Hindawi Publishing Corporation
EURASIP Journal on Applied Signal Processing
Volume 2006, Article ID 37296, Pages 1–8
DOI 10.1155/ASP/2006/37296

Fast Iterative Subspace Algorithms for Airborne STAP Radar

Hocine Belkacemi and Sylvie Marcos

Laboratoire des Signaux et Systèmes (LSS), CNRS, Supélec, 3 rue Joliot-Curie, Plateau du Moulon, Gif-sur-Yvette Cedex 91192, France

Received 16 December 2005; Revised 30 May 2006; Accepted 16 July 2006

Space-time adaptive processing (STAP) is a crucial technique for the new generation airborne radar for Doppler spread compensation caused by the platform motion. We here propose to apply range cell snapshots-based recursive algorithms in order to reduce the computational complexity of the conventional STAP algorithms and to deal with a possible nonhomogeneity of the data samples. Subspace tracking algorithms as PAST, PASTd, OPAST, and more recently the fast approximate power iteration (FAPI) algorithm, which are time-based recursive algorithms initially introduced in spectral analysis, array processing, are good candidates. In this paper, we more precisely investigate the performance of FAPI for interference suppression in STAP radar. Extensive simulations demonstrate the outperformance of FAPI algorithm over other subspace trackers of similar computational complexity. We demonstrate also its effectiveness using measured data from the multichannel radar measurements (MCARM) program.

1. INTRODUCTION

Space-time adaptive processing (STAP) is a technique for suppressing clutter and jamming in airborne radar [1]. Employing an adaptive array antenna (spatial dimension) and a coherent (pulse) processing interval (CPI), the joint spatiotemporal domain optimization can provide far superior interference mitigation compared to the classical moving target indicator (MTI) methods [2]. In the optimum processor, the weight vector which maximizes the signal-to-interference-plus-noise ratio (SINR) is given by $\mathbf{w}_{\text{opt}} = \kappa\mathbf{R}^{-1}\mathbf{s}$, where $\mathbf{R}$ is the covariance matrix of the interferences and κ a constant gain. Since the covariance matrix is not known, Brennan and Reed [3] proposed the sample matrix inversion (SMI) based on replacing $\mathbf{R}$ by the sample average estimate $\widehat{\mathbf{R}}$. In general, there are two computational criteria that a practical implementation should ideally possess to achieve sufficient interference suppression: a rapid convergence (i.e., sample support size) to reduce nonhomogenous samples that contribute for the interference covariance estimation and a low computational complexity for real-time processing. Thus the SMI is a poor technique for the weight computation because it converges slowly requiring a wide-sense stationary (WSS) sample support of $K = 2NM$ samples to obtain an SINR performance within 3 dB of the optimal one in the Gaussian case, with a computational load of $O((NM)^3)$. The STAP interference covariance matrix is in general of low-rank. Subspace techniques exploit the low rank property of the interference covariance matrix to suppress the interferences. The key idea is the separation of the overall space into interference subspace and noise subspace followed by a projection into the interference-free subspace to suppress the interference [4]. A common method to obtain these subspaces is via singular value decomposition (SVD) of the interference-plus-noise covariance matrix. Such methods can reduce the sample support requirement to $O(2r)$, where r is the rank of the covariance matrix, but at the expense of a considerable computational complexity due to the SVD $O((NM)^3)$. This complexity prevents real-time applications. To reduce the computational burden linked to the SVD, many algorithms have been proposed in spectral analysis and spatial array processing literature. Among them, we have recursive subspace tracking algorithms that update the subspace estimate as long as a new snapshot is received. They can be classified depending on their computational complexities into $O((NM)^2r)$, $O(NMr^2)$, and $O(NMr)$ operations at each iteration (update) [5]. In this paper, only the class of the lowest computational complexity referred to as the linear subspace tracking algorithms is considered. The application of such algorithms in STAP radar problems is novel, since to our knowledge, the transcription of the recursive updating to the range cells has not been successfully performed until now in STAP radar problems. We more precisely propose the application of the fast approximate power iteration algorithm (FAPI) [6] in order to deal with the interference suppression in STAP radar. A comparison of the suggested algorithm

