

# Un Algorithme PAST Orthogonal Rapide Pour l'Estimation et la Poursuite de Sous-Espaces

Karim Abed-Meraim<sup>1</sup>, Ammar Chkeif<sup>1</sup>, Yingbo Hua<sup>2</sup>

<sup>1</sup>ENST/TSI

46 rue Barrault 75634, Paris Cedex 13 France

<sup>2</sup>Elec. Eng. Dept. The University of Melbourne  
Parkville, Victoria, 3052 Australia

abed@tsi.enst.fr, chkeif@tsi.enst.fr

yhua@ee.mu.oz.au

**Résumé** – Le problème d'estimation et de poursuite de sous-espaces joue un rôle très important dans une variété d'applications modernes de traitement du signal. Cet article élabore une version orthogonale de l'algorithme PAST (Projection Approximation Subspace Tracking) pour l'évaluation et la poursuite rapide du sous-espace principal ou/et des composantes principales d'une séquence de vecteurs aléatoires. L'algorithme PAST orthogonal (OPAST) converge au moins aussi rapidement que le PAST. De plus, l'algorithme OPAST garantit l'orthogonalité de la matrice de poids à chaque itération. Nous proposons une implémentation rapide de l'algorithme OPAST en  $O(np)$  opérations par itération où  $n$  est la taille du vecteur d'observation et  $p$  est la dimension du sous-espace principal.

**Abstract** – Subspace estimation and tracking play an important role in a variety of modern signal processing applications. This paper elaborates on an orthogonal version of the PAST (projection approximation and subspace tracking) algorithm for fast estimation and tracking of principal subspace or/and principal components of a vector sequence. The orthogonal PAST (OPAST) algorithm has a faster convergence rate than the PAST algorithm. Moreover, the OPAST algorithm guarantees the orthogonality of the weight matrix at each iteration. Like the PAST algorithm, the OPAST algorithm can be implemented with only  $O(np)$  flops where  $n$  is the dimension of the vector sequence and  $p$  is the dimension of the principal subspace.

## 1 Introduction

Les techniques sous-espaces jouent un rôle fondamental en estimation statistique et en traitement d'antennes. En effet, ces techniques sont largement utilisées dans des applications telle que la compression de données, l'identification de système, le filtrage, l'estimation de paramètres, et la reconnaissance des formes. Dans les quelques années passées, nombreux algorithmes ont été proposés pour l'estimation et la poursuite du sous-espace principal, e.g., [1]-[9]. En particulier, la méthode Oja [2] a reçu une considération spéciale et a été utilisée dans certaines applications réelles [3]. Il a été établi dans [4] que la méthode Oja peut être vue comme une technique de gradient approximée pour la minimisation d'une certaine fonction d'erreur quadratique moyenne (EQM). Suite à cette méthode de gradient approximée, d'autres méthodes plus rapide d'estimation et de poursuite du sous-espace principal ont été développées. Parmi les méthodes les plus robustes et les plus efficaces, nous trouvons la méthode PAST (Projection Approximation Subspace Tracking) proposée dans [4]. La méthode PAST utilise la même fonction d'erreur quadratique moyenne que la méthode Oja plus une approximation simplificatrice dite 'approximation de projection'. La vitesse de convergence de la méthode PAST est en générale largement supérieure à celle de la méthode Oja. Aussi, dans presque toutes les situations la méthode

PAST converge vers une matrice orthogonale dont les vecteurs colonnes génèrent le sous-espace principal. Cependant dans certains cas particuliers, peut avoir un comportement oscillatoire et ne pas converger (voir [9] pour plus de détails). Pour résoudre ce problème (i.e., assurer la convergence globale de la méthode) et, plus important, pour garantir l'orthogonalité de la matrice de poids à chaque itération, nous présentons ici une nouvelle méthode PAST orthogonale (algorithme OPAST). L'algorithme OPAST s'implémente avec une complexité linéaire, i.e.,  $4np + O(p^2)$  flops au lieu de  $3np + O(p^2)$  flops pour l'algorithme PAST. Aussi, l'algorithme OPAST converge aussi rapidement (parfois même plus rapidement comme observé sur la Figure 2) que l'algorithme PAST original.

