



## UNE CLASSE D'ALGORITHMES D'ADAPTATION DE FILTRES RECURSIFS PRESENTANT DES PROPRIETES STABILISANTES

Christophe VIGNAT, Sylvie MARCOS et Odile MACCHI

GRECO TdSI ET LABORATOIRE DES SIGNAUX ET SYSTEMES

Plateau du Moulon, 91192 Gif sur Yvette, France

**Résumé.** L'application rigoureuse de la méthode du gradient au domaine du filtrage récursif adaptatif est irréalisable en pratique. Nous introduisons donc une simplification de cette méthode, consistant à estimer par un filtrage sur horizon temporel fixe -donc transverse- des quantités -l'erreur et ses gradients- théoriquement récurrentes. Nous aboutissons à une famille d'algorithmes intégrant certains algorithmes classiques. Puis nous montrons expérimentalement que plus l'estimation de ces variables est précise, meilleur est le comportement de l'algorithme en termes de stabilité et de rapidité de convergence.

**Abstract.** A rigorous application of the gradient optimization method to the adaptive recursive filtering field is not feasible in effect. We thus introduce a simplification of this method which consists in estimating the error and its gradients, which are theoretically recursive quantities, through a transversal filtering. This results in a class of algorithms which is proved to contain some classical algorithms. Then, a better accuracy of the estimation of these variables is experimentally shown to improve the algorithm behaviour in terms of stability and convergence rate.

### I INTRODUCTION

De nombreux domaines du traitement du signal, tels l'égalisation de canal, la prédiction temporelle ou l'annulation d'écho, justifient l'utilisation de filtres récursifs. De plus, la faculté d'adaptativité se révèle essentielle lorsque l'environnement non-stationnaire impose l'acquisition, puis la poursuite continue d'un état optimal. Pour cela, les paramètres du système sont modifiés de façon récursive à l'aide d'un algorithme d'optimisation.

Parmi les nombreuses procédures d'optimisation existantes, celle du gradient est souvent utilisée, notamment de par sa faible exigence en capacités de calculs. Cependant, aux problèmes spécifiques du filtrage récursif (non-unicité du filtre optimal, risque d'instabilité) s'ajoute la difficulté de l'évaluation des quantités -erreur et gradients d'erreur- nécessaires à l'actualisation, par un algorithme de gradient, des paramètres du filtre. Idéalement, chacune de ces quantités est obtenue, à l'instant (n), par filtrage récursif-donc par traitement sur un horizon temporel de longueur (n)- des entrées et des sorties retardées. Comme le vecteur paramètre nécessaire à ce filtrage est réactualisé en permanence, un nouveau filtrage sur horizon temporel de longueur (n+1) est nécessaire à l'instant suivant. La charge de calcul croît donc avec le temps, ce qui n'est pas admissible dans le cadre d'une utilisation en temps réel du dispositif. Aussi, nous proposons une estimation de ces quantités en remplaçant le filtre récursif qui les génère par un filtre transverse, composé de copies du filtre initial, menant à chaque instant (n) à un calcul sur horizon temporel de longueurs fixes L pour l'erreur et L' pour les gradients. Nous aboutissons ainsi à une famille d'algorithmes, paramétrés par les valeurs L et L', que nous appelons FMRLMS (L,L') - Finite Memory Recursive Least Mean Squares Algorithm [1]. Puis nous montrons que certains algorithmes bien connus du

filtrage récursif adaptatif tels le L.M.S. récursif, le R.P.E. [2]... sont des éléments de la famille d'algorithmes FMRLMS (L,L').

Dans une seconde partie, les comportements de ces algorithmes sont examinés expérimentalement, en termes de stabilité et de qualité de convergence. Notamment, nous mettons en évidence le phénomène d'autostabilisation, déjà étudié dans le cas du L.M.S. récursif [3]. L'influence des longueurs L et L' des horizons choisis est observée sur des simulations, dans le cas d'un prédicteur AR(1).

