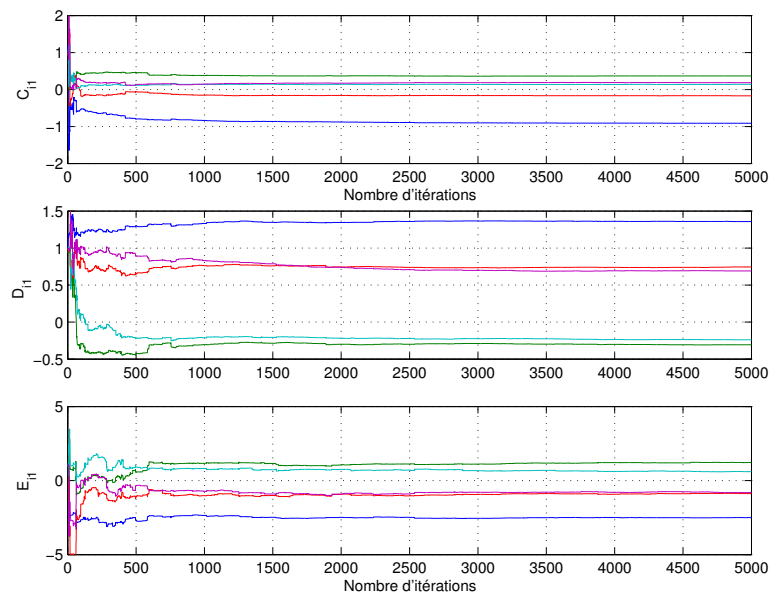


FIG. 4.8 – Les composantes estimées du modèle PARAFAC

4.4.2 Second système simulé

Le deuxième système simulé est un modèle simplifié d'un canal satellitaire non-linéaire (Lee et Mathews [40]) représenté dans la figure 4.9. Le canal filtre d'abord le signal d'entrée $u(k)$ avec un filtre linéaire passe-bas dont la fonction de transfert est $H_B(z)$. Ensuite, le signal est filtré par un dispositif non-linéaire sans mémoire défini par sa caractéristique entrée/sortie $A(r)$ définie dans l'équation (4.57) et représentée dans la figure 4.10. Enfin le signal passe dans un autre filtre passe-bas dont la fonction de transfert est $H_C(z)$. $H_B(z)$ et $H_C(z)$ sont deux filtres passe-bas d'ordre 4 définis dans les équations (4.58) et (4.59).

$$A(r) = 0.542r^3 - 2.13r^2 + 2.55r + 0.05 \quad (4.57)$$

$$H_B(z) = \frac{(0.078 + 0.1559z^{-1} + 0.078z^{-2})(0.0619 + 0.1238z^{-1} + 0.0619z^{-2})}{(1 - 1.3209z^{-1} + 0.6327z^{-2})(1 - 1.0486z^{-1} + 0.2961z^{-2})} \quad (4.58)$$

$$H_C(z) = \frac{(0.4638 - 0.4942z^{-1} + 0.4638z^{-2})(0.183 + 0.1024z^{-1} + 0.183z^{-2})}{(1 - 1.2556z^{-1} + 0.6891z^{-2})(1 - 0.7204z^{-1} + 0.1888z^{-2})} \quad (4.59)$$

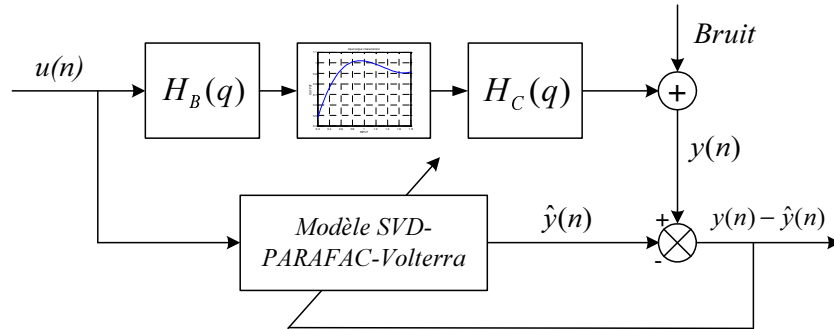


FIG. 4.9 – Application du modèle SVD-PARAFAC-Volterra d'ordre 3 pour l'identification d'un canal satellitaire non linéaire

La réponse impulsionnelle des filtres H_B et H_C devient inférieure à 10^{-2} après 20 points comme le montre la figure 4.11. Pour cela, on modélise le canal par un système de Wiener-Hammerstein cubique c'est à dire un filtre de Volterra d'ordre 3 symétrique et de mémoire $M = 39$. La complexité paramétrique de ce modèle est égale à $C_{Standard} = 11479$ paramètres. En utilisant un modèle SVD-PARAFAC-Volterra avec $M = 39$, $P = 1$, $R = 1$ et dont la complexité paramétrique est $C_{SVD-PARAFAC} = 234$,

on obtient une importante réduction de la complexité numérique :

$$TRC = \frac{C_{SVD-PARAFAC}}{C_{Standard}} = 2.04 \cdot 10^{-3}$$

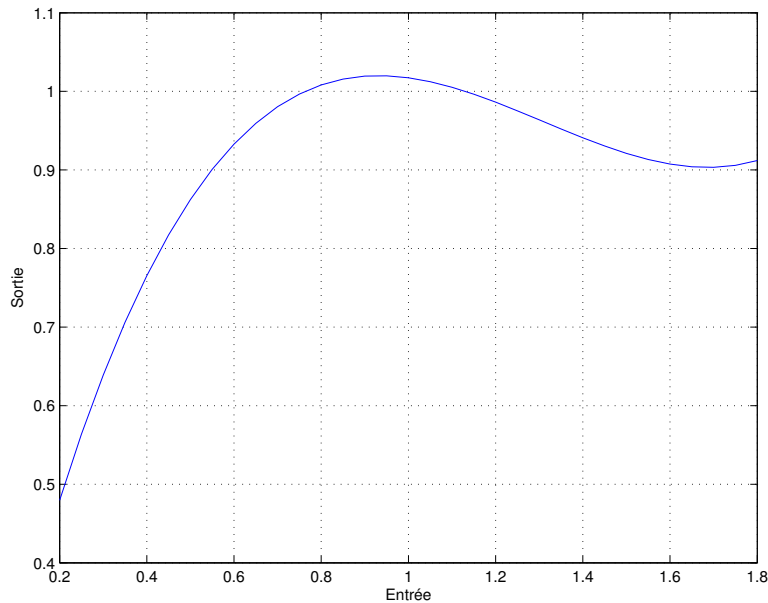
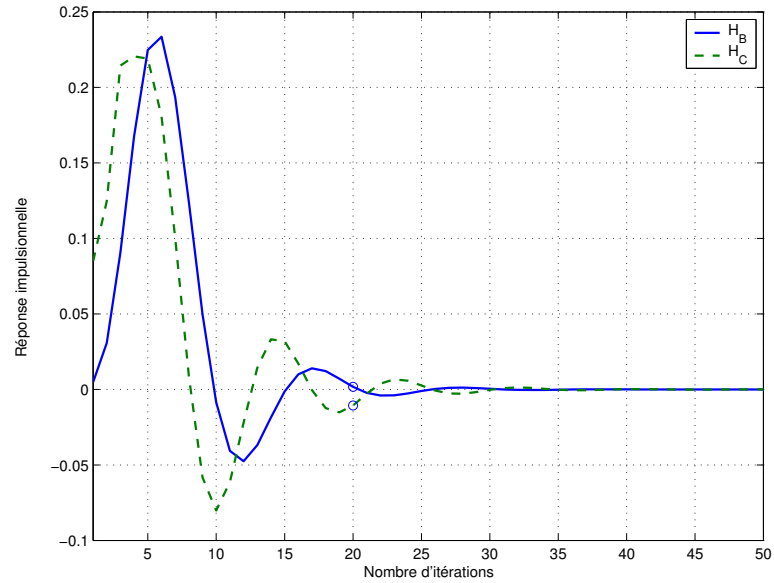


FIG. 4.10 – La caractéristique entrée-sortie du dispositif non -linéaire

La figure 4.12 montre les variations de $EQMN_{sortie}$ en fonction du nombre d'itérations pour trois RSBs différents avec $P = 1$, $R = 1$ et $M = 39$. L'entrée du système est une séquence de bruit blanc gaussien de longueur $N = 10000$, de moyenne nulle et de variance l'unité. Les résultats des simulations sont obtenus en utilisant la méthode de Monte Carlo avec 20 séquences différentes de bruits gaussiens additifs.

Le tableau 4.4 contient les valeurs de $EQMN_{sortie}$ en fonction du nombre de facteurs P de la décomposition PARAFAC et du nombre de facteurs R de la SVD pour un rapport signal sur bruit $RSB = 20 \text{ dB}$. Les valeurs de $EQMN_{sortie}$ décroient légèrement quand on augmente les valeurs de R et P , d'où l'intérêt de choisir des petites valeurs pour P et R afin de réduire la complexité numérique du modèle.

A partir de ces résultats de simulations, on peut conclure que l'approche de la décomposition SVD-PARAFAC permet de modéliser des systèmes de Volterra d'ordre 3 avec des Taux de Réduction de Complexité (TRC) intéressants. Cette approche nous a permis de représenter un canal satellitaire non linéaire avec des erreurs de modélisation relativement petites pour une réduction de complexité très importante. Le choix de la

FIG. 4.11 – Les réponses impulsionnelles des filtres H_B et H_C

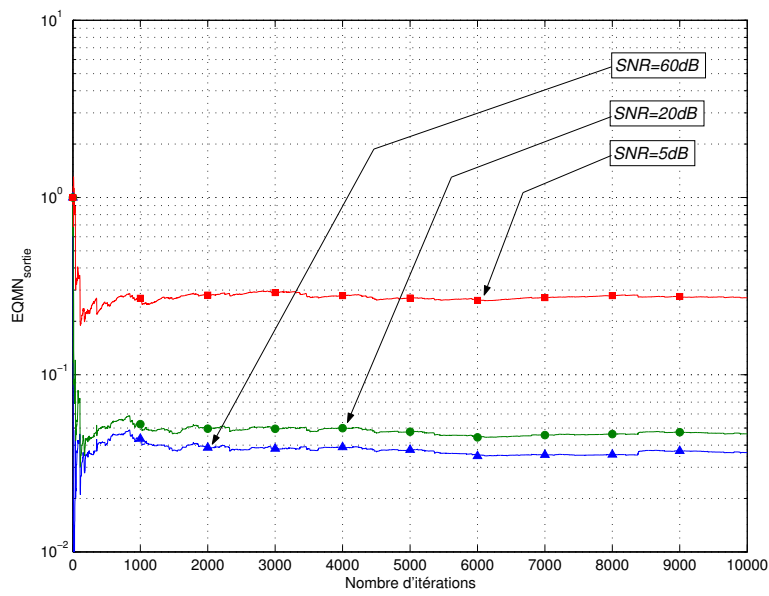
	R=1	R=2	R=3	R=4	R=5	R=6
P=1	$4.639 \cdot 10^{-2}$	$3.448 \cdot 10^{-2}$	$2.347 \cdot 10^{-2}$	$2.102 \cdot 10^{-2}$	$1.974 \cdot 10^{-2}$	$1.842 \cdot 10^{-2}$
P=2	$4.638 \cdot 10^{-2}$	$3.202 \cdot 10^{-2}$	$1.947 \cdot 10^{-2}$	$1.871 \cdot 10^{-2}$	$1.779 \cdot 10^{-2}$	$1.681 \cdot 10^{-2}$
P=3	$4.331 \cdot 10^{-2}$	$3.199 \cdot 10^{-2}$	$1.893 \cdot 10^{-2}$	$1.843 \cdot 10^{-2}$	$1.719 \cdot 10^{-2}$	$1.638 \cdot 10^{-2}$
P=4	$4.294 \cdot 10^{-2}$	$3.124 \cdot 10^{-2}$	$1.865 \cdot 10^{-2}$	$1.798 \cdot 10^{-2}$	$1.681 \cdot 10^{-2}$	$1.590 \cdot 10^{-2}$
P=5	$4.243 \cdot 10^{-2}$	$3.079 \cdot 10^{-2}$	$1.802 \cdot 10^{-2}$	$1.771 \cdot 10^{-2}$	$1.653 \cdot 10^{-2}$	$1.545 \cdot 10^{-2}$
P=6	$4.228 \cdot 10^{-2}$	$3.045 \cdot 10^{-2}$	$1.754 \cdot 10^{-2}$	$1.722 \cdot 10^{-2}$	$1.605 \cdot 10^{-2}$	$1.511 \cdot 10^{-2}$

TAB. 4.4 – $EQMN_{sortie}$ obtenue avec le modèle SVD-PARAFAC-Volterra d'ordre 3 pour différentes valeurs de R et P

séquence d'entrée n'influence pas beaucoup la performance de l'identification du modèle, de même que le choix des nombres P et R des facteurs des décompositions PARAFAC et SVD.

4.5 Conclusion

Dans ce chapitre, on a présenté une nouvelle approche pour représenter et identifier des modèles de Volterra d'ordre K en utilisant les décompositions SVD et PARAFAC. Ce qui nous a permis de réduire considérablement la complexité paramétrique des noyaux, spécialement quand ils sont séparables. Un nouvel algorithme ARLS a été proposé pour identifier de telles décompositions.

FIG. 4.12 – $EQMN_{sortie}$ pour trois RSBs différents

Chapitre 5

Identification robuste des systèmes

5.1 Introduction

Les approches d'identification à erreur bornée ne font pas d'hypothèses statistiques sur l'erreur à l'exception que celle-ci soit bornée et de bornes connues. L'identification consiste à déterminer une région dans l'espace des paramètres appelée domaine (ou ensemble) d'appartenance compatible avec les mesures, la structure du modèle et les bornes de l'erreur. L'ensemble exact est un polyèdre convexe (ou un polytope s'il est borné). La mise à jour de cet ensemble a été effectuée par (Walter et Lahanier [69]; Mo et Norton [52]). En raison de sa complexité, l'ensemble exact est approché par des régions de formes plus simples telles que l'ellipsoïde (Fogel et Huang [23]), l'orthotope (Messaoud et Favier [50]), le parallélotope (Vicino et Zappa [65]) ou des polytopes à complexité limitée (Broman et Shensa [7], Lahanier et Walter [39], Veres et al [64] et Maraoui et Messaoud [46]).

Dans ce chapitre on présente quelques unes des méthodes d'identifications à erreur bornée (approche polytopique, approche orthotopique et approche polytopique à complexité limitée). Ensuite, dans la partie simulations et pour faire le lien avec le deuxième chapitre, on modélise des systèmes linéaires à l'aide de modèles de Laguerre, de Kautz et des modèles issus de la BOG qu'on identifie par la suite en utilisant les approches à erreur bornée. On utilise enfin dans la dernière partie des simulations ces méthodes d'identification dans le cas non-linéaire afin d'estimer l'espace d'incertitude des paramètres de modèles de Volterra à complexité réduite.

5.2 Méthodes d'identification à erreur bornée

5.2.1 Approche UBBE (Unknown But Bounded Error)

On considère l'équation aux différences suivante :

$$y(k) = - \sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=0}^{n_b} b_i u(k-d-i) + e(k) \quad (5.1)$$

où $y(k)$ et $u(k)$ sont respectivement la sortie et l'entrée du système à l'instant k , $e(k)$ est l'erreur de modélisation supposée être bornée. a_i ; $i = 1, \dots, n_a$ et b_j ; $j = 0, \dots, n_b$ sont les paramètres du processus et d le retard.

L'équation (5.1) peut être réécrite sous la forme suivante :

$$y(k) = \phi_k^T \theta + e(k) \quad (5.2)$$

où ϕ_k est le vecteur de régression défini par :

$$\phi_k = \left[-y(k-1) \quad \dots \quad -y(k-n_a) \quad u(k-d) \quad \dots \quad u(k-d-n_b) \right]^T \quad (5.3)$$

et θ est le vecteur de paramètres suivant :

$$\theta = \left[a_1 \quad \dots \quad a_{n_a} \quad b_0 \quad \dots \quad b_{n_b} \right]^T \quad (5.4)$$

L'erreur $e(k)$ est bornée comme suit :

$$e_k^{min} \leq e(k) \leq e_k^{max} \quad (5.5)$$

où e_k^{min} et e_k^{max} sont respectivement les bornes inférieure et supérieure de l'erreur $e(k)$.

A partir des équations (5.2) et (5.5), on peut écrire :

$$y(k) - e_k^{max} \leq \phi_k^T \theta \leq y(k) - e_k^{min} \quad (5.6)$$

Ces bornes peuvent être rendues symétriques en effectuant le changement de variable suivant :

$$y(k) = y^{(s)}(k) + \frac{e_k^{max} + e_k^{min}}{2} \quad (5.7)$$

L'équation (5.6) devient :

$$y^{(s)}(k) - \delta_k \leq \phi_k^T \theta \leq y^{(s)}(k) + \delta_k \quad (5.8)$$

ou encore :

$$\left| y^{(s)}(k) - \phi_k^T \theta \right| \leq \delta_k \quad (5.9)$$

avec

$$\delta_k = \frac{e_k^{max} - e_k^{min}}{2} \quad (5.10)$$

Cette relation définit à chaque instant deux hyperplans H_{k_1} et H_{k_2} orthogonaux au vecteur ϕ_k limitant l'ensemble des paramètres. Ces deux hyperplans représentés dans la figure 5.1, sont définis par :

$$H_{k_1} = \{ \theta / \phi_k^T \theta = y(k) - e_k^{min} \} = \{ \theta / \phi_k^T \theta = y^{(s)}(k) + \delta_k \} \quad (5.11)$$

$$H_{k_2} = \{ \theta / \phi_k^T \theta = y(k) - e_k^{max} \} = \{ \theta / \phi_k^T \theta = y^{(s)}(k) - \delta_k \} \quad (5.12)$$

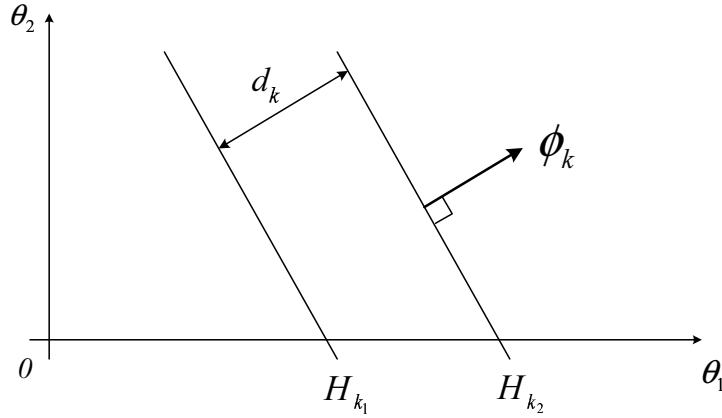


FIG. 5.1 – Les deux hyperplans H_{k_1} et H_{k_2}

La distance entre les deux hyperplans H_{k_1} et H_{k_2} est donnée par :

$$d_k = \frac{2\delta_k}{\|\phi_k\|} \quad (5.13)$$

où $\|\phi_k\|$ est la norme euclidienne du vecteur ϕ_k .

Chaque hyperplan H_{k_j} ($j = 1, 2$), engendre un demi-espace fermé négatif $H_{k_j}^-$; ($j = 1, 2$)

et un demi-espace fermé positif $H_{k_j}^+$; ($j = 1, 2$) tels que :

$$H_{k_1}^+ = \{\theta / \phi_k^T \theta \leq y(k) - e_k^{min}\} = \{\theta / \phi_k^T \theta \leq y^{(s)}(k) + \delta_k\} \quad (5.14)$$

$$H_{k_1}^- = \{\theta / \phi_k^T \theta \geq y(k) - e_k^{min}\} = \{\theta / \phi_k^T \theta \geq y^{(s)}(k) + \delta_k\} \quad (5.15)$$

$$H_{k_2}^+ = \{\theta / \phi_k^T \theta \geq y(k) - e_k^{max}\} = \{\theta / \phi_k^T \theta \geq y^{(s)}(k) - \delta_k\} \quad (5.16)$$

$$H_{k_2}^- = \{\theta / \phi_k^T \theta \leq y(k) - e_k^{max}\} = \{\theta / \phi_k^T \theta \leq y^{(s)}(k) - \delta_k\} \quad (5.17)$$

Le vecteur de paramètres θ satisfaisant les contraintes (5.6) appartient au domaine défini par l'intersection des demi-espaces fermés $H_{k_1}^+$ et $H_{k_2}^+$ engendrés respectivement par les hyperplans H_{k_1} et H_{k_2} . Soit :

$$H_k^+ = H_{k_1}^+ \cap H_{k_2}^+ \quad (5.18)$$

L'approche UBBE consiste donc à déterminer à chaque instant k , le plus petit ensemble de paramètres $S_k \in \mathbb{R}^n$ consistant avec les mesures, la connaissance a priori du modèle et les bornes de l'erreur. L'ensemble obtenu (appelé ensemble d'appartenance) doit satisfaire les contraintes associées à toutes les observations.

$$S_k = \bigcap_{t=1}^k H_{t_1}^+ \cap H_{t_2}^+ \quad (5.19)$$

où k est le nombre d'observations. On obtient alors :

$$S_k = \{\theta / y(t) - e_t^{max} \leq \phi_t^T \theta \leq y(t) - e_t^{min} ; t = 1, \dots, k\} \quad (5.20)$$

S_k peut être vide si les bornes de l'erreur sont mal choisies, la structure du modèle est mal définie ou le domaine initial est mal choisi. Dans le cas contraire S_k est un polytope s'il est borné. Dans la section suivante une méthode récursive de calcul du polytope est présentée.

5.2.2 L'approche polytopique

Définitions

- Soit X un espace de dimension n et P une partie de X . P est un polyèdre convexe de X s'il est l'intersection d'un nombre fini N de demi-espaces fermés. Le polyèdre est un polytope s'il est compact et d'intérieur non vide.

• On appelle face de dimension m d'un polytope P , l'intersection de $(n - m)$ hyperplans indépendants (n étant la dimension de P).

De cette définition on peut définir :

- Une arête comme étant une face de dimension 1.

- Un sommet comme étant une face de dimension 0. Et donc un sommet est l'intersection de n hyperplans.

D'après la relation (5.19) S_k est l'intersection de $2k$ demi-espaces fermés positifs qui forment donc un polyèdre convexe.

La détermination de l'ensemble d'appartenance des paramètres consistants avec les mesures consiste donc à définir le polytope obtenu par l'intersection de ces demi-espaces. Les contraintes (5.6) peuvent s'écrire sous la forme :

$$\phi_k^T \theta \geq y(k) - e_k^{max} \quad (5.21)$$

$$-\phi_k^T \theta \geq -y(k) + e_k^{min} \quad (5.22)$$

ou sous forme matricielle :

$$\begin{bmatrix} \phi_k^T \\ -\phi_k^T \end{bmatrix} \theta \geq \begin{bmatrix} y(k) - e_k^{max} \\ -y(k) + e_k^{min} \end{bmatrix} \quad (5.23)$$

Pour N mesures cette inégalité devient :

$$A\theta \geq c \quad (5.24)$$

où A est une matrice de dimension $(2N \times n)$ dont la $i^{\text{ème}}$ ligne est définie par :

$$A_i = (-1)^{i+1} \phi_{\mathbf{E}((i+1)/2)}^T \quad (5.25)$$

où l'opérateur $\mathbf{E}(\cdot)$ désigne la partie entière et $c \in \mathbb{R}^{2N}$ est un vecteur dont la $i^{\text{ème}}$ composante est donnée par :

$$c_i = \begin{cases} y((i+1)/2) - e_{(i+1)/2}^{max} & \text{si } i \text{ est impair} \\ -y(\mathbf{E}((i+1)/2)) + e_{\mathbf{E}((i+1)/2)}^{min} & \text{si } i \text{ est pair} \end{cases} \quad (5.26)$$

Supposons que pour chaque paramètre $\theta(i)$, il y ait une borne inférieure $\theta_{min}(i)$, telle que ;

$$\theta(i) \geq \theta_{min}(i) \quad \forall i = 1, \dots, n \quad (5.27)$$

On pose

$$x(i) = \theta(i) - \theta_{min}(i) \quad \forall i = 1, \dots, n \quad (5.28)$$

donc

$$x(i) \geq 0 \quad (5.29)$$

D'après (5.24) et (5.28), on a :

$$Ax + A\theta_{min} \geq c \quad (5.30)$$

où x et θ_{min} sont deux vecteurs de composantes respectives $x(i)$ et $\theta_{min}(i)$; $i = 1, \dots, n$.

Si on pose $b = c - A\theta_{min}$, on obtient :

$$Ax \geq b \quad \text{avec} \quad x(i) \geq 0 \quad (5.31)$$

Le problème revient donc à résoudre un système d'inéquations linéaires sous la contrainte que toutes les composantes du vecteur x soient positives ou nulles.

La résolution d'un tel système peut se faire selon deux approches différentes :

- Approche algébrique qui permet la résolution du système d'inéquations de manière analogue à celle d'un système d'équations. Cette approche nécessite le traitement de toutes les inéquations en même temps ce qui ne permet pas une résolution récursive et un traitement en temps réel.
- Approche géométrique qui consiste à étudier le système d'inéquations comme étant une forme géométrique caractérisée par sa frontière. Le polyèdre solution s'il existe est caractérisé par une représentation paramétrique déterminée à partir de ses sommets ou de ses arêtes. C'est cette approche qui va être considérée dans la suite.

On considère le système d'inéquations :

$$\sum_{j=1}^n a_{ij}x(j) \geq b(i) \quad i = 1, \dots, 2N \quad (5.32)$$

Si b est différent du vecteur nul ce système définit un polytope. Il existe deux méthodes pour calculer d'une manière récursive un polytope exact en déterminant ses sommets. La démarche est la même pour les deux méthodes : On part d'une forme initiale (orthotope). Une nouvelle contrainte génère un hyperplan qui divise l'espace en deux demi-espaces. L'un des deux demi-espaces est négatif et l'autre est positif. La nouvelle forme polytopique obtenue est égale à l'intersection de la forme initiale avec

le demi-espace fermé positif engendré par la nouvelle contrainte. Les sommets situés dans le demi-espace ouvert négatif sont rejetés, ceux situés dans le demi-espace fermé positif sont intégralement maintenus et d'autres nouveaux sommets sont formés à partir de l'intersection de l'hyperplan associé à la nouvelle contrainte avec le polytope précédent. Ces nouveaux sommets sont obtenus comme combinaisons convexes de sommets adjacents de ce polytope situés de part et d'autre de l'hyperplan.

Détermination du polytope exact

Cette approche permet de déterminer tous les sommets d'un polytope. Soient $\{S_{k-1}^i\}_{i=1,\dots,n_{k-1}}$ les sommets du polytope P_{k-1} , obtenu après $k-1$ mesures.

La nouvelle contrainte introduite par la nouvelle mesure $y(k)$ génère deux hyperplans H_{k_1} et H_{k_2} donnés par les relations (5.11) et (5.12), divisant l'espace en deux demi-espaces $H_{k_j}^+$ et $H_{k_j}^-$ ($j = 1, 2$) définis par les équations (5.14)-(5.17).

On classe les sommets du polytope P_{k-1} en trois classes selon leurs positions par :

$$S_+ = \left\{ S_{k-1}^i \in P_{k-1} / S_{k-1}^i \in H_{k_j}^+ \setminus H_{k_j}, i = 1, \dots, n_{k-1} \right\} \quad (5.33)$$

$$S_- = \left\{ S_{k-1}^i \in P_{k-1} / S_{k-1}^i \in H_{k_j}^- \setminus H_{k_j}, i = 1, \dots, n_{k-1} \right\} \quad (5.34)$$

$$S_0 = \left\{ S_{k-1}^i \in P_{k-1} / S_{k-1}^i \in H_{k_j}, i = 1, \dots, n_{k-1} \right\} \quad (5.35)$$

où $(H_{k_j}^+ \setminus H_{k_j})$ et $(H_{k_j}^- \setminus H_{k_j})$ désignent respectivement les demi-espaces ouverts positif et négatif engendrés par l'hyperplan H_{k_j} . De ce fait, S_+ est l'ensemble des sommets situés dans le demi-espace ouvert positif, S_- est l'ensemble des sommets situés dans le demi-espace ouvert négatif et S_0 est l'ensemble des sommets qui appartiennent à l'hyperplan.