## 2 L'algorithme OPAST

### 2.1 Rappel sur le PAST

Soit  $\{\mathbf{r}(k)\}$  une séquence de vecteurs aléatoires  $n \times 1$  de matrice de covariance  $\mathbf{C} = E(\mathbf{r}(k)\mathbf{r}(k)^H)$ . Considérons le problème d'estimation du sous-espace principal engendré par  $\{\mathbf{r}(k)\}$ , de dimension  $p < n$ , supposé coïncider avec le sous-espace engendré par les  $p$  vecteurs propres principaux de la matrice de covariance  $\mathbf{C}$ .

Considérons la fonction scalaire d'EQM (Erreur Qua-

dratique Moyenne)

$$\begin{aligned} J(\mathbf{W}) &= E(\|\mathbf{r} - \mathbf{W}\mathbf{W}^H\mathbf{r}\|^2) \\ &= \text{Tr}(\mathbf{C} - 2\mathbf{W}^H\mathbf{C}\mathbf{W}) + \mathbf{W}^H\mathbf{C}\mathbf{W}\mathbf{W}^H\mathbf{W}(1) \end{aligned}$$

avec  $\mathbf{W} \in \mathbb{C}^{n \times p}$ . Il est montré dans [4, 5] que

- $\mathbf{W}$  est un point stationnaire de  $J(\mathbf{W})$  si et seulement si  $\mathbf{W} = \mathbf{U}_s\mathbf{Q}$ , où  $\mathbf{U}_s$  est une matrice  $n \times p$  contenant  $p$  vecteurs propres distincts de  $\mathbf{C}$  et  $\mathbf{Q}$  est une matrice  $p \times p$  unitaire.
- Tous les points stationnaires de  $J(\mathbf{W})$  sont des points selles à moins que  $\mathbf{U}_s$  contienne les  $p$  vecteurs propres dominants de  $\mathbf{C}$ . Auquel cas,  $J(\mathbf{W})$  atteint son minimum global.

La minimisation itérative de (1) mène à la forme suivante de l'algorithme PAST [7]

$$\mathbf{W}(i) = \mathbf{C}\mathbf{W}(i-1)(\mathbf{W}^H(i-1)\mathbf{C}\mathbf{W}(i-1))^{-1}$$

où  $\mathbf{C}\mathbf{W}(i-1)$  est un terme de puissance et  $(\mathbf{W}^H(i-1)\mathbf{C}\mathbf{W}(i-1))^{-1}$  un terme de normalisation. Dans des applications de poursuite, on peut simplement remplacer la matrice de covariance  $\mathbf{C}$  par sa version récursive:

$$\mathbf{C}(i) = \alpha\mathbf{C}(i-1) + \mathbf{r}(i)\mathbf{r}^H(i)$$

où  $\mathbf{C}(i-1)$  est la matrice de covariance estimée à l'itération  $i-1$ , et  $\alpha$  est un facteur d'oubli choisi entre (0, 1].

Dans [4], une implémentation rapide de l'algorithme PAST est proposée basée sur l'approximation dite 'approximation de projection', c.-à-d.,  $\mathbf{C}(i)\mathbf{W}(i) \approx \mathbf{C}(i)\mathbf{W}(i-1)$  qui est valide lorsque la matrice de poids  $\mathbf{W}(i)$  change lentement avec  $i$ . Avec cette approximation, le produit  $\mathbf{C}(i)\mathbf{W}(i-1)$  et l'inversion  $(\mathbf{W}^H(i-1)\mathbf{C}(i)\mathbf{W}(i-1))^{-1}$  peuvent être calculés en  $O(np)$  opérations (voir [4] pour plus de détails). Plus précisément, en définissant