### II LA METHODE DU GRADIENT EN FILTRAGE RECURSIF ADAPTATIF

#### 1- La procédure d'optimisation du gradient

Soit  $y(n)$  la sortie d'un filtre constitué d'une partie transverse  $(b_j)$ ,  $j \in [1, N_b]$ , et d'une partie récursive  $(a_i)$ ,  $i \in [1, N_a]$ , et dont  $x(n)$  est l'entrée.

$$y(n) = \sum_{i=1}^{N_a} a_i y(n-i) + \sum_{j=1}^{N_b} b_j x(n-j) \quad (1)$$

Dans un contexte adaptatif, une réponse désirée  $d(n)$  est disponible, et il s'agit de réactualiser récursivement et continuellement les paramètres  $(a_i)$  et  $(b_j)$  afin de rendre la sortie  $y(n)$  aussi proche que possible, au sens d'un critère à définir, de cette réponse désirée  $d(n)$ . Nous considérons par la suite le critère L.M.S.:

$$J(n) \triangleq \frac{1}{2} E[e(n)^2] \quad (2)$$

où

$$e(n) \triangleq d(n) - y(n) \quad (3)$$

$E(\cdot)$  représente ici l'opérateur de moyennage statistique.



L'adaptation suivant l'algorithme du gradient consiste en une modification récursive - à savoir, consécutive à chaque nouvelle donnée  $x(n)$  - des paramètres  $(a_i)$  et  $(b_j)$  suivant:

$$\begin{cases} a_i(n+1) = a_i(n) - \mu \cdot \text{grad}_{a_i} [J(n)] | a_i(n), b_j(n) \\ b_j(n+1) = b_j(n) - \mu \cdot \text{grad}_{b_j} [J(n)] | a_i(n), b_j(n) \end{cases} \quad (4)$$

où  $\mu > 0$  est le pas d'adaptation.

Dans la plupart des applications, peu d'informations sont disponibles a priori sur la moyenne d'ensemble du gradient du critère. Une première approximation consiste donc à supprimer l'opérateur de moyennage, de sorte que (4) se réduit à:

$$\begin{cases} a_i(n+1) = a_i(n) - \mu \cdot e(n) \cdot \frac{\partial e(n)}{\partial a_i} | a_i(n), b_j(n) \\ b_j(n+1) = b_j(n) - \mu \cdot e(n) \cdot \frac{\partial e(n)}{\partial b_j} | a_i(n), b_j(n) \end{cases} \quad (5)$$

## 2 - La mémoire infinie de l'erreur et des gradients d'erreur

La difficulté des formules (5) réside dans l'évaluation de l'incrément à chaque instant  $(n)$  à l'aide des paramètres courants  $(a_i(n))$ ,  $(b_j(n))$ . En effet, l'erreur est récursive du fait que  $y(n)$  l'est. Quant aux gradients d'erreur, il est aisé de calculer :

$$\begin{cases} \alpha_k(n) \triangleq - \frac{\partial e(n)}{\partial a_k} = y(n-k) + \sum_{i=1}^{N_a} a_i \cdot \alpha_k(n-i) \\ \beta_l(n) \triangleq - \frac{\partial e(n)}{\partial b_l} = x(n-l) + \sum_{i=1}^{N_b} a_i \cdot \beta_l(n-i) \end{cases} \quad (6)$$

Ainsi,  $\alpha_k(n)$  et  $\beta_l(n)$  sont les réponses d'un filtre dont la partie récursive est la même qu'en (1). Or, comme il apparaît dans (5), la technique du gradient requiert un calcul de ces quantités avec les paramètres  $(a_i)$  et  $(b_j)$  fixés à leur dernière valeur disponible. Ceci signifie qu'à chaque nouvel instant  $(n)$ , disposant de paramètres  $(a_i)$  et  $(b_j)$ , l'erreur et ses gradients devraient être calculés par filtrage récursif, à savoir sur un horizon temporel théoriquement infini - en pratique, en remontant jusqu'à l'instant  $n=0$  de mise en œuvre du filtre - à l'aide des paramètres  $(a_i(n))$  et  $(b_j(n))$  maintenus fixés.