Le nouveau polytope P_k est donné par l'intersection du polytope P_{k-1} et des demi-espaces fermés positifs $H_{k_j}^+$ ($j = 1, 2$).

$$P_k = P_{k-1} \bigcap_{j=1}^2 H_{k_j}^+ \quad (5.36)$$

et les sommets sont donnés par :

$$S = S_+ \cup S_0 \cup X \quad (5.37)$$

où X désigne l'ensemble des nouveaux sommets formés. Ces sommets sont obtenus par combinaison convexe de sommets du polytope P_{k-1} appartenant aux demi-espaces

positif et négatif. L'ensemble X peut s'écrire comme :

$$X = \left\{ S_k^{l,i} / S_k^{l,i} = \lambda S_k^l + (1 - \lambda) S_k^i ; \left\{ \begin{array}{l} S_k^l \in H_{k_j}^+ ; \\ S_k^i \in H_{k_j}^- \text{ et} \\ 0 \leq \lambda \leq 1 \end{array} \right\} \right\} \quad (5.38)$$

Tous les sommets du demi-espace positif sont combinés avec tous les sommets du demi-espace négatif. Ceci implique nécessairement la formation de sommets parasites qui ne contribuent pas à la formation du nouveau polytope.

Un sommet constitue effectivement un sommet du polytope P_k s'il est la combinaison convexe de deux sommets S_k^l et S_k^i adjacents situés respectivement dans $H_{k_j}^+$ et $H_{k_j}^-$.

Pour éliminer ces sommets parasites et afin d'éviter d'alourdir les calculs, il serait plus intéressant d'éviter de les générer. Pour cela il est nécessaire de vérifier si deux sommets S_k^l et S_k^i du polytope sont adjacents c'est à dire sont reliés par une arête avant de les combiner.

Algorithme récursif

Cet algorithme a été énoncé par Mo et Norton [52].

- *Initialisation :*

Si aucune connaissance a priori sur le domaine d'appartenance des paramètres est disponible, on choisit comme polytope initial un orthotope centré à l'origine ayant une dimension assez importante et 2^n sommets ayant chacun n composantes à ± 100 ou ± 1000 .

- *Acquisition d'une nouvelle mesure :*

Supposant qu'on dispose du polytope P_{k-1} défini par les $k - 1$ premières mesures. P_{k-1} est caractérisé par ses sommets et les hyperplans d'appui de chaque sommet.

Soient n_{k-1} le nombre de sommets et V_{k-1} la matrice dont chaque colonne contient les coordonnées d'un sommet de P_{k-1} .

$$V_{k-1} = [S_{k-1}^1, \dots, S_{k-1}^{n_{k-1}}] \quad (5.39)$$

L'acquisition d'une $k^{\text{ème}}$ mesure engendre deux hyperplans H_{k_1} et H_{k_2} . Chaque hyperplan H_{k_j} ($j = 1, 2$) divise l'espace en deux demi-espaces $H_{k_j}^+$ et $H_{k_j}^-$; ($j = 1, 2$). Le nouveau polytope P_k est obtenu en faisant l'intersection du polytope P_{k-1} avec les demi-espaces fermés positifs $H_{k_j}^+$ ($j = 1, 2$).

Il est donc indispensable de déterminer la position des différents sommets du polytope P_{k-1} par rapport à l'hyperplan H_{k_j} ($j = 1, 2$) associés à la nouvelle mesure et l'algorithme est répété deux fois, une pour chaque hyperplan engendré par la mesure.

Pour déterminer la position de chaque sommet de P_{k-1} par rapport aux hyperplans H_{k_1} et H_{k_2} , on définit respectivement les vecteurs β_1 et β_2 tels que :

$$\beta_1(i) = y(k) - e_k^{min} - \phi_k^T S_{k-1}^i \quad (5.40)$$

$$\beta_2(i) = -y(k) + e_k^{max} + \phi_k^T S_{k-1}^i \quad (5.41)$$

S_{k-1}^i étant le $i^{ème}$ sommet de P_{k-1} . Le signe de $\beta_j(i)$ détermine la position du sommet S_{k-1}^i ; ($i = 1, \dots, n_{k-1}$) par rapport à l'hyperplan H_{k_j} ($j = 1, 2$). Trois cas de figures peuvent se présenter :

- Si $\beta_j(i)$ est négatif alors le sommet S_{k-1}^i appartient au demi-espace ouvert négatif $H_{k_j}^- \setminus H_{k_j}$.
- Si $\beta_j(i)$ est positif alors le sommet S_{k-1}^i appartient au demi-espace ouvert positif $H_{k_j}^+ \setminus H_{k_j}$.
- Si $\beta_j(i)$ est nul alors le sommet S_{k-1}^i appartient à l'hyperplan H_{k_j} .

a/ Si $\beta_j(\mathbf{i}) < 0$; $\mathbf{i} = 1, \dots, \mathbf{n}_{k-1}$

Tous les sommets du polytope P_{k-1} appartiennent au demi-espace ouvert négatif $H_{k_j}^- \setminus H_{k_j}$ et le polytope résultant est nul comme le montre la figure (5.2). En effet l'intersection du polytope avec le demi-espace fermé positif $H_{k_j}^+$ est vide. Ce cas peut avoir lieu lorsque le polytope initial P_0 ou les bornes de l'erreur sont mal choisies.

b/ Si $\beta_j(\mathbf{i}) > 0$; $\mathbf{i} = 1, \dots, \mathbf{n}_{k-1}$

Tous les sommets du polytope P_{k-1} appartiennent au demi-espace ouvert positif $H_{k_j}^+ \setminus H_{k_j}$ comme le montre la figure (5.3) et la nouvelle mesure ne modifie pas le polytope, elle est donc redondante d'où $P_k = P_{k-1}$.

c/ Si les $\beta_j(\mathbf{i})$ sont de signes différents ; $\mathbf{i} = 1, \dots, \mathbf{n}_{k-1}$

C'est le cas le plus général. Il est schématisé dans la figure (5.4). Les sommets de P_{k-1} sont répartis de part et d'autre de l'hyperplan H_{k_j} et le nouveau polytope P_k sera défini par :

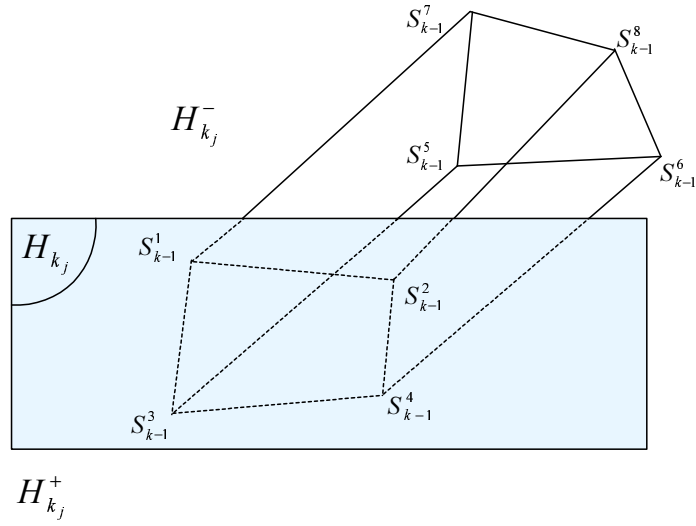


FIG. 5.2 – Acquisition d’une nouvelle mesure : Cas où $\beta_j(i) < 0 ; i = 1, \dots, n_{k-1}$

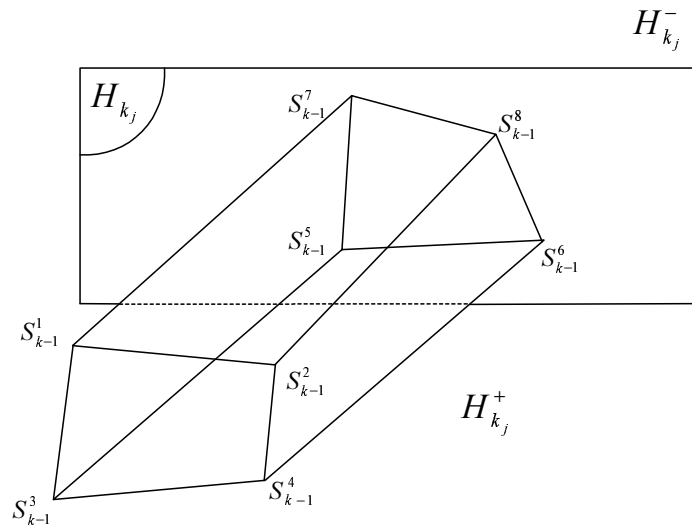


FIG. 5.3 – Acquisition d’une nouvelle mesure : Cas où $\beta_j(i) > 0 ; i = 1, \dots, n_{k-1}$

- Les sommets de P_{k-1} qui appartiennent à $H_{k_j}^+$
- Les nouveaux sommets formés obtenus par combinaisons convexes des sommets de P_{k-1} appartenant à $H_{k_j}^+$ et ceux appartenant à $H_{k_j}^-$.

Les sommets appartenant au demi- espace ouvert négatif sont éliminés.

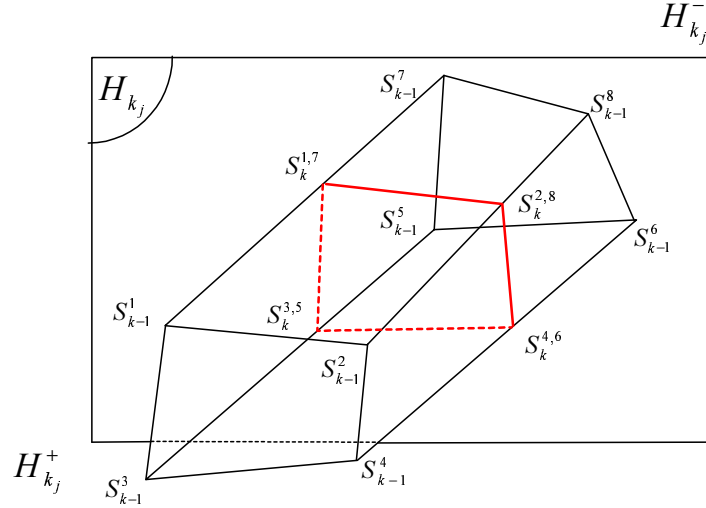


FIG. 5.4 – Acquisition d'une nouvelle mesure : Cas où les $\beta_j(i)$ sont de signes différents ; $i = 1, \dots, n_{k-1}$

- Calcul des nouveaux sommets formés :

Soient S_{k-1}^l et S_{k-1}^i deux sommets appartenant respectivement à $H_{k_j}^+$ et à $H_{k_j}^-$ et soit $S_k^{l,i}$ le nouveau sommet formé. $S_k^{l,i}$ s'écrit alors :

$$S_k^{l,i} = \lambda S_{k-1}^l + (1 - \lambda) S_{k-1}^i ; 0 \leq \lambda \leq 1 \quad (5.42)$$

Puisque $S_k^{l,i}$ appartient à l'hyperplan H_{k_j} et d'après l'équation (5.23), on a :

$$\phi_k^T S_k^{l,i} = d_{k_j} \quad (5.43)$$

avec

$$d_{k_j} = \begin{cases} y(k) - e_k^{\min} & \text{si } j = 1 \\ y(k) - e_k^{\max} & \text{si } j = 2 \end{cases} \quad (5.44)$$

A partir de (5.42) et (5.43), on obtient :

$$\lambda = \frac{d_{k_j} - \phi_k^T S_{k-1}^i}{\phi_k^T [S_{k-1}^l - S_{k-1}^i]} \quad \text{avec } 0 \leq \lambda \leq 1 \quad (5.45)$$

Avec les relations (5.42), (5.44) et (5.45), on détermine tous les nouveaux sommets formés qui, avec les sommets de P_{k-1} situés dans le demi-espace positif $H_{k_j}^+$; ($j = 1, 2$), forment l'ensemble des sommets du polytope P_k représenté dans la figure (5.5).

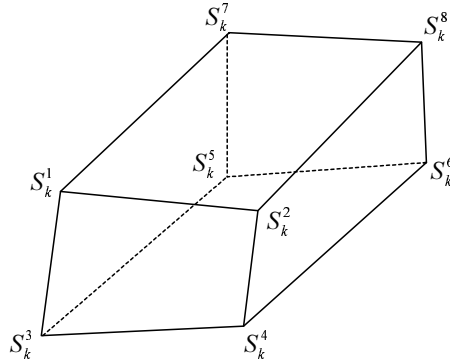


FIG. 5.5 – Le polytope P_k

- Test d'adjacence :

Définition : Deux sommets S^l et S^i sont adjacents si et seulement s'ils forment une arête, de plus aucun autre sommet S^r ; ($r \neq l$ et $r \neq i$) ne doit appartenir à cette arête.

Il faut noter que tout sommet est l'intersection de n hyperplans et toute arête est l'intersection de $(n - 1)$ hyperplans donc tout sommet est l'intersection de n arêtes. Soient S_{k-1}^l et S_{k-1}^i deux sommets appartenant respectivement aux demi-espaces $H_{k_j}^+$ et $H_{k_j}^-$. On définit $a_{k-1}^{l,i}$ comme étant l'arête qui joint ces deux sommets. On peut alors écrire :

$$a_{k-1}^{l,i} \cap H_{k_j} = S_k^{l,i} \quad (5.46)$$

Si les sommets S_{k-1}^l et S_{k-1}^i sont adjacents alors les sommets $S_k^{l,i}$ et S_{k-1}^l le sont aussi. Comme $S_k^{l,i}$ est l'intersection de n arêtes et comme chaque arête joint deux sommets, alors le sommet $S_k^{l,i}$ admet n sommets adjacents.

Le seul sommet appartenant au demi espace ouvert $H_{k_j}^+ \setminus H_{k_j}$ et adjacent au sommet $S_k^{l,i}$ est le sommet S_{k-1}^l . Par conséquent $n - 1$ sommets parmi les nouveaux sommets formés sont adjacents au sommet $S_k^{l,i}$. Soit S_k^h un tel sommet, alors S_k^h et $S_k^{l,i}$ ont $(n - 1)$ hyperplans en commun puisqu'ils forment une arête.

Ainsi pour chaque sommet, il faut chercher la liste des hyperplans d'appui. Si on note \mathcal{H}_l l'ensemble des indices des hyperplans d'appui du sommet S^l et \mathcal{H}_i l'ensemble d'indices des hyperplans d'appui du sommet S^i alors les sommets S^l et S^i sont adjacents si :

$$\text{Card}(\mathcal{H}_l \cap \mathcal{H}_i) = n - 1 \quad (5.47)$$

Il reste maintenant à calculer l'ensemble des hyperplans adjacents à chaque sommet. Pour les sommets $\in S^+$, la liste des hyperplans d'appui est maintenue. Cependant pour les nouveaux sommets formés $S_k^{l,i}$, la liste des hyperplans d'appui de chaque sommet est donnée par les $(n - 1)$ hyperplans communs aux sommets S^l et S^i à laquelle on ajoute l'hyperplan H_{k_j} .

Inconvénients de l'approche polytopique et solutions proposées

La méthode polytopique présentée dans la section précédente permet de caractériser le polytope exact par ses sommets. L'algorithme récursif permet de calculer le polytope à l'instant k à partir du polytope obtenu à l'instant $k - 1$ et de la mesure obtenue à l'instant k .

La taille du polytope est décroissante, mais la complexité des calculs augmente avec la dimension du vecteur de paramètres et le nombre de mesures. Ainsi l'utilisation de cet algorithme est limitée à des applications où l'ordre du système n'est pas grand et le nombre de mesures est réduit.

Une solution pour remédier à ce problème consiste à approcher le polytope exact par une forme géométrique plus simple et définie par un nombre de paramètres constant.

Plusieurs méthodes ont été proposées (approximation du polytope par un orthotope, un paralléloèdre ou un ellipsoïde). Ces formes simples peuvent contenir la forme exacte (bornitude extérieure ou Outer Bounding) et dans ce cas on cherche le plus petit domaine contenant la forme exacte pour réduire la marge d'erreur. De même ces formes simples peuvent être contenues dans le polytope exact et on procède à une bornitude intérieure (Inner Bounding) et dans ce cas on cherche le domaine maximal contenu dans la forme exacte, mais cette méthode comporte le risque que les paramètres réels soient situés en dehors de l'espace paramétrique ainsi déterminé.

5.2.3 L'approche orthotopique

Cette approche a été proposée par H. Messaoud et Favier [50]. Elle consiste à approcher le polytope exact par un orthotope dont les faces de dimension $(n-1)$ sont alignées avec les axes de coordonnées.

Définitions

- Un orthotope est un polytope dont toutes les arêtes sont parallèles aux axes de coordonnées.
- Soit un orthotope ϑ de dimension n centré autour d'un point C de l'espace paramétrique. Soient c et l les vecteurs qui contiennent respectivement les coordonnées du centre C de l'orthotope et les demi-longueurs des arêtes parallèles aux axes de coordonnées.
- Pour tout point x de l'orthotope, la $j^{\text{ème}}$ composante de x satisfait la double inégalité suivante :

$$\forall x \in \vartheta ; c(j) - l(j) \leq x(j) \leq c(j) + l(j) \quad (5.48)$$

où $c(j)$ désigne la $j^{\text{ème}}$ coordonnée du centre de l'orthotope et $l(j)$ désigne la demi-longueur de l'arête parallèle au $j^{\text{ème}}$ axe de coordonnée.

Soit une face f_i de dimension $p \leq n$, cette face est l'intersection de $n - p$ hyperplans linéairement indépendants. Comme tous les points d'un même hyperplan ont une coordonnée fixée à $c(j) - l(j)$ ou à $c(j) + l(j)$, les p autres coordonnées prennent leur valeur dans l'intervalle $]c(j) - l(j); c(j) + l(j)[$.

Algorithme récursif

- *Initialisation* :

Pour s'assurer que le vecteur de paramètres réels soit inclus dans l'orthotope initial ϑ_0 , on choisit ce dernier assez large si aucune information a priori est disponible sur le système. ϑ_0 sera un orthotope centré autour de l'origine et de demi-longueur d'axe élevée Δ . Les 2^n sommets de ϑ_0 numérotés de 0 à $2^n - 1$ et ayant pour coordonnées $\pm\Delta$ sont regroupés dans la matrice V_0 donnée par :

$$V_0 = \left[S_0^0 \quad \dots \quad S_0^{2^n-1} \right] \quad (5.49)$$

où S_k^i désigne le vecteur contenant les coordonnées du $i^{\text{ème}}$ sommet à l'instant k .

• Introduction d'une nouvelle mesure :

Supposant qu'à l'instant $(k - 1)$, on dispose de l'orthotope ϑ_{k-1} défini par ses sommets S_{k-1}^i , ($i = 0, \dots, 2^n - 1$). La matrice V_{k-1} des sommets est définie par :

$$V_{k-1} = [S_{k-1}^0 \dots S_{k-1}^{2^n-1}] \quad \text{avec} \quad \mathbf{dim}(V_{k-1}) = (n, 2^n) \quad (5.50)$$

Chaque nouvelle mesure engendre deux nouvelles contraintes comme le montre la relation (5.19), chacune étant associée à un hyperplan H_{k_j} , ($j = 1, 2$) et aux demi-espaces $H_{k_j}^+$ et $H_{k_j}^-$, ($j = 1, 2$). Les deux hyperplans sont donnés par :

$$\phi_k^T \theta = d_{k_j} \quad (5.51)$$

où ϕ_k , θ et d_{k_j} sont donnés respectivement par (5.3), (5.4) et (5.44).

L'algorithme d'actualisation contient deux étapes :

- Calcul du polytope intersection de l'orthotope ϑ_{k-1} avec les demi-espaces positifs $H_{k_j}^+$.
- Calcul de l'orthotope ϑ_k à partir de ce polytope.

Chaque sommet de l'orthotope est défini à partir des sommets extrêmes du polytope, c'est à dire les sommets dont l'une au moins des coordonnées prend sa valeur maximale ou minimale par rapport à l'ensemble des sommets du polytope.

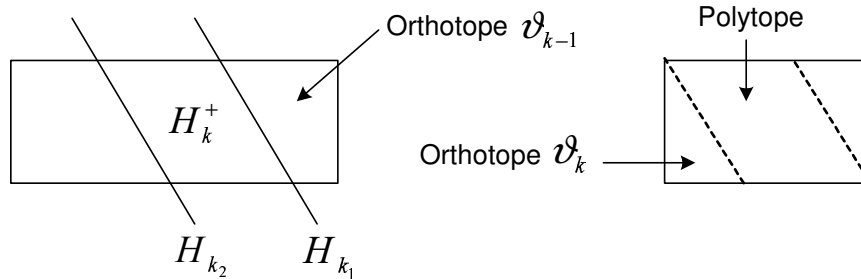


FIG. 5.6 – Mise à jour de l'orthotope ϑ_{k-1}

- Détermination du polytope intersection :

Pour trouver l'intersection de l'orthotope ϑ_{k-1} , avec les demi-espaces positifs $H_{k_j}^+$, il faut étudier la position de chaque sommet S_{k-1}^i par rapport à chaque hyperplan H_{k_j} . Cette position est indiquée par les composantes $\gamma_k(i)$ du vecteur γ_k défini comme suit :

$$\gamma_k = \phi_k^T V_{k-1} ; k = 0, \dots, 2^n - 1 \quad (5.52)$$

En tenant compte de la définition (5.44), on étudie la position des 2^n sommets de l'orthotope ϑ_{k-1} par rapports aux hyperplans H_{k_1} et H_{k_2} ,

- Pour l'hyperplan H_{k_1}
 - Si $\gamma_k(i) < d_{k_1} \Rightarrow S_{k-1}^i \in H_{k_1}^+ \setminus H_{k_1}$
 - Si $\gamma_k(i) > d_{k_1} \Rightarrow S_{k-1}^j \in H_{k_1}^- \setminus H_{k_1}$
 - Si $\gamma_k(i) = d_{k_1} \Rightarrow S_{k-1}^j \in H_{k_1}$
- Pour l'hyperplan H_{k_2}
 - Si $\gamma_k(i) < d_{k_2} \Rightarrow S_{k-1}^i \in H_{k_2}^- \setminus H_{k_2}$
 - Si $\gamma_k(i) > d_{k_2} \Rightarrow S_{k-1}^i \in H_{k_2}^+ \setminus H_{k_2}$
 - Si $\gamma_k(i) = d_{k_2} \Rightarrow S_{k-1}^i \in H_{k_2}$

\Rightarrow Il faut donc tester la position des 2^n sommets par rapport à chaque hyperplan H_{k_j} ; $j = 1, 2$. Trois cas sont possibles :

Premier cas : Tous les sommets de l'orthotope ϑ_{k-1} appartiennent au demi-espace négatif ouvert $H_{k_j}^- \setminus H_{k_j}$ ($j = 1, 2$) comme le montre la figure 5.7. Le polytope résultant est vide. Il faut donc arrêter l'algorithme. Ce cas peut arriver suite à un mauvais choix de l'orthotope initial ou des bornes de l'erreur.

Deuxième cas : Tous les sommets de l'orthotope appartiennent au demi-espace positif $H_{k_j}^+$ ($j = 1, 2$) comme le montre la figure 5.8. L'intersection entre ϑ_{k-1} et le demi-espace positif $H_{k_j}^+$ est l'orthotope ϑ_{k-1} lui même. La nouvelle mesure ne modifie pas l'orthotope de départ, elle est donc redondante.

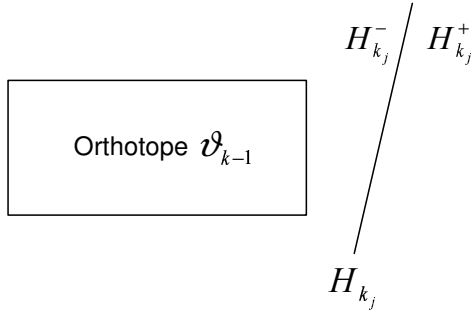


FIG. 5.7 – Détermination du polytope intersection : $\vartheta_{k-1} \cap H_{k_j}^+ = \emptyset$

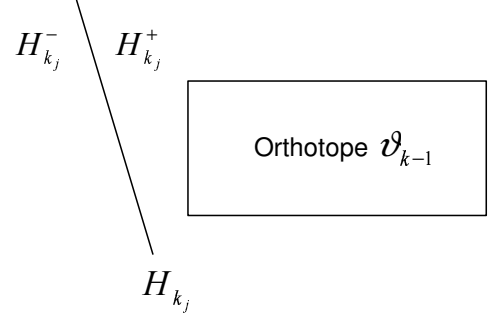


FIG. 5.8 – Détermination du polytope intersection : $\vartheta_{k-1} \cap H_{k_j}^+ = \vartheta_{k-1}$

Troisième cas : Certains sommets de l'orthotope appartiennent au demi-espace positif $H_{k_j}^+$ et d'autres appartiennent au demi-espace négatif $H_{k_j}^-$ comme le montre la figure 5.9.

On définit :

- V_{sp} : vecteur de longueur N_{k-1}^{sp} qui contient les numéros des sommets de ϑ_{k-1} situés dans le demi-espace ouvert positif $H_{k_j}^+$.
- V_{sn} : vecteur de longueur N_{k-1}^{sn} qui contient les numéros des sommets de ϑ_{k-1} situés dans le demi-espace ouvert négatif $H_{k_j}^-$.

Donc on a :

$$N_{k-1}^{sp} + N_{k-1}^{sn} = 2^n \quad (5.53)$$

Soit le polytope intersection entre ϑ_{k-1} et $H_{k_j}^+$ défini par ses sommets qui sont :

- Les sommets de ϑ_{k-1} appartenant à $H_{k_j}^+$.
- Les nouveaux sommets obtenus par l'intersection de l'hyperplan H_{k_j} ; ($j = 1, 2$) avec n'importe quelle arête de l'orthotope ϑ_{k-1} qui joint deux sommets adjacents S_{k-1}^l et S_{k-1}^i situés respectivement dans $H_{k_j}^+$ et $H_{k_j}^-$.

Soient $S_k^{l,i}$ les nouveaux sommets obtenus. Ils sont calculés comme une combinaison convexe des deux sommets adjacents S_{k-1}^l et S_{k-1}^i , soit :

$$S_k^{l,i} = \lambda S_{k-1}^l + (1 - \lambda) S_{k-1}^i ; 0 \leq \lambda \leq 1 \quad (5.54)$$

λ est donnée par l'équation (5.45).

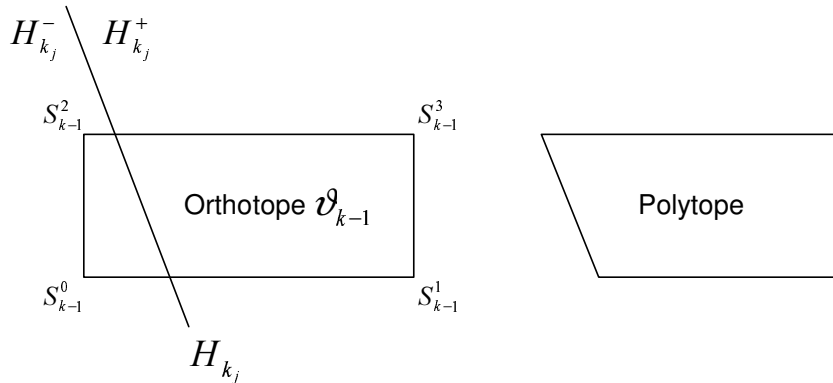


FIG. 5.9 – Détermination du polytope intersection : $\vartheta_{k-1} \cap H_{k_j}^+ = P_k \neq \vartheta_{k-1}$

Avant d'appliquer la relation (5.54), il faut détecter toutes les arêtes de ϑ_{k-1} coupées par l'hyperplan H_{k_j} ; ($j = 1, 2$), puis déduire toutes les paires de sommets adjacents (S_{k-1}^l, S_{k-1}^i) situés de part et d'autre de l'hyperplan H_{k_j} ; $j = 1, 2$ pour éviter le calcul de sommets parasites. Une fois que l'adjacence de deux sommets a été détectée, on calcule le nouveau sommet formé.

- Détermination de l'orthotope :

On effectue un test d'adjacence pour tous les sommets du polytope P_{k-1} afin de déterminer tous les sommets du polytope P_k . Une fois que tous ces derniers ont été déterminés, les 2^n sommets de l'orthotope sont déduits en prenant les valeurs maximales et minimales pour chaque coordonnée relativement à l'ensemble des sommets du polytope, soit :

$$\vartheta_k = \{ \theta / \theta_k^{min}(i) \leq \theta_k(i) \leq \theta_k^{max}(i) ; i = 1, \dots, n \} \quad (5.55)$$

où

- $\theta_k^{min}(i)$ est la valeur minimale de la $i^{\text{ème}}$ composante du vecteur θ , c'est à dire :

$$\theta_k^{min}(i) = \min_{j=1, \dots, n_k} S_k^j(i) ; i = 1, \dots, n \quad (5.56)$$

- $\theta_k^{max}(i)$ est la valeur maximale de la $i^{\text{ème}}$ composante du vecteur θ , c'est à dire :

$$\theta_k^{max}(i) = \max_{j=1, \dots, n_k} S_k^j(i) ; i = 1, \dots, n \quad (5.57)$$

- $S_k^j(i)$ représente la $i^{\text{ème}}$ coordonnée du sommet S_k^j et n_k , le nombre de sommets du polytope à l'instant k .

Il est à noter que les valeurs $\theta_k^{min}(i)$ et $\theta_k^{max}(i)$ obtenues de manière récursive fournissent les bornes des intervalles d'incertitude des paramètres estimés.