$$\mathbf{Z}(i) \stackrel{\text{def}}{=} (\mathbf{W}^H(i-1)\mathbf{C}(i)\mathbf{W}(i-1))^{-1}, \quad (2)$$

l'algorithme PAST est donné par les équations de mise à jour suivantes:

$$\begin{aligned} \mathbf{W}(i) &= \mathbf{W}(i-1) + \mathbf{p}(i)\mathbf{q}^H(i) \\ \mathbf{q}(i) &= \frac{1}{\alpha}\mathbf{Z}(i-1)\mathbf{y}(i) \\ \mathbf{y}(i) &= \mathbf{W}^H(i-1)\mathbf{r}(i) \\ \gamma(i) &= \frac{1}{1 + \mathbf{y}^H(i)\mathbf{q}(i)} \\ \mathbf{p}(i) &= \gamma(i)(\mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)) \\ \mathbf{Z}(i) &= \frac{1}{\alpha}\mathbf{Z}(i-1) - \gamma(i)\mathbf{q}(i)\mathbf{q}^H(i) \end{aligned}$$

## 2.2 Algorithme OPAST

L'algorithme OPAST est une modification de l'algorithme PAST où la matrice de poids  $\mathbf{W}(i)$  est forcée à être orthonormale à chaque itération. Avec des notations informelles, on écrit

$$\mathbf{W}(i) := \mathbf{W}(i)(\mathbf{W}^H(i)\mathbf{W}(i))^{-1/2} \quad (3)$$

où  $(\mathbf{W}^H(i)\mathbf{W}(i))^{-1/2}$  désigne l'inverse d'une racine carrée de  $(\mathbf{W}^H(i)\mathbf{W}(i))$ . Pour calculer cette dernière, nous utilisons l'équation de mise à jour de  $\mathbf{W}(i)$ . Grâce au fait que  $\mathbf{W}(i-1)$  est maintenant une matrice orthogonale, on obtient

$$\begin{aligned} \mathbf{W}^H(i)\mathbf{W}(i) &= \mathbf{I} + \|\mathbf{p}(i)\|^2\mathbf{q}(i)\mathbf{q}^H(i) \\ &= \mathbf{I} + \mathbf{x}\mathbf{x}^H, \end{aligned} \quad (4)$$

où nous avons utilisé le fait que  $\mathbf{W}^H(i-1)\mathbf{p}(i) = \mathbf{0}$ ,  $\mathbf{I}$  étant la matrice identité, et  $\mathbf{x} \stackrel{\text{def}}{=} \|\mathbf{p}(i)\|\mathbf{q}(i)$ . Ainsi

$$(\mathbf{W}^H(i)\mathbf{W}(i))^{-1/2} = (\mathbf{I} + \mathbf{x}\mathbf{x}^H)^{-1/2}$$

or

$$\begin{aligned} (\mathbf{I} + \mathbf{x}\mathbf{x}^H)^{-1/2} &= \mathbf{I} + \frac{1}{\|\mathbf{x}\|^2} \left( \frac{1}{\sqrt{1 + \|\mathbf{x}\|^2}} - 1 \right) \mathbf{x}\mathbf{x}^H \\ &= \mathbf{I} + \tau(i)\mathbf{q}(i)\mathbf{q}^H(i), \end{aligned} \quad (5)$$

où  $\tau(i) \stackrel{\text{def}}{=} \frac{1}{\|\mathbf{q}(i)\|^2} \left( \frac{1}{\sqrt{1 + \|\mathbf{p}(i)\|^2\|\mathbf{q}(i)\|^2}} - 1 \right)$ . En substituant (5) dans (3) et en utilisant l'équation de mise à jour de  $\mathbf{W}(i)$ , on obtient finalement

