# SLIDING WINDOW ORTHONORMAL PAST ALGORITHM

*R. Badeau, K. Abed-Meraim, G. Richard and B. David*

Ecole Nationale Supérieure des Télécommunications
46 rue Barrault, 75634 Paris Cedex 13 France
rbadeau, abed, grichard, bedavid@tsi.enst.fr

**ABSTRACT**

This paper introduces an orthonormal version of the sliding-window Projection Approximation Subspace Tracker (PAST). The new algorithm guarantees the orthonormality of the signal subspace basis at each iteration. Moreover, it has the same complexity as the original PAST algorithm, and like the more computationally demanding natural power (NP) method, it satisfies a global convergence property, and reaches an excellent tracking performance.

## 1. INTRODUCTION

Subspace tracking has been widely investigated in the fields of adaptive filtering, source localization or parameter estimation. One of the various approaches proposed in the literature consists in the iterative optimization of a specific cost function involving the estimated covariance matrix of the data, in combination with a projection approximation hypothesis [1–3].

Most of these fast subspace trackers are designed for exponential forgetting windows. Indeed, this choice tends to smooth the signal variations and thus allows a low-complexity adaptation at each time step. However, it is only suitable for slowly varying signals. Conversely, algorithms based on sliding windows often have a higher complexity, but offer a faster tracking response to sudden signal changes [4, 5].

In other respects, the various subspace trackers do not have the same behavior regarding the orthonormality of the estimated signal subspace basis. The need for orthonormality only depends on the post-processing method which uses the signal subspace estimate to extract the desired signal information. For instance, in the context of DOA or frequency estimation, the MUSIC [6] and the minimum-norm [7] estimators require an orthonormal basis, whereas this is not the case of the ESPRIT algorithm [8].

Among the most robust and efficient subspace trackers is the PAST algorithm [1], which converges to an orthonormal basis spanning the signal subspace, but does not guarantee the orthonormality at each iteration. To alleviate this drawback, orthonormal versions of PAST were proposed in [9] and [3]. Both of them have the same complexity as PAST, but the latter (referred to as OPAST) additionally satisfies a global convergence property.

Following these ideas, this paper introduces an orthonormal version of the sliding window PAST algorithm, which offers the fast tracking response of a sliding window and satisfies the interesting properties of the OPAST subspace tracker.

This paper is organized as follows. Section 2 presents a sliding window version of the PAST algorithm (SW-PAST), slightly different from that proposed in [1]. Section 3 introduces the new sliding window OPAST algorithm (SW-OPAST), whose properties are discussed in section 4. Section 5 compares the performance of this algorithm to that of SW-PAST and OPAST. Finally, section 6 summarizes the main conclusions of this paper.

## 2. SLIDING WINDOW PAST

A sliding window version of the PAST algorithm was briefly presented in [1]. It consists of two successive rank-one updates of the signal subspace basis at each time step. Equivalently, the SW-PAST algorithm summarized in this section makes only one rank-two update at each time step.

The dominant subspace estimation consists in minimizing the scalar cost function

$$J(\boldsymbol{U}(t)) = \sum_{i=t-l+1}^{t} \|\boldsymbol{x}(i) - \boldsymbol{U}(t)\,\boldsymbol{U}(t)^H \boldsymbol{x}(i)\|^2$$

where $\{\boldsymbol{x}(i)\}$ is a sequence of $n \times 1$ data vectors, $l$ is the length of the sliding window, and the superscript $^H$ denotes the transpose conjugate of a matrix. B. Yang showed that the solution $\boldsymbol{U}(t) \in \mathbb{C}^{n\times r}$ (with $r < n$) was given recursively by

$$\boldsymbol{U}(t) = \boldsymbol{C}_{xx}(t)\,\boldsymbol{U}(t-1)\left(\boldsymbol{U}(t-1)^H \boldsymbol{C}_{xx}(t)\,\boldsymbol{U}(t-1)\right)^{-1} \tag{1}$$

where $\boldsymbol{C}_{xx}(t)$ is the $n \times n$ covariance estimate

$$\boldsymbol{C}_{xx}(t) = \sum_{i=t-l+1}^{t} \boldsymbol{x}(i)\boldsymbol{x}(i)^H.$$