Par exemple, concernant l'erreur, il nous faudrait calculer, pour appliquer rigoureusement la méthode du gradient, la quantité  $y(n)$  de la façon suivante:

$$y(n) = \sum_{i=1}^{N_a} a_i(n) \cdot y(n-i) + \sum_{j=1}^{N_b} b_j(n) \cdot x(n-j) \quad (8)$$

sachant que tous les  $y(n-i)$  auront aussi été calculés avec  $a_i(n)$  et  $b_j(n)$  selon:

$$\begin{aligned} y(n-1) &= \sum_{i=1}^{N_a} a_i(n) \cdot y(n-i-1) + \sum_{j=1}^{N_b} b_j(n) \cdot x(n-j-1) \\ y(n-2) &= \sum_{i=1}^{N_a} a_i(n) \cdot y(n-i-2) + \sum_{j=1}^{N_b} b_j(n) \cdot x(n-j-2) \\ &\dots \\ y(N_a) &= \sum_{i=1}^{N_a} a_i(n) \cdot y(N_a-i) + \sum_{j=1}^{N_b} b_j(n) \cdot x(N_a-j) \end{aligned} \quad (9)$$

Le même raisonnement s'applique aux quantités  $\alpha_k(n)$  et  $\beta_l(n)$ . Aussi, la charge de calcul requise pour l'actualisation des paramètres à l'instant  $n$  augmenterait avec  $n$ , ce qui n'est pas acceptable dans le cadre d'une utilisation en temps réel du filtre récursif.

## 3 - De la mémoire infinie à la mémoire finie: la famille d'algorithmes FMRLMS(L,L')

Devant l'impossibilité pratique d'un calcul exact de l'erreur et de ses gradients, nous proposons d'estimer ces variables en tronquant la mémoire du filtre récursif.

Ceci revient à tronquer le système (8) (9), ne gardant que les  $(L+1)$  premières valeurs  $y(n)$ ,  $y(n-1), \dots, y(n-L)$  qui, étant calculées à  $a_i=a_i(n)$ ,  $b_j=b_j(n)$ , seront désormais notées:

$$y^{(n)}(n), y^{(n)}(n-1), \dots, y^{(n)}(n-L)$$

L'indice supérieur  $(n)$  indique les valeurs des paramètres -ici  $a(n)$  et  $b(n)$ - ayant servi à les calculer. Ainsi, nous remplacerons (8) et (9) par

$$y^{(n)}(n) = \sum_{i=1}^{N_a} a_i(n) \cdot y^{(n)}(n-i) + \sum_{j=1}^{N_b} b_j(n) \cdot x(n-j) \quad (10)$$

$$y^{(n)}(n-1) = \sum_{i=1}^{N_a} a_i(n) \cdot y^{(n)}(n-i-1) + \sum_{j=1}^{N_b} b_j(n) \cdot x(n-j-1)$$

...

$$y^{(n)}(n-L) = \sum_{i=1}^{N_a} a_i(n) \cdot \tilde{y}^{(n)}(n-i-L) + \sum_{j=1}^{N_b} b_j(n) \cdot x(n-j-L)$$

Les quantités  $\tilde{y}^{(n)}(n-i-L)$  ( $i \in [1, N_a]$ ) sont des valeurs d'initialisation du système (10), dont le choix sera discuté ci-dessous. La même méthode d'estimation, mais cette fois sur un horizon temporel de longueur  $L'$ , est proposée concernant les gradients d'erreur, conduisant aux systèmes (11) et (12) suivants:

$$\begin{cases} \alpha_k^{(n)}(n) = y^{(n)}(n-k) + \sum_{i=1}^{N_a} a_i(n) \cdot \alpha_k^{(n)}(n-i) \\ \alpha_k^{(n)}(n-1) = y^{(n)}(n-k-1) + \sum_{i=1}^{N_a} a_i(n) \cdot \alpha_k^{(n)}(n-i-1) \\ \dots \\ \alpha_k^{(n)}(n-L') = y^{(n)}(n-k-L') + \sum_{i=1}^{N_a} a_i(n) \cdot \tilde{\alpha}_k^{(n)}(n-i-L') \end{cases} \quad (11)$$

$$\begin{cases} \beta_l^{(n)}(n) = x(n-l) + \sum_{i=1}^{N_b} a_i(n) \cdot \beta_l^{(n)}(n-i) \\ \beta_l^{(n)}(n-1) = x(n-l-1) + \sum_{i=1}^{N_b} a_i(n) \cdot \beta_l^{(n)}(n-i-1) \\ \dots \\ \beta_l^{(n)}(n-L') = x(n-l-L') + \sum_{i=1}^{N_b} a_i(n) \cdot \tilde{\beta}_l^{(n)}(n-i-L') \end{cases} \quad (12)$$

Là encore apparaissent des variables d'initialisation  $\tilde{\alpha}_k^{(n)}(n-i-L')$  et  $\tilde{\beta}_l^{(n)}(n-i-L')$  dont le choix sera discuté ci-dessous.