$$\begin{aligned} \mathbf{W}(i) &= (\mathbf{W}(i-1) + \mathbf{p}(i)\mathbf{q}^H(i)) \times \\ &\quad (\mathbf{I} + \tau(i)\mathbf{q}(i)\mathbf{q}^H(i)) \\ &= \mathbf{W}(i-1) + \tau(i)\mathbf{W}(i-1)\mathbf{q}(i)\mathbf{q}^H(i) \\ &\quad + (1 + \tau(i)\|\mathbf{q}(i)\|^2)\mathbf{p}(i)\mathbf{q}^H(i) \\ &= \mathbf{W}(i-1) + \mathbf{p}'(i)\mathbf{q}^H(i) \end{aligned} \quad (6)$$

où  $\mathbf{p}'(i) = \tau(i)\mathbf{W}(i-1)\mathbf{q}(i) + (1 + \tau(i)\|\mathbf{q}(i)\|^2)\mathbf{p}(i)$ . L'algorithme OPAST se résume donc comme suit:

$$\begin{aligned} \mathbf{W}(i) &= \mathbf{W}(i-1) + \mathbf{p}'(i)\mathbf{q}^H(i) \\ \mathbf{q}(i) &= \frac{1}{\alpha}\mathbf{Z}(i-1)\mathbf{y}(i) \\ \mathbf{y}(i) &= \mathbf{W}^H(i-1)\mathbf{r}(i) \\ \gamma(i) &= \frac{1}{1 + \mathbf{y}^H(i)\mathbf{q}(i)} \\ \mathbf{p}(i) &= \gamma(i)(\mathbf{r}(i) - \mathbf{W}(i-1)\mathbf{y}(i)) \\ \tau(i) &= \frac{1}{\|\mathbf{q}(i)\|^2} \left( \frac{1}{\sqrt{1 + \|\mathbf{p}(i)\|^2\|\mathbf{q}(i)\|^2}} - 1 \right) \\ \mathbf{p}'(i) &= \tau(i)\mathbf{W}(i-1)\mathbf{q}(i) + (1 + \tau(i)\|\mathbf{q}(i)\|^2)\mathbf{p}(i) \\ \mathbf{Z}(i) &= \frac{1}{\alpha}\mathbf{Z}(i-1) - \gamma(i)\mathbf{q}(i)\mathbf{q}^H(i) \end{aligned}$$

## 3 Discussion

Nous discutons ici les performances de l'algorithme OPAST et établissons le lien entre l'OPAST et la méthode de puissance naturelle (Natural Power NP) développée dans [8, 9].

### 3.1 Évaluation de Performance

Ci-dessous une brève comparaison des performances du PAST et de l'OPAST. L'évaluation de performance porte sur l'erreur d'estimation, la (vitesse de) convergence, et la complexité numérique des 2 algorithmes considérés.

- **Erreur d'estimation:** En fait, nous avons 2 types d'erreur d'estimation. Il y a d'abord l'erreur sur l'estimation du sous espace qui est sensiblement la même pour les 2 algorithmes (voir les résultats de simulation). Il y a ensuite l'erreur sur l'orthogonalité de la matrice de poids. Cette erreur est nulle pour l'OPAST qui garantit l'orthogonalité de la matrice de poids à chaque itération, alors que pour le PAST nous avons seulement une convergence asymptotique de la matrice de poids vers une matrice orthogonale.
- **Convergence:** L'algorithme OPAST peut être vu comme un algorithme de puissance naturelle (voir 3.2). La convergence de la méthode OPAST découle donc naturellement de celle de la méthode NP établie dans [9]. Pour la méthode PAST la convergence de l'algorithme (bloc) n'est pas toujours garantie dans le sens où l'algorithme peut osciller indéfiniment entre deux valeurs de la matrice de poids (voir [9]). La vitesse de convergence asymptotique du PAST ainsi que celle de l'OPAST dépendent exponentiellement du rapport de la  $p$ ème et la  $(p+1)$ ème valeur propre de  $\mathbf{C}$  (voir [9]). Toutefois, pour les premières itérations de l'algorithme nous avons observé (voir Figure 2) que l'OPAST peut converger plus rapidement que le PAST selon le choix du point initial de l'algorithme.
- **Complexité:** La complexité de calcul de l'algorithme OPAST est de  $4np + O(p^2)$  flops par itération. Elle est donc légèrement supérieure à celle de l'algorithme PAST de  $3np + O(p^2)$  flops par itération.