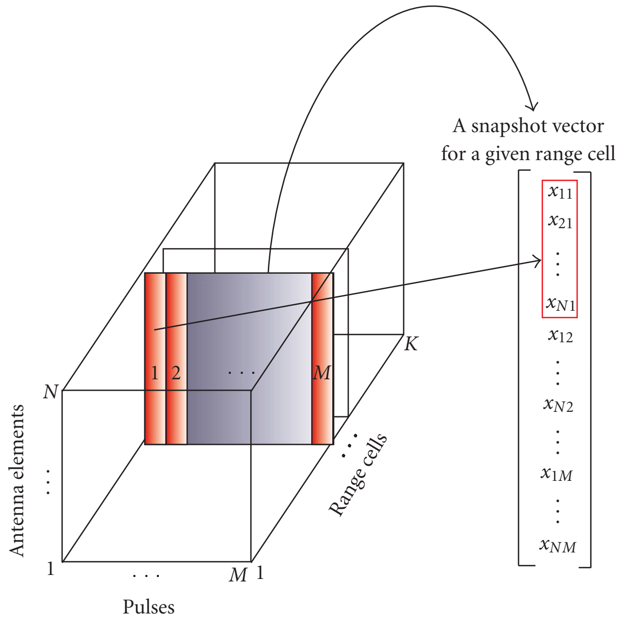

FIGURE 1: Illustration of the data cube and the construction of a STAP snapshot vector.

to other existing linear subspace tracking algorithms such as PAST, PASTd, OPAST [7, 8] is given. the proposed STAP-FAPI algorithm is also tested on the MCARM real data [9]. This paper is organized as follows. In sections 2 and 3, we introduce the data model and the STAP fundamentals, respectively. The concept of the projection approximate subspace tracking algorithms is revisited in Section 4. The fast approximate power iteration algorithm is presented in Section 5. After simulations both on synthetic and real data in Section 6, a few concluding remarks are drawn in Section 7.

2. DATA MODEL

We consider a pulse-Doppler radar mounted on an airborne platform moving at a constant speed v_p. The radar antenna is a uniformly spaced linear array consisting of N elements. The radar transmits a coherent burst of M pulses at a constant pulse-repetition frequency (PRF) $f_r = 1/T_r$. The returned signals are collected over K range directions of interest. The collected data in a coherent processing interval (CPI) can be represented by a 3D data cube as shown in Figure 1. The returned MN-dimensional space-time vector at one range of interest t represents the concatenated elements of a slice in the data cube (see Figure 1). It may consist of the target echo embedded into interferences such as jammer, clutter, and thermal noise, and is given by [10]

$$\mathbf{x}(t) = \alpha_t \mathbf{v}(\varpi_t, \nu_t) + \mathbf{x}_c(t) + \mathbf{x}_j(t) + \mathbf{n}(t), \quad (1)$$

where the following hold.

(i) $\mathbf{x}(t) = [x_1(t), \ldots, x_{MN}(t)]^T$ is the array output vector.

(ii) α_t and $\mathbf{v}(\varpi_t, \nu_t) = \mathbf{b}(\varpi_t) \otimes \mathbf{a}(\nu_t)$ are the complex target attenuation factor and target steering signal vector, respectively, associated with the spatial and Doppler parameters ϖ_t, ν_t [10] with

  (1) $\mathbf{a}(\nu_t) = [1 \ \ e^{j2\pi\nu_t} \ \cdots \ e^{j2\pi(M-1)\nu_t}]^T$ is the temporal steering vector ($\nu_t = f_t/f_r$, f_t is the target's Doppler frequency);

  (2) $\mathbf{b}(\varpi_t) = [1 \ \ e^{j2\pi\varpi_t} \ \cdots \ e^{j2\pi(N-1)\varpi_t}]^T$ is the spatial steering vector ($\varpi_t = (d/\lambda)\sin(\theta_t)$, d is the element separation distance and λ is the wavelength, and θ_t is the target's azimuth angle).