In [1], a fast implementation is proposed based on the projection approximation $\boldsymbol{U}(t-1)^H \boldsymbol{x}(i) \simeq \boldsymbol{y}(i)$, where $\boldsymbol{y}(i) = \boldsymbol{U}(i-1)^H \boldsymbol{x}(i)$. Substituting this approximation in equation (1) yields

$$\boldsymbol{U}(t) = \boldsymbol{C}_{xy}(t)\,\boldsymbol{C}_{yy}(t)^{-1} \tag{2}$$

where the sliding window covariance estimates $\boldsymbol{C}_{yy}(t)$ and $\boldsymbol{C}_{xy}(t)$ are recursively defined by

$$\boldsymbol{C}_{yy}(t) = \boldsymbol{C}_{yy}(t-1) + \boldsymbol{Y}(t)\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\boldsymbol{Y}(t)^H \tag{3}$$

where $\boldsymbol{Y}(t) = \left[\ \boldsymbol{y}(t)\ \middle|\ \boldsymbol{y}(t-l)\ \right]$ is a $r \times 2$ matrix, and

$$\boldsymbol{C}_{xy}(t) = \boldsymbol{C}_{xy}(t-1) + \boldsymbol{X}(t)\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\boldsymbol{Y}(t)^H \tag{4}$$

where $\boldsymbol{X}(t) = \left[\ \boldsymbol{x}(t)\ \middle|\ \boldsymbol{x}(t-l)\ \right]$ is a $n \times 2$ matrix.

From now on, suppose that $\boldsymbol{C}_{yy}(t)$ is non-singular, and consider the $r \times r$ hermitian matrix $\boldsymbol{Z}(t) = \boldsymbol{C}_{yy}(t)^{-1}$. The following matrix inversion lemma [10, pp. 18-19] will lead to a recursion involving $\boldsymbol{Z}(t)$.

**Lemma 1** *Let $\boldsymbol{A}$ be a $r \times r$ non-singular complex matrix. Consider the $r \times r$ matrix $\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{M}\,\boldsymbol{R}\,\boldsymbol{N}$, where $\boldsymbol{M}$ is $r \times m$, $\boldsymbol{N}$ is $m \times r$, and $\boldsymbol{R}$ is $m \times m$ and non-singular. Then $\boldsymbol{B}$ is non-singular if and only if $\boldsymbol{R}^{-1} + \boldsymbol{N}\,\boldsymbol{A}^{-1}\,\boldsymbol{M}$ is non-singular, and in this case*

$$\boldsymbol{B}^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\,\boldsymbol{M}\left(\boldsymbol{R}^{-1} + \boldsymbol{N}\,\boldsymbol{A}^{-1}\,\boldsymbol{M}\right)^{-1}\boldsymbol{N}\,\boldsymbol{A}^{-1}.$$

Lemma 1 applied to equation (3) shows that

$$\boldsymbol{Z}(t) = \boldsymbol{Z}(t-1) - \boldsymbol{H}(t)\,\boldsymbol{\Gamma}(t)\,\boldsymbol{H}(t)^H \tag{5}$$

where $\boldsymbol{H}(t) = \boldsymbol{Z}(t-1)\,\boldsymbol{Y}(t)$ is a $r \times 2$ matrix, and $\boldsymbol{\Gamma}(t)$ is the $2 \times 2$ hermitian matrix[1]

$$\boldsymbol{\Gamma}(t) = \left(\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + \boldsymbol{H}(t)^H\boldsymbol{Y}(t)\right)^{-1}.$$

Substituting equations (4) and (5) into equation (2) yields

$$\boldsymbol{U}(t) = \boldsymbol{U}(t-1) + \boldsymbol{E}(t)\,\boldsymbol{G}(t)^H \tag{6}$$

where $\boldsymbol{E}(t) = \boldsymbol{X}(t) - \boldsymbol{U}(t-1)\,\boldsymbol{Y}(t)$ is a $n \times 2$ matrix, and $\boldsymbol{G}(t) = \boldsymbol{H}(t)\,\boldsymbol{\Gamma}(t)$ is a $r \times 2$ matrix.

Finally, the complete SW-PAST algorithm is presented[2] in table 1. Its overall computational cost is $O(nr)$.

## 3. SLIDING WINDOW OPAST

Compared to the compact form of SW-PAST given in equation (1), SW-OPAST additionally involves an orthonormalization step:

$$\boldsymbol{U}(t) = \boldsymbol{C}_{xx}(t)\boldsymbol{W}(t-1)\left(\boldsymbol{W}(t-1)^H\boldsymbol{C}_{xx}(t)\boldsymbol{W}(t-1)\right)^{-1} \tag{7}$$

$$\boldsymbol{W}(t) = \boldsymbol{U}(t)\,\boldsymbol{S}(t) \tag{8}$$

where $\boldsymbol{S}(t)$ is a $r \times r$ inverse square root of $\boldsymbol{U}(t)^H\boldsymbol{U}(t)$:

$$\boldsymbol{S}(t)\,\boldsymbol{S}(t)^H = \left(\boldsymbol{U}(t)^H\boldsymbol{U}(t)\right)^{-1}.$$

Note that here, it is supposed that $\boldsymbol{U}(t)$ is always full-rank, so that $\boldsymbol{U}(t)^H\boldsymbol{U}(t)$ is always non-singular (the rank deficiency case will be discussed later). Consequently, the $n \times r$ matrix $\boldsymbol{W}(t)$ defined in equation (8) is orthonormal. Applying the projection approximation to equation (7) yields equation (2), where $\boldsymbol{C}_{xy}(t)$ and $\boldsymbol{C}_{yy}(t)^{-1}$ are defined as in equations (4) and (3), except that $\boldsymbol{y}(t) = \boldsymbol{W}(t-1)^H\boldsymbol{x}(t)$ instead of $\boldsymbol{y}(t) = \boldsymbol{U}(t-1)^H\boldsymbol{x}(t)$. Therefore, the first step of the SW-OPAST algorithm is exactly the same as the SW-PAST main section[3] with $\boldsymbol{y}(t) = \boldsymbol{W}(t-1)^H\boldsymbol{x}(t)$.

[1] Note that lemma 1 also proves that the $2 \times 2$ matrix $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + \boldsymbol{H}(t)^H\boldsymbol{Y}(t)$ is non-singular if and only if $\boldsymbol{C}_{yy}(t)$ is non-singular.

[2] $\boldsymbol{I}_r$ denotes the $r \times r$ identity matrix.

[3] Note that the exponential forgetting window OPAST algorithm [3] consists in storing the orthonormalized matrix in the matrix variable $\boldsymbol{U}(t)$ itself at the end of each iteration. This method proved to outperform the original PAST subspace tracker. On the contrary, in the sliding window context, the orthonormalization of $\boldsymbol{U}(t)$ leads to a loss of stability. This degradation of the tracking performance was already observed in [1]. Here, to solve this problem, the orthonormalized matrix is stored in a matrix variable distinct from $\boldsymbol{U}(t)$, so that the orthonormalization step leaves the main part of the SW-PAST algorithm unchanged, leading to a stable subspace tracking method.

**Table 1**. Sliding Window PAST Algorithm

**SW − PAST initialization :**

$$[\boldsymbol{x}(-l+1), \ldots, \boldsymbol{x}(0)] = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{0}_{r\times(l-r)} \\ \boldsymbol{0}_{(n-r)\times r} & \boldsymbol{0}_{(n-r)\times(l-r)} \end{bmatrix},$$

$$[\boldsymbol{y}(-l+1), \ldots, \boldsymbol{y}(0)] = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{0}_{r\times(l-r)} \end{bmatrix},$$

$$\boldsymbol{U}(0) = \begin{bmatrix} \boldsymbol{I}_r \\ \boldsymbol{0}_{(n-r)\times r} \end{bmatrix},\ \boldsymbol{Z}(0) = \boldsymbol{I}_r$$

for each time step do

input vector : $\boldsymbol{x}(t)$

$\boldsymbol{y}(t) = \boldsymbol{U}(t-1)^H\boldsymbol{x}(t)$

**SW − PAST main section :**

$\boldsymbol{X}(t) = \left[\ \boldsymbol{x}(t)\ \middle|\ \boldsymbol{x}(t-l)\ \right]$

$\boldsymbol{Y}(t) = \left[\ \boldsymbol{y}(t)\ \middle|\ \boldsymbol{y}(t-l)\ \right]$

$\boldsymbol{H}(t) = \boldsymbol{Z}(t-1)\,\boldsymbol{Y}(t)$

$\boldsymbol{\Gamma}(t) = \left(\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + \boldsymbol{H}(t)^H\boldsymbol{Y}(t)\right)^{-1}$

$\boldsymbol{G}(t) = \boldsymbol{H}(t)\,\boldsymbol{\Gamma}(t)$

$\boldsymbol{Z}(t) = \boldsymbol{Z}(t-1) - \boldsymbol{G}(t)\,\boldsymbol{H}(t)^H$

$\boldsymbol{E}(t) = \boldsymbol{X}(t) - \boldsymbol{U}(t-1)\,\boldsymbol{Y}(t)$

$\boldsymbol{U}(t) = \boldsymbol{U}(t-1) + \boldsymbol{E}(t)\,\boldsymbol{G}(t)^H$

The orthonormalization step normally requires $O(nr^2)$ operations, but a fast $O(nr)$ update can be used instead. Consider the $r \times r$ non-singular hermitian matrix

$$\boldsymbol{\Phi}(t) = \boldsymbol{U}(t)^H\boldsymbol{U}(t). \tag{9}$$

The following developments aim at updating an inverse square root of $\boldsymbol{\Phi}(t)$. Substituting equation (6) into equation (9) yields

$$\boldsymbol{\Phi}(t) = \boldsymbol{\Phi}(t-1) + \boldsymbol{F}(t)\,\boldsymbol{J}(t)\,\boldsymbol{F}(t)^H \tag{10}$$

where $\boldsymbol{J}(t)$ is the $4 \times 4$ non-singular hermitian matrix

$$\boldsymbol{J}(t) = \left[\begin{array}{c|c} \boldsymbol{E}(t)^H\boldsymbol{E}(t) & \boldsymbol{I}_2 \\ \hline \boldsymbol{I}_2 & \boldsymbol{0}_2 \end{array}\right]$$

and $\boldsymbol{F}(t)$ is the $r \times 4$ matrix

$$\boldsymbol{F}(t) = \left[\ \boldsymbol{G}(t)\ \middle|\ \boldsymbol{U}(t-1)^H\,\boldsymbol{E}(t)\ \right]. \tag{11}$$

Let $\boldsymbol{S}(t-1)$ be an inverse square root of $\boldsymbol{\Phi}(t-1)$. In the particular case $\boldsymbol{F}(t) = \boldsymbol{0}$, $\boldsymbol{\Phi}(t) = \boldsymbol{\Phi}(t-1)$, so that $\boldsymbol{S}(t-1)$ is also an inverse square root of $\boldsymbol{\Phi}(t)$. From now on, suppose that $\boldsymbol{F}(t)$ has rank $0 < p \leq 4$. Consider the SVD of $\boldsymbol{S}(t-1)^H\boldsymbol{F}(t)$:

$$\boldsymbol{S}(t-1)^H\boldsymbol{F}(t) = \boldsymbol{Q}(t)\,\boldsymbol{\Sigma}(t)\,\boldsymbol{O}(t)^H$$

where $\boldsymbol{Q}(t)$ is a $r \times p$ orthonormal matrix, $\boldsymbol{\Sigma}(t)$ is a $p \times p$ positive diagonal matrix and $\boldsymbol{O}(t)$ is a $4 \times p$ orthonormal matrix. Then consider the $4 \times p$ matrix $\boldsymbol{L}(t) = \boldsymbol{O}(t)\,\boldsymbol{\Sigma}(t)$, so that