Finalement, l'algorithme FMRLMS(L,L') consiste en les étapes suivantes:

**a-** à l'instant  $(n)$ , on dispose des quantités  $y^{(n-1)}(n-i-1)$ ,  $\alpha_k^{(n-1)}(n-i-1)$ ,  $\beta_l^{(n-1)}(n-i-1)$   $i, k \in [1, N_a]$   $l \in [1, N_b]$

**b-** une entrée  $x(n)$  est disponible, permettant de calculer  $y^{(n)}(n-i)$  à l'aide de (10)  $i \in [1, N_a]$   $\alpha_k^{(n)}(n-i)$  à l'aide de (11)  $k \in [1, N_a]$   $\beta_l^{(n)}(n-i)$  à l'aide de (12)  $l \in [1, N_b]$

**c-** les nouvelles valeurs  $a_i(n+1)$ ,  $b_j(n+1)$  sont ensuite estimées, selon (5,6,7), par:

$$\begin{cases} \Delta a_i(n) = \mu (d(n)-y^{(n)}(n)).\alpha_i^{(n)}(n) & (13) \\ \Delta b_j(n) = \mu (d(n)-y^{(n)}(n)).\beta_j^{(n)}(n) & (14) \end{cases}$$

4- Initialisation des systèmes (10), (11) et (12)

Il s'agit ici de choisir des valeurs pour les variables  $\tilde{y}^{(n)}(n-i-L)$ ,  $\tilde{\alpha}_k^{(n)}(n-i-L')$  et  $\tilde{\beta}_l^{(n)}(n-i-L')$  apparaissant dans les systèmes (10), (11) et (12). Deux approches sont classiques en filtrage adaptatif récursif [2]:

a - approche de type equation error:

$$\begin{cases} \tilde{y}^{(n)}(n-i-L) = d(n-i-L) \\ \tilde{\alpha}_k^{(n)}(n-i-L') = 0 \\ \tilde{\beta}_l^{(n)}(n-i-L') = 0 \end{cases} \quad (15)$$

b - approche de type output error:

$$\begin{cases} \tilde{y}^{(n)}(n-i-L) = y^{(n-1)}(n-i-L) \\ \tilde{\alpha}_k^{(n)}(n-i-L') = 0 \\ \tilde{\beta}_l^{(n)}(n-i-L') = 0 \end{cases} \quad (16)$$

Nous montrons dans [1] que les algorithmes classiques du filtrage récursif adaptatif sont des cas particuliers (correspondant à des choix précis de L et L') de la classe FMRLMS(L,L').

5- Une interprétation plus générale de la famille des algorithmes FMRLMS(L,L')

L'idée du principe des algorithmes FMRLMS est apparue lors de l'étude, en collaboration avec G. Dreyfus et L. Personnaz du Laboratoire d'Electronique de l'E.S.P.C.I., de l'apprentissage de réseaux neuronaux temporellement bouclés [4]. Considérant ainsi le cas plus général d'un système quelconque (non-nécessairement linéaire par rapport à ses entrées) descriptible par une représentation d'état, comme illustré Fig.1, ce système est représentable de façon équivalente comme indiqué Fig.2.

L'adaptation du système consiste à trouver un jeu de paramètres ( $\Sigma$ ) minimisant un certain critère. La troncature proposée pour le FMRLMS revient à ne considérer, à l'instant (n), qu'une cascade de (L+1) cellules élémentaires. Ainsi, les différentes variables  $y^{(n)}(n-i)$  ne sont autres que les sorties des différentes cellules composant cette cascade. Quant à la valeur d'initialisation proposée en (16) pour  $\tilde{u}^{(n)}(n-L)$  -pour un filtre linéaire, les variables u et y sont confondues- elle découle naturellement de la considération des deux cascades aux instants consécutifs (n) et (n+1), comme illustré Fig. 3.