### 3.2 Lien entre l'OPAST et la méthode NP

Dans [8, 9], une méthode de puissance naturelle (NP) pour l'estimation et la poursuite de sous espaces a été introduite. La méthode NP est une méthode itérative où la mise à jour de la matrice de poids est donnée par

$$\mathbf{W}(i) = \mathbf{C}\mathbf{W}(i-1)(\mathbf{W}(i-1)^H \mathbf{C}^2 \mathbf{W}(i-1))^{-1/2} \quad (7)$$

où la normalisation par  $(\mathbf{W}(i-1)^H \mathbf{C}^2 \mathbf{W}(i-1))^{-1/2}$  garantit l'orthogonalité de  $\mathbf{W}(i)$ . dans les applications de poursuite,  $\mathbf{C}$  est remplacée par sa version récursive  $\mathbf{C}(i)$  à l'itération  $i$ .

On montre ici que l'algorithme OPAST peut être vu comme une implémentation particulière de la méthode NP. En effet, l'équation de mise à jour de l'OPAST s'écrit

$$\begin{aligned} \mathbf{W}(i) &= \mathbf{C}(i)\mathbf{W}(i-1)\mathbf{Z}(i) \\ \mathbf{W}(i) &:= \mathbf{W}(i)(\mathbf{W}(i)^H \mathbf{W}(i))^{-1/2}. \end{aligned} \quad (8)$$

où  $\mathbf{Z}(i)$  est définie par (2). En substituant (8) dans (8), on obtient

$$\begin{aligned} \mathbf{W}(i) &= \mathbf{C}(i)\mathbf{W}(i-1)\mathbf{Z}(i) \times \\ & \quad [(\mathbf{Z}(i)(\mathbf{W}(i-1)^H \mathbf{C}^2 \mathbf{W}(i-1))\mathbf{Z}(i))]^{-1/2} \end{aligned}$$

or nous avons<sup>1</sup>

$$(\mathbf{A}\mathbf{B}\mathbf{A})^{-1/2} = \mathbf{A}^{-1}\mathbf{B}^{-1/2}$$

En appliquant cette relation avec  $\mathbf{A} = \mathbf{Z}(i)$  et  $\mathbf{B} = \mathbf{W}(i-1)^H \mathbf{C}^2 \mathbf{W}(i-1)$ , nous obtenons l'équation (7) de la méthode NP.

Il est important ici de souligner les avantages de l'algorithme OPAST par rapport à l'algorithme NP3 proposé dans [8]. L'OPAST est plus rapide que NP3 ( $4np + O(p^2)$  flops pour l'OPAST au lieu de  $8np + O(p^2)$  flops pour NP3). L'OPAST est toujours stable (du moins sur l'ensemble des simulations réalisées) alors que NP3 peut être instable (cette instabilité a été observée dans certains contextes notamment celui de l'exemple 2 de simulation). Enfin, l'expression de  $\mathbf{W}(i)$  donnée par l'OPAST est plus compacte (i.e., somme de 2 termes seulement) que celle donnée par NP3. Cette propriété est importante lorsque l'algorithme considéré est cascadié avec d'autres algorithmes adaptatives, e.g., [10, 11].

## 4 Simulations

Nous présentons dans cette section deux exemples de simulation pour illustrer les performances de l'OPAST.