$$\boldsymbol{S}(t-1)^H\boldsymbol{F}(t) = \boldsymbol{Q}(t)\,\boldsymbol{L}(t)^H, \tag{12}$$

and the $p \times p$ matrix

$$\boldsymbol{T}(t) = \left(\boldsymbol{I}_p + \boldsymbol{L}(t)^H \boldsymbol{J}(t)\, \boldsymbol{L}(t)\right)^{-\frac{1}{2}} - \boldsymbol{I}_p.$$

Note that $\boldsymbol{I}_p + \boldsymbol{L}(t)^H \boldsymbol{J}(t)\, \boldsymbol{L}(t)$ is non-singular if and only if $\boldsymbol{U}(t)$ is full-rank[4]. A direct calculation shows that[5]

$$\boldsymbol{S}(t) \triangleq \boldsymbol{S}(t-1)\left(\boldsymbol{I}_r + \boldsymbol{Q}(t)\, \boldsymbol{T}(t)\, \boldsymbol{Q}(t)^H\right) \tag{13}$$

is a square root of $\boldsymbol{\Phi}(t)^{-1}$. Now consider the $n \times p$ matrix

$$\boldsymbol{P}(t) = \left[\begin{array}{c|c} \boldsymbol{E}(t) & \boldsymbol{0}_{n\times 2} \end{array}\right] \boldsymbol{L}(t)$$

so that equations (6), (8), (11) and (12) yield

$$\boldsymbol{U}(t)\, \boldsymbol{S}(t-1) = \boldsymbol{W}(t-1) + \boldsymbol{P}(t)\, \boldsymbol{Q}(t)^H. \tag{14}$$

Substituting equation (13) and (14) into equation (8) yields

$$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) + \boldsymbol{P}'(t)\, \boldsymbol{Q}(t)^H \tag{15}$$

where $\boldsymbol{P}'(t)$ is the $n \times p$ matrix

$$\boldsymbol{P}'(t) = \boldsymbol{W}(t-1)\, \boldsymbol{Q}(t)\, \boldsymbol{T}(t) + \boldsymbol{P}(t)\left(\boldsymbol{I}_p + \boldsymbol{T}(t)\right).$$

In equation (15), the update of $\boldsymbol{W}$ is just a rank $p$ modification. The whole processing requires only $O(nr)$ operations. Finally, this fast SW-OPAST algorithm[6] is summarized in table 2.

Note that if $\boldsymbol{U}(t)$ is rank deficient, $\boldsymbol{W}(t)$ can no longer be updated with equation (15). In this case, the algorithm must be re-initialized. In practice, we never encountered the rank deficiency case with our test signals, synthesized with various levels of noise (from noise free to a SNR of 0 dB).

## 4. LINK WITH THE NATURAL POWER METHOD

As for the OPAST algorithm [3], a link can be made between the SW-OPAST and the natural power (NP) method, introduced in [11,12]. This iterative method updates the signal subspace basis according to the following scheme:

$$\boldsymbol{W}(t) = \boldsymbol{C}_{xx}(t)\, \boldsymbol{W}(t-1)\left(\boldsymbol{W}(t-1)^H \boldsymbol{C}_{xx}(t)^2\, \boldsymbol{W}(t-1)\right)^{-\frac{1}{2}}$$

where the scaling $\left(\boldsymbol{W}(t-1)^H \boldsymbol{C}_{xx}(t)^2\, \boldsymbol{W}(t-1)\right)^{-\frac{1}{2}}$ guarantees that $\boldsymbol{W}(t)$ is an orthonormal matrix. Substituting equation (7) into equation (8) yields

$$\boldsymbol{W}(t) = \left(\boldsymbol{C}_{xx}(t)\, \boldsymbol{W}(t-1)\right) \boldsymbol{S}'(t)$$

where $\boldsymbol{S}'(t) = \left(\boldsymbol{W}(t-1)^H \boldsymbol{C}_{xx}(t) \boldsymbol{W}(t-1)\right)^{-1} \boldsymbol{S}(t)$. In particular, the orthonormality of $\boldsymbol{W}(t)$ proves that $\boldsymbol{S}'(t)$ is an inverse square root of $\boldsymbol{W}(t-1)^H \boldsymbol{C}_{xx}(t)^2\, \boldsymbol{W}(t-1)$. Therefore, the SW-OPAST algorithm can be seen as an implementation of the NP method, faster than the $O(nr^2)$ NP2 algorithm presented in [12].

Consequently, SW-OPAST satisfies the same global convergence property. As shown in [12], if the matrix $\boldsymbol{C}_{xx}(t)$ is static ($\boldsymbol{C}_{xx}(t) = \boldsymbol{C}_{xx}\ \forall t$), and if the first $r$ eigenvalues of $\boldsymbol{C}_{xx}$ are strictly larger than the $n-r$ other ones, the SW-OPAST algorithm converges globally and exponentially to the principal subspace. The convergence rate is governed by the ratio of the $r^{\text{th}}$ and the $(r+1)^{\text{th}}$ largest eigenvalues of $\boldsymbol{C}_{xx}$. Moreover, SW-OPAST avoids the stability problem of the original sliding window PAST (which can oscillate between two matrices without converging).

**Table 2**. Sliding Window orthonormal PAST algorithm

**SW − PAST initialization** (*cf.* table 1)

**SW − OPAST initialization :**

$$\boldsymbol{W}(0) = \begin{bmatrix} \boldsymbol{I}_r \\ \boldsymbol{0}_{(n-r)\times r} \end{bmatrix}, \ \boldsymbol{S}(0) = \boldsymbol{I}_r$$

for each time step do

- input vector : $\boldsymbol{x}(t)$
- $\boldsymbol{y}(t) = \boldsymbol{W}(t-1)^H \boldsymbol{x}(t)$
- **SW − PAST main section** (*cf.* table 1)
- **SW − OPAST main section :**
- $\boldsymbol{F}(t) = \left[\begin{array}{c|c} \boldsymbol{G}(t) & \boldsymbol{U}(t-1)^H\, \boldsymbol{E}(t) \end{array}\right]$
- $\boldsymbol{Q}(t)\, \boldsymbol{\Sigma}(t)\, \boldsymbol{O}(t)^H = \boldsymbol{S}(t-1)^H \boldsymbol{F}(t)$
- $\boldsymbol{L}(t) = \boldsymbol{O}(t)\, \boldsymbol{\Sigma}(t)$
- $\boldsymbol{J}(t) = \left[\begin{array}{c|c} \boldsymbol{E}(t)^H \boldsymbol{E}(t) & \boldsymbol{I}_2 \\ \hline \boldsymbol{I}_2 & \boldsymbol{0}_2 \end{array}\right]$
- $\boldsymbol{P}(t) = \left[\begin{array}{c|c} \boldsymbol{E}(t) & \boldsymbol{0}_{n\times 2} \end{array}\right] \boldsymbol{L}(t)$
- $\boldsymbol{T}(t) = \left(\boldsymbol{I}_p + \boldsymbol{L}(t)^H \boldsymbol{J}(t)\, \boldsymbol{L}(t)\right)^{-\frac{1}{2}} - \boldsymbol{I}_p$
- $\boldsymbol{S}(t) = \boldsymbol{S}(t-1) + \left(\boldsymbol{S}(t-1)\, \boldsymbol{Q}(t)\right) \boldsymbol{T}(t)\, \boldsymbol{Q}(t)^H$
- $\boldsymbol{P}'(t) = \boldsymbol{W}(t-1)\, \boldsymbol{Q}(t)\, \boldsymbol{T}(t) + \boldsymbol{P}(t)\left(\boldsymbol{I}_p + \boldsymbol{T}(t)\right)$
- $\boldsymbol{W}(t) = \boldsymbol{W}(t-1) + \boldsymbol{P}'(t)\, \boldsymbol{Q}(t)^H$

## 5. SIMULATION RESULTS

In this section, the performance of the subspace estimation is analyzed in the context of frequency estimation, in terms of the maximum principal angle between the true dominant subspace of the covariance matrix $\boldsymbol{C}_{xx}(t)$ (obtained via an exact eigenvalue decomposition), and the estimated dominant subspace of the same covariance matrix (obtained with the subspace tracker). This error criterion was proposed by P. Comon and G.H. Golub as a measure of the distance between equidimensional subspaces [13].

The test signal of Figure 1-a is a sum of $r = 4$ complex sinusoidal sources plus a complex white gaussian noise (the SNR is 5.7 dB). The frequencies of the sinusoids vary according to a jump scenario (originally proposed by P. Strobach in the context of DOA estimation [14]): their values abruptly change at different time in-

[4] This can be shown by applying lemma 1 to equation (10): $\boldsymbol{\Phi}(t)$ is non-singular if and only if the matrix $\boldsymbol{J}(t)^{-1} + \boldsymbol{F}(t)^H\, \boldsymbol{\Phi}(t)^{-1}\, \boldsymbol{F}(t)$ is non-singular. Since this last matrix is equal to $\boldsymbol{J}(t)^{-1} + \boldsymbol{L}(t)\, \boldsymbol{L}(t)^H$, lemma 1 also proves that it is non-singular if and only if $\boldsymbol{I}_p + \boldsymbol{L}(t)^H\, \boldsymbol{J}(t)\, \boldsymbol{L}(t)$ is non-singular.

[5] This can be proved by verifying that $\left(\boldsymbol{S}(t)\, \boldsymbol{S}(t)^H\right) \boldsymbol{\Phi}(t) = \boldsymbol{I}_r$.

[6] Note that if $\boldsymbol{y}(t)$ is computed like in table 1, the whole SW-PAST section is unchanged. In this case, the orthonormalization step just becomes a post-processing of the SW-PAST algorithm.

stants, between which they remain constant. Their variations are represented on Figure 1-b.

Figure 2-a shows the subspace tracking result, with parameters $n = 80$ and $l = 120$. It can be seen that the algorithm robustly tracks abrupt frequency variations. This result can be compared to that shown in figure 2-b, obtained with the SW-PAST subspace tracker. The performance is quite similar, but at the very beginning of the signal, it can be seen that SW-OPAST converges faster than SW-PAST. This can be explained by the global and exponential convergence property of the NP method. Figure 2-c shows the tracking result of the OPAST algorithm[7]. The tracking response to abrupt signal changes is slower. This is mainly due to the exponential forgetting nature of the analysis window used in OPAST, which tends to smooth the signal variations.

Finally, the orthonormality of the subspace basis estimates can be measured by means of the error criterion $\|\boldsymbol{W}(t)^H\boldsymbol{W}(t)-\boldsymbol{I}\|_F^2$ [9]. We observed on our test signal that the PAST subspace tracker reached a maximum error of -3.5 dB, whereas the SW-OPAST and OPAST algorithms never exceeded -200 dB and -290 dB.

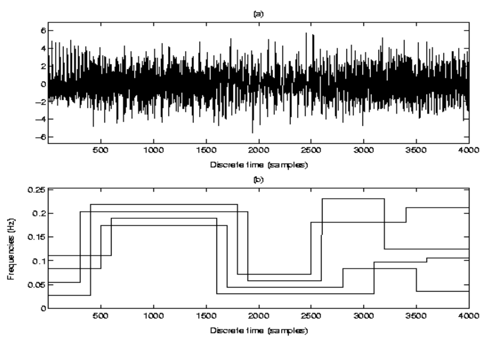


**Fig. 1**. (a): Test signal; (b): Normalized frequencies of the sinusoids.

## 6. CONCLUSIONS

In this paper, an orthonormal version of the sliding-window PAST subspace tracker was presented, which guarantees the orthonormality of the signal subspace basis at each time step. This algorithm reaches the linear complexity $O(nr)$. It can be seen as a fast implementation of the more computationally demanding NP2 algorithm, and therefore satisfies the same global convergence property. In the context of frequency estimation, the technique proved to robustly track abrupt frequency variations. It outperforms SW-PAST in terms of convergence speed, and offers a faster tracking response to sudden signal changes than OPAST.

[7]The forgetting factor $\alpha \simeq 0.99$ was chosen to get an effective window length equal to $L$.

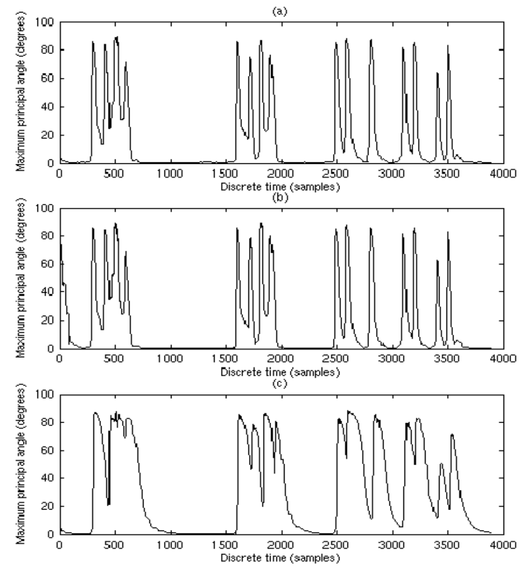


**Fig. 2**. Maximum principal angle trajectory: (a): SW-OPAST; (b): SW-PAST; (c): OPAST.

## 7. REFERENCES


[1] B. Yang, "Projection Approximation Subspace Tracking," *IEEE Trans. on Signal Proc.*, vol. 44, no. 1, january 1995.

[2] Y. Miao and Y. Hua, "Fast subspace tracking by a novel information criterion," in *Conf. record of the 31st Asilomar Conf. on Signals, Systems, and Comp.*, november 1997, vol. 2.

[3] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," *IEEE Sig. Pr. Letters*, vol. 7, no. 3, 2000.

[4] P. Strobach, "Square hankel SVD subspace tracking algorithms," *Signal Proc.*, vol. 57, no. 1, pp. 1–18, february 1997.

[5] E.C. Real, D.W. Tufts, and J.W. Cooley, "Two algorithms for fast approximate subspace tracking," *IEEE Trans. on Signal Proc.*, vol. 47, no. 7, pp. 1936–1945, july 1999.

[6] R.O. Schmidt, *A signal subspace approach to multiple emitter location and spectral estimation*, Ph.D. thesis, 1981.

[7] R. Kumaresan and D.W. Tufts, "Estimating the angles of arrival of multiple plane waves," *IEEE Trans. on Aerospace Electron. Syst.*, vol. AES-19, pp. 134–139, 1983.

[8] R. Roy and T. Kailath, "ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques," *IEEE Trans. on Acous., Speech, and Sig. Proc.*, vol. 37, no. 7, july 1989.

[9] S.C. Douglas, "Numerically-robust adaptive subspace tracking using Householder transformations," in *Proc. of IEEE Sensor Array and Multichannel Sig. Proc. Workshop*, 2000.

[10] R.A. Horn and C.R. Johnson, *Matrix analysis*, Cambridge University Press, Cambridge, 1985.

[11] Y. Hua, T. Chen, and Y. Miao, "A unifying view of a class of subspace tracking methods," in *Proc. Symp. Image, Speech, Sig. Proc. and Robotics*, Hong Kong, september 1998, vol. 2.

[12] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim, and Y. Miao, "A new look at the power method for fast subspace tracking," *Digital Signal Processing*, october 1999.

[13] P. Comon and G.H. Golub, "Tracking a few extreme singular values and vectors in signal processing," in *Proc. IEEE*, august 1990, vol. 78, pp. 1327–1343.

[14] P. Strobach, "Fast recursive subspace adaptive ESPRIT algorithms," *IEEE Trans. on Signal Proc.*, vol. 46, no. 9, 1998.