III ETUDE DU COMPORTEMENT DES FMRLMS(L,L')

1 - Stabilité des filtres récursifs : le phénomène d'autostabilisation

Par la suite, nous considérons la notion de stabilité au sens B.I.B.O. (Bounded Input Bounded Output). Dans le cas d'un filtre récursif non-adaptatif, cette stabilité est assurée par l'appartenance des coefficients à un domaine de stabilité bien défini. Or, il convient de remarquer que, dans un contexte adaptatif, les coefficients du filtre peuvent parfaitement quitter momentanément ce même domaine sans pour autant que la sortie n'explose. Ce type d'instabilité, appelé **instabilité locale** du filtre, est rendu possible par le fait que la sortie considérée est celle d'un système global composé non seulement du filtre linéaire, mais aussi du processus d'adaptation non-linéaire, rendant ainsi cette sortie non-linéaire par rapport aux entrées.

Le phénomène d'autostabilisation, révélé dans [3], apparaît par exemple dans le cas de l'adaptation d'un filtre récursif par

l'algorithme L.M.S. récursif. La courbe 1 montre la trajectoire du paramètre  $a_1$  d'un AR(1) à entrée constante adapté par cet algorithme. L'entrée vaut  $s=0.5$  tandis que  $\mu=0.02$ . On y voit le paramètre  $a_1$  sortir quasi-périodiquement du domaine de stabilité ( $|a_1|<1$ ) et, ayant culminé à une valeur  $a_1 \approx 1.3$ , réintégrer rapidement ce domaine avant d'en ressortir progressivement pour culminer à nouveau à la même valeur, et ainsi de suite.

2 - Autostabilisation dans le cas des FMRLMS: résultats expérimentaux

Les résultats expérimentaux présentés ici concernent le filtre (1) avec  $N_a=N_b=1$ ,  $b_1=1$ ,  $x(n)=s=0.5$  et  $\mu=0.02$ .

a - Influence de l'horizon de calcul L de l'erreur

Dans ce sous-paragraphe, aucun gradient n'est estimé (L' est inexistant). On note FMRLMS(L,\*) cet algorithme. La courbe 1 montre les trajectoires du paramètre  $a_1$  pour  $L \in \{0,1,3,5,15\}$ . Deux phénomènes apparaissent:

- dans tous les cas, ces algorithmes montrent la propriété d'autostabilisation.
- de plus, tandis que pour le L.M.S. étendu (L=0), le paramètre culmine à une valeur fixe, l'augmentation de L fait que  $a_1$  se rapproche de plus en plus rapidement de la valeur limite  $a_1=1$ .

Remarquons que la solution atteinte à la convergence est le cycle d'ordre 2 suivant:

$$\begin{matrix} (n) & & (n+1) & & (n+2) \\ \left[ \begin{matrix} e=s \\ a_1=1 \end{matrix} \right] & \rightarrow & \left[ \begin{matrix} e=0 \\ a_1=1 \end{matrix} \right] & \rightarrow & \left[ \begin{matrix} e=s \\ a_1=1 \end{matrix} \right] & \rightarrow \dots \end{matrix} \quad (17)$$

Ainsi, tout en conservant la propriété d'autostabilisation, l'augmentation de l'horizon de calcul de l'erreur (même sans estimation du gradient) accroît la qualité de convergence du paramètre  $a_1$ .

b - Influence de l'horizon de calcul L' des gradients de l'erreur

Partant d'une valeur de L fixée, deux cas ont été observés :

- \*  $L=0$ . Dans ce cas, les FMRLMS(0,L') (L'≥0) ne présentent pas la propriété d'autostabilisation, et mènent toujours à un comportement divergent du paramètre  $a_1$  (instabilité globale).
- \*  $L \neq 0$ . Dans ce cas, les FMRLMS(L,L') présentent tous la propriété d'autostabilisation, comme l'illustre la courbe 2 dans le cas L=5 et  $\mu=0.05$ . Il est à remarquer que la nature de la convergence est, dans ce cas, différente: au cycle limite d'ordre 2 décrit en (17) se substitue, dès que L'≥0, un autre cycle limite d'ordre 2:

$$\begin{matrix} (n) & & (n+1) & & (n+2) \\ \left[ \begin{matrix} e=0.22 \\ a_1=0.97 \end{matrix} \right] & \rightarrow & \left[ \begin{matrix} e=0.29 \\ a_1=0.95 \end{matrix} \right] & \rightarrow & \left[ \begin{matrix} e=0.22 \\ a_1=0.97 \end{matrix} \right] \rightarrow \dots \end{matrix} \quad (18)$$