**Exemple 1:** On considère ici  $p = 4$  signaux sources indépendants et à bande étroite reçus sur une antenne linéaire uniforme à  $n = 10$  capteurs. La distance entre deux capteurs voisins est égale à la moitié de la longueur d'onde. Le rapport signal sur bruit (SNR) des signaux sources est de 20dB, 20dB, 10dB, et 5dB, respectivement. Le facteur d'oubli est fixé à  $\alpha = 0.99$  et à l'initialisation  $\mathbf{W}(0)$  est une matrice  $n \times p$  orthogonale choisie aléatoirement. Nous avons réalisé 100 tirages de Monte-Carlo et à chaque tirage, les directions d'arrivées des sources (DOA) sont choisies aléatoirement suivant une loi de distribution uniforme sur  $[0 \pi]$ . Sur la figure 1, on compare l'erreur d'estimation du sous espace pour les algorithmes PAST et OPAST. L'erreur d'estimation du sous espace est définie par

$$\| \{ \mathbf{I} - \mathbf{W}(i)(\mathbf{W}(i)^H \mathbf{W}(i))^{-1} \mathbf{W}(i)^H \} \mathbf{U}_s \mathbf{U}_s^H \|^2 / p$$

où  $\mathbf{U}_s$  dénote une matrice orthogonale  $n \times p$  générant le sous espace principal de  $\mathbf{C}$ . Sur cette exemple, l'OPAST et le PAST ont (en moyenne) la même erreur d'estimation du sous espace.

**Exemple 2:** Nous considérons ici un système CDMA (code division multiple access). Le récepteur est constitué par une antenne uniforme à 3 éléments. Nous considérons un ensemble de  $p = 4$  utilisateurs utilisant des séquences d'étalement GOLD de longueur  $N = 7$  [10]. Ainsi, la dimension du vecteur d'observation est  $n = 3N = 21$ . Les SNRs et les DOAs des utilisateurs sont (8dB, 0°), (16dB, 40°), (16dB, 50°), et (16dB, -30°), respectivement. Dans ce contexte, l'estimation sous espace est utilisée pour

<sup>1</sup>On a  $(\mathbf{A}^{-1}\mathbf{B}^{-1/2})^H(\mathbf{A}\mathbf{B}\mathbf{A})(\mathbf{A}^{-1}\mathbf{B}^{-1/2}) = \mathbf{I}$  où  $\mathbf{A}$  et  $\mathbf{B}$  sont deux matrices Hermitiennes inversibles données.

la détection multi-utilisateurs en aveugle (voir [10, 12] pour plus de détails). Comme dans l'exemple précédent, le facteur d'oubli est  $\alpha = 0.99$  et  $\mathbf{W}(0)$  est une matrice orthogonale aléatoire. La figure 2 montre les erreurs d'estimation du sous espace en fonction du nombre d'itérations pour les algorithmes PAST et OPAST. Comme on peut le voir, dans cette situation l'algorithme OPAST converge plus rapidement (du moins pour sur les premières itérations) que l'algorithme PAST.

## 5 Conclusion

Nous avons élaboré une version orthogonale rapide de l'algorithme PAST. L'algorithme proposé (OPAST) garantit l'orthogonalité exacte de la matrice de poids à chaque itération. L'algorithme OPAST est stable et converge globalement (avec une vitesse de convergence exponentielle) vers la solution exacte. Nous avons établie le lien entre l'algorithme OPAST et la méthode de puissance naturelle (NP). Cependant, il est montré que l'algorithme OPAST est meilleur que la version rapide (algorithme NP3) de la méthode de puissance naturelle.

## Références

- [1] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing", *Proc. of the IEEE*, vol. 78, no. 8, pp. 1327–1343, Aug. 1990.
- [2] E. Oja, "A simplified neuron model as a principal component analyser", *J. Math. Biology*, vol. 15, pp. 267–273, 1982
- [3] S. Y. Kung, K. I. Diamantras, and J. S. Taur, "Adaptive principal component extraction (APEX) and applications", *IEEE-T-SP*, vol. 42, no. 5, pp. 1202–1217, May 1994.
- [4] B. Yang, "Projection Approximation Subspace Tracking," *IEEE-T-SP*, vol. 44, no. 1, pp. 95-107, Jan. 1995
- [5] B. Yang, "Asymptotic convergence analysis of the projection approximation subspace tracking algorithms", *Sig. Processing*, vol. 50, pp. 123–136, 1996.
- [6] C. Riou, and T. Chonavel, "Fast adaptive eigenvalue decomposition: a maximum likelihood approach," in *Proc. IEEE ICASSP*, pp. 3565–3568, 1997.
- [7] Y. Hua, T. Chen, and Y. Miao, "A unifying view of a class of subspace tracking methods", *Proc of the 1998 Symposium on Image, Speech, Signal Processing and Robotics*, Vol. 2, pp. 27-32, Hong Kong, 3-4 Sept 1998.
- [8] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim, and Y. Miao, "Natural power method for fast subspace tracking", *to appear in Proc. NNSP*, 1999.
- [9] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim, and Y. Miao, "A new look at the power method for fast subspace tracking", *submitted to IEEE-T-SP*, Jan. 1999.
- [10] A. Chkeif, K. Abed-Meraim, G. K. Kaleh, and Y. Hua, "Blind adaptive multiuser detection with use of