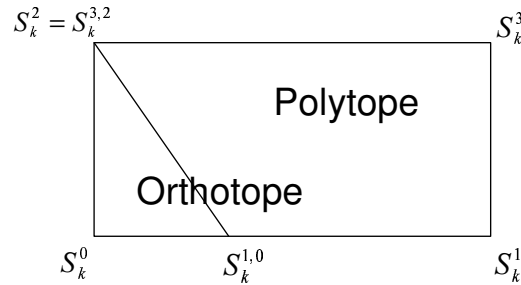


FIG. 5.10 – L'orthotope final ϑ_k

Le centre C_k de l'orthotope ϑ_k qui peut être utilisé comme estimateur du vecteur de paramètres réels θ à l'instant k et la demi-longueur L_k relative au $k^{\text{ème}}$ axe de coordonnées qui exprime la taille géométrique de l'orthotope à l'itération k sont déterminés par les relations suivantes :

$$C_k(i) = \frac{\theta_k^{max}(i) + \theta_k^{min}(i)}{2} ; i = 1, \dots, n \quad (5.58)$$

$$L_k(i) = \frac{\theta_k^{max}(i) - \theta_k^{min}(i)}{2} ; i = 1, \dots, n \quad (5.59)$$

L'orthotope final est donné par la figure 5.10.

5.2.4 L'approche polytopique à complexité limitée

Introduction

Dans les sections précédentes on a présenté les algorithmes récursifs qui mettent à jour le polytope exact ou bien l'une de ses approximations de forme orthotopique. Ces approximations qui sont conçues pour contourner les problèmes posés par la complexité

numérique de la mise à jour du polytope exact peuvent mettre à jour plus rapidement des régions de formes simples mais plus larges. L'augmentation du volume de la région résulte de l'accumulation de l'erreur d'approximation tout au long de l'opération de mise à jour de la région. Cette erreur provient de la bornitude extérieure du polytope exact.

Dans ce qui suit, on procède à une approximation du polytope exact par un polytope dont la complexité est limitée. Cette limitation consiste à fixer le nombre de faces de l'approximation du polytope exact sans contraindre la forme et l'orientation de cette approximation.

Le nombre de faces étant fixé, la mise à jour du polytope exact s'effectue normalement jusqu'à ce que le nombre de faces dépasse le seuil fixé. Dans ce cas on adopte une procédure qui permet de calculer un polytope modifié de complexité limitée. Plusieurs approches ont été mises au point afin de mettre à jour le polytope exact jusqu'à ce qu'un seuil de limitation de la complexité du polytope soit atteint.

La première approche introduite par Broman et Shensa [7] consiste à approcher le polytope P_{k-1}^{cl} par un orthotope ϑ_k puis on considère les hyperplans H_{k_1} et H_{k_2} ainsi que ceux qui constituent le polytope P_{k-1}^{cl} . Ces hyperplans sont classés par ordre croissant en fonction de la distance qui les sépare des différents sommets de ϑ_k . On coupe ensuite l'orthotope ϑ_k par chacun de ces hyperplans jusqu'à atteindre le seuil fixé pour le nombre d'arêtes du nouveau polytope à Complexité Limitée (C.L.). Cette approche est illustrée dans la figure 5.11.

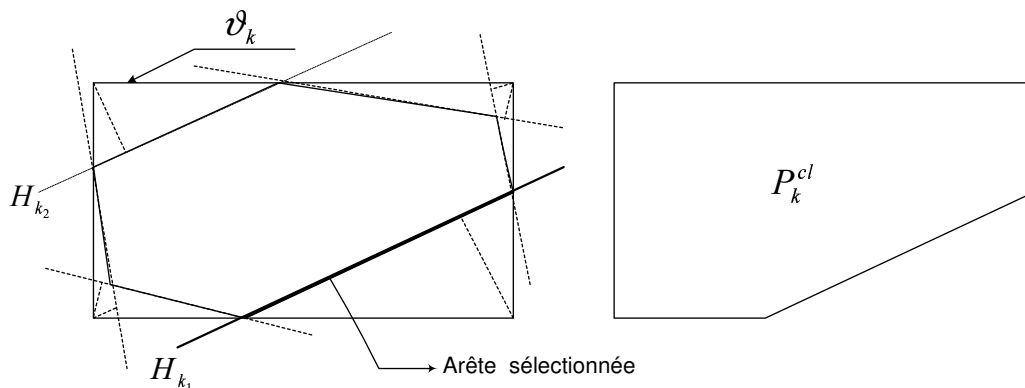


FIG. 5.11 – Mise à jour du polytope à complexité limitée selon Broman et Shensa

La deuxième approche est proposée par Lahanier et Walter [39]. Elle consiste à définir tous les hyperplans qui caractérisent le polytope exact P_k , les classer suivant leurs distances par rapport au centre c_k du polytope exact puis éliminer ceux d'entre eux qui sont les plus éloignés de c_k jusqu'à atteindre le seuil fixé. Cette approche est illustrée dans la figure 5.12.

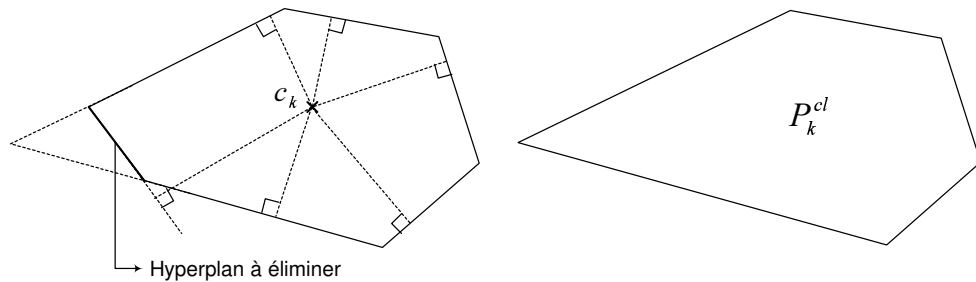


FIG. 5.12 – Mise à jour du polytope à complexité limitée selon Lahanier et Walter

La troisième approche est celle proposée par Veres et al [64]. Dans ce cas lorsque le nombre d'hyperplans caractérisant le polytope P dépasse le seuil fixé, les directions des hyperplans de la nouvelle mesure sont modifiées de façon à ce qu'elles appartiennent à un ensemble de directions fixé et que l'approximation contienne le polytope exact.

Dans ce qui suit, on présente une approche proposée par Maraoui et Messaoud [46]. C'est une combinaison des deux dernières approches.

Approche de Maraoui et Messaoud [46]

Les trois approches présentées précédemment pour le calcul du polytope à complexité limitée permettent de limiter le nombre de faces ou le nombre d'hyperplans de l'approximation. Cependant, ces approches présentent certains inconvénients. En effet, dans l'approche de Broman et Shensa [7], pour déterminer le polytope modifié, on doit d'abord calculer un orthotope autour du polytope dont le nombre de faces a dépassé le seuil N_f . Cet orthotope constitue une première bornitude extérieure dont on veut diminuer la taille en déterminant les hyperplans qui le coupent au mieux afin d'en tirer le plus petit polytope. La limitation du nombre de faces est accompagnée par une augmentation de la taille du polytope due à la partie non éliminée de l'orthotope. La figure 5.13 illustre cet inconvénient. Cette erreur s'accumule à chaque fois qu'on calcule

le polytope modifié, ce qui peut mener à une région de dimension plus large.

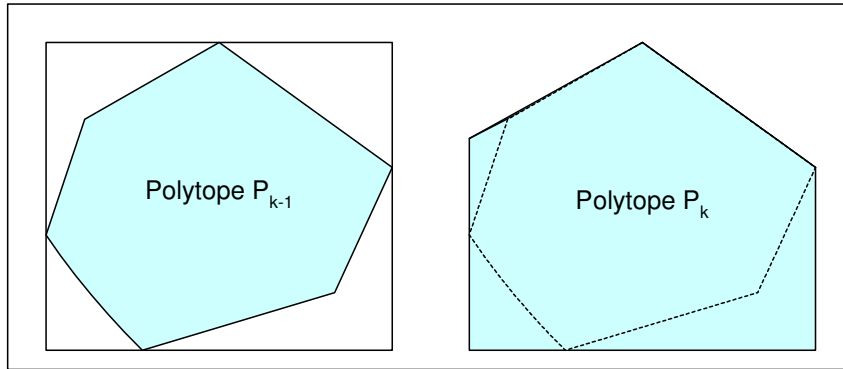


FIG. 5.13 – Augmentation du volume du polytope à complexité limitée lors de sa mise à jour par la méthode de Broman et Shensa

Dans l'approche de Lahanier et Walter [39], ce problème n'est pas rencontré puisqu'on ne calcule pas une approximation extérieure du polytope intersection tel que l'orthotope, mais on élimine les hyperplans les plus éloignés du centre tout en gardant les autres hyperplans formant le polytope à complexité limitée. Cette procédure présente le risque d'avoir occasionnellement une grande augmentation de la taille du polytope modifié. De plus on peut obtenir une région non bornée dans le cas où les hyperplans choisis ne forment pas un polytope mais un polyèdre. La figure 5.14 illustre cette situation.

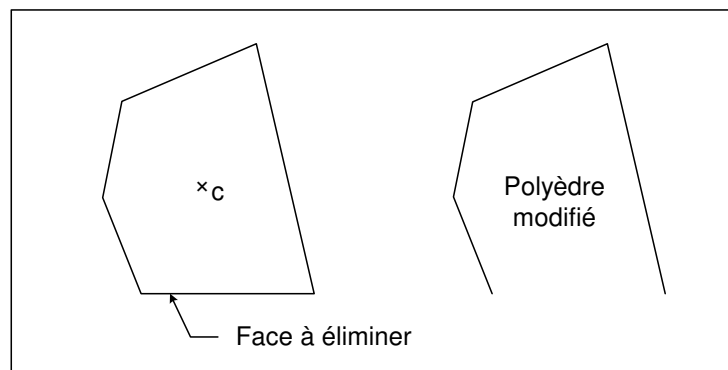


FIG. 5.14 – Inconvénient de la méthode de Lahanier et Walter

Pour l'approche de Veres et al [64], un bon choix de l'ensemble de directions D

permet d'éviter ce problème. Cependant, puisque le nombre maximal de faces dépend essentiellement du nombre de directions permises formulé par l'ensemble D , l'augmentation du cardinal de D entraîne l'augmentation du nombre de faces de l'approximation, ce qui multiplie le nombre de mises à jour du polytope exact sans toutefois limiter sa complexité. De plus la détermination des nouvelles bornes de l'erreur nécessite le calcul des ensembles de sommets, ce qui nécessite le calcul de l'intersection du polytope P_{k-1} avec chacun des demi-espaces positifs engendrés par la nouvelle mesure, ce qui alourdit énormément les calculs.

Une nouvelle approche développée par Maraoui et Messaoud [46] vise à pallier les inconvénients des approches précédentes et à en garder les avantages. Elle s'inspire des approches précédentes et en particulier celle de Lahanier et Walter et celle de Veres et al. L'algorithme proposé suit les étapes suivantes :

- Comme pour les autres approches, on calcule le polytope exact jusqu'à ce que le nombre de faces atteigne un seuil N_f .
- Une fois que le nombre de faces dépasse N_f on calcule les sommets, les hyperplans et le centre du polytope exact P_k résultant de l'intersection du polytope P_{k-1} à l'instant antérieur $k - 1$ et des deux demi-espaces positifs relatifs à la nouvelle mesure.
- On calcule et on stocke par ordre croissant les distances dH_j entre le centre C de P_k et chacun des hyperplans H_j caractérisant P_k .
 - L'équation d'un hyperplan peut être écrite sous la forme :

$$H_j = \{x/a_j^T X = b_j\} \quad \text{avec} \quad a_j, X \in \mathbb{R}^n \quad (5.60)$$

- Les composantes de C sont calculées comme suit :

$$C(i) = \frac{1}{2} [\min S^j(i) + \max S^j(i)] \quad (5.61)$$

où S^j est le $j^{\text{ème}}$ sommet de P_k et $S^j(i)$ la $i^{\text{ème}}$ composante de S^j .

- La distance dH_j entre le sommet S^j et le centre C du polytope est calculée comme suit :

$$dH_j = \frac{|a_j^T C - b_j|}{\|a_j\|} \quad (5.62)$$

- On prend les hyperplans un par un en commençant par le plus proche du centre C . Soit H_p cet hyperplan. Il est donné par :

$$H_p = \{X/a_p^T X = b_p\} \quad (5.63)$$

- On construit la bande formée par H_p et l'hyperplan qui lui est parallèle défini par :

$$H_m = \{X/a_p^T X = b_m\} \quad (5.64)$$

avec

$$b_m = \max \{a_p^T s / s \in S\} \quad (5.65)$$

S étant l'ensemble des sommets de l'intersection.

- On recommence avec les autres hyperplans formant le polytope et ceci en prenant soin que les hyperplans pris soient linéairement indépendants. A chaque fois on construit une nouvelle bande et on détermine l'intersection des bandes obtenues jusqu'à atteindre le seuil N_f .

Cette procédure est illustrée par la figure 5.15 où le nombre d'hyperplans de l'intersection est égal à 5 et le nombre seuil de faces du polytope à C.L. est $N_f = 4$. Les hyperplans H_2 et H_3 sont les hyperplans les plus proches du centre C . On calcule la bande formée par H_{2p} et H_{2m} et celle formée par H_{3p} et H_{3m} . Les deux bandes forment l'approximation à complexité limitée.

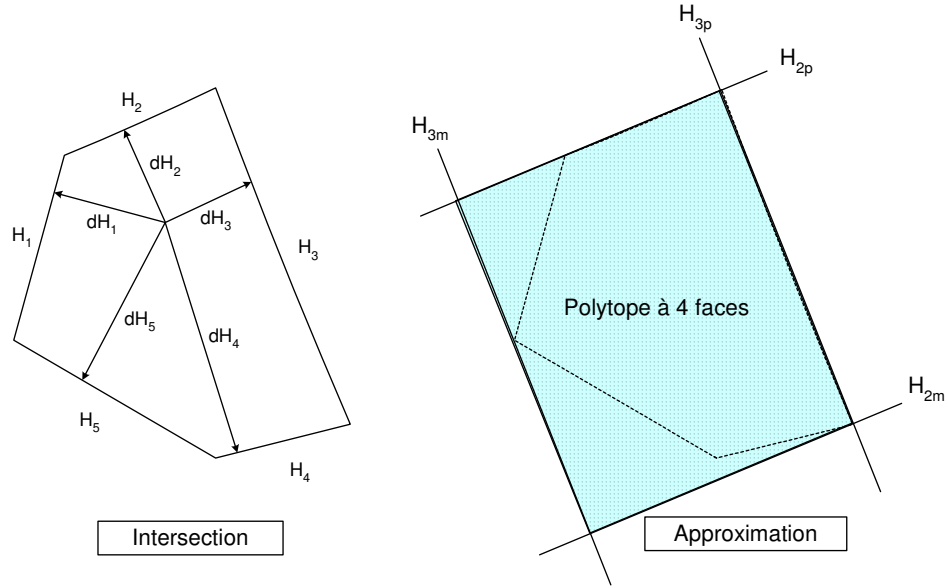
5.3 Identification robuste des systèmes définis à l'aide des BOG

Ninness et Gustafsson [53] ont montré que le développement d'une fonction absolument sommable $G(z)$ à l'aide d'une BOG s'écrit sous la forme suivante :

$$G(z) = \sum_{n=1}^{\infty} g_n B_n(z) \quad (5.66)$$

avec

$$B_n(z) = \frac{\sqrt{1 - |\xi_n|^2}}{z - \xi_n} \prod_{k=1}^{n-1} \frac{1 - \xi_{k-1}^* z}{z - \xi_k} ; n = 1, \dots \quad (5.67)$$


 FIG. 5.15 – Mise à jour du polytope à complexité limitée ($N_f = 4$)

L'écriture de la sortie de ce système à l'instant k peut être tronquée à un ordre N . Elle s'écrit donc sous la forme :

$$y(k) = \sum_{i=1}^N g_i x_i(k) + e(k) \quad (5.68)$$

N étant l'ordre de troncature du modèle, $\{g_i\}_{k=i,\dots,N}$ sont les coefficients de Fourier du développement sur la BOG, $x_i(k) = \sum_{l=1}^i b_l(k)u(k-l)$ sont les sorties des filtres de la BOG à l'instant k et $e(k)$ est l'erreur de modélisation.

On définit ϕ_k le vecteur de régression suivant :

$$\phi_k = \begin{bmatrix} x_1(k) & x_2(k) & \cdots & x_N(k) \end{bmatrix}^T \quad (5.69)$$

et θ le vecteur des coefficients de Fourier à estimer.

$$\theta = \begin{bmatrix} g_1 & g_2 & \cdots & g_N \end{bmatrix}^T \quad (5.70)$$

L'équation (5.68) s'écrit alors sous la forme suivante :

$$y(k) = \phi_k^T \theta + e(k) \quad (5.71)$$

Cette écriture de la sortie du modèle issu de la BOG sous forme de régression linéaire nous permet d'utiliser les techniques d'identification à erreur bornée, afin d'estimer les coefficients de Fourier du développement.

Dans ce qui suit on applique les approches à erreur bornée présentées dans ce chapitre (polytopique, orthotopique et à complexité limitée) au cas des modèles issus des BOG (Laguerre et BOG).

5.4 Résultats de simulations

Dans cette partie, on présente les résultats de simulation des différents algorithmes à erreur bornée présentés précédemment. Ces simulations ont été mises au point en utilisant l'outil GBT (Geometric Bounding Toolbox) intégré au logiciel Matlab.

Ces simulations ont pour buts :

- de valider les méthodes d'identification à erreur bornée présentées,
- de faire une étude comparative des différentes approches présentées.
- de comparer les algorithmes à erreur bornée à d'autres méthodes d'identification dont la méthode des moindres carrés.

5.4.1 Présentation de l'outil GBT

Cet outil présenté par Veres et al en 1996 offre plusieurs routines qui facilitent la programmation des différents algorithmes. Parmi ces routines on peut citer celles qui mettent à jour un polytope. Ces régions qui peuvent être visualisées sont décrites par des matrices qui contiennent des informations sur les paramètres caractéristiques de la région mise à jour. Dans le cas où cette région est un polytope par exemple, la matrice contient les hyperplans et les sommets du polytope, ainsi que la liste d'hyperplans adjacents à chaque sommet. D'autres routines permettent de calculer des paramètres qui facilitent l'évaluation des performances des différents algorithmes et l'extraction d'informations sur les régions obtenues. On note plus particulièrement le calcul du volume de la région et la détermination de son centre, l'extraction des faces et des sommets d'un polyèdre, et bien d'autres opérations sur les différentes régions qu'on peut calculer.

5.4.2 Critères d'évaluation

L'évaluation des performances consiste à déterminer les critères suivants :

- Le centre de la région finale qui peut être considéré comme un estimateur du vecteur de paramètres réels,
- la taille de la région finale caractérisée par le volume de cette région et par les intervalles d'incertitude des paramètres estimés,
- le taux d'actualisation (TA), exprimé en %, qui représente le rapport du nombre de mesures qui actualise la région, par le nombre total de mesures traitées,
- le temps de convergence (TCo) exprimé en secondes (s), correspond au temps nécessaire pour que le volume de la région devienne inférieur à un seuil, fixé à $V_{seuil} = 10^{-2}$ dans notre cas.

Les bornes de l'erreur utilisées dans les simulations suivantes sont calculées sur la base du bruit additif $e(k)$ telles que $\delta_k = 110\% e(k)$.

Les critères de performance précédents sont présentés dans les tableaux 5.1 et 5.3. Dans les tableaux 5.2 et 5.4, on présente le centre de la région des paramètres estimés.

5.4.3 Modélisation et identification d'un système linéaire

Les approches d'identification à erreur inconnue mais bornée sont appliquées aux modèles des BOG représentant le système discret d'ordre 3 défini par sa fonction de transfert $G(z)$.

$$G(z) = \frac{b(z)}{(z - 0.7) a(z)} \quad (5.72)$$

avec

$$a(z) = z^2 - 2r \cos(\varphi)z + r^2 \quad (5.73)$$

$$b(z) = z^2 - 2r \cos(\varphi + \Delta\varphi)z + r^2 \quad (5.74)$$

$$\varphi = 1.3\pi/4 \quad (5.75)$$

$$r = 0.9 \quad (5.76)$$

$$\Delta\varphi = 0.2\pi/4 \quad (5.77)$$

Les pôles de $a(z)$ sont $0.392 \pm j0.639$

Le signal d'entrée est une séquence de bruit blanc gaussien de longueur $L = 100$, de moyenne nulle et avec une variance égale à l'unité.

L'ordre de troncature des développements sur les bases orthonormales (Laguerre et BOG) est $N=4$. Dans les tableaux qui suivent, on présente les trois coefficients de

Fourier les plus importants (g_i) ainsi que leurs intervalles d'incertitude (Δg_i), les quatrièmes coefficients étant non significatifs et n'ayant pas d'influence sur l'étude comparative. Dans ces mêmes tableaux on présente L'Erreur Quadratique Moyenne Normalisée $EQMN_{sortie}$ entre la sortie du système et celle du modèle issu de la base de Laguerre ou de la BOG.

Les coefficients de Fourier des développements sur des bases orthonormales sont les centres des espaces d'incertitude (polytope, orthotope et polytope à C.L.) estimés à l'aide des différentes approches à erreur bornée.

Dans les figures 5.16, 5.17, 5.20, 5.21, 5.24, 5.25, 5.28 et 5.29, on trace l'évolution des valeurs maximales, des valeurs minimales et des valeurs moyennes des coefficients de Fourier estimés du développement sur les bases de fonctions de Laguerre et les BOG à l'aide de l'approche polytopique, orthotopique et polytopique à complexité limitée. Dans les figures 5.18, 5.19, 5.22, 5.23, 5.26, 5.27, 5.30 et 5.31 est tracée l'évolution des volumes des régions d'incertitude estimées (polytope, polytope à C.L. et orthotope) en fonction du nombre d'itérations.

- Identification robuste des systèmes linéaires modélisés à l'aide des bases de fonctions de Laguerre (modèle de Laguerre incertain)

Le pôle optimal de Laguerre, calculé en utilisant l'algorithme de Fu et Fumont [24] est $\xi = 0.703$.

RSB = 10dB						
Approche	Volume final	Intervalles d'incertitude			TA(%)	TCo(s)
		Δg_1	Δg_2	Δg_3		
Polytopique	$3.531 \cdot 10^{-9}$	$1.755 \cdot 10^{-2}$	$1.735 \cdot 10^{-2}$	$1.372 \cdot 10^{-2}$	37	0.881
Poly. à C.L. ($N_f = 12$)	$2.538 \cdot 10^{-7}$	$3.697 \cdot 10^{-2}$	$3.025 \cdot 10^{-2}$	$4.232 \cdot 10^{-2}$	76	1.152
Poly. à C.L. ($N_f = 10$)	$5.155 \cdot 10^{-7}$	$7.831 \cdot 10^{-2}$	$4.987 \cdot 10^{-2}$	$7.361 \cdot 10^{-2}$	86	1.729
Orthotopique	$5.064 \cdot 10^{-3}$	$3.071 \cdot 10^{-1}$	$2.529 \cdot 10^{-1}$	$2.349 \cdot 10^{-1}$	100	7.543
RSB = 3dB						
Polytopique	$2.344 \cdot 10^{-6}$	$7.441 \cdot 10^{-2}$	$8.476 \cdot 10^{-2}$	$7.914 \cdot 10^{-2}$	42	1.092
Poly. à C.L. ($N_f = 12$)	$4.716 \cdot 10^{-5}$	$1.752 \cdot 10^{-2}$	$1.093 \cdot 10^{-2}$	$1.496 \cdot 10^{-2}$	80	1.342
Poly. à C.L. ($N_f = 10$)	$4.185 \cdot 10^{-4}$	$3.797 \cdot 10^{-1}$	$2.273 \cdot 10^{-1}$	$2.271 \cdot 10^{-1}$	90	1.829
Orthotopique	$2.255 \cdot 10^{-2}$	$4.517 \cdot 10^{-1}$	$3.998 \cdot 10^{-1}$	$3.690 \cdot 10^{-1}$	98	-

TAB. 5.1 – Volume des régions estimées, intervalles d'incertitude (Δg_i), taux d'actualisation (TA) et temps de convergence (Tco) pour différents algorithmes UBBE (Modèle de Laguerre)

Approche	Centre de la région (RSB = 10dB)			$EQMN_{sortie}$
	\hat{g}_1	\hat{g}_2	\hat{g}_3	
Moindres carrés	$3.267 \cdot 10^{-1}$	$2.130 \cdot 10^{-1}$	$6.608 \cdot 10^{-2}$	$5.408 \cdot 10^{-3}$
Polytopique	$3.258 \cdot 10^{-1}$	$2.179 \cdot 10^{-1}$	$6.769 \cdot 10^{-2}$	$6.727 \cdot 10^{-3}$
Polytopique à C.L. ($N_f = 12$)	$3.254 \cdot 10^{-1}$	$2.034 \cdot 10^{-1}$	$7.496 \cdot 10^{-2}$	$7.534 \cdot 10^{-3}$
Polytopique à C.L. ($N_f = 10$)	$3.453 \cdot 10^{-1}$	$2.373 \cdot 10^{-1}$	$6.525 \cdot 10^{-2}$	$9.413 \cdot 10^{-3}$
Orthotopique	$3.270 \cdot 10^{-1}$	$2.401 \cdot 10^{-1}$	$4.963 \cdot 10^{-2}$	$1.281 \cdot 10^{-2}$
Centre de la région (RSB = 3dB)				
	\hat{g}_1	\hat{g}_2	\hat{g}_3	$EQMN_{sortie}$
Moindres carrés	$3.321 \cdot 10^{-1}$	$1.841 \cdot 10^{-1}$	$8.582 \cdot 10^{-2}$	$1.353 \cdot 10^{-2}$
Polytopique	$3.278 \cdot 10^{-1}$	$1.764 \cdot 10^{-1}$	$7.929 \cdot 10^{-2}$	$1.483 \cdot 10^{-2}$
Polytopique à C.L. ($N_f = 12$)	$3.125 \cdot 10^{-1}$	$1.921 \cdot 10^{-1}$	$6.648 \cdot 10^{-2}$	$1.526 \cdot 10^{-2}$
Polytopique à C.L. ($N_f = 10$)	$3.252 \cdot 10^{-1}$	$1.687 \cdot 10^{-1}$	$5.482 \cdot 10^{-2}$	$2.137 \cdot 10^{-2}$
Orthotopique	$3.357 \cdot 10^{-1}$	$1.662 \cdot 10^{-1}$	$5.471 \cdot 10^{-2}$	$2.686 \cdot 10^{-2}$

TAB. 5.2 – Coordonnées des centres des régions estimées avec les différentes approches UBBE (Modèle de Laguerre)

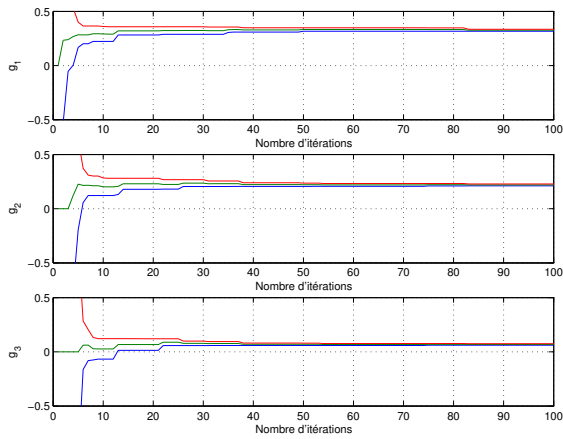


FIG. 5.16 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope ($RSB = 10dB$) (Modèle de Laguerre)

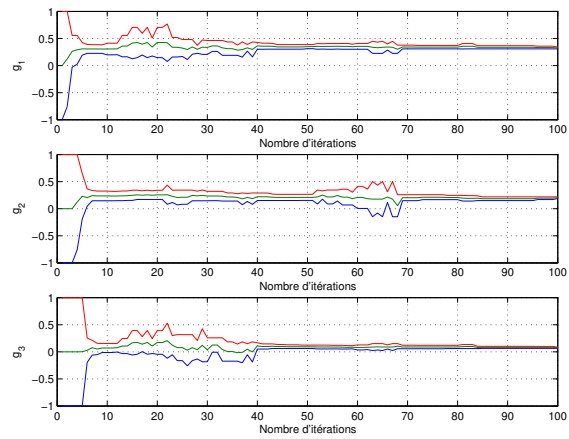


FIG. 5.17 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope à complexité limitée ($N_f = 12$ et $RSB = 10dB$) (Modèle de Laguerre)

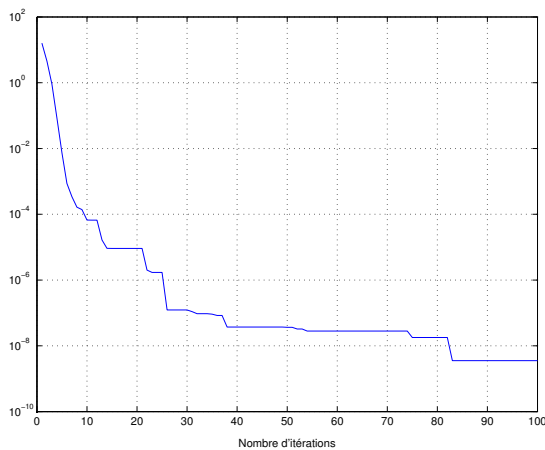


FIG. 5.18 – Volume du Polytope ($RSB = 10dB$) (Modèle de Laguerre)

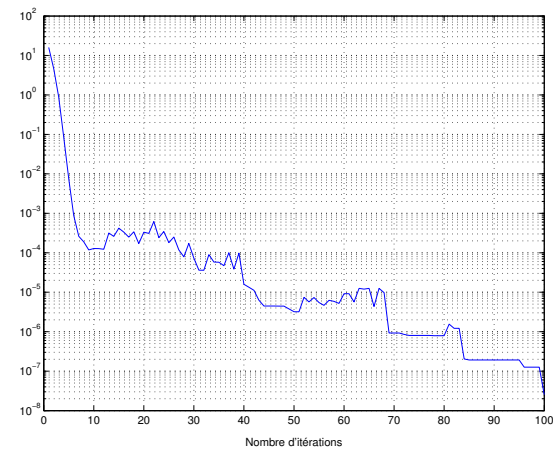


FIG. 5.19 – Volume du polytope à complexité limitée ($N_f = 12$ et $RSB = 10dB$) (Modèle de Laguerre)

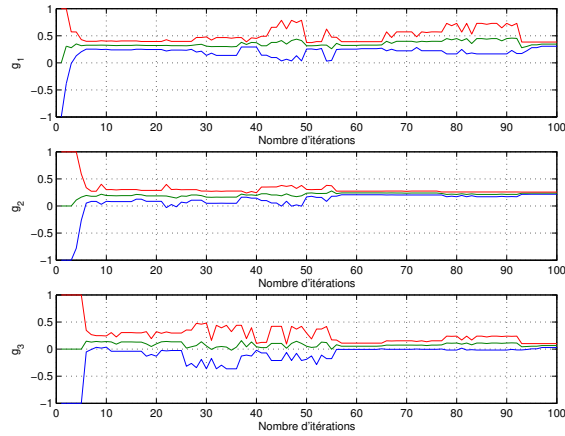


FIG. 5.20 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope à complexité limitée ($N_f = 10$ et $RSB = 10dB$) (Modèle de Laguerre)

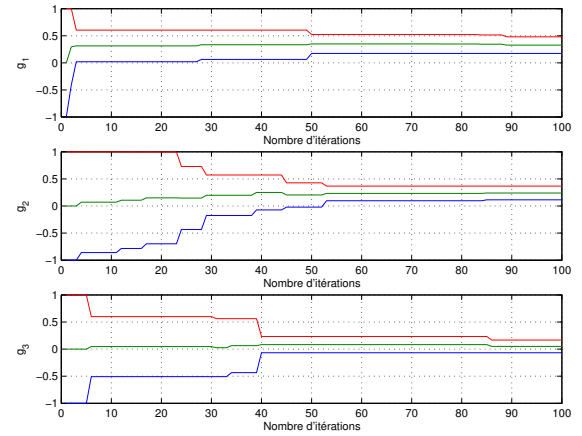


FIG. 5.21 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets de l'orthotope ($RSB = 10dB$) (Modèle de Laguerre)

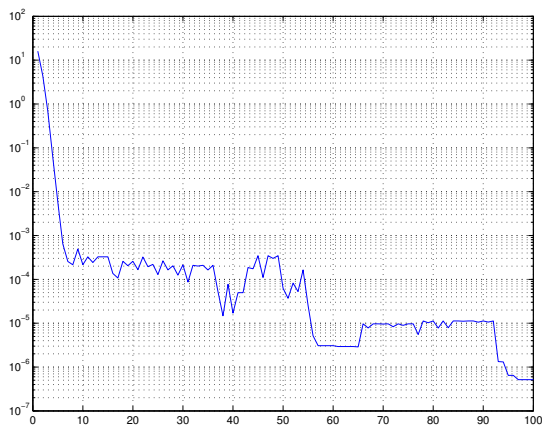


FIG. 5.22 – Volume du polytope à complexité limitée ($N_f = 10$ et $RSB = 10dB$) (Modèle de Laguerre)

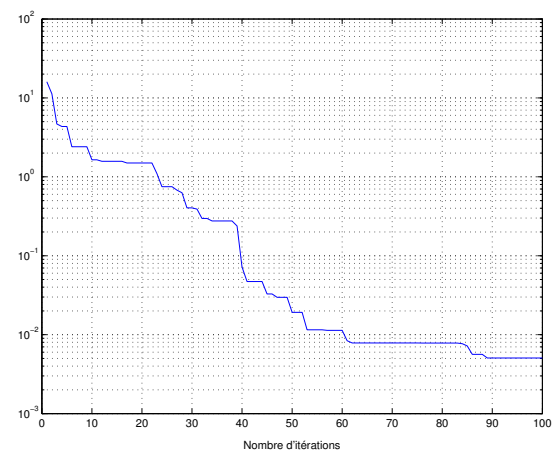


FIG. 5.23 – Volume de l'orthotope ($RSB = 10dB$) (Modèle de Laguerre)

- Identification robuste des systèmes linéaires modélisés à l'aide de la base orthogonale généralisée (modèle de la BOG incertain)

Les pôles optimaux de la BOG, calculés en utilisant l'algorithme présenté au premier chapitre [34] sont :

$$\xi_1 = 0.599$$

$$(\xi_2, \xi_2^*) = 0.394 \pm j 0.771$$

RSB = 10dB						
Approche	Volume final	Intervalle d'incertitude			TA (%)	TCo(s)
		Δg_1	Δg_2	Δg_3		
Polytopique	$5.387 \cdot 10^{-10}$	$7.657 \cdot 10^{-3}$	$1.938 \cdot 10^{-2}$	$1.609 \cdot 10^{-2}$	39	0.80
Poly. à C.L. ($N_f = 12$)	$6.188 \cdot 10^{-8}$	$2.791 \cdot 10^{-2}$	$7.647 \cdot 10^{-2}$	$1.919 \cdot 10^{-2}$	68	0.86
Poly. à C.L. ($N_f = 10$)	$6.332 \cdot 10^{-7}$	$8.039 \cdot 10^{-2}$	$1.233 \cdot 10^{-1}$	$7.583 \cdot 10^{-2}$	93	0.87
Orthotopique	$9.494 \cdot 10^{-1}$	$9.231 \cdot 10^{-1}$	$1.166 \cdot 10^{-0}$	$8.728 \cdot 10^{-1}$	99	-
RSB = 3dB						
Polytopique	$5.252 \cdot 10^{-6}$	$6.206 \cdot 10^{-2}$	$1.323 \cdot 10^{-1}$	$9.939 \cdot 10^{-2}$	48	1.23
Poly. à C.L. ($N_f = 12$)	$8.626 \cdot 10^{-5}$	$1.827 \cdot 10^{-1}$	$1.520 \cdot 10^{-1}$	$5.666 \cdot 10^{-1}$	49	1.59
Poly. à C.L. ($N_f = 10$)	$1.463 \cdot 10^{-4}$	$5.087 \cdot 10^{-1}$	$8.913 \cdot 10^{-2}$	$5.827 \cdot 10^{-1}$	92	1.87
Orthotopique	$3.565 \cdot 10^{-0}$	$1.374 \cdot 10^{-0}$	$1.446 \cdot 10^{-0}$	$1.248 \cdot 10^{-0}$	98	-

TAB. 5.3 – Volume des régions estimées, intervalles d'incertitude (Δg_i), taux d'actualisation (TA) et temps de convergence (Tco) pour différents algorithmes UBBE (Modèle issu de la BOG)

Approche	Centre de la région (RSB = 10dB)			$EQMN_{sortie}$
	\hat{g}_1	\hat{g}_2	\hat{g}_3	
Moindres carrés	$2.737 \cdot 10^{-1}$	$4.375 \cdot 10^{-1}$	$7.075 \cdot 10^{-3}$	$5.293 \cdot 10^{-3}$
Polytopique	$2.829 \cdot 10^{-1}$	$4.404 \cdot 10^{-1}$	$1.871 \cdot 10^{-3}$	$4.336 \cdot 10^{-3}$
Polytopique à C.L. ($N_f = 12$)	$2.846 \cdot 10^{-1}$	$4.158 \cdot 10^{-1}$	$2.648 \cdot 10^{-2}$	$1.598 \cdot 10^{-2}$
Polytopique à C.L. ($N_f = 10$)	$2.642 \cdot 10^{-1}$	$4.448 \cdot 10^{-1}$	$2.660 \cdot 10^{-3}$	$1.933 \cdot 10^{-2}$
Orthotopique	$2.747 \cdot 10^{-1}$	$4.374 \cdot 10^{-1}$	$4.877 \cdot 10^{-2}$	$4.713 \cdot 10^{-2}$
	Centre de la région (RSB = 3dB)			$EQMN_{sortie}$
	\hat{g}_1	\hat{g}_2	\hat{g}_3	
Moindres carrés	$2.674 \cdot 10^{-1}$	$5.015 \cdot 10^{-1}$	$9.768 \cdot 10^{-3}$	$1.967 \cdot 10^{-2}$
Polytopique	$2.916 \cdot 10^{-1}$	$4.934 \cdot 10^{-1}$	$2.097 \cdot 10^{-2}$	$1.919 \cdot 10^{-2}$
Polytopique à C.L. ($N_f = 12$)	$2.706 \cdot 10^{-1}$	$4.524 \cdot 10^{-1}$	$1.213 \cdot 10^{-2}$	$2.115 \cdot 10^{-2}$
Polytopique à C.L. ($N_f = 10$)	$3.076 \cdot 10^{-1}$	$3.988 \cdot 10^{-1}$	$5.671 \cdot 10^{-2}$	$5.321 \cdot 10^{-2}$
Orthotopique	$3.971 \cdot 10^{-1}$	$3.164 \cdot 10^{-1}$	$6.678 \cdot 10^{-2}$	$1.883 \cdot 10^{-1}$

TAB. 5.4 – Coordonnées des centres des régions estimées avec les différentes approches UBBE (Modèle issu de la BOG)

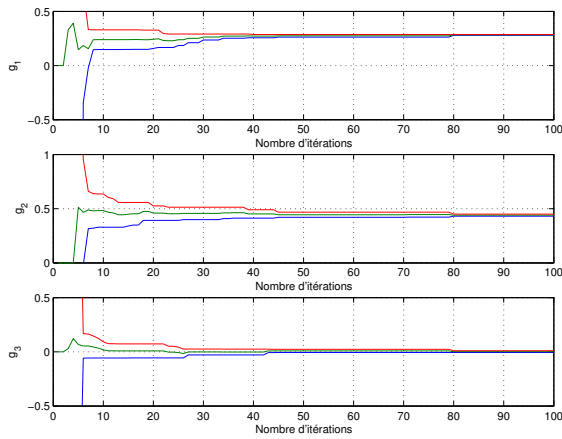


FIG. 5.24 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope ($RSB = 10dB$) (modèle issu de la BOG)

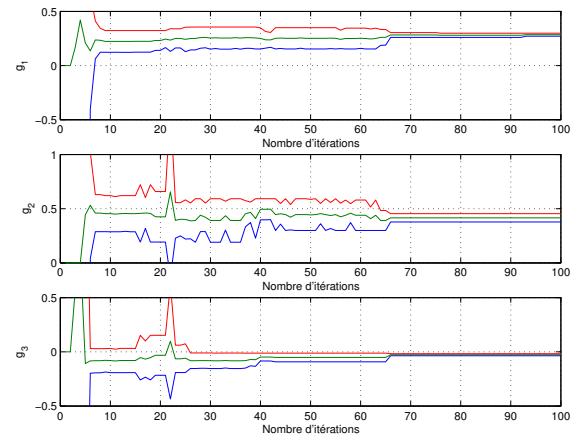


FIG. 5.25 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope à complexité limitée ($N_f = 12$ et $RSB = 10dB$) (Modèle issu de la BOG)

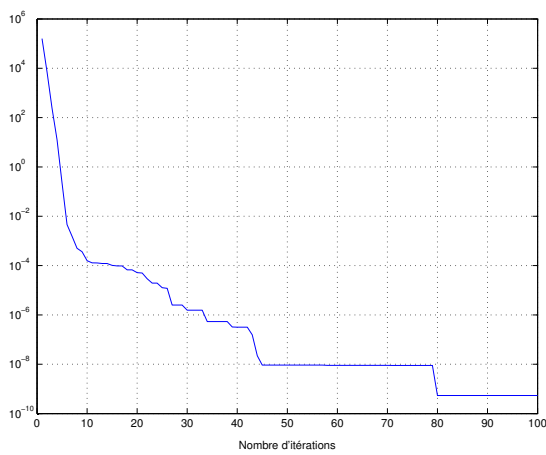


FIG. 5.26 – Volume du Polytope ($RSB = 10dB$) (Modèle issu de la BOG)

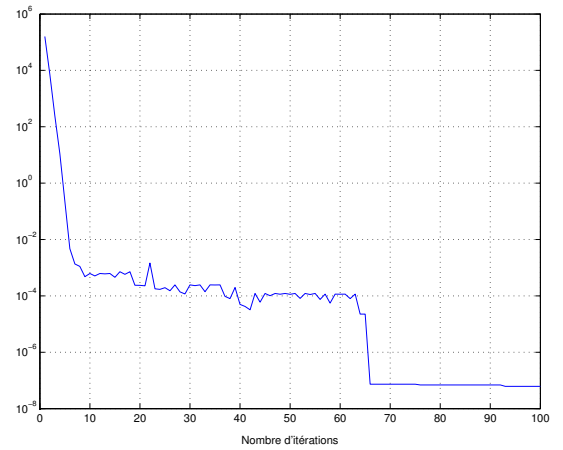


FIG. 5.27 – Volume du polytope à complexité limitée ($N_f = 12$ et $RSB = 10dB$) (Modèle issu de la BOG)

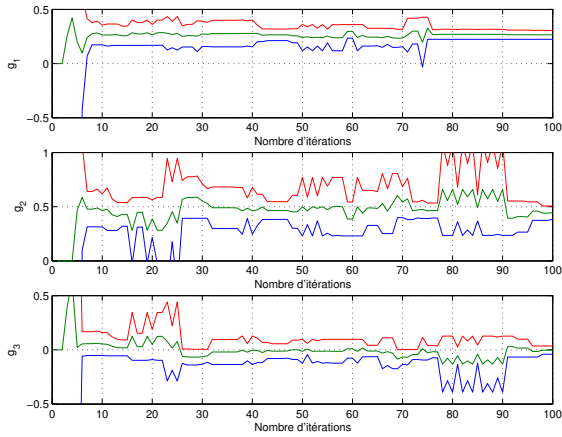


FIG. 5.28 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope à complexité limitée ($N_f = 10$ et $RSB = 10dB$) (Modèle issu de la BOG)

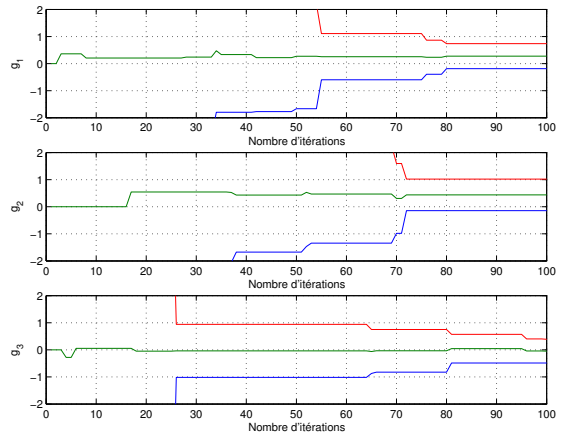


FIG. 5.29 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets de l'orthotope ($RSB = 10dB$) (Modèle issu de la BOG)

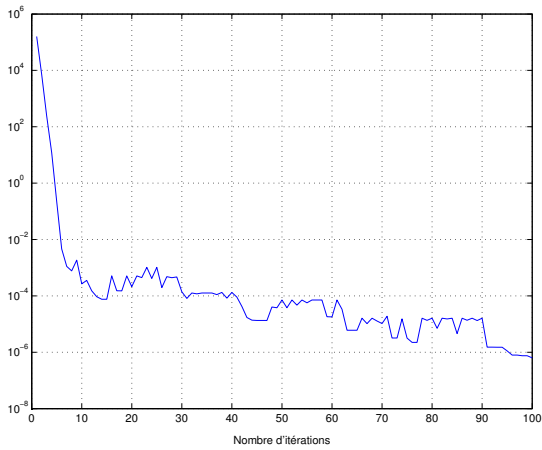


FIG. 5.30 – Évolution du volume du Polytope à complexité limitée ($RSB = 10dB$ et $N_f = 10$) (Modèle issu de la BOG)

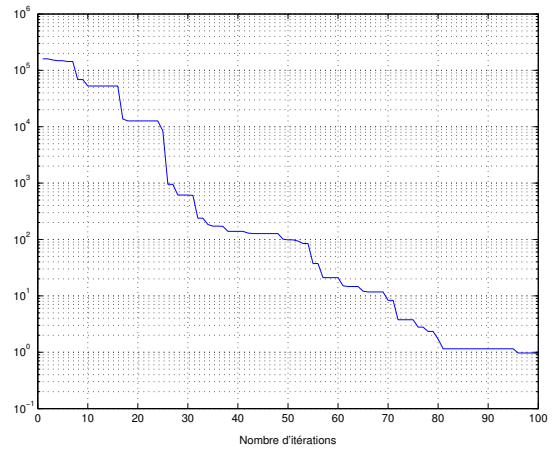


FIG. 5.31 – Évolution du volume de l'orthotope ($RSB = 10dB$) (Modèle issu de la BOG)

5.4.4 Analyse des résultats

Les tableaux 5.1 et 5.3 résument les performances des différents algorithmes de mise à jour du polytope exact ainsi que ses approximations du type orthotopique et polytopique à complexité limitée. Les tableaux 5.2 et 5.4 présentent les coefficients de Fourier estimés par les différents algorithmes et l' $EQMN_{sortie}$.

- **Approche polytopique**

Parmi les différentes approches de l'erreur inconnue mais bornée, l'approche polytopique donne toujours les meilleurs résultats du point de vue de la taille des domaines d'incertitude obtenus (volume minimal et intervalles d'incertitude minimaux). Ceci est évidemment le résultat attendu puisque cette approche détermine le domaine d'appartenance exact des paramètres. Cette approche présente aussi l'avantage de converger rapidement, c'est à dire que les intervalles d'incertitude sur les paramètres diminuent rapidement et on s'approche de la région de volume minimal.

Les figures 5.16 et 5.24 présentent l'évolution des coordonnées du centre du polytope qui représentent ici les coefficients de Fourier du développement à l'aide de la base de fonctions de Laguerre et de la BOG, ainsi que celle des bornes des intervalles d'incertitude de ces coefficients. Elles illustrent l'évolution du centre de la région ainsi que celles des valeurs maximales et minimales pour chaque paramètre. Les figures 5.18 et 5.26 montrent l'évolution du volume du polytope mis à jour.

La convergence rapide de cette approche implique un taux d'actualisation faible. On peut expliquer ceci par le fait que plus la région est petite, moins on a de chance d'avoir une intersection entre le polytope et les hyperplans engendrés par les différentes mesures. Le nombre de mesures redondantes augmente ce qui réduit le nombre d'actualisation.

Le nombre de faces et de sommets augmente avec chaque nouvelle mesure non redondante, ce qui engendre un polytope de plus en plus complexe. Ceci nous permet de déduire que cette méthode n'est pas pratique dans des procédures en temps réel par exemple.

- **Approche orthotopique**

L'approche orthotopique présente l'avantage d'être simple et fournit une région ayant un nombre de directions et de faces fixes. Les figures 5.21 et 5.29 donnent l'évolution des coordonnées du centre de l'orthotope qui représentent ici les coefficients de Fourier du développement à l'aide de la base de fonctions de Laguerre et de la BOG, ainsi que celles des bornes des intervalles d'incertitude de ces coefficients. Les figures 5.23 et 5.31 montrent l'évolution du volume de ces orthotopes. On note avec cette approche, la lenteur de la convergence. En effet, on s'approche des paramètres réels par paliers. Ceci est dû au grand nombre de mesures redondantes, ce qui entraîne le faible taux d'actualisation. La redondance des mesures est due essentiellement au nombre réduit de sommets appartenant au demi espace positif. Elle a lieu si ce nombre de sommets est au moins égal à $2^{n-1} + 1$ [49], ce qui est fréquent, contrairement au cas du polytope où pour avoir une redondance tous les sommets du polytope doivent se situer dans le demi espace positif ce qui est beaucoup moins probable.

Puisque l'orthotope admet un nombre de sommets fixe sa complexité reste constante à chaque itération de l'algorithme. Donc le temps de calcul d'une mise à jour non redondante est, contrairement à l'approche polytopique, pratiquement constant. Les orthotopes obtenus présentent des volumes et des intervalles d'incertitude assez larges comparés au polytope.

- **Approche polytopique à complexité limitée**

Les figures 5.17 et 5.25 relatives aux polytopes à complexité limitée avec un nombre de faces seuil $N_f = 12$ ainsi que les figures 5.20 et 5.28 relatives aux polytopes à complexité limitée avec un nombre de faces seuil $N_f = 10$ présentent l'évolution des coordonnées du centre du polytope à C.L. qui représentent ici les coefficients de Fourier du développement à l'aide de la base de fonctions de Laguerre et de la BOG, ainsi que celles des bornes des intervalles d'incertitude de ces coefficients. Les figures 5.19 et 5.27 ainsi que les figures 5.22 et 5.30 montrent l'évolution du volume du polytope mis à jour.

L'augmentation du nombre maximum de faces implique une amélioration de certaines performances, c'est à dire, on obtient des volumes finaux plus petits, des intervalles d'incertitude plus étroits, un taux d'actualisation plus faible et une convergence

plus rapide. C'est un résultat prévisible puisque plus on augmente le nombre de faces du polytope à C.L., plus on s'approche du polytope exact. En contre partie, le temps de calcul augmente avec le nombre de faces. Bien entendu, les résultats dépendent fortement du choix de N_f .

Les pics observés dans les figures 5.19, 5.22, 5.27 et 5.30 représentent une augmentation du volume du polytope à C.L. au cours du déroulement de l'algorithme. Ce phénomène ne se produit pas en utilisant les polytopes classiques ou les orthotopes. C'est un inconvénient de cette méthode [46]. Mais on remarque qu'à partir d'un certain nombre de mesures, le volume du polytope à C.L. se stabilise pour converger.

La diminution du RSB (et donc l'augmentation des bornes de l'erreur), conduit à une augmentation de la taille des régions obtenues et des temps de convergence.

5.5 Identification robuste des systèmes de Volterra

Les approches d'identification à erreur inconnue mais bornée peuvent aussi être utilisées afin d'identifier des systèmes non linéaires, à condition bien sûr d'écrire la sortie de ces modèles sous forme de régression linéaire. Dans cette partie on va identifier des modèles de Volterra classiques, des modèles de Volterra représentés à l'aide d'une BOG ainsi que des modèles SVD-PARAFAC-Volterra cubiques.

Les conditions de ces simulations sont les mêmes que pour les simulations précédentes dans le cas linéaire. La borne de l'erreur est choisie égale à $\Delta = 120\% \max\{e(k)\}$. $e(k)$ étant le bruit additif au signal de sortie du système à l'instant k et N le nombre de mesures.

Au cours des simulations, on s'intéressera à l'approche polytopique classique. On comparera les paramètres du modèle de Volterra estimés à l'aide de la méthode des moindres carrés aux coordonnées du centre du polytope estimé à l'aide de l'approche à erreur bornée (les coordonnées du centre du polytope étant considérés comme les paramètres estimés du modèle simulé). Les critères d'évaluation seront :

- l'erreur quadratique moyenne normalisée $EQMN_{sortie}$ entre la sortie du système et celle du modèle.
- l'erreur quadratique moyenne normalisée $EQMN_{coef}$ entre les coefficients du modèle et ceux estimés avec les deux méthodes (*i.e. les moindres carrés et l'approche*

polytopique à erreur bornée).

5.5.1 Système 1

Le premier système simulé est un cas très particulier des systèmes de Volterra d'ordre 3. Il est défini par la figure 5.32 où $\alpha = 0.5$ et $\beta = 0.3$. Le noyau linéaire est de mémoire $M = 3$. Les noyaux quadratique et cubique sont de mémoire nulle, ce qui nous donne respectivement une non linéarité statique d'ordre 2 et d'ordre 3.

Le noyau linéaire h_1 est défini comme suit :

$$h_1(1) = 1 ; h_1(2) = -0.45 ; h_1(3) = 0.2775 \quad (5.78)$$

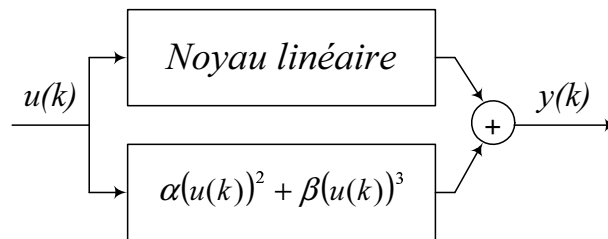


FIG. 5.32 – Cas des systèmes de Volterra d'ordre 3

Résultats de simulations

Les figures 5.33 et 5.34 montrent l'évolution des coordonnées du centre du polytope ainsi que celle des valeurs maximales et des valeurs minimales pour chaque paramètre du modèle estimé pour $RSB = 10dB$ et $RSB = 3dB$.

En analysant les figures 5.35 et 5.36, on note qu'en utilisant l'approche polytopique ou la méthode des moindres carrés on estime des coefficients du modèle qui approchent bien les paramètres réels du système.

On présente dans le tableau 5.5 les coefficients du modèle estimés par la méthode des moindres carrés et par l'approche polytopique. On remarque que les coefficients estimés sont relativement proches des vrais coefficients du système. La performance de cette estimation est évaluée à l'aide de l'erreur $EQMN_{coef}$ dans le tableau 5.6.

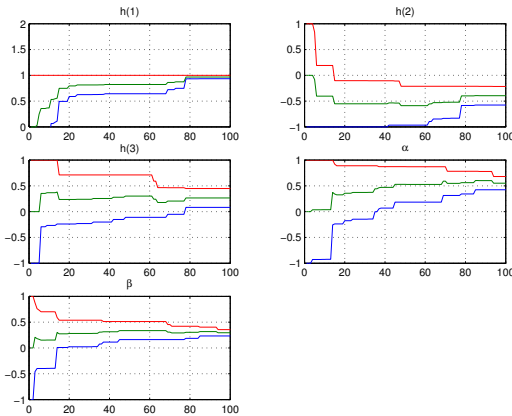


FIG. 5.33 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope ($RSB = 10dB$)

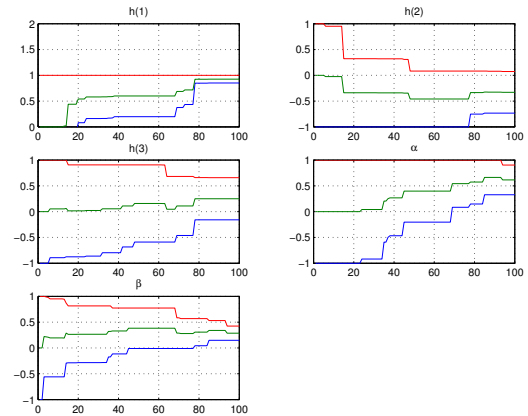


FIG. 5.34 – Coordonnées du centre et valeurs maximales et minimales des coordonnées des sommets du polytope ($RSB = 3dB$)

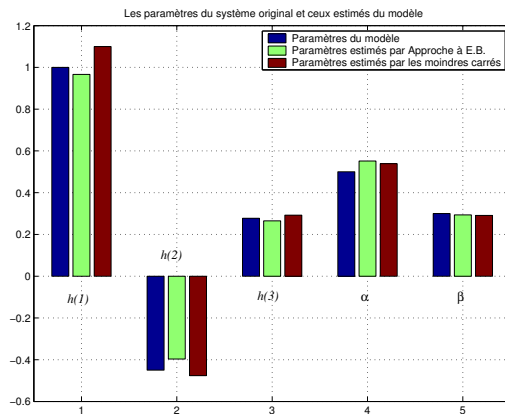


FIG. 5.35 – Paramètres du système comparés à leurs estimées par deux méthodes différentes ($RSB = 10dB$)

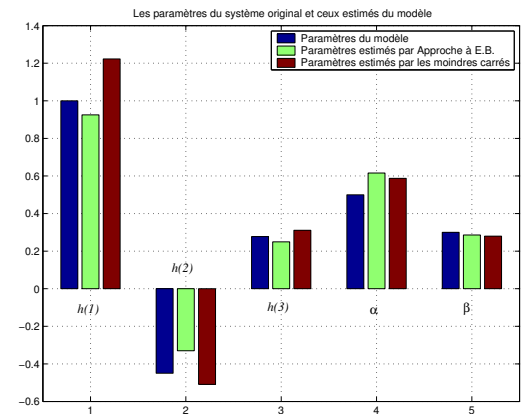


FIG. 5.36 – Paramètres du système comparés à leurs estimées par deux méthodes différentes ($RSB = 3dB$)

Le tableau 5.6 présente les valeurs de l' $EQMN_{sortie}$ et de l' $EQMN_{coef}$ pour deux méthodes d'identification et pour deux valeurs différentes du RSB. On remarque que pour des RSB faibles ($RSB = 3dB$), on arrive à reconstituer la sortie du système non-linéaire simulé. On obtient ainsi des valeurs d'erreurs sur les paramètres du modèle relativement faibles. On conclut qu'une bonne identification est accomplie en utilisant l'approche polytopique ainsi que la méthode des moindres carrés.

	$\mathbf{h(1) = 1}$	$\mathbf{h(2) = -0.45}$	$\mathbf{h(3) = 0.2775}$	$\alpha = \mathbf{0.5}$	$\beta = \mathbf{0.3}$
RSB = 10dB					
	$\widehat{\mathbf{h(1)}}$	$\widehat{\mathbf{h(2)}}$	$\widehat{\mathbf{h(3)}}$	$\widehat{\alpha}$	$\widehat{\beta}$
M. C.	1.099	-0.476	0.292	0.539	0.291
Polytopique	0.967	-0.396	0.265	0.552	0.294
RSB = 3dB					
	$\widehat{\mathbf{h(1)}}$	$\widehat{\mathbf{h(2)}}$	$\widehat{\mathbf{h(3)}}$	$\widehat{\alpha}$	$\widehat{\beta}$
M. C.	1.223	-0.509	0.311	0.587	0.280
Polytopique	0.925	-0.330	0.249	0.616	0.286

TAB. 5.5 – Paramètres du système et du modèle estimés par deux méthodes

RSB = 10dB		
	$\mathbf{EQMN}_{\text{sortie}}$	$\mathbf{EQMN}_{\text{coef}}$
M.C.	$4.10 \cdot 10^{-3}$	$4.67 \cdot 10^{-3}$
Polytopique	$4.94 \cdot 10^{-3}$	$6.22 \cdot 10^{-3}$
RSB = 3dB		
	$\mathbf{EQMN}_{\text{sortie}}$	$\mathbf{EQMN}_{\text{coef}}$
M.C.	$2.45 \cdot 10^{-2}$	$4.84 \cdot 10^{-2}$
Polytopique	$2.07 \cdot 10^{-2}$	$2.11 \cdot 10^{-2}$

TAB. 5.6 – $EQMN_{\text{sortie}}$ et $EQMN_{\text{coef}}$ pour les deux méthodes d'identification et pour deux valeurs différentes du RSB

5.5.2 Système 2

Le second système simulé est un système du second ordre de Volterra homogène, discret, invariant dans le temps de mémoire $M = 3$. Le noyau quadratique du système simulé peut toujours être ramené à un noyau triangulaire défini dans le tableau 5.7 suivant :

	$\mathbf{n_1 = 1}$	$\mathbf{n_1 = 2}$	$\mathbf{n_1 = 3}$
$\mathbf{n_2 = 1}$	1	-0.5	0.415
$\mathbf{n_2 = 2}$	0	0.2475	-0.20063
$\mathbf{n_2 = 3}$	0	0	0.15332

TAB. 5.7 – Coefficients du noyau de Volterra quadratique triangulaire H_2^{tri}

La sortie de ce système de Volterra simulé s'écrit comme suit :

$$y(n) = \sum_{n_1=1}^3 \sum_{n_2=n_1}^3 H_2^{tri}(n_1, n_2) u(n - n_1) u(n - n_2) \quad (5.79)$$

Afin que la sortie du filtre de Volterra s'écrive sous forme de régression linéaire, on arrange les éléments du noyau quadratique dans un vecteur Φ_2 et on définit le vecteur $U(n)$ suivant :

$$U(n) = \begin{bmatrix} u^2(n-1) \\ u(n-1)u(n-2) \\ u(n-1)u(n-3) \\ u^2(n-2) \\ u(n-2)u(n-3) \\ u^2(n-3) \end{bmatrix} \quad \text{et} \quad \Phi_2 = \begin{bmatrix} H_2^{tri}(1,1) \\ H_2^{tri}(1,2) \\ H_2^{tri}(1,3) \\ H_2^{tri}(2,2) \\ H_2^{tri}(2,3) \\ H_2^{tri}(3,3) \end{bmatrix} \quad (5.80)$$

L'équation 5.79 s'écrit alors :

$$y(n) = \Phi_2^T U(n) \quad (5.81)$$

Dans les simulations qui suivent, on s'intéresse aux estimées ainsi qu'aux intervalles d'incertitude des coefficients du noyau quadratique H_2^{tri}

Résultats de simulations

Dans le tableau 5.8, on présente les paramètres du noyau de Volterra quadratique original, leurs estimées à l'aide de la méthode des Moindres Carrés (M.C.) et les coordonnées du centre du polytope considérés comme les estimées des paramètres originaux par la méthode de l'Erreur Bornée (E.B.). La borne de l'erreur est $\Delta = 120\% \max\{e(k)\}$, $e(k)$ étant le bruit additif à l'instant k .

Les intervalles d'incertitude des paramètres estimés dans le tableau 5.9 deviennent plus petits quand le RSB augmente. Ceci s'explique par le fait que le RSB est fonction de la borne de l'erreur Δ . Cette dernière fixe la distance entre les hyperplans H_{k_j} $j = 1, 2$. Ce qui explique que les polytopes deviennent plus petits quand le RSB augmente.

Le tableau 5.10 présente les valeurs de l' $EQMN_{sortie}$ et de l' $EQMN_{coef}$ pour deux méthodes d'identification et pour deux différentes valeurs du RSB .

	$\mathbf{H}_2(1, 1)$ = 1	$\mathbf{H}_2(1, 2)$ = -0.5	$\mathbf{H}_2(1, 3)$ = 0.415	$\mathbf{H}_2(2, 2)$ = 0.2475	$\mathbf{H}_2(2, 3)$ = -0.20063	$\mathbf{H}_2(3, 3)$ = 0.15332
RSB = 10dB						
	$\widehat{\mathbf{H}}_2(1, 1)$	$\widehat{\mathbf{H}}_2(1, 2)$	$\widehat{\mathbf{H}}_2(1, 3)$	$\widehat{\mathbf{H}}_2(2, 2)$	$\widehat{\mathbf{H}}_2(2, 3)$	$\widehat{\mathbf{H}}_2(3, 3)$
M.C.	1.030	-0.496	0.421	0.244	-0.242	0.136
Polytopique	0.907	-0.501	0.423	0.262	-0.193	0.169
RSB = 3						
	$\widehat{\mathbf{H}}_2(1, 1)$	$\widehat{\mathbf{H}}_2(1, 2)$	$\widehat{\mathbf{H}}_2(1, 3)$	$\widehat{\mathbf{H}}_2(2, 2)$	$\widehat{\mathbf{H}}_2(2, 3)$	$\widehat{\mathbf{H}}_2(3, 3)$
M.C.	1.068	-0.492	0.427	0.241	-0.292	0.115
Polytopique	0.792	-0.501	0.433	0.279	-0.184	0.191

TAB. 5.8 – Paramètres du système et paramètres estimés par deux méthodes

RSB	$\Delta\widehat{\mathbf{H}}_2(1, 1)$	$\Delta\widehat{\mathbf{H}}_2(1, 2)$	$\Delta\widehat{\mathbf{H}}_2(1, 3)$	$\Delta\widehat{\mathbf{H}}_2(2, 2)$	$\Delta\widehat{\mathbf{H}}_2(2, 3)$	$\Delta\widehat{\mathbf{H}}_2(3, 3)$
10 dB	$1.86 \cdot 10^{-1}$	$4.43 \cdot 10^{-1}$	$3.14 \cdot 10^{-1}$	$1.83 \cdot 10^{-1}$	$3.40 \cdot 10^{-1}$	$2.51 \cdot 10^{-1}$
3 dB	$4.16 \cdot 10^{-1}$	$9.93 \cdot 10^{-1}$	$7.03 \cdot 10^{-1}$	$4.11 \cdot 10^{-1}$	$7.62 \cdot 10^{-1}$	$5.62 \cdot 10^{-1}$

TAB. 5.9 – Intervalles d'incertitude des paramètres estimés pour deux RSBs différents

Les figures 5.37 et 5.38 montrent la sortie du système simulé, celle du modèle estimé à l'aide des M.C. et celle du modèle estimé à l'aide de l'approche à E.B., respectivement pour $RSB = 10dB$ et $RSB = 3dB$. Pour les mêmes valeurs du rapport signal sur bruit, on trace dans les figures 5.39 et 5.40 l'erreur quadratique moyenne normalisée $EQMN_{sortie}$.

L'évolution des coordonnées du centre du polytope estimé et des bornes des intervalles d'incertitude par l'approche polytopique est présentée dans les figures 5.41 et 5.42 respectivement pour $RSB = 10dB$ et $RSB = 3dB$.

Dans les figures 5.43 et 5.44 sont comparés les coefficients du noyau quadratique de Volterra original et leurs estimées à l'aide de la méthode des M.C et de l'approche à E.B. pour deux RSBs différents.

RSB = 10dB		
	EQMN _{sortie}	EQMN _{coef}
M.C.	$1.76 \cdot 10^{-3}$	$1.91 \cdot 10^{-3}$
Polytopique	$3.79 \cdot 10^{-3}$	$5.97 \cdot 10^{-3}$
RSB = 3dB		
	EQMN _{sortie}	EQMN _{coef}
M.C.	$8.83 \cdot 10^{-3}$	$9.55 \cdot 10^{-3}$
Polytopique	$1.90 \cdot 10^{-2}$	$2.99 \cdot 10^{-2}$

TAB. 5.10 – EQMN_{sortie} et EQMN_{coef} pour les deux méthodes d'identification et pour deux différentes valeurs du RSB

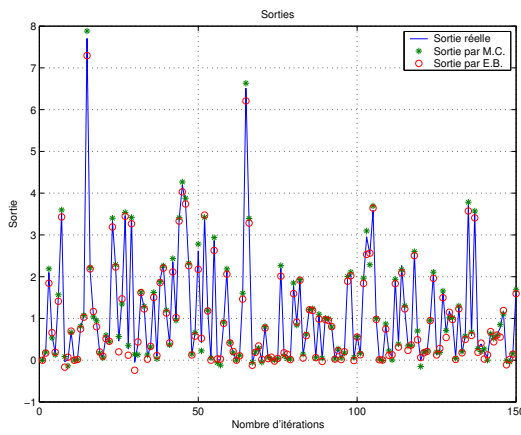


FIG. 5.37 – Sortie du système simulé et celles des deux modèles estimés par deux méthodes différentes (RSB = 10dB)

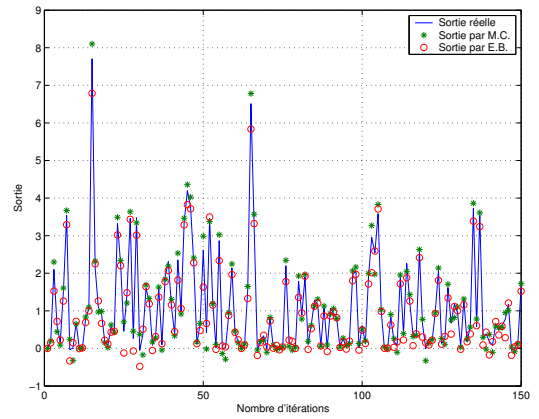


FIG. 5.38 – Sortie du système simulé et celles des deux modèles estimés par deux méthodes différentes (RSB = 3dB)

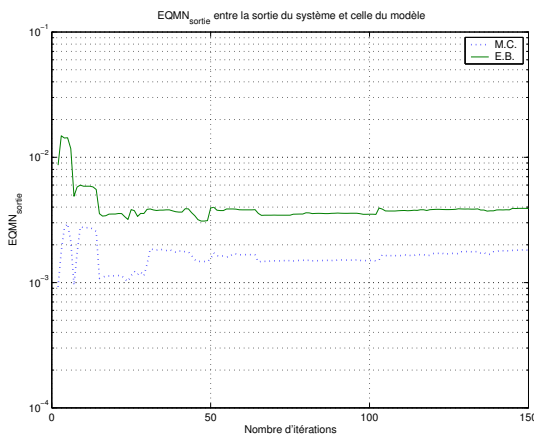


FIG. 5.39 – EQMN_{Sortie} (RSB = 10dB)

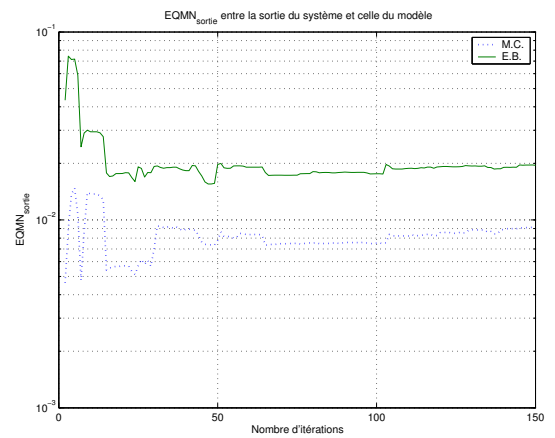


FIG. 5.40 – EQMN_{Sortie} (RSB = 3dB)

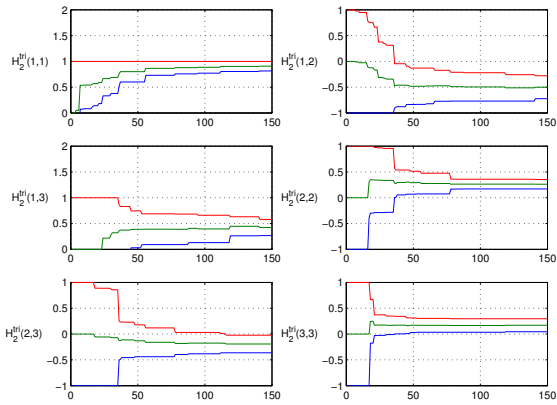


FIG. 5.41 – Coordonnées du centre du polytope et bornes des intervalles d’incertitude ($RSB = 10dB$)

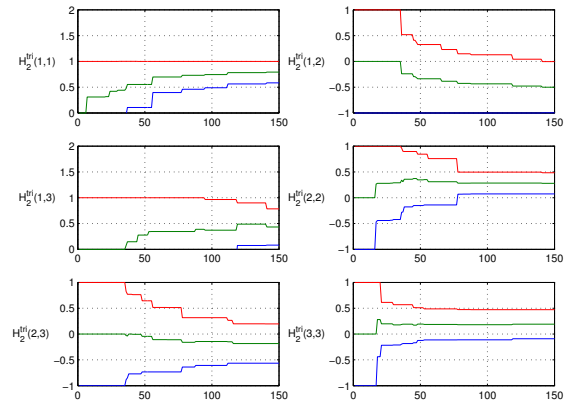


FIG. 5.42 – Coordonnées du centre du polytope et bornes des intervalles d’incertitude ($RSB = 3dB$)

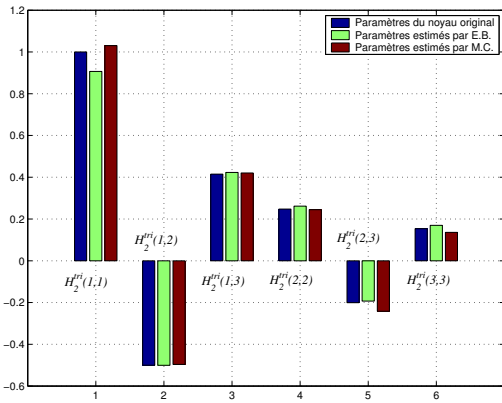


FIG. 5.43 – Paramètres du système comparés à leurs estimées par deux méthodes différentes ($RSB = 10dB$)

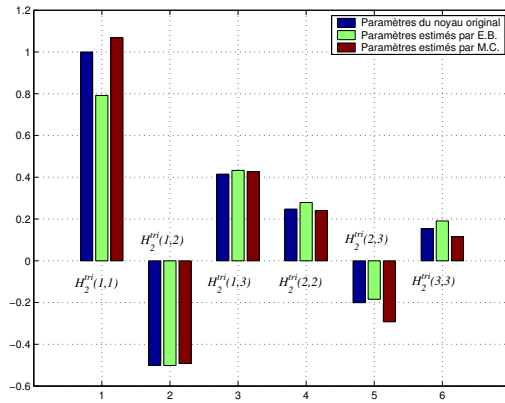


FIG. 5.44 – Paramètres du système comparés à leurs estimées par deux méthodes différentes ($RSB = 3dB$)

5.5.3 Système 3

Ce système est défini dans la figure 5.45 par un filtre oscillant du second ordre défini par sa fonction de transfert $H(z)$:

$$H(z) = \frac{z + 1}{(z + 0.74)(z + 0.62)} \quad (5.82)$$

suivi d'une non linéarité statique définie par un polynôme de degré 3 : $P(u) = 0.99u - 0.35u^3$.

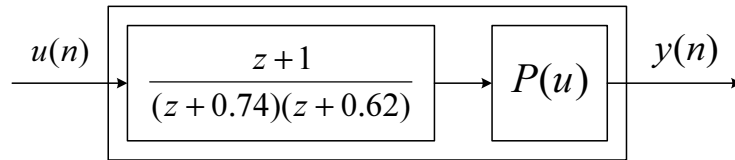


FIG. 5.45 – Système 3

Pour modéliser ce système on utilise un filtre de Volterra linéaire-cubique de mémoire $M = 2$, égale aussi au nombre de dynamiques du système simulé.

Afin de réduire la complexité du modèle de Volterra utilisé, on représente son noyau à l'aide d'une Base Orthogonale Généralisée. L'ordre de troncature du développement du modèle sur la BOG est $N = 2$ et les pôles de la base sont estimés à l'aide d'une méthode exhaustive mise au point par (A. Kibangu et al [36]). On trouve $\xi_1 = -0.731$ et $\xi_2 = -0.609$. Le modèle issu de la BOG appelé dorénavant BOG-Volterra linéaire cubique, est présenté dans la figure 5.46.

L'estimation des coefficients de Fourier du développement sur la BOG est réalisée en utilisant deux méthodes : la méthode des moindres carrés et l'approche polytopique des méthodes d'identification à erreur bornée. Les simulations de ce système comportent :

- la détermination des intervalles d'incertitude des coefficients de Fourier,
- la comparaison des coordonnées du centre du polytope estimé avec les coefficients de Fourier estimés à l'aide de la méthode des moindres carrés.
- l'étude de l'influence du choix de la borne d'erreur sur les performances de l'approche polytopique.

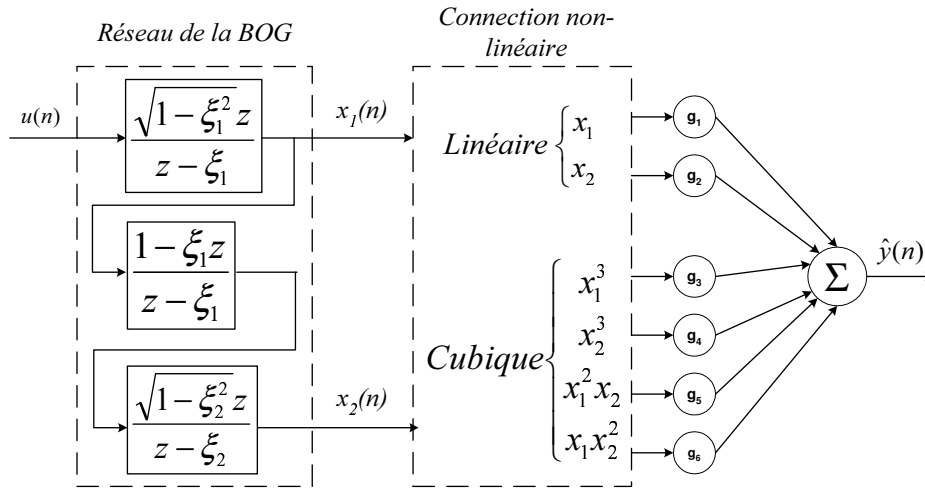


FIG. 5.46 – Le Modèle BOG-Volterra

Résultats de simulations

Le tableau 5.11, présente les valeurs des coefficients de Fourier du développement sur la BOG du modèle, estimés par deux méthodes différentes (La méthode de Moindres Carrés (M.C.) et la méthode polytopique à Erreur Bornée (E.B.)). Dans la première ligne de ce tableau, on présente les coefficients de Fourier estimés dans des conditions idéales (sans bruit) par la méthode des moindres carrés.

M.C. s/bruit	0.707	0.886	-0.127	-0.251	-0.160	-0.200
RSB = 10dB						
	\hat{g}_1	\hat{g}_2	\hat{g}_3	\hat{g}_4	\hat{g}_5	\hat{g}_6
M.C.	0.725	0.896	-0.132	-0.253	-0.162	-0.199
E.B.	0.674	0.854	-0.123	-0.248	-0.152	-0.205
RSB = 3dB						
	\hat{g}_1	\hat{g}_2	\hat{g}_3	\hat{g}_4	\hat{g}_5	\hat{g}_6
M.C.	0.723	0.928	-0.124	-0.273	-0.157	-0.199
E.B.	0.612	0.834	-0.105	-0.261	-0.135	-0.213

TAB. 5.11 – Coefficients de Fourier du modèle BOG-Volterra estimés par deux méthodes

Pour étudier l'influence de la borne de l'erreur sur la performance de l'approche polytopique, on utilise une même séquence d'entrée bruit blanc gaussien et une même séquence de bruit additif. La borne de l'erreur prend les valeurs Δ , 2Δ , 3Δ et 4Δ pour un $RSB = 10dB$, ($\Delta = 120\% \{e(k)\}$, $e(k)$ étant le bruit additif).

Dans ce qui suit on étudie l'influence de la borne de l'erreur pour un $RSB = 10dB$. Dans le tableau 5.12, on présente pour différentes bornes de l'erreur, les intervalles d'incertitude des coefficients de Fourier du développement sur la BOG du système simulé.

	Δg_1	Δg_2	Δg_3	Δg_4	Δg_5	Δg_6
Borne de l'erreur = Δ	0.733	0.886	0.151	0.217	0.116	0.093
Borne de l'erreur = 2Δ	2.228	2.573	0.349	0.518	0.234	0.198
Borne de l'erreur = 3Δ	3.191	2.982	0.741	1.212	0.711	0.581
Borne de l'erreur = 4Δ	3.792	3.531	1.342	1.918	0.981	0.854

TAB. 5.12 – Intervalles d'incertitude des coefficients de Fourier de la BOG pour différentes bornes de l'erreur

A chaque nouvelle mesure, en augmentant la borne de l'erreur, la direction des deux hyperplans H_{k_1} et H_{k_2} reste inchangée puisque la séquence d'entrée est la même. Par contre la distance entre ces deux derniers augmente. Ce qui engendre l'élargissement de la bande entre les deux hyperplans. D'où un volume important du polytope et une actualisation moins fréquente de celui-ci. Ceci implique des intervalles d'incertitude des paramètres plus grands. D'où, on peut conclure qu'une augmentation de la borne de l'erreur induit une baisse de la performance de l'algorithme polytopique.

5.6 Identification robuste de modèle SVD-PARAFAC-Volterra

Rappelons que la sortie d'un modèle SVD-PARAFAC-Volterra d'ordre 3 s'écrit comme suit :

$$\begin{aligned} \hat{y}(n) &= \sum_{n_1=1}^M h_1(n_1)u(n-n_1) + \sum_{n_1=1}^M \sum_{n_2=1}^M \left(\sum_{r=1}^R a_{n_1 r} b_{n_2 r} \right) u(n-n_1)u(n-n_2) \\ &+ \sum_{n_1=1}^M \sum_{n_2=1}^M \sum_{n_3=1}^M \left(\sum_{p=1}^P c_{n_1 p} d_{n_2 p} e_{n_3 p} \right) u(n-n_1)u(n-n_2)u(n-n_3) \end{aligned} \quad (5.83)$$

$$\begin{aligned} &= \sum_{n_1=1}^M h_1(n_1)u(n-n_1) \\ &+ \sum_{r=1}^R \left(\sum_{n_1=1}^M a_{n_1 r} u(n-n_1) \right) \left(\sum_{n_2=1}^M b_{n_2 r} u(n-n_2) \right) \\ &+ \sum_{p=1}^P \left(\sum_{n_1=1}^M c_{n_1 p} u(n-n_1) \right) \left(\sum_{n_2=1}^M d_{n_2 p} u(n-n_2) \right) \left(\sum_{n_3=1}^M e_{n_3 p} u(n-n_3) \right) \end{aligned} \quad (5.84)$$

et que cette sortie peut s'écrire de 3 manières différentes comme suit :

$$\hat{y}(n) = \Theta_1^T P_1(n) \quad (5.85)$$

$$= \Theta_2^T P_2(n) \quad (5.86)$$

$$= \Theta_3^T P_3(n) \quad (5.87)$$

Les vecteurs Θ_1 , Θ_2 et Θ_3 contiennent les composantes du modèle SVD-PARAFAC-Volterra d'ordre 3. Ils sont définis ainsi que les vecteurs $P_1(n)$, $P_2(n)$ et $P_3(n)$ dans les équations (4.35), (4.40) et (4.46) du chapitre précédent.

L'inéquation 5.9 devient alors :

$$y^{(s)}(n) - \delta_n \leq \theta_1^T P_1(n) \leq y^{(s)}(n) + \delta_n \quad (5.88)$$

$$y^{(s)}(n) - \hat{y}_L(n) - \delta_n \leq \theta_2^T P_2(n) \leq y^{(s)}(n) - \hat{y}_L(n) + \delta_n \quad (5.89)$$

$$y^{(s)}(n) - \hat{y}_L(n) - \hat{y}_Q(n) - \delta_n \leq \theta_3^T P_3(n) \leq y^{(s)}(n) - \hat{y}_L(n) - \hat{y}_Q(n) + \delta_n \quad (5.90)$$

où $\hat{y}_L(n)$ et $\hat{y}_Q(n)$ définies dans les équations (4.42) et (4.48), sont respectivement la sortie linéaire et la sortie quadratique du modèle SVD-PARAFAC-Volterra d'ordre 3.

En se basant sur les 3 inéquations précédentes et en utilisant l'approche polytopique à erreur bornée, on définit l'algorithme polytopique à Erreur Bornée Alterné (EBA) qui admet à chacune de ses itérations trois étapes. A chaque étape, on met à jour et d'une manière alternée les polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3 comme le montrent les figures 5.47.

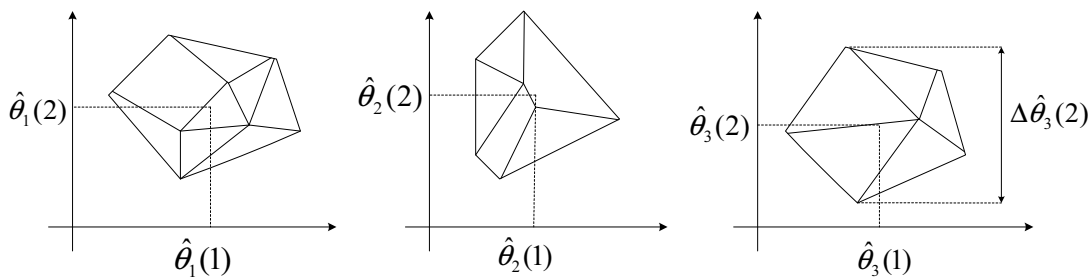


FIG. 5.47 – Mise à jour des polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3

A la convergence de l'algorithme EBA on trouve les estimées ainsi que les espaces d'incertitude des estimées des composantes du modèle SVD-PARAFAC-Volterra d'ordre 3, c'est à dire le noyau h_1 , les matrices A et B de la SVD du noyau quadratique et les matrices C , D et E de la décomposition PARAFAC du noyau cubique.

5.6.1 Choix de la borne d'erreur

On remarque dans les inéquations (5.88), (5.89) et (5.90) que les sorties utilisées pour la mise à jour des polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3 respectivement ne sont pas les mêmes. Les sorties linéaire $\widehat{y}_L(n)$ et quadratique $\widehat{y}_Q(n)$ du modèle SVD-PARAFAC-Volterra d'ordre 3 rajoutent chacune une erreur de modélisation supplémentaire. Pour prendre en compte cette erreur de modélisation supplémentaire, on rajoute à la borne d'erreur principale δ_n une incertitude δ_n^L relative à la sortie $\widehat{y}_L(n)$ et une incertitude δ_n^Q relative à la sortie $\widehat{y}_Q(n)$. C'est à dire, qu'à la $n^{\text{ème}}$ mesure, pour mettre à jour θ_2 il faut tenir compte de la borne d'erreur $\delta_n + \delta_n^L$ et pour mettre à jour θ_3 il faut tenir compte de la borne d'erreur $\delta_n + \delta_n^L + \delta_n^Q$.

On rappelle que $\widehat{y}_L(n)$ est la sortie linéaire du modèle SVD-PARAFAC-Volterra d'ordre 3 calculée en prenant comme noyau linéaire $\widehat{h}_1(n)$ les M premières coordonnées du vecteur $\theta_1(n)$ qui représentent les coordonnées du centre du polytope \mathcal{P}_1 à l'instant n .

Soit $\widehat{y}_L^{S_i^1}(n)$ la sortie linéaire du modèle SVD-PARAFAC-Volterra d'ordre 3 calculée en prenant comme noyau linéaire les M premières coordonnées du sommet S_i^1 du polytope \mathcal{P}_1 à l'instant n .

Pour la mise à jour du polytope \mathcal{P}_2 , la borne de l'erreur à l'instant n s'écrit alors $\delta_n + \delta_n^L$ avec :

$$\delta_n^L = \max_i \left| \widehat{y}_L(n) - \widehat{y}_L^{S_i^1}(n) \right| \quad (5.91)$$

On procède de la même manière pour la mise à jour du polytope \mathcal{P}_3 . Soient

- $\widehat{y}_Q(n)$ la sortie quadratique du modèle SVD-PARAFAC-Volterra d'ordre 3 calculée en prenant comme coefficients de la matrice \widehat{A} de la décomposition SVD les MR coordonnées comprises entre les composantes $M + 1$ et $M(1 + R)$ du centre du polytope \mathcal{P}_1 à l'instant n et en prenant comme coefficients de la matrice \widehat{B} les MR premières coordonnées du centre du polytope \mathcal{P}_2 à l'instant n .
- $\widehat{y}_Q^{S_i^1, S_j^2}(n)$ la sortie quadratique du modèle SVD-PARAFAC-Volterra d'ordre 3 calculée en prenant comme coefficients de la matrice \widehat{A} de la décomposition SVD les MR coordonnées comprises entre les composantes $M + 1$ et $M(1 + R)$ du sommet S_i^1 du polytope \mathcal{P}_1 à l'instant n et en prenant comme coefficients de la matrice \widehat{B} les MR premières coordonnées du sommet S_j^2 du polytope \mathcal{P}_2 à l'instant n .

La borne de l'erreur à l'instant n s'écrit alors $\delta_n + \delta_n^L + \delta_n^Q$ avec :

$$\delta_n^Q = \max_{i,j} \left| \widehat{y}_Q(n) - \widehat{y}_Q^{S_i^1, S_j^2}(n) \right| \quad (5.92)$$

5.6.2 Résultats de simulations

Le système simulé est un système de Volterra d'ordre 3, non homogène, discret et invariant dans le temps. Il est caractérisé par ses trois noyaux H_1 , H_2 et H_3 respectivement linéaire, quadratique et cubique.

$$\begin{aligned} H_1 &= \begin{pmatrix} 1 \\ -0,7 \end{pmatrix}; \quad H_2 = \begin{pmatrix} 1 & -1.1 \\ -0,6 & 0.66 \end{pmatrix}; \\ H_3(1, \dots) &= \begin{pmatrix} 1 & -0.1 \\ -0,9 & 0.09 \end{pmatrix} \quad \text{et} \quad H_3(2, \dots) = \begin{pmatrix} -0.3 & 0.03 \\ 0.27 & -0.027 \end{pmatrix} \end{aligned} \quad (5.93)$$

Le système étant d'ordre 3, le modèle SVD-PARAFAC-Volterra à identifier est d'ordre 3 et de mémoire $M = 2$ aussi. Le nombre de paramètres à estimer est alors de $M(1 + 2 + 3) = 12$.

Le signal d'entrée est une séquence de bruit blanc gaussien de longueur $L = 100$ de moyenne nulle et de variance unité. On utilise l'approche polytopique afin d'estimer les espaces d'incertitude des composantes du modèle SVD-PARAFAC-Volterra d'ordre 3.

A la $n^{\text{ème}}$ itération de l'algorithme :

- La première étape de l'identification alternée consiste à mettre à jour le polytope \mathcal{P}_1 de dimension égale à 6. La borne de l'erreur est égale à :

$$\Delta_1 = 120\% \max_{n=1, \dots, N} \{e_n\} \quad (5.94)$$

où $e(n)$ est le bruit additif à la sortie du système simulé à la $n^{\text{ème}}$ itération et N le nombre de mesures.

- La deuxième étape consiste à mettre à jour le polytope \mathcal{P}_2 de dimension égale à 4. La borne de l'erreur est égale à :

$$\Delta_2 = \Delta_1 + \max_{n=1, \dots, N} \{\delta_n^L\} \quad (5.95)$$

où δ_n^L est l'erreur de modélisation relative à la sortie $\widehat{y}_L(n)$ définie dans (5.91).

- La troisième étape consiste à mettre à jour le polytope \mathcal{P}_3 de dimension égale à 2. La borne de l'erreur est égale à :

$$\Delta_3 = \Delta_2 + \max_{n=1,\dots,N} \{\delta_n^Q\} \quad (5.96)$$

où δ_n^Q est l'erreur de modélisation relative à la sortie $\hat{y}_Q(n)$ définie dans (5.92).

Dans le tableau 5.13 on présente les intervalles d'incertitude des coefficients du noyau linéaire h_1 , des coefficients des matrices A , B relatives au noyau quadratique du modèle et de ceux des matrices C , D et E relatives au noyau cubique et les volumes des polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3 pour deux RSB différents.

	RSB = 10dB	RSB=3dB
$\Delta h_1(1)$	$1.65 \cdot 10^{-2}$	$7.66 \cdot 10^{-1}$
$\Delta h_1(2)$	$5.84 \cdot 10^{-2}$	$3.42 \cdot 10^{-1}$
$\Delta A(1)$	$5.79 \cdot 10^{-2}$	$9.83 \cdot 10^{-2}$
$\Delta A(2)$	$7.29 \cdot 10^{-2}$	$2.83 \cdot 10^{-1}$
$\Delta B(1)$	$2.54 \cdot 10^{-2}$	$7.36 \cdot 10^{-2}$
$\Delta B(2)$	$3.47 \cdot 10^{-1}$	$6.63 \cdot 10^{-1}$
$\Delta C(1)$	$2.39 \cdot 10^{-2}$	$8.57 \cdot 10^{-2}$
$\Delta C(2)$	$6.58 \cdot 10^{-2}$	$9.21 \cdot 10^{-1}$
$\Delta D(1)$	$2.23 \cdot 10^{-1}$	$4.45 \cdot 10^{-1}$
$\Delta D(2)$	$6.54 \cdot 10^{-2}$	$3.89 \cdot 10^{-1}$
$\Delta E(1)$	$3.49 \cdot 10^{-2}$	$8.51 \cdot 10^{-2}$
$\Delta E(2)$	$2.72 \cdot 10^{-2}$	$4.51 \cdot 10^{-1}$
Volume(\mathcal{P}_1)	$7.83 \cdot 10^{-8}$	$3.06 \cdot 10^{-6}$
Volume(\mathcal{P}_2)	$8.74 \cdot 10^{-5}$	$1.98 \cdot 10^{-4}$
Volume(\mathcal{P}_3)	$6.72 \cdot 10^{-3}$	$8.73 \cdot 10^{-2}$

TAB. 5.13 – Intervalles d'incertitude des coefficients du noyau linéaire h_1 et des matrices A , B , C , D et E et volume des polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3

Dans le tableau 5.13, on remarque que les intervalles d'incertitude des paramètres du modèle ainsi que les volumes des polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3 deviennent plus grands en augmentant le bruit. Ceci est dû à la relation de proportionnalité entre le bruit additif et la borne de l'erreur. Donc, plus le bruit est important, plus la borne de l'erreur est grande et donc moins bonne est la précision sur les paramètres du modèle SVD-PARAFAC-Volterra. Les paramètres du modèle sont estimés en tant que les coordonnées du centre du polytope. Ils sont présentés dans le tableau 5.14 avec l'Erreur Quadratique Moyenne

	RSB = 10dB	RSB=3dB
$\widehat{h}_1(1)$	3.830	3.536
$\widehat{h}_1(2)$	3.262	2.452
$\widehat{A}(1)$	-2.687	-2.452
$\widehat{A}(2)$	0.324	0.423
$\widehat{B}(1)$	2.691	2.202
$\widehat{B}(2)$	-1.295	-1.659
$\widehat{C}(1)$	5.839	5.633
$\widehat{C}(2)$	2.602	1.571
$\widehat{D}(1)$	-1.819	-1.971
$\widehat{D}(2)$	1.732	0.701
$\widehat{E}(1)$	-0.289	-0.248
$\widehat{E}(2)$	0.493	0.495
EQMN_{sortie}	$1.72 \cdot 10^{-1}$	$7.28 \cdot 10^{-1}$

TAB. 5.14 – Coefficients du noyau linéaire h_1 et des matrices A , B , C , D et E à l'aide de l'approche polytopique et $EQMN_{sortie}$

Normalisée $EQMN_{sortie}$ entre la sortie du système simulé et celle du modèle SVD-PARAFAC-Volterra d'ordre 3 dont les composantes sont estimées à l'aide de l'approche polytopique pour deux RSB différents.

5.7 Conclusions

Dans ce chapitre a été accomplie l'identification de modèles linéaires ainsi que des modèles non-linéaires en utilisant des techniques d'identification des intervalles d'incertitude des paramètres appelées techniques d'identification à erreur bornée.

Plusieurs techniques à E.B. ont été présentées : l'approche polytopique classique, l'approche orthotopique, et plusieurs approches polytopiques à complexité limitée.

Les conclusions tirées de la comparaison des différentes approches à E.B et qui sont valables pour les modèles linéaires aussi bien que pour les modèles non-linéaires sont les suivantes :

- l'approche polytopique donne toujours les meilleurs résultats du point de vue de la taille des polytopes obtenus (volume minimal et intervalles d'incertitude minimaux). Cette approche présente aussi l'avantage de converger rapidement. Le nombre de faces et de sommets du polytope augmente avec chaque nouvelle mesure

non redondante, ce qui engendre un polytope de plus en plus complexe. Ceci nous permet de déduire que cette méthode n'est pas pratique dans des procédures en temps réel par exemple.

- l'approche orthotopique présente l'avantage d'être simple et fournit une région ayant un nombre de directions et de faces fixes. On note avec cette approche, la lenteur de la convergence et un nombre important de mesures redondantes. La simplicité de cette approche est due essentiellement au nombre réduit et fixe de sommets de l'orthotope qui engendre une complexité constante à chaque itération de l'algorithme orthotopique. Donc le temps de calcul d'une mise à jour non redondante est, contrairement à l'approche polytopique, pratiquement constant. Les orthotopes obtenus présentent des volumes et des intervalles d'incertitude assez larges comparés au polytope.
- les approches polytopiques à complexité limitée sont un compromis entre l'approche polytopique classique et l'approche orthotopique. Les volumes des zones d'incertitude sont moins importants que ceux des orthotopes. La complexité numérique est moindre que celle de l'algorithme polytopique classique puisque le polytope à C.L admet un nombre fixe de sommets et de faces.

Les simulations faites dans ce chapitre, concernent en premier lieu l'identification de modèles linéaires de Laguerre et des modèles issus de la BOG. L'identification basée sur des approches à erreur bornée, consiste à estimer les intervalles paramétriques d'incertitude des coefficients de Fourier de ces modèles.

Les critères de performance étudiés sont : les intervalles d'incertitude des coefficients, le taux d'actualisation de l'algorithme à E.B., son temps de convergence, les coordonnées du centre de l'espace estimé et l' $EQMN_{sortie}$ entre la sortie du système simulé et celles des modèles identifiés.

A l'arrêt de l'algorithme à erreur bornée, les coordonnées du centre de l'espace paramétrique final sont considérées comme les coefficients de Fourier du modèle estimé. Ceux-ci, comparés aux coefficients de Fourier estimés par la méthode des moindres carrés montrent que les performances données par les deux techniques sont proches. La supériorité des performances de l'approche polytopique par rapport aux autres approches à erreur bornée est due au volume réduit du polytope final comparé aux volumes des

autres espaces estimés.

L' $EQMN_{sortie}$ entre la sortie du système simulé et celles des modèles identifiés à l'aide des techniques à E.B. converge vers des valeurs assez faibles ce qui montre qu'on reconstitue bien le signal de sortie du système même pour des valeurs assez faibles du RSB .

La deuxième partie des simulations faites dans ce chapitre concerne l'identification des modèles non-linéaires de Volterra. Plusieurs systèmes non linéaires ont été simulés. Ils ont été d'abord représentés par des modèles de Volterra à complexité réduite (VBFO et SVD-PARAFAC-Volterra). Ensuite, on a estimé les paramètres de ces modèles à savoir les coefficients de Fourier pour les modèles développés sur la BOG ou les coefficients des matrices des décompositions tensorielles pour le modèle SVD-PARAFAC-Volterra d'ordre 3.

Les modèles non linéaires estimés reproduisent bien la sortie du système. Ceci est confirmé par le critère de performance $EQMN_{sortie}$ qui converge vers des valeurs relativement petites. Les paramètres de ces modèles sont proches des vrais coefficients du système simulé dans le cas du premier et deuxième systèmes de Volterra). D'autre part dans le troisième système, les coefficients de Fourier estimés à l'aide de l'approche polytopique sont proches de ceux estimés par les moindres carrés estimés sans la présence de bruit.

L'influence de la borne de l'erreur sur la performance de l'identification a été traitée dans le cas du système 3 de Volterra. En augmentant la borne de l'erreur de Δ à 4Δ , ($\Delta = 120\% \max\{e(k)\}$), on augmente la distance entre les hyperplans H_{k_1} et H_{k_2} à chaque nouvelle mesure. D'où le volume du polytope final est plus grand et les intervalles d'incertitude du domaine identifié sont de plus en plus larges.

Un autre critère de performance est utilisé pour le premier et deuxième systèmes : l' $EQMN_{coef}$ entre les paramètres du système simulé (Coefficients du noyau de Volterra) et les coordonnées du centre du volume paramétrique d'incertitude identifié. Ce critère nous permet de conclure qu'on estime les coefficients du noyau de Volterra original avec des valeurs d'erreurs relativement faibles et dans des conditions de simulation assez bruitées ($RSB = 3dB$).

Une nouvelle méthode d'identification des intervalles d'incertitude a été présentée dans ce chapitre, c'est l'utilisation de l'approche polytopique d'une manière alternée

afin de mettre à jour les composantes du modèle SVD-PARAFAC-Volterra d'ordre 3. Il a été tenu compte de 3 bornes de l'erreur différentes :

- la première est relative au bruit additif $e(t)$.
- la deuxième est relative à l'erreur de modélisation de la partie linéaire du système
- et la troisième est relative à l'erreur de modélisation de la partie quadratique du système.

Cette méthode nous a permis d'identifier les intervalles d'incertitude des éléments des décompositions SVD et PARAFAC d'un modèle non-linéaire. La résolution du problème à l'aide des BOG aurait nécessité l'identification d'un polytope unique dans un espace de dimension $M + \frac{M(M+1)}{2} + \frac{M(M+1)(M+2)}{6}$, ce qui aurait été très lourd à cause du grand nombre de paramètres à estimer. Avec l'approche SVD-PARAFAC-Volterra et en choisissant $R = 1$ et $P = 1$, on a identifié et d'une manière alternée trois polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3 dans trois espaces différents de dimensions respectives $3M$, $2M$ et M . Ce qui a considérablement réduit la complexité numérique et le temps de calcul de l'algorithme d'identification.

Chapitre 6

Conclusions et Perspectives

Cette thèse a été consacrée à l'étude de nouvelles méthodes de modélisation et d'identification de systèmes non linéaires et plus particulièrement des modèles de type Volterra. Quoi qu'ils aient l'avantage d'être linéaires vis-à-vis de leurs paramètres, le majeur inconvénient de ces modèles reste leur complexité paramétrique beaucoup plus grande que celle des modèles linéaires, ce qui rend difficile leur identification. Pour pallier à cet inconvénient on s'est intéressé à de nouveaux modèles de Volterra dits à complexité réduite. De nouvelles techniques d'identification ont été explorées afin d'estimer les paramètres de ces modèles de Volterra de complexité réduite.

Modélisation

La première approche considérée pour la modélisation a consisté à utiliser des développements en série des différents noyaux sur des bases de fonctions orthogonales (Laguerre, Kautz et Bases Orthogonales Généralisées (BOG)). Ces bases de fonctions ont tout d'abord été étudiées en vue de leur utilisation pour la modélisation des systèmes linéaires. Elles ont été ensuite utilisées pour la représentation des noyaux d'un modèle de Volterra.

La détermination des pôles caractéristiques des bases de fonctions orthogonales a été étudiée dans la littérature [24], [62], [61]. De nouvelles méthodes itératives basées sur la discrétisation de l'espace des paramètres (pôles des bases de fonctions orthonormales) et sur un processus de filtrage ont été développées dans le deuxième chapitre.

Ces méthodes [33], [34] sont convergentes et présentent l'avantage de ne pas utiliser les réponses impulsionnelles des systèmes étudiés comme les méthodes précédemment citées. La détermination de l'ordre des développements des différents noyaux, des coefficients de ces développements appelés coefficients de Fourier du développement et de l'incertitude relative à ces coefficients a été aussi traitée.

Une fois les pôles optimaux des bases orthonormales choisis, ces dernières ont été comparées entre-elles en utilisant différents critères de performance afin de déduire quels types de bases utiliser pour la représentation d'un système donné. La base de Laguerre par exemple est plus adaptée aux systèmes à dynamique dominante réelle, celle de Kautz aux systèmes à dynamique dominante complexe et enfin les BOG sont utilisées dans le cas général puisqu'elles sont caractérisées par plusieurs pôles réels et complexes. L'utilisation des BOG dans le cas non linéaire a permis de réduire considérablement la complexité numérique des modèles de Volterra classiques.

Une représentation d'état associée à un développement sur une base de fonctions orthogonales généralisées a été développée puis utilisée pour la construction de prédicteurs de la sortie du système ainsi modélisé.

Dans un deuxième temps, on s'est intéressé aux décompositions tensorielles (PARAFAC, PARAFAC2, TUCKER1, TUCKER2, TUCKER3, PARATUCK). L'étude de chacune de ces techniques a été détaillée. Les différentes écritures et représentations des décompositions (scalaires, matricielles ou tensorielles) ont été détaillées et unifiées. Une étude comparative entre ces techniques de décomposition a été effectuée afin de dégager les avantages et les inconvénients de chacune d'elle. Afin d'estimer les paramètres de ces décompositions, un algorithme Alternating Least Squares (ALS) a été défini. Cet algorithme nous permet d'identifier les composantes des différents modèles (PARAFAC, TUCKER,...) dans le sens des moindres carrés et d'une manière alternée.

Dans la suite, on s'est intéressé à la décomposition PARAFAC dans le but de mettre au point un nouveau modèle de Volterra à complexité réduite. Ce nouveau modèle appelé PARAFAC-Volterra cubique a été défini [35] en décomposant le noyau cubique en trois matrices à l'aide de la décomposition tensorielle PARAFAC. L'algorithme ALS-PARAFAC nous a ensuite permis d'identifier les éléments de cette décomposition. Il a été montré (en utilisant comme critère de performance le Taux de Réduction de Complexité (TRC)), en comparant la représentation d'un système de Volterra à l'aide d'une BOG

et sa représentation à l'aide de la décomposition PARAFAC, que cette dernière est plus précise et réduit plus la complexité numérique de modèle [35].

Un autre nouveau modèle de Volterra à complexité réduite qui généralise PARAFAC-Volterra a été défini dans le quatrième chapitre. En considérant le noyau quadratique comme une matrice et les autres noyaux comme des tenseurs d'ordres supérieurs à deux, nous avons utilisé une décomposition en valeurs singulières (SVD) pour le noyau quadratique et la décomposition tensorielle PARAFAC pour les noyaux d'ordres supérieurs à deux afin de construire le modèle réduit de Volterra. Le nouveau modèle présenté est appelé SVD-PARAFAC-Volterra [31].

Identification

Une fois les deux nouveaux modèles de Volterra à complexité réduite (VBFO et SVD-PARAFAC-Volterra) mis au point, on s'est intéressé à leur identification.

Pour les modèles VBFO, leurs sorties s'expriment sous forme de régressions linéaires en fonction des coefficients de Fourier à estimer. Pour identifier ces modèles, on a utilisé la technique des moindres carrés classique. Celle-ci nous a permis d'estimer les coefficients de Fourier des développements des noyaux de Volterra sur les bases de fonctions étudiées (Laguerre, Kautz et BOG). Lors de la minimisation de la fonction de coût dans le sens des moindres carrés, on n'a pas de problèmes de minima locaux puisque celle-ci est convexe. Les coefficients de Fourier des différents modèles ont été comparés ainsi que les erreurs $EQMN_{sortie}$ entre la sortie du système et celles des modèles VBFO.

Pour le modèle SVD-PARAFAC-Volterra d'ordre N , la sortie s'écrit de N manières différentes sous forme de régressions linéaires. Les N vecteurs de paramètres à estimer représentent les éléments de la décomposition SVD et des différentes décompositions PARAFAC d'ordres $3, \dots, N$. Pour estimer ces composantes, un nouvel algorithme appelé ARLS (Alternating Recursive Least Squares algorithm) a été présenté [32]. Cet algorithme est essentiellement basé sur la technique RLS appliquée d'une manière alternée afin d'estimer les paramètres de tels modèles de Volterra. Un canal satellitaire simplifié a été modélisé en utilisant le nouveau modèle non linéaire SVD-PARAFAC-Volterra d'ordre 3, puis identifié à l'aide du nouvel algorithme ARLS.

Enfin, dans le cinquième chapitre, de nouvelles techniques d'identification robuste

appelées techniques d'identification à erreur bornée ont été présentées. Ces méthodes ont été comparées entre elles (Complexité de calcul, temps d'exécution, taux d'actualisation, précision,...), afin de dégager les avantages et les inconvénients de chacune d'elles.

Ces méthodes ont été ensuite utilisées lors des simulations afin d'identifier les modèles de Volterra à complexité réduite (VBFO et SVD-PARAFAC-Volterra) définis dans les chapitres précédents, travail qui vise à étendre au cas des systèmes non linéaires incertains des résultats obtenus récemment pour des systèmes linéaires incertains.

D'abord, dans le cas des modèles VBFO, ces méthodes nous ont permis d'estimer les intervalles d'incertitude des coefficients de Fourier du développement sur les différentes bases orthogonales étudiées. Ensuite, ces mêmes méthodes d'identification à erreur bornée ont été utilisées afin d'identifier les intervalles d'incertitude des éléments du modèle SVD-PARAFAC-Volterra d'ordre 3. L'utilisation de l'approche polytopique d'une manière alternée a permis d'obtenir des résultats d'identification satisfaisants. Les critères de performance pris en considération sont le volume final du polytope, la largeur des intervalles d'incertitude et l'Erreur Quadratique Moyenne Normalisée $EQMN_{sortie}$ entre la sortie du système simulé et celle du modèle SVD-PARAFAC-Volterra d'ordre 3.

La résolution du problème à l'aide des BOG aurait nécessité l'utilisation des techniques d'identification à E.B. classiques, donc l'estimation d'un polytope unique, ce qui aurait été très lourd à cause du grand nombre de paramètres à estimer. Avec l'approche SVD-PARAFAC-Volterra et en utilisant l'approche polytopique alternée, on a estimé successivement trois polytopes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3 dans trois espaces différents. Ce qui a considérablement réduit la complexité numérique et le temps de calcul de l'algorithme d'identification robuste.

Perspectives

Dans la continuité de nos recherches sur les VBFO et après le calcul de la représentation d'état associée à un développement sur une BOG et celui des prédicteurs de la sortie d'un système donné, on envisage de développer de nouvelles techniques de commande prédictive robuste, travail qui visera à étendre au cas des systèmes non linéaires incertains des résultats obtenus pour des systèmes linéaires incertains [14].

Ensuite, les techniques de décompositions tensorielles étudiées seront utilisées pour

l'identification aveugle de canaux de communication mobile linéaires et de canaux de communication non linéaires de type Wiener. Ces travaux ont déjà commencé et font l'objet de recherche au sein de l'équipe.

Enfin, la possibilité de mettre au point de nouvelles techniques de commande robuste est envisageable. Ces techniques pourront utiliser le modèle de Volterra à complexité réduite SVD-PARAFAC-Volterra mis au point ainsi que les techniques d'identification à Erreur Bornée Alternées (EBA).

Annexe A

Représentation dans l'espace d'état d'un système à l'aide des bases de Kautz

On utilise dans cet annexe la définition des fonctions de Kautz de Wahlberg [67] et de Lindskog [42]. Le réseau de fonctions de Kautz correspondant est représenté dans la figure A.1. Elle exprime le dénominateur ainsi qu'une partie du numérateur de ces fonctions comme une multiplication de polynômes d'ordre 2, dont les racines, imposées par le choix des paramètres b et c , sont complexes et conjuguées. Elle se présente ainsi :

$$\Psi_{2p}(z, b, c) = \frac{\sqrt{1-c^2}(z-b)}{z^2 + b(c-1)z - c} \left[\frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c} \right]^p$$
$$\Psi_{2p+1}(z, b, c) = \frac{\sqrt{(1-c^2)(1-b^2)}}{z^2 + b(c-1)z - c} \left[\frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c} \right]^p$$

avec $p = 1, 2, \dots$; $-1 < b < 1$ et $-1 < c < 1$.

Soit le vecteur d'état :

$$XX(k+1) = \begin{pmatrix} XX_0(k+1) \\ XX_1(k+1) \\ \vdots \\ XX_{2p}(k+1) \\ XX_{2p+1}(k+1) \end{pmatrix}$$

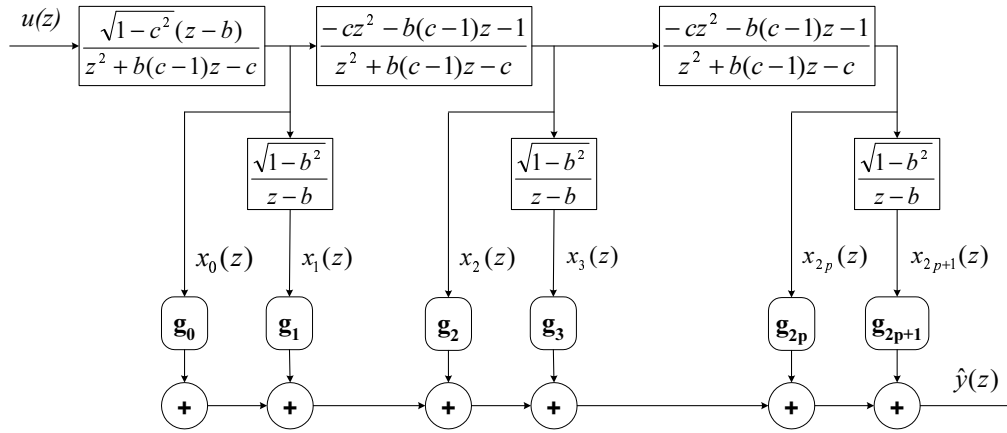


FIG. A.1 – Réseau de filtres de la Base des fonctions de Kautz

Remarque : Les éléments de ce vecteur d'indices pairs sont de dimension 2×1 et ceux d'indices impairs de dimension 1×1 .

L'équation d'état s'écrit comme suit :

$$\begin{cases} XX(k+1) = AX(k) + Bu(k) \\ \hat{y}(k) = CXX(k) \end{cases}$$

A partir de la figure A.1, on peut déterminer les relations suivantes :

1/ Equation d'état relative au premier bloc

$$\frac{x_0(z)}{u(z)} = \frac{\sqrt{1-c^2}(z-b)}{z^2+b(c-1)z-c}$$

$$\Rightarrow \begin{cases} XX_0(k+1) = \begin{pmatrix} 0 & c \\ 1 & -b(c-1) \end{pmatrix} XX_0(k) + \sqrt{1-c^2} \begin{pmatrix} -b \\ 1 \end{pmatrix} u(k) \\ x_0(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k) \end{cases}$$

Le passage précédent d'une fonction de transfert à une équation d'état est démontrée dans l'annexe B.

2/ Equation d'état relative au deuxième bloc

$$\frac{x_1(z)}{x_0(z)} = \frac{\sqrt{1-b^2}}{z-b}$$

$$XX_1(k+1) = bXX_1(k) + \sqrt{1-b^2}x_0(k)$$

$$\Rightarrow \begin{cases} XX_1(k+1) = bXX_1(k) + \sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k) \\ x_1(k) = XX_1(k) \end{cases}$$

3/ Equation d'état relative au troisième bloc

$$\frac{x_2(z)}{x_0(z)} = \frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c}$$

$$\begin{aligned} XX_2(k+1) &= \begin{pmatrix} 0 & c \\ 1 & -b(c-1) \end{pmatrix} XX_2(k) + (1-c^2) \begin{pmatrix} 1 \\ -b \end{pmatrix} x_0(k) \\ &= \begin{pmatrix} 0 & c \\ 1 & -b(c-1) \end{pmatrix} XX_2(k) \\ &+ (1-c^2) \begin{pmatrix} 1 \\ -b \end{pmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k) \end{aligned}$$

et

$$x_2(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} XX_2(k) - cx_0(k)$$

$$\Rightarrow \begin{cases} XX_2(k+1) = \begin{pmatrix} 0 & c \\ 1 & -b(c-1) \end{pmatrix} XX_2(k) + (1-c^2) \begin{pmatrix} 0 & 1 \\ 0 & -b \end{pmatrix} XX_0(k) \\ x_2(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} XX_2(k) - c \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k) \end{cases}$$

4/ Equation d'état relative au quatrième bloc

$$\frac{x_3(z)}{x_2(z)} = \frac{\sqrt{1-b^2}}{z-b}$$

$$XX_3(k+1) = b.XX_3(k) + \sqrt{1-b^2}x_2(k)$$

$$\Rightarrow \begin{cases} XX_3(k+1) = b.XX_3(k) + (-c)^0\sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_2(k) \\ \quad \quad \quad + -c\sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k) \\ x_3(k) = XX_3(k) \end{cases}$$

5/ Equation d'état relative au cinquième bloc

$$\begin{aligned} XX_4(k+1) &= \begin{pmatrix} 0 & c \\ 1 & -b(c-1) \end{pmatrix} XX_4(k) + (1-c^2) \begin{pmatrix} 1 \\ -b \end{pmatrix} x_2(k) \\ &= \begin{pmatrix} 0 & c \\ 1 & -b(c-1) \end{pmatrix} XX_4(k) \\ &+ (1-c^2) \begin{pmatrix} 0 & 1 \\ 0 & -b \end{pmatrix} XX_2(k) \\ &+ (-c)^1(1-c^2) \begin{pmatrix} 0 & 1 \\ 0 & -b \end{pmatrix} XX_0(k) \end{aligned}$$

$$\begin{aligned} x_4(k) &= \begin{bmatrix} 0 & 1 \end{bmatrix} XX_4(k) - cx_2(k) \\ &= (-c)^0 \begin{bmatrix} 0 & 1 \end{bmatrix} XX_4(k) \\ &+ (-c)^1 \begin{bmatrix} 0 & 1 \end{bmatrix} XX_2(k) \\ &+ (-c)^2 \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k) \end{aligned}$$

6/ Equation d'état relative au sixième bloc

$$\begin{aligned} XX_5(k+1) &= b.XX_5(k) + \sqrt{1-b^2}x_4(k) \\ &= b.XX_5(k) \\ &+ (-c)^0\sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_4(k) \\ &+ (-c)^1\sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_2(k) \\ &+ (-c)^2\sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k) \end{aligned}$$

$$x_5(k) = XX_5(k)$$

d'où, dans le cas général :

Equation d'état relative au $(2p + 1)^{\text{ème}}$ bloc

$$\begin{aligned}
 XX_{2p}(k+1) &= \begin{pmatrix} 0 & c \\ 1 & -b(c-1) \end{pmatrix} XX_{2p}(k) \\
 &+ (1-c^2) \begin{pmatrix} 1 \\ -b \end{pmatrix} x_{2p-2}(k) \\
 &= \begin{pmatrix} 0 & c \\ 1 & -b(c-1) \end{pmatrix} XX_{2p}(k) \\
 &+ (-c)^0(1-c^2) \begin{pmatrix} 0 & 1 \\ 0 & -b \end{pmatrix} XX_{2p-2}(k) \\
 &+ \dots\dots \\
 &+ (-c)^{p-2}(1-c^2) \begin{pmatrix} 0 & 1 \\ 0 & -b \end{pmatrix} XX_0(k)
 \end{aligned}$$

$$\begin{aligned}
 x_{2p}(k) &= (-c)^0 \begin{bmatrix} 0 & 1 \end{bmatrix} XX_{2p}(k) \\
 &+ (-c)^1 \begin{bmatrix} 0 & 1 \end{bmatrix} XX_{2p-2}(k) \\
 &+ \dots\dots \\
 &+ (-c)^{p-1} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k)
 \end{aligned}$$

Equation d'état relative au $(2p + 2)^{\text{ème}}$ bloc

$$\begin{aligned}
 XX_{2p+1}(k+1) &= b.XX_{2p+1}(k) + \\
 &+ (-c)^0\sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_{2p}(k) \\
 &+ \dots\dots \\
 &+ (-c)^{p-1}\sqrt{1-b^2} \begin{bmatrix} 0 & 1 \end{bmatrix} XX_0(k)
 \end{aligned}$$

$$x_{2p+1}(k) = XX_{2p+1}(k)$$

Annexe B

Passage d'une fonction de transfert à une équation d'état

Considérons la fonction de transfert discrète :

$$\frac{Y(z)}{U(z)} = \frac{\sum_{i=0}^n b_i z^{-i}}{\sum_{i=0}^n a_i z^{-i}} = \frac{B(z)}{A(z)} \quad ; \quad a_0 = 1$$

Dans ce qui suit, on détaille le cas de la forme compagne observable.

$$Y(z) = \sum_{i=1}^n z^{-i} [b_i U(z) - a_i Y(z)] + b_0 U(z)$$

ou de façon développée :

$$\begin{aligned} Y(z) = & z^{-1} \{ [b_1 U(z) - a_1 Y(z)] + z^{-1} \{ [b_2 U(z) - a_2 Y(z)] + \dots \\ & + z^{-1} \{ [b_{n-1} U(z) - a_{n-1} Y(z)] + z^{-1} \{ [b_n U(z) - a_n Y(z)] \} \dots \} \} \\ & + b_0 U(z) \end{aligned}$$

Le vecteur d'état est alors défini à l'aide des équations suivantes :

$$\left\{ \begin{array}{l} X_1(z) = z^{-1} [b_n U(z) - a_n Y(z)] \\ X_2(z) = z^{-1} \{ [b_{n-1} U(z) - a_{n-1} Y(z)] + X_1(z) \} \\ \vdots \\ X_{n-1}(z) = z^{-1} \{ [b_2 U(z) - a_2 Y(z)] + X_{n-2}(z) \} \\ X_n(z) = z^{-1} \{ [b_1 U(z) - a_1 Y(z)] + X_{n-1}(z) \} \end{array} \right.$$

Ce qui correspond dans le domaine temporel, aux équations récurrentes suivantes :

$$\begin{cases} x_1(k+1) &= b_n u(k) - a_n y(k) \\ x_2(k+1) &= x_1(k) + b_{n-1} u(k) - a_{n-1} y(k) \\ &\vdots \\ x_{n-1}(k+1) &= x_{n-2}(k) + b_2 u(k) - a_2 y(k) \\ x_n(k+1) &= x_{n-1}(k) + b_1 u(k) - a_1 y(k) \end{cases}$$

avec :

$$y(k) = x_n(k) + b_0 u(k)$$

En posant :

$$M = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \\ 0 & \cdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} b_n \\ b_{n-1} \\ \vdots \\ \vdots \\ b_1 \end{pmatrix}, \quad a = \begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ \vdots \\ a_1 \end{pmatrix} \text{ et}$$

$$C = [0 \quad 0 \quad \cdots \quad 0 \quad 1]$$

et en remplaçant $y(k)$ par son expression dans l'équation d'évolution d'état, on obtient un modèle d'état suivant :

$$\begin{cases} x(k+1) = (M - aC)x(k) + (B - ab_0)u(k) \\ y(k) = Cx(k) + b_0 u(k) \end{cases}$$

Ces équations constituent la forme compagne observable des équations d'état discrètes.

Annexe C

Représentation d'état d'un système à l'aide des BOG

La nature des pôles (réels ou complexes) d'une BOG ainsi que leur ordre de succession sont importants. C'est pour cela que nous allons étudier les différents cas de succession des pôles séparément pour pouvoir à la fin conclure dans le cas général.

C.1 Cas d'un pôle réel précédé par un pôle réel (Figure C.1)

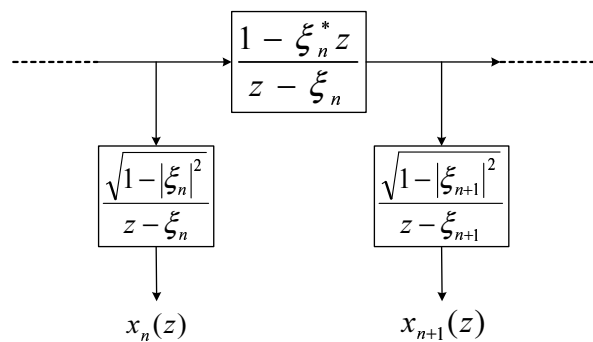


FIG. C.1 – ξ_n réel et ξ_{n+1} réel

La relation entre $x_n(z)$ et $x_{n+1}(z)$ s'écrit comme suit :

$$x_{n+1}(z) = \frac{\sqrt{1 - |\xi_{n+1}|^2}}{\sqrt{1 - |\xi_n|^2}} \frac{1 - \xi_n^* z}{z - \xi_{n+1}} x_n(z) \quad (\text{C.1})$$

$$= \frac{\sqrt{1 - |\xi_{n+1}|^2}}{\sqrt{1 - |\xi_n|^2}} \frac{-\xi_n^* + z^{-1}}{1 - \xi_{n+1} z^{-1}} x_n(z) \quad (\text{C.2})$$

D'où la mise en équation d'état :

$$\begin{cases} xx_{n+1}(k+1) = \xi_{n+1} xx_{n+1}(k) + \frac{\sqrt{1 - |\xi_{n+1}|^2}}{\sqrt{1 - |\xi_n|^2}} (1 - \xi_n^* \xi_{n+1}) x_n(k) \\ x_{n+1}(k) = xx_{n+1}(k) - \xi_n^* \frac{\sqrt{1 - |\xi_{n+1}|^2}}{\sqrt{1 - |\xi_n|^2}} x_n(k) \end{cases} \quad (\text{C.3})$$

C.2 Cas de deux pôles complexes conjugués précédés par un pôle réel. (Figure C.2)

A partir de la définition des fonctions de la BOG donnés par les équations (2.23) et (2.24), on déduit la figure C.2.

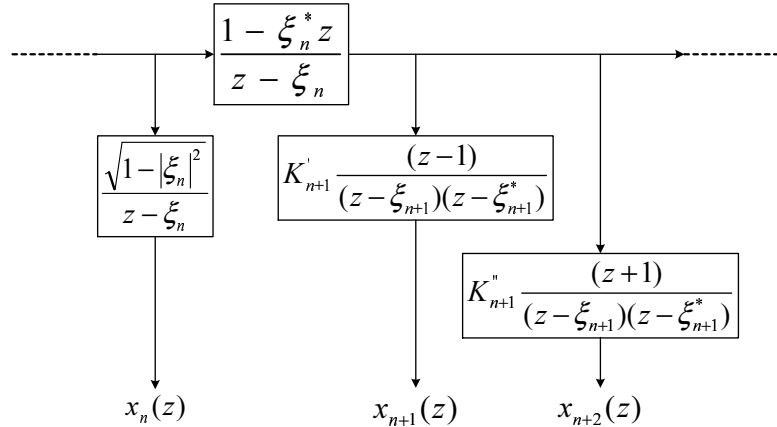


FIG. C.2 - ξ_n réel, ξ_{n+1} et ξ_{n+2} complexes

La relation entre $x_n(z)$ et $x_{n+1}(z)$ s'écrit comme suit :

$$x_{n+1}(z) = \frac{K'_{n+1}}{\sqrt{1 - |\xi_n|^2}} \frac{(z-1)(1 - \xi_n^* z)}{(z - \xi_{n+1})(z - \xi_{n+1}^*)} x_n(z) \quad (\text{C.4})$$

$$= \frac{K'_{n+1}}{\sqrt{1 - |\xi_n|^2}} \frac{-\xi_n^* + (1 + \xi_n^*)z^{-1} - z^{-2}}{1 - 2\text{Re}(\xi_{n+1})z^{-1} + |\xi_{n+1}|^2 z^{-2}} x_n(z) \quad (\text{C.5})$$

La mise en équation d'état s'écrit alors comme suit (voir démonstration dans l'annexe B) :

$$\Rightarrow \begin{cases} xx_{n+1}(k+1) = \begin{pmatrix} 0 & -|\xi_{n+1}|^2 \\ 1 & 2\text{Re}(\xi_{n+1}) \end{pmatrix} xx_{n+1}(k) \\ \quad + \frac{K'_{n+1}}{\sqrt{1-|\xi_n|^2}} \begin{pmatrix} -1 + \xi_n^* |\xi_{n+1}|^2 \\ 1 + \xi_n^* - 2\xi_n^* \text{Re}(\xi_{n+1}) \end{pmatrix} x_n(k) \\ \\ xx_{n+1}(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} xx_{n+1}(k) - K'_{n+1} \frac{\xi_n^*}{\sqrt{1-|\xi_n|^2}} x_n(k) \end{cases} \quad (\text{C.6})$$

De même, la relation entre $x_n(z)$ et $x_{n+2}(z)$ s'écrit come suit :

$$x_{n+2}(z) = \frac{K''_{n+1}}{\sqrt{1-|\xi_n|^2}} \frac{-\xi_n^* + (1 - \xi_n^*)z^{-1} + z^{-2}}{1 - 2\text{Re}(\xi_{n+1})z^{-1} + |\xi_{n+1}|^2 z^{-2}} x_n(z) \quad (\text{C.7})$$

d'où

$$\begin{cases} xx_{n+2}(k+1) = \begin{pmatrix} 0 & -|\xi_{n+1}|^2 \\ 1 & 2\text{Re}(\xi_{n+1}) \end{pmatrix} xx_{n+2}(k) \\ \quad + \frac{K''_{n+1}}{\sqrt{1-|\xi_n|^2}} \begin{pmatrix} 1 + \xi_n^* |\xi_{n+1}|^2 \\ 1 - \xi_n^* - 2\xi_n^* \text{Re}(\xi_{n+1}) \end{pmatrix} x_n(k) \\ \\ xx_{n+2}(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} xx_{n+2}(k) - K''_{n+1} \frac{\xi_n^*}{\sqrt{1-|\xi_n|^2}} x_n(k) \end{cases} \quad (\text{C.8})$$

C.3 Cas d'un pôle réel précédé d'un pôle complexe (Figure C.3)

La relation entre $x_{n+1}(z)$ et $x_n(z)$ s'écrit comme suit :

$$x_{n+1}(z) = \frac{\sqrt{1-|\xi_{n+1}|^2}}{K''_n} \frac{(1 - \xi_n z)(1 - \xi_n^* z)}{(z+1)(z - \xi_{n+1})} x_n(z) \quad (\text{C.9})$$

$$= \frac{\sqrt{1-|\xi_{n+1}|^2}}{K''_n} \frac{1 - 2\text{Re}(\xi_n)z^{-1} + |\xi_n|^2 z^{-2}}{1 + (1 - \xi_{n+1})z^{-1} - \xi_{n+1}z^{-2}} x_n(z) \quad (\text{C.10})$$

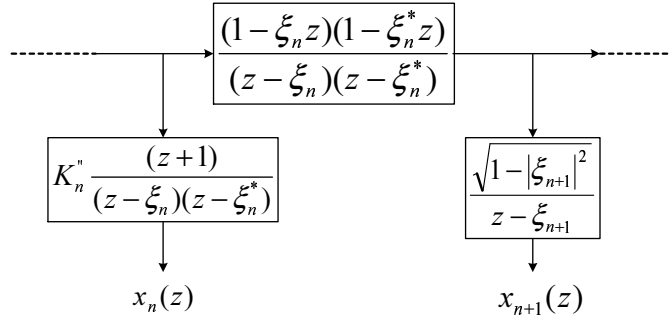


FIG. C.3 – ξ_n complexe et ξ_{n+1} réel

D'où la mise en équation d'état :

$$\begin{cases} xx_{n+1}(k+1) = \begin{pmatrix} 0 & \xi_{n+1} \\ 1 & \xi_{n+1} - 1 \end{pmatrix} xx_{n+1}(k) \\ \quad + \frac{\sqrt{1 - |\xi_{n+1}|^2}}{K_n''} \begin{pmatrix} |\xi_n|^2 + \xi_{n+1} \\ -2\text{Re}(\xi_n) + \xi_{n+1} - 1 \end{pmatrix} x_n(k) \\ \\ x_{n+1}(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} xx_{n+1}(k) + \frac{\sqrt{1 - |\xi_{n+1}|^2}}{K_n''} x_n(k) \end{cases} \quad (\text{C.11})$$

C.4 Cas d'un pôle complexe précédé par un pôle complexe Figure C.4)

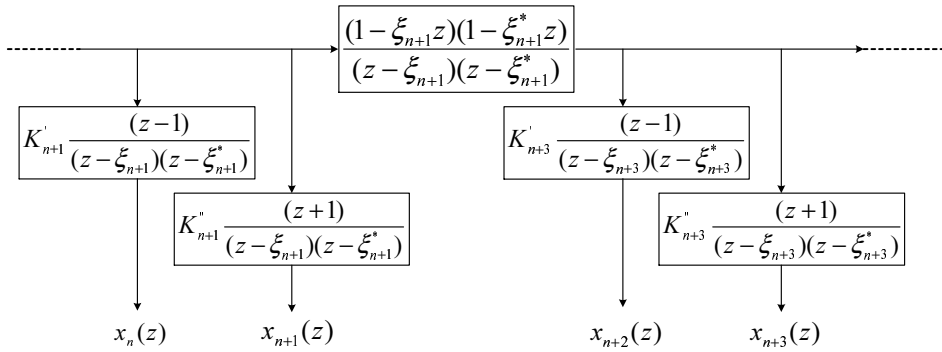


FIG. C.4 – ξ_n complexe et ξ_{n+1} complexe

La relation entre $x_{n+1}(z)$ et $x_{n+3}(z)$ s'écrit comme suit :

$$x_{n+3}(z) = \frac{K''_{n+3} (1 - \xi_{n+1}z)(1 - \xi_{n+1}^*z)}{K''_{n+1} (z - \xi_{n+3})(z - \xi_{n+3}^*)} x_{n+1}(z) \quad (\text{C.12})$$

$$= \frac{K''_{n+3} (|\xi_{n+1}|^2 - 2\text{Re}(\xi_{n+1})z^{-1} + z^{-2})}{K''_{n+1} (1 - 2\text{Re}(\xi_{n+3})z^{-1} + |\xi_{n+3}|^2 z^{-2})} x_{n+1}(z) \quad (\text{C.13})$$

D'où la mise en équation d'état :

$$\left\{ \begin{array}{l} xx_{n+3}(k+1) = \begin{pmatrix} 0 & -|\xi_{n+3}|^2 \\ 1 & 2\text{Re}(\xi_{n+3}) \end{pmatrix} xx_{n+3}(k) \\ + \frac{K''_{n+3}}{K''_{n+1}} \begin{pmatrix} 1 - |\xi_{n+1}|^2 |\xi_{n+3}|^2 \\ -2\text{Re}(\xi_{n+1}) + 2\text{Re}(\xi_{n+3}) |\xi_{n+1}|^2 \end{pmatrix} x_{n+1}(k) \\ \\ xx_{n+3}(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} xx_{n+3}(k) + \frac{K''_{n+3}}{K''_{n+1}} |\xi_{n+1}|^2 x_{n+1}(k) \end{array} \right. \quad (\text{C.14})$$

D'autre part la relation entre $x_n(z)$ et $x_{n+2}(z)$ s'écrit comme suit :

$$x_{n+2}(z) = \frac{K'_{n+3} (1 - \xi_{n+1}z)(1 - \xi_{n+1}^*z)}{K'_{n+1} (z - \xi_{n+3})(z - \xi_{n+3}^*)} x_n(z) \quad (\text{C.15})$$

$$= \frac{K'_{n+3} (|\xi_{n+1}|^2 - 2\text{Re}(\xi_{n+1})z^{-1} + z^{-2})}{K'_{n+1} (1 - 2\text{Re}(\xi_{n+3})z^{-1} + |\xi_{n+3}|^2 z^{-2})} x_n(z) \quad (\text{C.16})$$

D'où la mise en équation d'état :

$$\left\{ \begin{array}{l} xx_{n+2}(k+1) = \begin{pmatrix} 0 & -|\xi_{n+3}|^2 \\ 1 & 2\text{Re}(\xi_{n+3}) \end{pmatrix} xx_{n+2}(k) \\ + \frac{K'_{n+3}}{K'_{n+1}} \begin{pmatrix} 1 - |\xi_{n+1}|^2 |\xi_{n+3}|^2 \\ -2\text{Re}(\xi_{n+1}) + 2\text{Re}(\xi_{n+3}) |\xi_{n+1}|^2 \end{pmatrix} x_n(k) \\ \\ xx_{n+2}(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} xx_{n+2}(k) + \frac{K'_{n+3}}{K'_{n+1}} |\xi_{n+1}|^2 x_n(k) \end{array} \right. \quad (\text{C.17})$$

Annexe D

Deuxième représentation du modèle PARAFAC2

Montrons qu'une deuxième représentation matricielle du modèle PARAFAC2 s'écrit :

$$\mathcal{Y}_{N_2} = (C^T | \otimes | C^T)^T \mathbf{diag}(\mathbf{vec} \Phi) (B \otimes B)^T \quad (\text{D.1})$$

Théorème 1 Soient U , X et V trois matrices, on a :

$$\mathbf{vec}(UXV) = (V^T \otimes U) \mathbf{vec}(X) \quad (\text{D.2})$$

Démonstration du théorème 1

Supposant que V est de dimension $m \times N$, la $k^{\text{ème}}$ colonne de la matrice UXV s'écrit comme suit :

$$(UXV)_{.k} = UXV_{.k} = U \sum_{i=1}^m X_i v_{ik} \quad (\text{D.3})$$

$$= \left[\begin{array}{cccc} v_{1k}U & v_{2k}U & \cdots & v_{mk}U \end{array} \right] \underbrace{\begin{bmatrix} X_{.1} \\ X_{.2} \\ \vdots \\ X_{.m} \end{bmatrix}}_{\mathbf{vec}(X)} \quad (\text{D.4})$$

$$= \underbrace{\left[\begin{array}{cccc} v_{1k} & v_{2k} & \cdots & v_{mk} \end{array} \right]}_{(V_{.k})^T} \otimes U \mathbf{vec}(X) \quad (\text{D.5})$$

En assemblant les colonnes de la matrice (UXV) on trouve :

$$\mathbf{vec}(UXV) = \begin{bmatrix} (UXV)_{.1} \\ (UXV)_{.2} \\ \vdots \\ (UXV)_{.m} \end{bmatrix} \quad (\text{D.6})$$

$$= \begin{bmatrix} (V_{.1})^T \otimes U \\ (V_{.2})^T \otimes U \\ \vdots \\ (V_{.N})^T \otimes U \end{bmatrix} \mathbf{vec}(X) \quad (\text{D.7})$$

$$= (V^T \otimes U) \mathbf{vec}(X) \quad \square \quad (\text{D.8})$$

En utilisant les équations (3.38) et (3.41), on peut écrire :

$$(\mathbf{vec}(Y_{..n_3}))^T = (\mathbf{vec}(H_{..n_3}^T H_{..n_3}))^T \quad (\text{D.9})$$

$$= (\mathbf{vec}(BD_{n_3}^C \Phi D_{n_3}^C B^T))^T \quad (\text{D.10})$$

D'après le théorème 1 on peut écrire :

$$(\mathbf{vec}(Y_{..n_3}))^T = ((B \otimes B) \mathbf{vec}(D_{n_3}^C \Phi D_{n_3}^C))^T \quad (\text{D.11})$$

$$= ((B \otimes B)(D_{n_3}^C \otimes D_{n_3}^C) \mathbf{vec}(\Phi))^T \quad (\text{D.12})$$

$$= (\mathbf{vec}(\Phi))^T (D_{n_3}^C \otimes D_{n_3}^C)(B \otimes B)^T \quad (\text{D.13})$$

$$= (D_{n_3}^C \otimes D_{n_3}^C) \mathbf{vec}(\Phi)^T (B \otimes B)^T \quad (\text{D.14})$$

$$= (C_{n_3.} \otimes C_{n_3.}) \mathbf{diag}(\mathbf{vec}(\Phi))(B \otimes B)^T \quad (\text{D.15})$$

où $C_{n_3.}$ représente la $n_3^{\text{ème}}$ ligne de la matrice C et $D_{n_3}^C = \text{diag}(C_{n_3.})$

En assemblant les lignes de la matrice \mathcal{Y}_{N_2} on trouve :

$$\mathcal{Y}_{N_2} = \begin{bmatrix} (\mathbf{vec} Y_{..1})^T \\ (\mathbf{vec} Y_{..2})^T \\ \vdots \\ (\mathbf{vec} Y_{..N_3})^T \end{bmatrix} \quad (\text{D.16})$$

$$= \begin{bmatrix} (C_{1.} \otimes C_{1.}) \mathbf{diag}(\mathbf{vec}(\Phi))(B \otimes B)^T \\ (C_{2.} \otimes C_{2.}) \mathbf{diag}(\mathbf{vec}(\Phi))(B \otimes B)^T \\ \vdots \\ (C_{N_3.} \otimes C_{N_3.}) \mathbf{diag}(\mathbf{vec}(\Phi))(B \otimes B)^T \end{bmatrix} \quad (\text{D.17})$$

$$= (C^T | \otimes | C^T)^T \mathbf{diag}(\mathbf{vec}(\Phi))(B \otimes B)^T \quad \blacksquare \quad (\text{D.18})$$

Annexe E

Trois formes de l'algorithme

ALS-PARAFAC

Au cours de la minimisation des différentes fonctions de coût, on aura besoin de la définition de la dérivée par rapport à une matrice de la trace d'un produit de matrices.

E.1 Définition :

Soient Q , R , S et X quatre matrices. On définit $\frac{\partial(\cdot)}{\partial X}$ comme la dérivée partielle d'un scalaire (*d'une trace de matrice dans ce cas*) par rapport à une matrice X quelconque, alors :

$$\frac{\partial}{\partial X}(\text{tr}(Q^T X R^T)) = \frac{\partial}{\partial X}(\text{tr}(R X^T Q)) = QR \quad (\text{E.1})$$

et

$$\frac{\partial}{\partial X}(\text{tr}(Q X R X^T S)) = Q^T S^T X R^T + S Q X R \quad (\text{E.2})$$

E.2 ALS-PARAFAC 1

On rappelle que dans la première représentation du modèle PARAFAC, les matrices $(H_{n_1.})$, $(H_{.n_2.})$ et $(H_{..n_3})$ s'écrivent :

$$H_{n_1.} = B D_{n_1}^A C^T \quad (\text{E.3})$$

$$H_{.n_2.} = C D_{n_2}^B A^T \quad (\text{E.4})$$

$$H_{..n_3} = A D_{n_3}^C B^T \quad (\text{E.5})$$

avec

$$D_{n_1}^A = \mathbf{diag} (A_{n_1.}) \quad (\text{E.6})$$

$$D_{n_2}^B = \mathbf{diag} (B_{n_2.}) \quad (\text{E.7})$$

$$D_{n_3}^C = \mathbf{diag} (C_{n_3.}) \quad (\text{E.8})$$

L'algorithme ALS-PARAFAC 1 met à jour les matrices A , B et C en minimisant d'une manière alternée les fonctions de coût $\eta_1^{N_1}$, $\eta_1^{N_2}$ et $\eta_1^{N_3}$ suivantes.

$$\eta_1^{N_1} = \sum_{n_1=1}^{N_1} \|H_{n_1..} - BD_{n_1}^A C^T\|_F^2 \quad (\text{E.9})$$

$$\eta_1^{N_2} = \sum_{n_2=1}^{N_2} \|H_{..n_2} - CD_{n_2}^B A^T\|_F^2 \quad (\text{E.10})$$

$$\eta_1^{N_3} = \sum_{n_3=1}^{N_3} \|H_{..n_3} - AD_{n_3}^C B^T\|_F^2 \quad (\text{E.11})$$

On détaillera la minimisation de $\eta_1^{N_3}$ afin de mettre à jour la matrice C . La minimisation des deux autres fonctions se fera de la même manière.

On écrit d'abord la fonction de coût $\eta_1^{N_3}$ en fonction de $D_{n_3}^C$ avec A et B fixés.

$$\eta_1^{N_3} = \sum_{n_3=1}^{N_3} \|H_{..n_3} - AD_{n_3}^C B^T\|_F^2 \quad (\text{E.12})$$

$$= \sum_{n_3=1}^{N_3} \text{tr} ((H_{..n_3} - AD_{n_3}^C B^T)^T (H_{..n_3} - AD_{n_3}^C B^T)) \quad (\text{E.13})$$

$$\begin{aligned} &= \sum_{n_3=1}^{N_3} \text{tr} (H_{..n_3}^T H_{..n_3}) - 2 \sum_{n_3=1}^{N_3} \text{tr} (H_{..n_3}^T AD_{n_3}^C B^T) \\ &+ \sum_{n_3=1}^{N_3} \text{tr} (BD_{n_3}^C A^T AD_{n_3}^C B^T) \end{aligned} \quad (\text{E.14})$$

On calcule le gradient de $\eta_1^{N_3}$ par rapport à $D_{n_3}^C$.

$$\begin{aligned} \nabla_{D_{n_3}^C} (\eta_1^{N_3}) &= -2 \sum_{n_3=1}^{N_3} \frac{\partial}{\partial D_{n_3}^C} \text{tr} (H_{..n_3}^T AD_{n_3}^C B^T) \\ &+ \sum_{n_3=1}^{N_3} \frac{\partial}{\partial D_{n_3}^C} \text{tr} (BD_{n_3}^C A^T AD_{n_3}^C B^T) \end{aligned} \quad (\text{E.15})$$

$$= -2B^T H_{..n_3}^T A + 2B^T BD_{n_3}^C A^T A \quad (\text{E.16})$$

$$= -2 (B^T H_{..n_3}^T A - B^T BD_{n_3}^C A^T A) \quad (\text{E.17})$$

L'annulation de $\nabla_{D_{n_3}^C} (\eta_1^{N_3})$ donne :

$$D_{n_3}^C = (A^T A)^{-1} A^T (H_{..n_3}) B (B^T B)^{-1}; n_3 = 1, \dots, N_3 \quad (\text{E.18})$$

La diagonale de $D_{n_3}^C$ n'est autre que $(C_{n_3.})$ qui est la $n_3^{\text{ème}}$ ligne de la matrice C . De la même manière on met à jour les matrices A et B en minimisant les fonctions de coût $\eta_1^{N_1}$ et $\eta_1^{N_2}$.

Résumé :

L'algorithme 1 met à jour les lignes des matrices A , B et C en minimisant respectivement au sens des moindres carrés l'une des fonctions de coût $\eta_1^{N_1}$, $\eta_1^{N_2}$ et $\eta_1^{N_3}$.

$$D_{n_1}^A = (B^T B)^{-1} B^T (H_{n_1..}) C (C^T C)^{-1} \quad n_1 = 1, \dots, N_1 \quad (\text{E.19})$$

$$D_{n_2}^B = (C^T C)^{-1} C^T (H_{..n_2.}) A (A^T A)^{-1} \quad n_2 = 1, \dots, N_2 \quad (\text{E.20})$$

$$D_{n_3}^C = (A^T A)^{-1} A^T (H_{..n_3}) B (B^T B)^{-1} \quad n_3 = 1, \dots, N_3 \quad (\text{E.21})$$

avec

$$D_{n_1}^A = \mathbf{diag}(A_{n_1.}) \quad (\text{E.22})$$

$$D_{n_2}^B = \mathbf{diag}(B_{n_2.}) \quad (\text{E.23})$$

$$D_{n_3}^C = \mathbf{diag}(C_{n_3.}) \quad (\text{E.24})$$

E.3 ALS-PARAFAC 2

Cet algorithme, mis au point par Bro et Sidiropoulos [6], utilise l'expression de l'information du tenseur à l'aide de la famille des matrices $\{H_{..n_3}\}_{n_3=1\dots N_3}$ données par l'équation (3.6) :

$$H_{..n_3} = A D_{n_3}^C B^T \quad (\text{E.25})$$

On utilise la fonction de coût $\eta_1^{N_3}$ déjà utilisée dans l'algorithme 1. On met à jour les matrices A , B et C en minimisant cette même fonction, respectivement par rapport à A , B et $D_{n_3}^C$.

Soit

$$Z_{n_3} = BD_{n_3}^C = (D_{n_3}^C B^T)^T. \quad (\text{E.26})$$

La fonction de coût $\eta_1^{N_3}$ s'écrit alors :

$$\eta_1^{N_3} = \sum_{n_3=1}^{N_3} \|H_{..n_3} - AD_{n_3}^C B^T\|_F^2 \quad (\text{E.27})$$

$$= \sum_{n_3=1}^{N_3} \|H_{..n_3} - AZ_{n_3}^T\|_F^2 \quad (\text{E.28})$$

$$= \sum_{n_3=1}^{N_3} \text{tr}((H_{..n_3} - AZ_{n_3}^T)^T (H_{..n_3} - AZ_{n_3}^T)) \quad (\text{E.29})$$

$$(\text{E.30})$$

Soit $\nabla_{A^T}(\eta_1^{N_3})$ le gradient de $\eta_1^{N_3}$ par rapport à A^T .

$$\nabla_{A^T}(\eta_1^{N_3}) = -2 \sum_{n_3=1}^{N_3} \frac{\partial}{\partial A^T} \text{tr}(Z_{n_3} A^T H_{..n_3}) \quad (\text{E.31})$$

$$+ \sum_{n_3=1}^{N_3} \frac{\partial}{\partial A^T} \text{tr}(Z_{n_3} A^T A Z_{n_3}^T) \quad (\text{E.32})$$

$$= -2 \sum_{n_3=1}^{N_3} Z_{n_3}^T H_{..n_3}^T + 2 \sum_{n_3=1}^{N_3} Z_{n_3}^T Z_{n_3} A^T \quad (\text{E.33})$$

$$= -2 \left(\sum_{n_3=1}^{N_3} Z_{n_3}^T H_{..n_3}^T - \sum_{n_3=1}^{N_3} Z_{n_3}^T Z_{n_3} A^T \right) \quad (\text{E.34})$$

L'annulation de $\nabla_{A^T}(\eta_1^{N_3})$ donne la matrice A suivante :

$$A = \left(\sum_{n_3=1}^{N_3} H_{..n_3} B D_{n_3}^C \right) \left(\sum_{n_3=1}^{N_3} D_{n_3}^C B^T B D_{n_3}^C \right)^{-1} \quad (\text{E.35})$$

de la même manière en annulant le gradient $\nabla_B(\eta_1^{N_3})$ avec A et C fixés, on trouve :

$$B = \left(\sum_{n_3=1}^{N_3} H_{..n_3}^T A D_{n_3}^C \right) \left(\sum_{n_3=1}^{N_3} D_{n_3}^C A^T A D_{n_3}^C \right)^{-1} \quad (\text{E.36})$$

Pour le calcul de C , on reprend la même démonstration faite dans l'algorithme 1. En annulant $\nabla_{D_{n_3}^C}(\eta_1^{N_3})$ on obtient :

$$D_{n_3}^C = (A^T A)^{-1} A^T (H_{..n_3}) B (B^T B)^{-1}; n_3 = 1, \dots, N_3 \quad (\text{E.37})$$

Résumé :

L'algorithme 2 met à jour les matrices A , B et les lignes de la matrice C en minimisant au sens des moindres carrés la même fonction de coût $\eta_1^{N_3}$.

$$A = \left(\sum_{n_3=1}^{N_3} H_{..n_3} B D_{n_3}^C \right) \left(\sum_{n_3=1}^{N_3} D_{n_3}^C B^T B D_{n_3}^C \right)^{-1} \quad (\text{E.38})$$

$$B = \left(\sum_{n_3=1}^{N_3} H_{..n_3}^T A D_{n_3}^C \right) \left(\sum_{n_3=1}^{N_3} D_{n_3}^C A^T A D_{n_3}^C \right)^{-1} \quad (\text{E.39})$$

et

$$D_{n_3}^C = (A^T A)^{-1} A^T (H_{..n_3}) B (B^T B)^{-1}; n_3 = 1, \dots, N_3 \quad (\text{E.40})$$

E.4 ALS-PARAFAC 3

Cet algorithme a été mis au point par Harshman [26] et par Carroll et Chang [12]. On reprend la définition des matrices H_{N_1} , H_{N_2} et H_{N_3} de dimensions respectives $(N_1 \times N_2 N_3)$, $(N_2 \times N_1 N_3)$ et $(N_3 \times N_1 N_2)$.

$$H_{N_1} = \begin{bmatrix} H_{..1} & H_{..2} & \cdots & H_{..N_3} \end{bmatrix} \quad (\text{E.41})$$

$$H_{N_2} = \begin{bmatrix} H_{1..} & H_{2..} & \cdots & H_{N_1..} \end{bmatrix} \quad (\text{E.42})$$

$$H_{N_3} = \begin{bmatrix} H_{.1.} & H_{.2.} & \cdots & H_{.N_2.} \end{bmatrix} \quad (\text{E.43})$$

Selon la deuxième représentation matricielle du modèle PARAFAC et en utilisant le produit de Khatri-Rao ($| \otimes |$), les matrices H_{N_1} , H_{N_2} et H_{N_3} s'écrivent comme suit :

$$H_{N_1} = A(C| \otimes |B)^T \quad (\text{E.44})$$

$$H_{N_2} = B(A| \otimes |C)^T \quad (\text{E.45})$$

$$H_{N_3} = C(B| \otimes |A)^T \quad (\text{E.46})$$

Définissons maintenant les fonctions de coût $\eta_2^{N_1}$, $\eta_2^{N_2}$ et $\eta_2^{N_3}$ à minimiser :

$$\eta_2^{N_1} = \|H_{N_1} - A(C| \otimes |B)^T\|_F^2 \quad (\text{E.47})$$

$$\eta_2^{N_2} = \|H_{N_2} - B(A| \otimes |C)^T\|_F^2 \quad (\text{E.48})$$

$$\eta_2^{N_3} = \|H_{N_3} - C(B| \otimes |A)^T\|_F^2 \quad (\text{E.49})$$

soit $\Psi_{N_1} = (C| \otimes |B)$, la fonction de coût $\eta_2^{N_1}$ s'écrit :

$$\eta_2^{N_1} = \|H_{N_1} - A\Psi_{N_1}^T\|_F^2 \quad (\text{E.50})$$

$$= \mathbf{tr}((H_{N_1} - A\Psi_{N_1}^T)^T(H_{N_1} - A\Psi_{N_1}^T)) \quad (\text{E.51})$$

$$(\text{E.52})$$

Soit $\nabla_{A^T}(\eta_2^{N_1})$ le gradient de $\eta_2^{N_1}$ par rapport à A^T .

$$\nabla_{A^T}(\eta_2^{N_1}) = -2(\Psi_{N_1}^T H_{N_1}^T - \Psi_{N_1}^T \Psi_{N_1} A^T) \quad (\text{E.53})$$

L'annulation de $\nabla_{A^T}(\eta_2^{N_1})$ donne la matrice A suivante :

$$A = H_{N_1} \Psi_{N_1} (\Psi_{N_1}^T \Psi_{N_1})^{-1} \quad (\text{E.54})$$

De la même manière en posant $\Psi_{N_2} = (A| \otimes |C)$ et $\Psi_{N_3} = (B| \otimes |A)$ et en annulant les gradients $\nabla_{B^T}(\eta_2^{N_2})$ et $\nabla_{C^T}(\eta_2^{N_3})$ on trouve :

$$B = H_{N_2} \Psi_{N_2} (\Psi_{N_2}^T \Psi_{N_2})^{-1} \quad (\text{E.55})$$

$$C = H_{N_3} \Psi_{N_3} (\Psi_{N_3}^T \Psi_{N_3})^{-1} \quad (\text{E.56})$$

Remarque :

Afin d'éviter la manipulation du produit de Khatri-Rao [26], [12], on peut calculer autrement les matrices $(H_{N_1} \Psi_{N_1})$ et $(\Psi_{N_1}^T \Psi_{N_1})$ pour mettre à jour la matrice A .

Démonstration :

$$H_{N_1} \Psi_{N_1} = H_{N_1} (C| \otimes |B) \quad (\text{E.57})$$

$$= \begin{bmatrix} H_{..1} & \cdots & H_{..N_3} \end{bmatrix} \begin{bmatrix} C_{.1} \otimes B_{.1} & \cdots & C_{.P} \otimes B_{.P} \end{bmatrix} \quad (\text{E.58})$$

$$= \begin{bmatrix} H_{..1} & \cdots & H_{..N_3} \end{bmatrix} \begin{bmatrix} c_{11}B_{.1} & \cdots & c_{1P}B_{.P} \\ \vdots & & \vdots \\ c_{N_3 1}B_{.1} & \cdots & c_{N_3 P}B_{.P} \end{bmatrix} \quad (\text{E.59})$$

donc

$$H_{N_1} \Psi_{N_1} = H_{..1} B D_1^C + H_{..2} B D_2^C + \dots + H_{..N_3} B D_{N_3}^C \quad (\text{E.60})$$

et

$$\Psi_{N_1}^T \Psi_{N_1} = (C| \otimes |B)^T (C| \otimes |B) \quad (\text{E.61})$$

$$= \begin{pmatrix} (C_{.1})^T \otimes (B_{.1})^T \\ \vdots \\ (C_{.P})^T \otimes (B_{.P})^T \end{pmatrix} \begin{pmatrix} C_{.1} \otimes B_{.1} & \dots & C_{.P} \otimes B_{.P} \end{pmatrix} \quad (\text{E.62})$$

$$= \begin{pmatrix} (C_{.1})^T C_{.1} \otimes (B_{.1})^T B_{.1} & \dots & (C_{.1})^T C_{.P} \otimes (B_{.1})^T B_{.P} \\ \vdots & \dots & \vdots \\ (C_{.P})^T C_{.1} \otimes (B_{.P})^T B_{.1} & \dots & (C_{.P})^T C_{.P} \otimes (B_{.P})^T B_{.P} \end{pmatrix} \quad (\text{E.63})$$

$$= \begin{pmatrix} ((C_{.1})^T C_{.1}) ((B_{.1})^T B_{.1}) & \dots & ((C_{.1})^T C_{.P}) ((B_{.1})^T B_{.P}) \\ \vdots & \dots & \vdots \\ ((C_{.P})^T C_{.1}) ((B_{.P})^T B_{.1}) & \dots & ((C_{.P})^T C_{.P}) ((B_{.P})^T B_{.P}) \end{pmatrix} \quad (\text{E.64})$$

$$= (C^T C) \circ (B^T B) \quad (\text{E.65})$$

$$= (B^T B) \circ (C^T C) \quad (\text{E.66})$$

où l'opérateur \circ représente le produit de Hadamard.

De la même manière on peut calculer $H_{N_2} \Psi_{N_2}$, $\Psi_{N_2}^T \Psi_{N_2}$, $H_{N_3} \Psi_{N_3}$ et $\Psi_{N_3}^T \Psi_{N_3}$ sans passer par le produit de Khatri-Rao.

Résumé :

L'algorithme 3 met à jour les matrices A , B et C en minimisant respectivement au sens des moindres carrés les fonctions de coût $\eta_2^{N_1}$, $\eta_2^{N_2}$ et $\eta_2^{N_3}$.

$$\Psi_{N_1} = (C| \otimes |B) \implies A = H_{N_1} \Psi_{N_1} (\Psi_{N_1}^T \Psi_{N_1})^{-1} \quad (\text{E.67})$$

$$\Psi_{N_2} = (A| \otimes |C) \implies B = H_{N_2} \Psi_{N_2} (\Psi_{N_2}^T \Psi_{N_2})^{-1} \quad (\text{E.68})$$

$$\Psi_{N_3} = (B| \otimes |A) \implies C = H_{N_3} \Psi_{N_3} (\Psi_{N_3}^T \Psi_{N_3})^{-1} \quad (\text{E.69})$$

Bibliographie

- [1] A. Alper. "A consideration of the discrete Volterra series". *IEEE Trans. Automatic Control*, 10 (3) : 322–327, 1965.
- [2] J. M. F. Ten Berge and H. A. L. Kiers. "Some uniqueness results for PARAFAC2". *Psychometrika*, 61 : 123–132, 1996.
- [3] S. A. Billings. "Identification of nonlinear systems - a survey". *IEE Proceedings - Part D*, 127(6) : 272–285, 1980.
- [4] S. Boyd and L.O. Chua. "Fading memory and the problem of approximating nonlinear operators with Volterra series". *IEEE Tr. Circuits and Systems*, 32 (11) : 1150–1171, 1985.
- [5] R. Bro. "*Multi-way Analysis in the Food Industry : Models, Algorithms ans Applications*". PhD thesis, University of Amsterdam - Holland, 1998.
- [6] R. Bro and N. D. Sidiropoulos. "Least squares algorithms under unimodality and non-negativity constraints". *Journal of Chemometrics*, 12 (4) : 223–247, 1998.
- [7] V. Broman and M. J. Shensa. "A compact algorithm for the intersection and approximation of N-dimensional polytopes". *Mathematics and Computers in Simulation*, 32 : 469–480, 1990.
- [8] P. W. Broome. "Discrete orthonormal sequences". *Journal of the association for computing machinery*, 12 (2) : 151–168, 1965.
- [9] R. J. G. B. Campello, W. Amaral, and G. Favier. "Optimal expansions of discrete-time Volterra models using Laguerre functions". *Automatica*, 40 : 815–822, 2004.
- [10] R. J. G. B. Campello, W. C. Amaral, and G. Favier. "Optimal Laguerre series expansion of discrete Volterra models". In *ECC'2001, European Control Conference*, pages 272–277, September 2001.

- [11] R. J. G. B. Campello, G. Favier, and W. C. Amaral. "Optimal expansions of discrete-time Volterra models using Laguerre functions". In *IFAC Symp. SYSID, Rotterdam, The Netherlands*, pages 151–168, 2003.
- [12] J.D. Carroll and J. J. Chang. "Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition". *Psychometrika*, 35 : 283–319, 1970.
- [13] G. J. Clowes. "Choice of the time-scaling factor for linear system approximations using orthonormal Laguerre functions". *IEEE Trans. Automatic Control*, 10 : 487–489, 1965.
- [14] G. Costa-Oliveira, W. Amaral, G. Favier, and G. Dumont. "Constrained robust predictive controller for uncertain processes modeled by orthonormal series functions". *Automatica*, 36 (4) :563–572, 2000.
- [15] A. C. den Brinker. "Optimality conditions for a specific class of truncated Kautz series". *IEEE Tr. Circuits and Systems - II Analog and Digital Signal Processing*, 63 (8) : 597–599, 1996.
- [16] A. C. den Brinker, F. P. A. Benders, and T. O. e Silva. "Optimality conditions for truncated Kautz series". *IEEE Tr. Circuits and Systems - II Analog and Digital Signal Processing*, 34 (2) : 117–122, 1996.
- [17] W. A. Van der Kloot and P. M. Kroonenberg. "External analysis with three-mode principal component models". *Psychometrika*, 50 : 479–498, 1985.
- [18] G. A. Dumont and Y. Fu. "Nonlinear adaptive control via Laguerre expansion of Volterra kernels". *Int. J. Adaptive Control and Signal Processing*, 7 : 367–382, 1993.
- [19] T. O. e Silva. "On the determination of the optimal pole position of Laguerre filters". *IEEE Tr. Signal Processing*, 43 (9) : 2079–2087, 1995.
- [20] T. O. e Silva. "Optimality conditions for truncated Kautz networks with two periodically repeating complex conjugate poles". *IEEE Tr. Automatic control*, 40 (2) : 342–346, 1995.
- [21] G. Favier. *Estimation paramétrique de modèles entrée-sortie, Chapitre 7 de Signaux aléatoires, modélisation, estimation, détection, Coordination M. Guglielmi*. Hermes, 2004.

- [22] G. Favier, A. Kibangou, and R. J. G. B. Campello. "Nonlinear systems modelling by means of Generalized Orthonormal Basis functions". In *SSD'03, IEEE Conference on Signals, Systems, Decision and information technology (Invited paper)*, Sousse, Tunisia, 2003.
- [23] E. Fogel, Y. F. Huang, J. Lee, Z. Yu, and C. Garcia. "On the value of information in system identification : bounded noise case". *Automatica*, 18 (2) :229–238, 1982.
- [24] Y. Fu and G. A. Dumont. "An optimum time scale for discrete Laguerre network". *IEEE Trans. on Automatic Control*, 38 (6) : 934–938, 1993.
- [25] S. Gunnarsson and B. Wahlberg. "Some asymptotic results in recursive identification using Laguerre models". *Int. Journal of Adaptive Control and Signal Processing*, 5 : 313–333, 1991.
- [26] R. A. Harshman. "Foundation of the PARAFAC procedure : Models and conditions for an explanatory multimodal factor analysis". *UCLA working papers in phonetics*, 16 : 1–84, 1970.
- [27] R. A. Harshman. "Determination and proof of minimum uniqueness conditions for PARAFAC". *UCLA working papers in phonetics*, 22 : 111–117, 1972.
- [28] R. A. Harshman and M. E. Lundy. "PARAFAC : Parallel factor analysis". *Comp. Stat. and Data Anal.*, 18 : 39–72, 1994.
- [29] R. A. Harshman and M. E. Lundy. "Uniqueness proof for a family of models sharing features of TUCKER's three-mode factor analysis and PARAFAC/CANDECOMP". *Psychometrika*, 61 : 133–154, 1996.
- [30] P. S. C. Heuberger, P. M. J. Van den Hof, and O. H. Bosgra. "A generalized orthonormal basis for linear dynamical systems". *IEEE Trans. Automatic Control*, 40 (3) : 451–465, 1995.
- [31] A. Khouaja and G. Favier. "Third-order Volterra system identification using a PARAFAC approach". *submitted to IEEE Tr. Signal Processing*, 2004.
- [32] A. Khouaja and G. Favier. "Identification of SVD-PARAFAC based third-order Volterra models using an ARLS algorithm". *Accepted for publication at IFAC World Symposium, Prague, Czech Republic, July 2005*.

- [33] A. Khouaja, G. Favier, and H. Messaoud. "System identification using generalized orthonormal bases". In *SSD'03, IEEE Conference on Signal, Systems, Decision and information theory, Sousse, Tunisia, Mars 2003*.
- [34] A. Khouaja, G. Favier, and H. Messaoud. "Pole estimation for constructing generalized orthonormal bases in the context of system identification". *submitted to Systems, Analysis, Modeling and Simulation journal, SAMS Special Issue, 2004*.
- [35] A. Khouaja, A. Kibangou, and G. Favier. "Third-order Volterra kernels complexity reduction using PARAFAC". In *ISCCSP'04 IEEE International Symposium on Control, Communications and Signal Processing, Hammamet, Tunisia, pages 857–860, Marsh 2004*.
- [36] A. Kibangou, G. Favier, and M. M. Hassani. "A growing approach for selecting generalized orthonormal basis functions in the context of system modeling". In *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, NSIP'03, Grado, Italy, 2003*.
- [37] P. M. Kroonenberg. *"Three-mode principal component analysis - Theory and applications"*, volume 18. DSWO Press, M-T series, 1983.
- [38] J. B. Kruskal. "Three-way arrays : Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics". *Linear algebra and its applications*, 18 : 95–138, 1977.
- [39] H. P. Lahanier and E. Walter. Limited-complexity polyhedral tracking. *Int. Journal of Control*, 15 : 1417–1426, 1996.
- [40] J. Lee and V. J. Mathews. "A fast recursive least squares adaptive second-order Volterra filter and its performance analysis". *IEEE Trans. Signal Processing*, 41 (3) : 1087–1102, 1993.
- [41] C. L. De Ligny and al. "Three-mode factor analysis of data on retention in normal-phase high performance liquid chromatography". *Journal of Chromatography*, 301 : 311, 1984.
- [42] P. Lindskog. *"Methods, algorithms and tools for system identification based on prior knowledge"*. PhD thesis, Linköping University, Sweden, 1996.
- [43] L. Ljung. *"System identification : Theory for user"*. Prentice-Hall, 1987.

- [44] R. Malti. "*Représentation de systèmes discrets sur la base des filtres orthogonaux : Application à la modélisation des systèmes multi-variables*". PhD thesis, INPL, Nancy, France, 1999.
- [45] R. Malti, D. Maquin, and J. Ragot. "Optimality conditions for the truncated network of the generalized discrete orthonormal basis having real poles". In *37th IEEE Conf. on Decision and Control, Tampa, Floride (USA)*, pages 2189–2194, 1998.
- [46] S. Maraoui and H. Messaoud. "Limited complexity bounding error algorithms : design and comparative study". In *1st IFAC/IEEE Symposium On System Structure and Control, Prague, Czech Republic*, 2001.
- [47] V. Z. Marmarelis. "Identification of nonlinear biological systems using Laguerre expansions of kernels". *Anal. of Biomedical Engineering*, 21 : 573–589, 1993.
- [48] M. A. Masnadi-Shirazi and M. Ghasemi. "Adaptive Laguerre network realization". *Signal Processing*, 80 (10) : 2169–2186, 2000.
- [49] H. Messaoud. "*Identification et commande robustes : Etude et comparaison d'algorithmes*". PhD thesis, Ecole Normale Supérieure de l'Enseignement Technique de Tunis, 1993.
- [50] H. Messaoud and G. Favier. "Recursive determination of parameter uncertainty intervals for linear models with unknown but bounded errors". In *10th IFAC Symposium on System Identification, Copenhagen, Denmark*, pages 365–370, 1994.
- [51] F. Michaud. *Méthodes adaptatives pour le signal, outils mathématiques et mise en oeuvre des algorithmes*. Traité des nouvelles technologies, série Traitement du signal, Hermes, Paris, 1992.
- [52] S. H. Mo and J. P. Norton. "Fast and robust algorithm to compute exact polytope parameter bounds". In *Mathematics and Computers in Simulation, North Holland*, volume 32, pages 481–493, 1990.
- [53] B. Ninness and F. Gustafsson. "A unifying construction of orthonormal bases for system identification". *IEEE Trans. Automatic Control*, 40 (3) : 451–465, 1997.
- [54] G. H. C. Oliveira, W. C. Amaral, G. Favier, and G. A. Dumont. "Constrained robust predictive controller for uncertain processes modeled by orthonormal series functions". *Automatica*, 36 (4) : 563–574, 2000.

- [55] T. M. Panicker and V. J. Mathews. "Parallel-cascade realizations and approximations of truncated Volterra systems". *IEEE Tr. on Signal Processing*, 46 (10) : 2829–2832, 1998.
- [56] B. Picinbono. "*Signaux aléatoires, Tome 3, Bases du traitement statistique du signal avec problèmes résolus*". DUNOD, Paris, 1995.
- [57] M. Schetzen. *The Volterra and Wiener theories of nonlinear systems*. New York : Wiley, 1980.
- [58] N. D. Sidiropoulos and R. Bro. "On the uniqueness of multilinear decomposition of N-way arrays". *Journal of Chemometrics*, 14 : 229–239, 2000.
- [59] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hjalmarsson, and A. Juditsky. "Nonlinear black-box modeling in system identification : A unified overview". *Automatica*, 31 (12) :1691–1724, 1995.
- [60] T. Soderstrom and P. Stoica. *System Identification*. Prentice-Hall International, 1989.
- [61] N. Tanguy, R. Morvan, P. Vilbé, and L. C. Calvez. "Pertinent choice of parameters for discrete Kautz approximation". *IEEE Tr. Automatic Control*, 47 (5) : 783–787, 2002.
- [62] N. Tanguy, P. Vilbé, and L. C. Calvez. "Optimum choice of free parameter in orthonormal approximations". *IEEE Tr. Automatic Control*, 40 (10) : 1811–1813, 1995.
- [63] L. R. Tucker. "Some mathematical notes on three-mode factor analysis". *Psychometrika*, 31 (3) : 279–311, 1966.
- [64] S. M. Veres, H. Messaoud, and J. P. Norton. "Limited-complexity model unfalsifying adaptive tracking control". *Int. Journal of Control*, 15 : 1417–1426, 1999.
- [65] A. Vicino and G. Zappa. "Sequential approximation of feasible parameter sets for identification with set membership uncertainty". *IEEE Trans. Automatic Control*, 41 (6) : 774–784, 1996.
- [66] B. Wahlberg. "System identification using Laguerre models". *IEEE Tr. Automatic Control*, 36 : 551–562, 1991.
- [67] B. Wahlberg. "Laguerre and Kautz models". In *IFAC SYSID symposium, 1994, Copenhagen*, volume 3, pages 1–12, 1994.

- [68] B. Wahlberg and P. Mäkilä. "On approximation of stable linear systems using Laguerre and Kautz functions". *Automatica*, 32 (5) : 693–708, 1996.
- [69] E. Walter and H. Piet-Lahanier. "Exact recursive polyhedral description of the feasible parameter set for bounded-error models". *IEEE Tr. Automatic Control*, 34 (8) : 911–914, 1989.
- [70] N. Wiener. "*Nonlinear problems in random theory*". New York : Wiley, 1958.