De plus, la convergence du paramètre  $a_1$  est moins rapide, mais aboutit à des erreurs de prédiction moyennes inférieures à celles obtenues pour L'=0: la courbe 3 montre l'amélioration sur le gain de prédiction apportée par une estimation récursive des gradients (L'≥0). Cette propriété est justement d'un grand intérêt dans certaines applications, comme par exemple le codage prédictif de la parole.

Ainsi, le calcul de l'erreur et de ses gradients sur des horizons de longueurs non-nulles apporte ici un avantage indéniable sur la qualité de convergence du processus d'adaptation.



**IV CONCLUSION**

Les simulations présentées dans cet article montrent l'intérêt d'une évaluation la plus précise possible de l'erreur et de ses gradients nécessaires à l'actualisation des paramètres du filtre. Il s'agit ensuite, selon l'application considérée, de trouver un compromis entre le temps de calcul dédié à ces approximations d'une part, et la précision et la rapidité de convergence souhaitées pour les paramètres du filtre d'autre part. De plus, une estimation précise des gradients en même temps que des erreurs conduit, comme on l'a vu, à une amélioration non négligeable du gain de prédiction.

Il faut enfin noter que cette méthode, comme nous l'avons souligné, est parfaitement applicable à des systèmes non-linéaires, laissant entrevoir des applications par exemple dans le domaine du codage de la parole par prédiction et quantification, ou encore dans celui du traitement du signal à l'aide de réseaux de neurones récurrents.

**Bibliographie**

- [1] S. Marcos, O. Macchi, C. Vignat et al. "A unified framework for gradient algorithms in adaptive filters and in learning neural networks", to appear in IJCTA.
- [2]: J.J. Shynk, "Adaptive I.I.R. filtering", I.E.E.E. A.S.S.P. Magazine, p.4-21, 1989.
- [3]: M. Jaïdane, O. Macchi, "Stability of adaptive recursive filters", Proc. ICASSP, New-York, p.1503-1505, 1988.
- [4]: G. Dreyfus et al., "Neural network training schemes for non-linear adaptive filtering and modelling", IJCNN 91, Seattle.

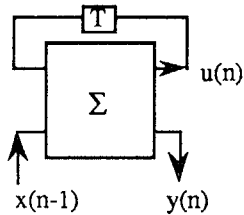


Fig. 1: le dispositif bouclé sous sa représentation d'état

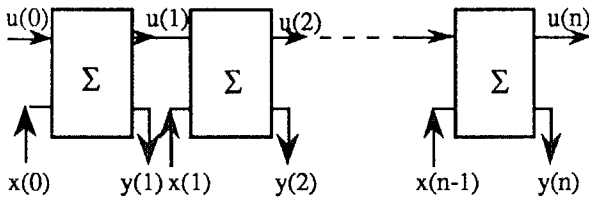


Fig. 2: le même dispositif sous sa représentation développée

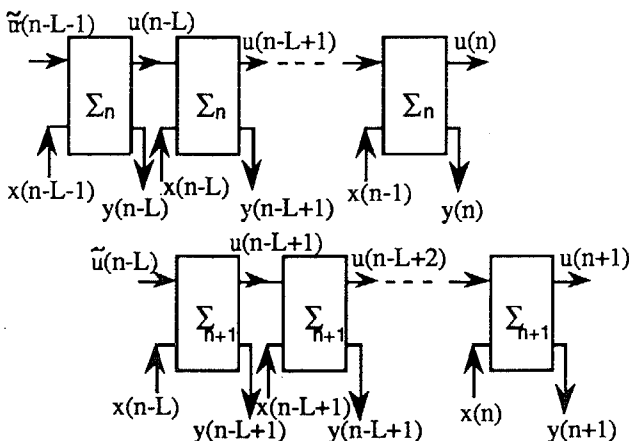
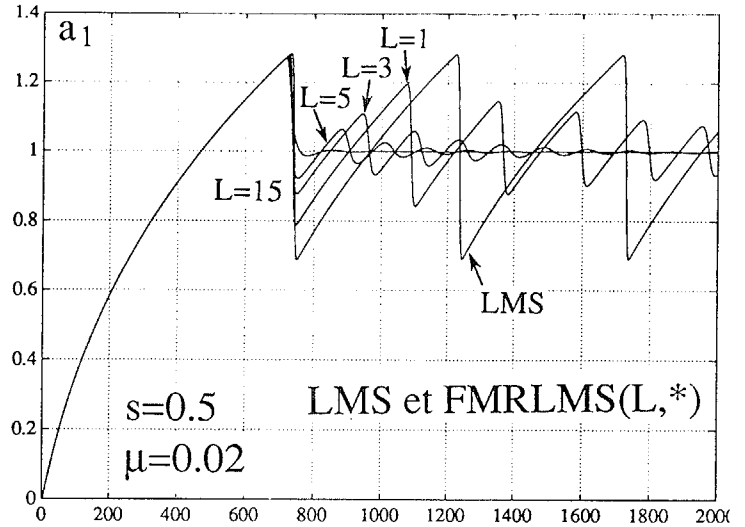
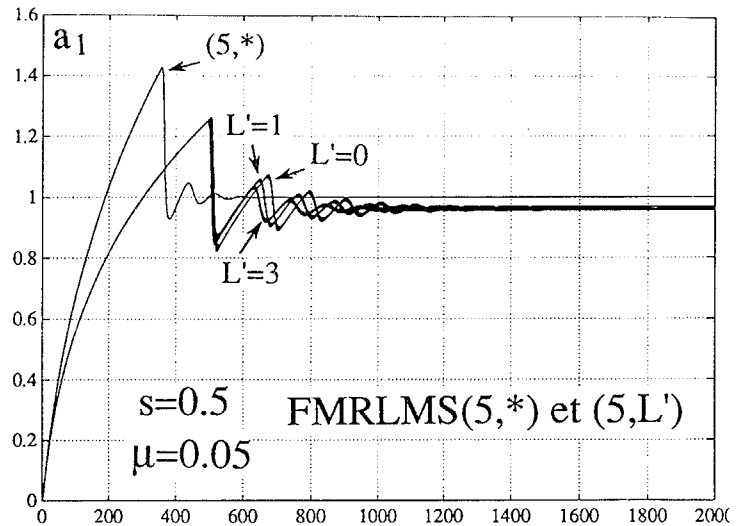


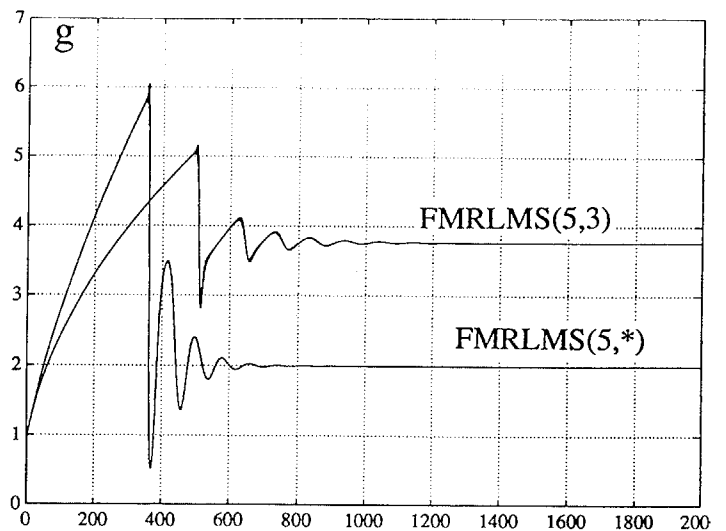
Fig. 3: le même dispositif sous sa représentation tronquée, représenté à deux instants consécutifs (n) et (n+1)



Courbe 1



Courbe 2



Courbe 3