linear antenna arrays in CDMA systems", *submitted to IEEE-T-Comm.*, Dec. 1998.

- [11] A. Chkeif, K. Abed-Meraim, and G. Kawas Kaleh, "Adaptive Multiuser Interference Cancellation: Algorithms and Performance Analysis," *to appear in Proc. GLOBECOM*, 1999.
- [12] X. Wang and V. H. Poor, "Blind Adaptive Multiuser Detection in Multipath CDMA Channels Based on Subspace Tracking," *IEEE Trans. on Sig. Process.*, vol.46, no. 11, pp. 3030-3044, Nov. 1998.

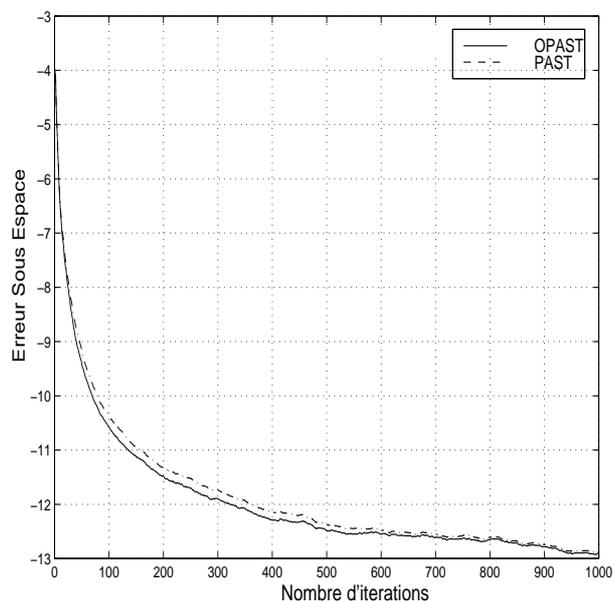


FIG. 1:

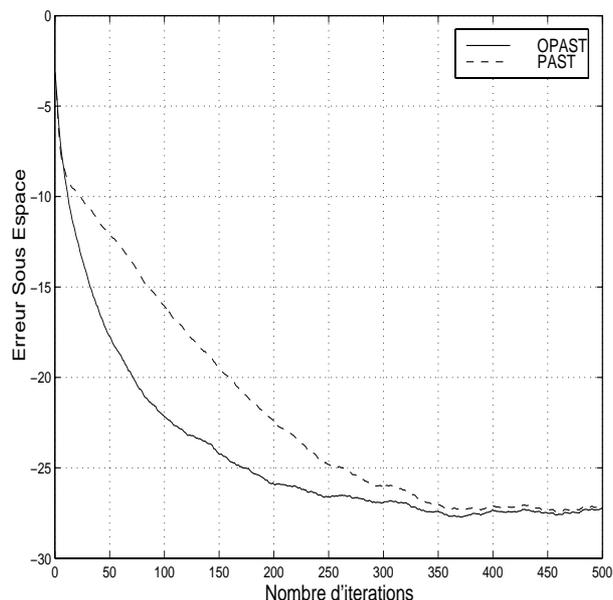


FIG. 2: