

Identification de modèles de Volterra basée sur la décomposition PARAFAC

G rard FAVIER, Thomas BOUILLOC

Laboratoire I3S, Universit  de Nice Sophia Antipolis, CNRS,
Les Algorithmes- B t. Euclide B, 2000 Route des lucioles, B.P. 121 - 06903 Sophia Antipolis Cedex, France
favier@i3s.unice.fr, bouilloc@i3s.unice.fr

R sum  – Les mod les de Volterra sont tr s utilis s dans de nombreux domaines d’application. Le principal inconv nient de ces mod les est leur complexit  param trique due au tr s grand nombre de coefficients qui caract risent leurs noyaux. Cet article pr sente une nouvelle m thode permettant de r duire cette complexit  param trique en consid rant les noyaux de Volterra d’ordre sup rieur   un comme des tenseurs sym triques et en les d composant   l’aide de la d composition PARAFAC. Les mod les de Volterra-PARAFAC ainsi obtenus peuvent  tre vus comme une s rie de mod les de Wiener mis en parall le. En exploitant cette nouvelle formulation des mod les de Volterra, nous proposons un algorithme d’identification r cursif du type filtre de Kalman  tendu. Des r sultats de simulation illustrent le comportement de cet algorithme pour l’identification de syst mes de Volterra cubiques.

Abstract – Discrete-time Volterra models play an important role in many application areas. The main drawback of these models is their parametric complexity due to the huge number of their parameters, the kernel coefficients. Using the symmetry property of the Volterra kernels, these ones can be viewed as symmetric tensors. In this paper, we apply the PARAFAC decomposition for reducing the kernel parametric complexity. The so obtained Volterra-PARAFAC models can be viewed as Wiener models in parallel. We propose to use an extended Kalman filter for recursively estimating the parameters of these models. Some simulation results illustrate the behaviour of the proposed identification method for cubic Volterra systems.

1 Introduction

Les mod les de Volterra   temps discret, de dimension finie, appel s aussi d veloppements en s ries de Volterra tronqu s ou encore mod les polynomiaux non r cursifs, peuvent  tre utilis s pour repr senter tout syst me non lin aire   m moire  vanouissante [4]. Ces mod les ont fait l’objet de tr s nombreuses applications dans diff rents domaines, comme par exemple l’annulation d’ cho et de bruit [1], [19], la lin arisation de haut-parleur [13], la mod lisation, l’ galisation et la pr distortion de canaux de communication non lin aires [2], [3], [6], [10], [17], la mod lisation de syst mes physiologiques [7], ou encore la commande de syst mes non lin aires [8].

Les mod les de Volterra qui peuvent  tre vus comme une extension non lin aire du mod le lin aire   r ponse impulsionnelle finie (RIF), sont interpr tables en termes de convolutions multidimensionnelles. Ils poss dent deux propri t s int ressantes : lin arit  vis- -vis de leurs param tres, les coefficients des noyaux, et stabilit  garantie au sens entr e born e-sortie born e. Leur principal inconv nient est leur grande complexit  param trique en raison du tr s grand nombre de coefficients contenus dans leurs noyaux. Il existe deux principales fa ons de r duire cette complexit  : une premi re approche consiste   d velopper les noyaux de Volterra sur des bases de fonctions orthonor-

males telles que les fonctions de Laguerre [5], [9], [16], [18], ou les bases orthonormales g n ralis es, i.e. des fonctions de bases orthonormales caract ris es par plusieurs p les [15]. Une autre approche, introduite dans [14], consiste   consid rer les noyaux de Volterra comme des tenseurs et   utiliser des d compositions tensorielles telles que la d composition PARAFAC [12]. Il en r sulte des mod les de type Volterra-PARAFAC qui peuvent  tre vus comme la mise en parall le d’une s rie de mod les de Wiener. En combinant cette nouvelle formulation des mod les de Volterra avec une m thode de lin arisation, nous proposons un algorithme d’identification r cursif du type filtre de Kalman  tendu pour estimer les param tres des d compositions PARAFAC des noyaux.

La suite de cet article est organis e comme suit. Dans le paragraphe 2, nous pr sentons le mod le de Volterra et indiquons comment ses noyaux peuvent  tre mis sous forme sym trique. Dans le paragraphe 3, nous rappelons la d composition PARAFAC d’un tenseur puis exploitons cette d composition pour d velopper le mod le Volterra-PARAFAC. Les  quations du filtre de Kalman  tendu sont d crites dans le paragraphe 4 pour l’estimation r cursive des param tres d’un tel mod le. Des r sultats de simulation sont fournis dans le paragraphe 5 pour illustrer le comportement de cet algorithme dans le cas de mod les de Volterra cubiques. Enfin, nous concluons cet article en indi-

quant quelques perspectives de ce travail.

2 Les modèles de Volterra

Un modèle de Volterra d'ordre P d'un système non linéaire, mono entrée-mono sortie, causal, stable, à mémoire évanouissante et donc finie, est décrit à l'aide de la relation entrée-sortie suivante :

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

$$= h_0 + \sum_{p=1}^P y_p(k)$$

où $u(k)$ et $y(k)$ désignent respectivement les signaux d'entrée et de sortie, P est le degré de non linéarité du modèle de Volterra, M_p est la mémoire du terme homogène d'ordre p représenté par $y_p(k)$, et $h_p(m_1, \dots, m_p)$ est un coefficient du noyau d'ordre p . Ce coefficient étant caractérisé par p indices, il peut être vu comme un élément d'un tenseur $\mathbb{H}_p \in \mathcal{K}^{M_p \times M_p \times \dots \times M_p}$, d'ordre p , avec $\mathcal{K} = \mathcal{R}$ ou \mathcal{C} , selon que les coefficients des noyaux sont à valeurs réelles ou complexes. Le noyau d'ordre p est caractérisé par M_p^p coefficients.

Comme toute permutation des indices m_1, \dots, m_p correspond au même produit $\prod_{i=1}^p u(k - m_i)$ d'entrées retardées, on peut sommer tous les coefficients associés à ces permutations pour obtenir un noyau symétrique défini comme :

$$h_{p, \text{sym}}(m_1, \dots, m_p) = \frac{1}{p!} \sum_{\pi(\cdot)} h_p(m_{\pi(1)}, \dots, m_{\pi(p)})$$

où $\pi(\cdot)$ désigne une permutation des indices m_1, \dots, m_p . Le nombre de coefficients indépendants contenus dans le noyau symétrique d'ordre p est égal à $C_p^{M_p + p - 1}$. Ainsi, pour un noyau cubique de mémoire $M_3 = 10$, le nombre de coefficients est égal à $M_3^3 = 10^3$ dans le cas non symétrique et $C_3^{12} = 220$ dans le cas symétrique, ce qui correspond à un taux de réduction de complexité de 78%.

Le noyau d'ordre p est dit séparable s'il peut s'exprimer comme le produit de p noyaux d'ordre un :

$$h_p(m_1, \dots, m_p) = \prod_{i=1}^p h_{m_i}^{(i)} \quad (2)$$

où $h_{m_i}^{(i)}$ représente le m_i ème élément du noyau d'ordre un $\mathbf{h}^{(i)}$. Si le noyau séparable est aussi symétrique, alors les p noyaux d'ordre un $\mathbf{h}^{(i)}$ sont identiques et (2) devient :

$$h_p(m_1, \dots, m_p) = \prod_{i=1}^p h_{m_i}$$

3 Les modèles de Volterra-PARAFAC

Dans ce paragraphe, nous allons tout d'abord rappeler la décomposition PARAFAC d'un tenseur, puis nous exploiterons cette décomposition pour développer les noyaux d'un modèle de Volterra, ce qui donnera le modèle Volterra-PARAFAC.

La décomposition PARAFAC d'un tenseur $\mathbb{H} \in \mathcal{K}^{M_1 \times M_2 \times \dots \times M_P}$, d'ordre P et de dimensions $M_1 \times M_2 \times \dots \times M_P$, s'écrit sous forme scalaire comme :

$$h_{m_1 m_2 \dots m_P} = \sum_{r=1}^R \prod_{p=1}^P a_{m_p r}^{(p)} \quad m_p = 1, \dots, M_p \quad (3)$$

où $a_{m_p r}^{(p)}$ est un élément du facteur matriciel $\mathbf{A}^{(p)} \in \mathcal{K}^{M_p \times R}$, $p = 1, \dots, P$, et R est le rang du tenseur.

L'équation (3) peut être réécrite comme :

$$\mathbb{H} = \sum_{r=1}^R \mathbf{A}_{\cdot r}^{(1)} \circ \mathbf{A}_{\cdot r}^{(2)} \circ \dots \circ \mathbf{A}_{\cdot r}^{(P)} \quad (4)$$

où \circ désigne le produit extérieur et $\mathbf{A}_{\cdot r}^{(p)}$ la r ème colonne de la matrice $\mathbf{A}^{(p)}$. Cette écriture montre que PARAFAC correspond à une décomposition du tenseur d'ordre P et de rang R , en une somme de R tenseurs de rang un, i.e. une somme de R produits extérieurs de P vecteurs.

Nous pouvons noter que pour un tenseur \mathbb{H} d'ordre P de rang un, (3) devient :

$$h_{m_1 m_2 \dots m_P} = \prod_{p=1}^P a_{m_p}^{(p)}, \quad m_p = 1, \dots, M_p \quad (5)$$

où $a_{m_p}^{(p)}$ est un élément du facteur vectoriel $\mathbf{a}^{(p)} \in \mathcal{K}^{M_p \times 1}$.

En comparant (5) avec (2), nous pouvons donc conclure que la décomposition PARAFAC d'un noyau de Volterra de rang un équivaut à celle d'un noyau séparable.

Dans le cas d'un tenseur \mathbb{H} symétrique, d'ordre P , de rang R , les P facteurs matriciels $\mathbf{A}^{(p)}$, sont identiques et égaux à \mathbf{A} . De façon similaire, pour un tenseur \mathbb{H} symétrique, d'ordre P , de rang un, les P facteurs vectoriels $\mathbf{a}^{(p)}$, $p = 1, \dots, P$, sont identiques et égaux à \mathbf{a} .

L'utilisation de la décomposition PARAFAC d'un noyau de Volterra d'ordre p , de rang un, symétrique permet de réécrire le terme homogène $y_p(k)$ de (1) comme :



FIG. 1 – Réalisation du terme homogène d'ordre p d'un modèle de Volterra comme un modèle de Wiener. Cas d'un noyau de rang un.

$$\begin{aligned}
 y_p(k) &= \sum_{m_1=1}^{M_p} \cdots \sum_{m_p=1}^{M_p} h_p(m_1, \dots, m_p) \prod_{i=1}^p u(k - m_i) \\
 &= \sum_{m_1=1}^{M_p} \cdots \sum_{m_p=1}^{M_p} \prod_{i=1}^p a_{m_i}^{(p)} u(k - m_i) \\
 &= \left(\mathbf{u}^T(k) \mathbf{a}^{(p)} \right)^p \quad (6)
 \end{aligned}$$

où $\mathbf{u}^T(k) = [u(k-1) \ \cdots \ u(k-M_p)] \in \mathcal{K}^{1 \times M_p}$ est le vecteur de régression linéaire associé au noyau de Volterra d'ordre p , de mémoire M_p , et $\mathbf{a}^{(p)T} = [a_1^{(p)} \ \cdots \ a_{M_p}^{(p)}] \in \mathcal{K}^{1 \times M_p}$ est formé des coefficients du facteur vectoriel (vecteur générateur) de la décomposition PARAFAC de ce noyau. Nous déduisons que (6) peut être vu comme la sortie d'un modèle de Wiener obtenu en concaténant un modèle linéaire RIF, de mémoire M_p , avec une non linéarité sans mémoire de degré p , comme illustré à l'aide de la Fig. 1.

Pour un noyau de Volterra d'ordre p , de rang r_p , symétrique, (6) devient :

$$y_p(k) = \sum_{r=1}^{r_p} \left(\mathbf{u}^T(k) \mathbf{A}_{.r}^{(p)} \right)^p \quad (7)$$

Le terme homogène d'ordre p peut donc être réalisé en parallélisant r_p modèles de Wiener, chacun de ces modèles étant associé avec une colonne du facteur matriciel de la décomposition PARAFAC du noyau. Par suite, la sortie (1) du modèle de Volterra peut être obtenue comme la somme du terme constant h_0 , et des sorties de $\sum_{p=1}^P r_p$ modèles de Wiener en parallèle, comme illustré à l'aide de la Fig. 2 pour un modèle de Volterra cubique, où $\mathbf{A}_{.r}^{(p)}$ désigne la $r^{\text{ème}}$ colonne du facteur matriciel de la décomposition PARAFAC du noyau d'ordre p , $p = 2, 3$, $\mathbf{A}_{.1}^{(1)} = [h_1(1) \ \cdots \ h_1(M_1)]^T$ et $\mathbf{u}_p^T(k) = [u(k-1) \ \cdots \ u(k-M_p)]$, où $p = 1, 2$ et 3 .

En utilisant la décomposition PARAFAC des noyaux, la sortie du modèle de Volterra-PARAFAC s'écrit comme :

$$y(k) = h_0 + \sum_{p=1}^P \sum_{r=1}^{r_p} \left(\mathbf{u}_p^T(k) \mathbf{A}_{.r}^{(p)} \right)^p \quad (8)$$

Dans le paragraphe suivant, nous allons exploiter cette écriture du modèle de Volterra-PARAFAC pour dériver les équations du

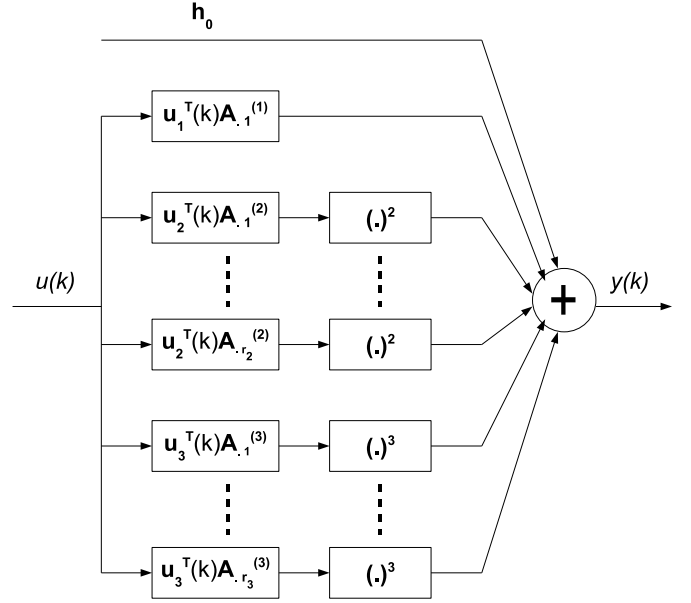


FIG. 2 – Réalisation d'un modèle de Volterra cubique comme des modèles de Wiener en parallèle.

filtre de Kalman étendu permettant d'estimer de façon récursive les paramètres des décompositions PARAFAC des différents noyaux, c'est-à-dire les éléments des facteurs vectoriels $\mathbf{A}_{.r}^{(p)}$, $r = 1, \dots, r_p$, $p = 1, \dots, P$. Pour simplifier la présentation, nous considérerons le cas d'un modèle de Volterra-PARAFAC cubique, l'extension au cas d'un modèle d'ordre quelconque ne posant pas de problème.

4 Algorithme d'identification récursif d'un modèle de Volterra-PARAFAC

Dans le cas d'un modèle de Volterra-PARAFAC cubique, la relation entrée-sortie s'écrit :

$$y(k) = h_0 + \sum_{p=1}^3 \sum_{r=1}^{r_p} \left(\mathbf{u}_p^T(k) \mathbf{A}_{.r}^{(p)} \right)^p \quad (9)$$

Définissons les vecteurs suivants :

$$\boldsymbol{\theta}^T = [h_0 \ \boldsymbol{\theta}_1^T \ \boldsymbol{\theta}_2^T \ \boldsymbol{\theta}_3^T]$$

$$\boldsymbol{\theta}_1^T = \mathbf{A}_{.1}^{(1)T} = [h_1(1) \ \cdots \ h_1(M_1)]^T$$

$$\boldsymbol{\theta}_p = \text{vec}(\mathbf{A}^{(p)}), \quad p = 2, 3$$

L'équation (9) donnant la sortie du modèle peut alors être réécrite comme :

$$y(k) = f(k, \boldsymbol{\theta})$$

et la sortie mesurée du système non linéaire à identifier s'écrit donc :

$$s(k) = f(k, \boldsymbol{\theta}) + \nu(k)$$

où le bruit additif $\nu(k)$, incluant à la fois le bruit de mesure et l'erreur de modélisation, est supposé être blanc, centré, de variance σ^2 .

En définissant :

$$\boldsymbol{\varphi}(k) = \left. \frac{\partial f(k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}(k-1)}$$

on a :

$$\boldsymbol{\varphi}^T(k) = [1 \quad \boldsymbol{\varphi}_1^T(k) \quad \boldsymbol{\varphi}_2^T(k) \quad \boldsymbol{\varphi}_3^T(k)]$$

avec

$$\boldsymbol{\varphi}_1^T(k) = \mathbf{u}_1(k)$$

et

$$\boldsymbol{\varphi}_p^T(k) = p \left[\alpha_{p,1}^{p-1}(k) \quad \cdots \quad \alpha_{p,r_p}^{p-1}(k) \right] \otimes \mathbf{u}_p^T(k) \quad p = 2, 3$$

où \otimes désigne le produit de Kronecker, et :

$$\alpha_{p,r}(k) = \left(\mathbf{u}_p^T(k) \hat{\mathbf{A}}_{.,r}^{(p)}(k-1) \right), \quad r = 1, \dots, r_p, \quad p = 2, 3$$

$$\mathbf{u}_p^T(k) = [u(k-1) \quad \cdots \quad u(k-M_p)], \quad p = 1, 2 \text{ et } 3$$

En notant que :

$$f(k, \hat{\boldsymbol{\theta}}(k-1)) = \boldsymbol{\Psi}^T(k) \hat{\boldsymbol{\theta}}(k-1) \quad (10)$$

où

$$\boldsymbol{\Psi}^T(k) = [1 \quad \boldsymbol{\varphi}_1^T(k) \quad \frac{1}{2} \boldsymbol{\varphi}_2^T(k) \quad \frac{1}{3} \boldsymbol{\varphi}_3^T(k)] \quad (11)$$

et en supposant les paramètres constants ($\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k)$), les équations du filtre de Kalman étendu sont résumées ci après :

$$\boldsymbol{\varepsilon}(k) = s(k) - \boldsymbol{\Psi}^T(k) \hat{\boldsymbol{\theta}}(k-1)$$

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{G}(k) \boldsymbol{\varepsilon}(k)$$

$$\mathbf{G}(k) = \mathbf{P}(k-1) \boldsymbol{\varphi}(k) [\boldsymbol{\varphi}^T(k) \mathbf{P}(k-1) \boldsymbol{\varphi}(k) + \sigma^2]^{-1}$$

$$\mathbf{P}(k) = \lambda^{-1} (\mathbf{I} - \mathbf{G}(k) \boldsymbol{\varphi}^T(k)) \mathbf{P}(k-1)$$

où λ désigne le facteur d'oubli.

5 Résultats de simulation

Dans ce paragraphe, nous présentons quelques résultats de simulations pour illustrer le comportement de l'algorithme d'identification proposé. Le système simulé est un système de Volterra cubique, dont les mémoires et les rangs des noyaux sont $M_1 = M_2 = M_3 = 3$, $R_2 = R_3 = 2$. Dix modèles ont été générés en tirant aléatoirement les coefficients des noyaux selon une distribution uniforme. D'autre part, les simulations de type Monte Carlo ont été réalisées à l'aide de 20 séquences de bruit et 10 séquences d'entrée, chacune d'elles étant une séquence de bruit blanc, pour un rapport signal sur bruit (RSB) fixé. Les résultats de simulation sont fournis en termes des critères de

performance suivants : erreur quadratique moyenne normalisée (EQMN) sur la sortie d'une part et sur les noyaux estimés d'autre part :

$$\text{EQMN}_{\mathbf{s}} = 10 \log_{10} \left(\frac{\|\hat{\mathbf{s}} - \mathbf{s}\|_2^2}{\|\mathbf{s}\|_2^2} \right) \quad (12)$$

$$\text{EQMN}_{\mathbb{H}} = 10 \log_{10} \left(\frac{\|\hat{\mathbb{H}} - \mathbb{H}\|_F^2}{\|\mathbb{H}\|_F^2} \right) \quad (13)$$

où $\|\cdot\|_F$ désigne la norme de Frobenius, $\hat{\mathbf{s}} = [\hat{s}(1) \cdots \hat{s}(N)]^T$ et $\hat{\mathbb{H}}$ étant le vecteur de sortie et le noyau (linéaire, quadratique ou cubique) reconstruits à partir des paramètres estimés à l'aide de N données entrée-sortie. L'EQMN sur la sortie est tracée en fonction du nombre d'itérations pour 5 valeurs de RSB différentes (Fig 3), et en fonction du RSB (Fig 4), pour différentes valeurs des rangs d'approximation $R_1 = R_2 = R = 1, 2, 3$. D'autre part, les EQMN sur les noyaux linéaire, quadratique et cubique estimés sont tracées sur la Fig. 5

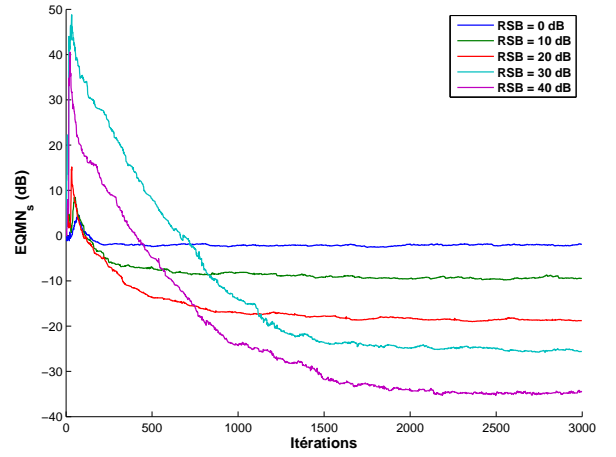


FIG. 3 – EQMN_s sur la sortie en fonction des itérations

A partir de ces résultats de simulations, nous pouvons conclure que, comme attendu, l'EQMN sur la sortie diminue lorsque le RSB augmente. A partir de la Fig. 4, on constate que l'EQMN sur la sortie augmente lorsque le rang R est sous-estimé, et cela d'autant plus que le RSB augmente. Par contre, une surestimation de R ne se traduit pas par une modification sensible de l'EQMN sur la sortie. La convergence de l'algorithme dépend de manière assez sensible du RSB. D'autre part, on constate que l'estimation des noyaux quadratique et cubique est nettement meilleure que celle du noyau linéaire.

6 Conclusion

Dans cet article, nous avons présenté une nouvelle formulation des modèles de Volterra basée sur une décomposition PA-

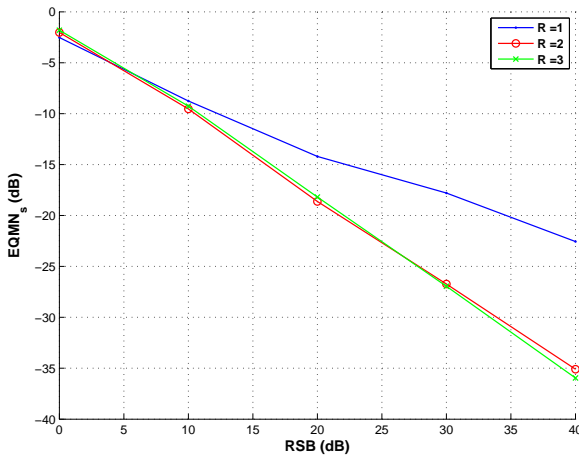


FIG. 4 – EQMN_s sur la sortie en fonction du RSB pour différentes valeurs de $R_2 = R_3 = R$

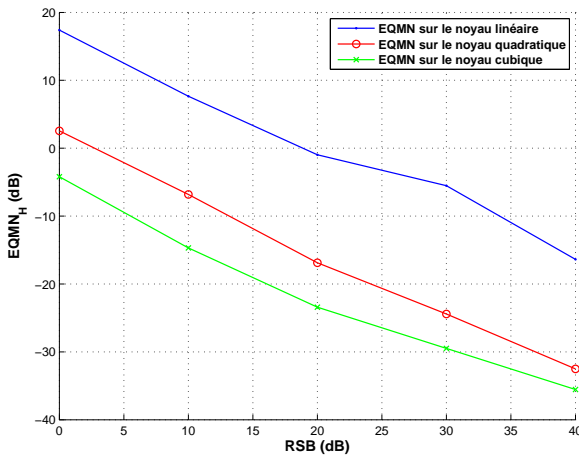


FIG. 5 – EQMN_H sur les noyaux en fonction du RSB

RAFAC de leurs noyaux considérés comme des tenseurs symétriques. Les modèles de Volterra-PARAFAC ainsi obtenus peuvent être vus comme la mise en parallèle de modèles de Wiener. L'utilisation de la décomposition PARAFAC permet de réduire de façon très significative la complexité paramétrique des modèles de Volterra comme nous le montrons dans [11]. Le filtre de Kalman étendu appliqué à ces modèles de Volterra-PARAFAC fournit un algorithme d'estimation paramétrique récursif, facile à mettre en oeuvre et efficace comme l'illustrent les résultats de simulations présentés. Parmi les perspectives de ce travail, nous envisageons de développer des méthodes d'identification structurelle en vue de déterminer la mémoire et le rang des noyaux, ce qui reste un problème ouvert concernant la recherche du rang d'un tenseur. L'utilisation d'autres décompositions tensorielles est aussi à l'étude pour réduire la

complexité des modèles de Volterra, et une comparaison avec des méthodes basées sur des développement des noyaux sur des bases de fonctions orthonormales est en cours.

Références

- [1] O. Agazzi, D. Messerschmitt, and D. Hodges. *IEEE Tr. Commun.*, 30(11) :2421–2433, Nov 1982.
- [2] S. Benedetto and E. Biglieri. *IEEE J. Select. Areas Commun.*, 1(1) :57–62, Jan 1983.
- [3] S. Benedetto, E. Biglieri, and R. Daffara. *IEEE Tr. Aerosp. Electron. Syst.*, 15(4) :494–507, July 1979.
- [4] S. Boyd and L. Chua. *IEEE Trans. Circ. and Syst. CAS*, 32(11) :1150–1161, Nov 1985.
- [5] R.J. Campello, G. Favier, and W.C. Amaral. *Automatica*, 40(5) :815–822, 2004.
- [6] Chi-Hao Cheng and E. J. Powers. *IEEE Tr. Signal Proc.*, 49(1) :147–163, Jan. 2001.
- [7] K. H. Chon, N. H. Holstein-Rathlou, D. J. Marsh, and V. Z. Marmarelis. *IEEE Tr. Neural Networks*, 9(3) :430–435, May 1998.
- [8] F.J. Doyle III, R.K. Pearson, and B.A. Ogunnaike. *Springer-Verlag*, 2002.
- [9] G.A. Dumont and Y. Fu. *Int.J. of Adaptive Control and Signal Proc.*, 7 :367–382, 1993.
- [10] C. Eun and E. J. Powers. *IEEE Tr. Signal Proc.*, 45(1) :223–227, Jan. 1997.
- [11] G. Favier and T. Bouilloc. *EUSIPCO, Glasgow, Scotland*, 2009.
- [12] R. Harshman. *UCLA working papers in phonetics*, 16 :1–84, 1970.
- [13] Y. Kajikawa. *EUSIPCO, Lausanne, Switzerland*, 2008.
- [14] A. Khouaja and G. Favier. *EUSIPCO, Wien, Austria*, pages 1903–1906, 2004.
- [15] A. Kibangou, G. Favier, and M. Hassani. *Signal Process.*, 85 :2371–2385, 2005.
- [16] V. Z. Marmarelis. *Annals of Biomedical Eng.*, 21 :573–589, 1993.
- [17] A.J. Redfern and G.T. Zhou. *Signal Process.*, 81 :1041–1052, 2001.
- [18] N. Wiener. *MIT Press, Cambridge, Massachusetts, and Wiley, New-York*, 1958.
- [19] M. Zeller and W. Kellerman. *EUSIPCO, Poznan, Poland*, pages 1605–1609, 2007.