(iii) The radar clutter returns vector $\mathbf{x}_c(t)$ is generated according to Ward's model [10]. It consists of a superposition of a large number N_c of clutter sources that are evenly distributed in a circular ring about the radar platform. The location of the ith clutter patch is described by its azimuth θ_i and normalized Doppler frequency ν_i, the clutter component of the space-time snapshot is given by

$$\mathbf{x_c} = \sum_{i=1}^{N_c} \gamma_i \mathbf{v}_i(\varpi_i, \nu_i), \quad (2)$$

where $\mathbf{v}_i(\varpi_i, \nu_i)$ is the space-time steering vector of the ith clutter patch, and γ_i is its random amplitude assumed to be Gaussian distributed (note that the range index t is omitted in the following notations in the model).

(iv) The component $\mathbf{x}_j$ represents the narrowband noise jamming signals. A commonly employed model for such N_j jamming signals is [10]

$$\mathbf{x_j} = \sum_{m=1}^{N_j} \boldsymbol{\gamma}_m \otimes \mathbf{b}(\varpi_m), \quad (3)$$

where $\boldsymbol{\gamma}_m$ contains the mth jammer amplitudes taken at a pulse repetition interval (PRI)

(v) The component $\mathbf{n}$ is due to the thermal noise and it is spatially and temporally white.

If we suppose that these components are uncorrelated then the interference (clutter + jammer + thermal noise) space-time covariance matrix $\mathbf{R}$ is

$$\mathbf{R} = \mathbf{R_c} + \sum_{i=1}^{N_j} \mathbf{R}_j(i) + \sigma^2 \mathbf{I}_{NM}, \quad (4)$$

where $\mathbf{R_c}$ is the clutter covariance matrix, N_j is the number of jammers, $\mathbf{R}_j(i)$ is the covariance matrix of the ith jammer, σ^2 is the noise variance, and $\mathbf{I}_{NM}$ denotes the identity matrix of dimension $NM \times NM$.

3. STAP FUNDAMENTALS

STAP is a two-dimensional adaptive filtering technique proposed as an alternative for one-dimensional techniques (spatial or temporel methods) to suppress interferences (clutter and jamming) and to achieve both target detection and parameter estimation in airborne or spaceborne radar [10].

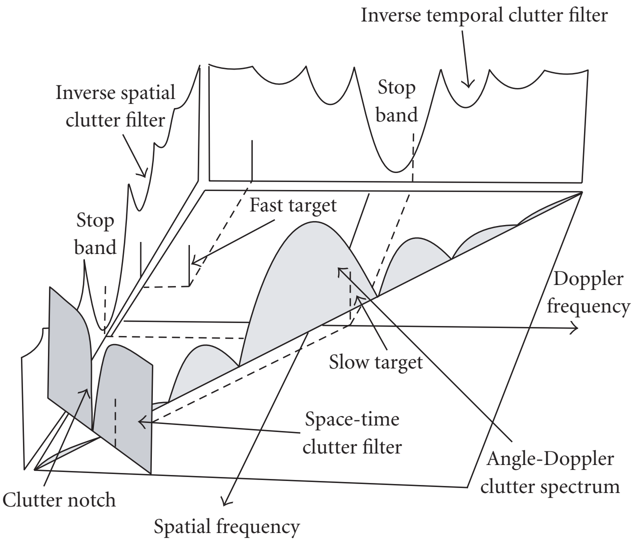

Figure 2: Illustration of the principle of the STAP filtering for a sidelooking radar [1].

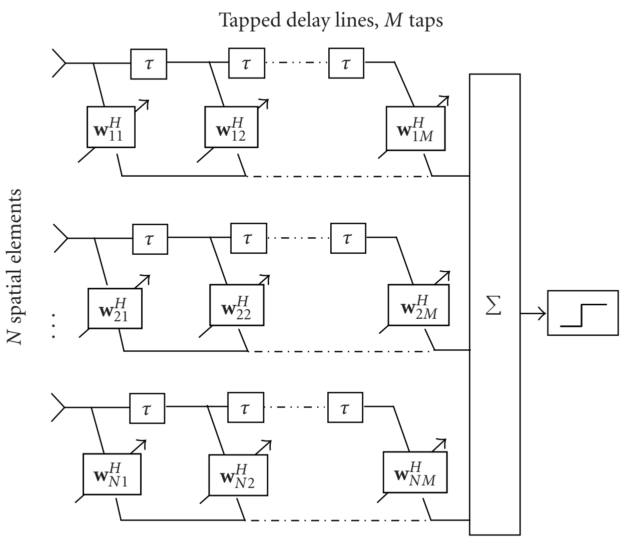

Figure 3: Space-time filter.

Figure 2 shows clearly that by using a 1D conventional filter, the slow moving target is masked by the clutter. The data cube is generally interrogated for presence or absence of targets. This is done by utilizing an adaptive 2D STAP beamformer at each range cell t to maximize the SINR (see Figure 3). The optimum weight vector which maximizes the SINR is given by

$$\mathbf{w}_{\text{opt}} = \kappa \mathbf{R}^{-1}\mathbf{v}_t, \tag{5}$$

where $\mathbf{R}$ is the interference + noise covariance matrix as defined in (4), κ is a constant gain, and $\mathbf{v}_t$ is the space-time target search steering vector.

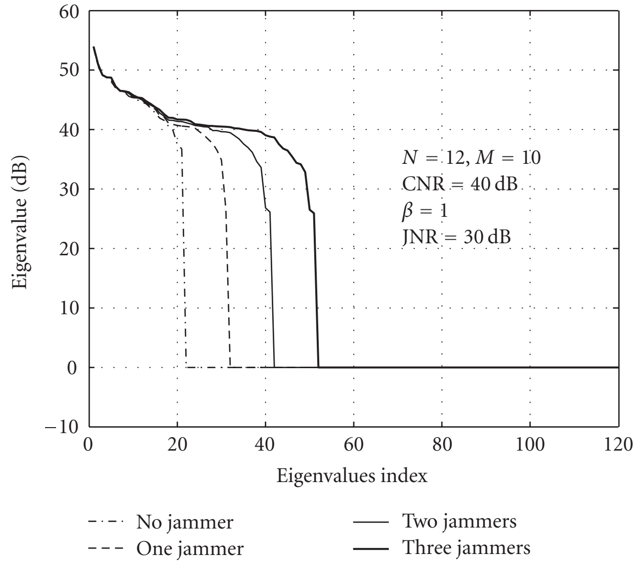

Figure 4: Eigenspectra for known STAP covariance matrices with target azimuth angle at 0° and normalized Doppler frequency $\nu_t = 0.25$; $N = 12$, $M = 10$, CNR = 40 dB, $\beta = 1$, JNR = 30 dB.

In practice, the covariance matrix $\mathbf{R}$ is not known and must be estimated from a given number of snapshots. A commonly used technique is the sample covariance estimation

$$\hat{\mathbf{R}} = \frac{1}{K}\sum_{t=1,\, t\neq l}^{K} \mathbf{x}(t)\mathbf{x}^H(t), \tag{6}$$

index l corresponds to the cell under test (CUT) which must be excluded from the covariance estimation to avoid target cancellation. Substituting $\mathbf{R}$ by $\hat{\mathbf{R}}$ in (5), we get the well-known SMI method

$$\mathbf{w}_{\text{SMI}} = \kappa \hat{\mathbf{R}}^{-1}\mathbf{v}_t. \tag{7}$$

This method has many drawbacks as a high computational complexity due to the matrix inversion, a slow convergence (it requires $2NM$ i.i.d snapshots to achieve 3 dB of SINR below the optimum), and high sidelobes. Thus this technique is not suitable for a real radar application. To overcome this problem, the authors in [4] proposed a subspace technique known as eigencanceller (EC). It requires $O(2r)$ snapshots (r is the rank of the interference covariance matrix) to achieve an SINR of 3 dB below the optimum. EC exploits the low-rank property of the interference covariance matrix to calculate the STAP weight vector. It is based on decomposing the observation space into interference subspace and noise subspace and then calculating a STAP weight vector that is orthogonal to the interference subspace. Figure 4 shows the eigenspectra of known STAP covariance matrices. We note a clear distinction between the interferences eigenvalues and the floor of identical eigenvalues (noise subspace). Brennan and Reed [3] found the following rule (Brennan's

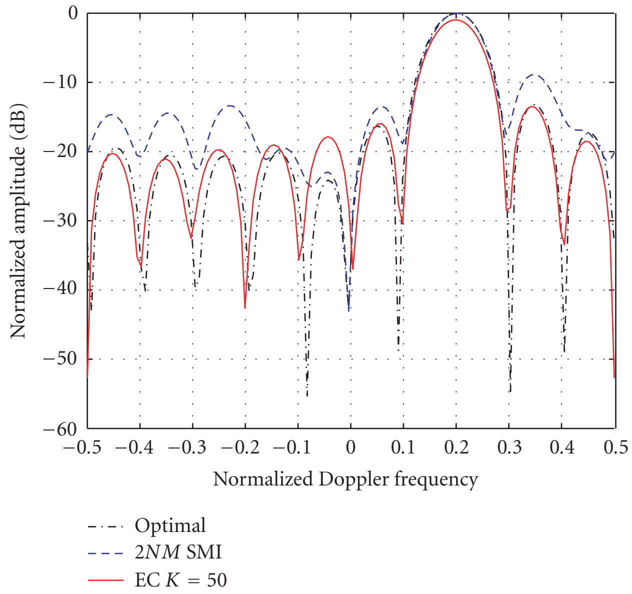

Figure 5: A cut of the 2D STAP pattern at the radar look angle (0°) for EC, SMI, and optimum STAP. $N = 12$, $M = 10$, $\theta_t = 0°$, $\nu_t = 0.2$.

rule)[1] for the number of the effective ranks of the space-time interfernce STAP covariance matrix of a side-looking radar $r = N + \beta(M - 1)$.

The STAP covariance matrix can be written as

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H + \sigma^2\mathbf{V}\mathbf{V}^H, \tag{8}$$

where the diagonal of the $r \times r\mathbf{\Lambda}$ matrix consists of the dominant eigenvalues and $\mathbf{U}$ is the matrix of the corresponding eigenvectors, $\mathbf{V}$ is the matrix of the remaining eigenvectors. The weight vector of the eigencanceller is given by [4]

$$\mathbf{w}_{\text{ec}} = (\mathbf{I} - \mathbf{U}\mathbf{U}^H)\mathbf{v}_t. \tag{9}$$

Figure 5 shows that the EC using the covariance estimate (6) has a performance near the optimum with low sidelobes and short sample support compared to the SMI. Our objective is to calculate an estimate of the interference subspace $\hat{\mathbf{U}}$ that does not resort to the eigendecomposition of $\hat{\mathbf{R}}$ with lower computational cost.

4. PROJECTION APPROXIMATE SUBSPACE TRACKING ALGORITHMS

Projection approximation subspace tracking (PAST) [7] is one successful subspace tracking algorithm due to its simplicity and efficiency. Consider the following optimization criterion:

$$J(\mathbf{W}) = E\left(\left\|\mathbf{x} - \mathbf{W}\mathbf{W}^H\mathbf{x}\right\|^2\right) \tag{10}$$

[1] An extension to Brenann's rule has been derived recently [11] and is given by $r = N + (\beta + J)(M - 1)$, where J denotes the number of jammers.

Initialization:
$\mathbf{W}(0) = \begin{bmatrix}\mathbf{I}_r \\ \mathbf{O}_{(NM-r)\times r}\end{bmatrix}, \quad \mathbf{Z}(0) = \mathbf{I}_r$
For $t = 1, 2, \ldots$ do
  $\mathbf{y}(t) = \mathbf{W}(t-1)^H\mathbf{x}(t)$
  $\mathbf{h}(t) = \mathbf{Z}(t-1)\mathbf{y}(t)$
  $\mathbf{g}(t) = \mathbf{h}(t)/(\mu + \mathbf{y}(t)^H\mathbf{h}(t))$
  $\tilde{\mathbf{h}}(t) = \mathbf{y}^H(t)\mathbf{Z}(t-1)$
  $\mathbf{Z}(t) = (1/\mu)(\mathbf{Z}(t-1) - \mathbf{g}(t)\tilde{\mathbf{h}}(t))$
  $\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{W}(t-1)\mathbf{y}(t)$
  $\mathbf{W}(t) = \mathbf{W}(t-1) + \mathbf{e}(t)\mathbf{g}(t)^H$
End for

Algorithm 1: PAST algorithm.

with $\mathbf{W} \in \mathbb{C}^{NM\times r}$. The criterion in (10) has the following properties which are summarized in the following theorem.

Theorem 1. *The matrix $\mathbf{W}$ is a stationary point of $J(\mathbf{W})$ if and only if $\mathbf{W} = \mathbf{U}_d\mathbf{Q}$, where $\mathbf{U}_d \in \mathbb{C}^{NM\times d}$ contains any distinct eigenvectors of $\mathbf{R}$, and $\mathbf{Q} \in \mathbb{C}^{d\times d}$ is an arbitrary unitary matrix. At each stationary point, $J(\mathbf{W})$ is equal to the sum of the eigenvalues whose eigenvectors are not involved in $\mathbf{U}_d$. Moreover, all stationary points of $J(\mathbf{W})$ are saddle points, except when $\mathbf{U}_d = \mathbf{U}$. In this case, $J(\mathbf{W})$ attains the global minimum.*

It is shown in [5] that minimizing (10) results in the following batch form of the PAST method:

$$\begin{aligned}\mathbf{W}(t) &= \mathbf{R}(t)\mathbf{W}(t-1)\left(\mathbf{W}^H(t-1)\mathbf{R}(t)\mathbf{W}(t-1)\right)^{-1} \\ &= \mathbf{C}(t)\mathbf{Z}(t),\end{aligned} \tag{11}$$

where $\mathbf{C}(t) = \mathbf{R}(t)\mathbf{W}(t-1)$ is the compression step as in the power iteration method and $\mathbf{Z}(t) = (\mathbf{W}^H(t-1)\mathbf{R}(t)\mathbf{W}(t-1))^{-1}$ corresponds to the orthonormalization step [5]. A fast recursive implementation of (11) is proposed in [7]. It is based on the following projection approximation:

$$\mathbf{W}(t) \approx \mathbf{W}(t-1). \tag{12}$$

The complete pseudo code is depicted in Algorithm 1. It results from replacing the covariance matrix $\mathbf{R}(t)$ with its recursive version $\mathbf{R}(t) = \mu\mathbf{R}(t-1) + \mathbf{x}(t)\mathbf{x}^H(t)$, and implementing recursively the inverse matrix $\mathbf{Z}(t)$ via the matrix inversion lemma. Note that the author in [7] proposed another version of the algorithm based on the deflation technique and referred to as the PASTd algorithm [7].

The PAST algorithm does generally not converge to an orthonormal basis. The classic methods for imposing orthonormality are based on a QR or Gram-Schmidt procedure. Such methods require at least $O(NMr^2)$ operations per update. In [8], the authors proposed a fast ($O(NMr)$) orthonormal version of the PAST algorithm denoted by OPAST. It consists of the PAST (11) plus an orthonormalization step via the square root inverse [12] to force the orthonormality of $\mathbf{W}(t)$ at each iteration

$$\mathbf{W}(t) := \mathbf{W}(t)\left(\mathbf{W}^H(t)\mathbf{W}(t)\right)^{-1/2}. \tag{13}$$

Initialization:
$\mathbf{W}(0) = \begin{matrix}\mathbf{I}_r \\ \mathbf{O}_{(NM-r)\times r}\end{matrix}, \quad \mathbf{Z}(0) = \mathbf{I}_r$
For $t = 1, 2, \ldots$ do
Past main section:
$\quad \mathbf{y}(t) = \mathbf{W}^H(t-1)\mathbf{x}(t)$
$\quad \mathbf{h}(t) = \mathbf{Z}(t-1)\mathbf{y}(t)$
$\quad \gamma(t) = \mathbf{1}/(\mu + \mathbf{y}^H(t)\mathbf{h}(t))$
$\quad \mathbf{g}(t) = \gamma(t)\mathbf{h}(t)$
OPAST main section:
$\quad \boldsymbol{\epsilon}(t) = \gamma(t)(\mathbf{x}(t) - \mathbf{W}(t-1)\mathbf{y}(t))$
$\quad \tau(t) = (1/(1/\mu^2)\|\mathbf{h}(t)\|^2)\left(1/\sqrt{1+(1/\mu^2)\|\boldsymbol{\epsilon}(t)\|^2\|\mathbf{h}(t)\|^2}\right)$
$\quad \mathbf{e}(t) = (\tau(t)/\mu)\mathbf{W}(t-1)\mathbf{h}(t) + (1 + (\tau(t)/\mu^2)\|\mathbf{h}(t)\|^2)\boldsymbol{\epsilon}(t)$
$\quad \widetilde{\mathbf{h}}(t) = \mathbf{y}^H(t)\mathbf{Z}(t-1)$
$\quad \mathbf{Z}(t) = (1/\mu)\mathbf{Z}(t-1) - (1/\mu)\mathbf{g}(t)\widetilde{\mathbf{h}}(t)$
$\quad \mathbf{W}(t) = \mathbf{W}(t-1) + \mathbf{e}(t)\mathbf{g}^H(t)$
End for

Algorithm 2: OPAST algorithm.

Initialization:
$\mathbf{W}(0) = \begin{matrix}\mathbf{I}_r \\ \mathbf{O}_{(NM-r)\times r}\end{matrix}, \quad \mathbf{Z}(0) = \mathbf{I}_r$
For $t = 1, 2, \ldots$ do
PAST main section:
$\quad \mathbf{y}(t) = \mathbf{W}(t-1)^H\mathbf{x}(t)$
$\quad \mathbf{h}(t) = \mathbf{Z}(t-1)\mathbf{y}(t)$
$\quad \mathbf{g}(t) = \mathbf{h}(t)/(\mu + \mathbf{y}(t)^H\mathbf{h}(t))$
FAPI main section:
$\quad \varepsilon^2(t) = \|\mathbf{x}(t)\|^2 - \|\mathbf{y}(t)\|^2$
$\quad \tau(t) = \varepsilon^2(t)/\left(1 + \varepsilon^2(t)\|\mathbf{g}(t)\|^2 + \sqrt{1+\varepsilon^2(t)\|\mathbf{g}(t)\|^2}\right)$
$\quad \eta(t) = 1 - \tau(t)\|\mathbf{g}(t)\|^2$
$\quad \widetilde{\mathbf{y}}(t) = \eta(t)\mathbf{y}(t) + \tau(t)\mathbf{g}(t)$
$\quad \widetilde{\mathbf{h}}(t) = \mathbf{Z}(t-1)\widetilde{\mathbf{y}}(t)$
$\quad \boldsymbol{\epsilon}(t) = (\tau(t)/\eta(t))(\mathbf{Z}(t-1)\mathbf{g}(t) - (\widetilde{\mathbf{h}}(t)^H\mathbf{g}(t))\mathbf{g}(t))$
$\quad \mathbf{Z}(t) = (1/\mu)(\mathbf{Z}(t-1) - \mathbf{g}(t)\widetilde{\mathbf{h}}(t)^H + \boldsymbol{\epsilon}(t)\mathbf{g}(t)^H)$
$\quad \mathbf{e}(t) = \eta(t)\mathbf{x}(t) - \mathbf{W}(t-1)\widetilde{\mathbf{y}}(t)$
$\quad \mathbf{W}(t) = \mathbf{W}(t-1) + \mathbf{e}(t)\mathbf{g}(t)^H$
End for

Algorithm 3: FAPI algorithm.

Using the fact that $\mathbf{W}(t-1)$ is an orthonormal matrix and employing the orthogonality of $\mathbf{e}(t)$ with $\mathbf{W}(t-1)$, (13) can be written as

$$\mathbf{W}(t) := \mathbf{W}(t)\left(\mathbf{I}_r + \|\mathbf{e}(t)\|^2\mathbf{g}(t)\mathbf{g}^H(t)\right)^{-1/2}. \tag{14}$$

A fast implementation of the inverse square root in (14) is described in detail in [8][2]. The complete pseudocode of OPAST is shown in Algorithm 2.

Note that the PAST algorithm in its first version by B. Yang which inspired some other works [5, 8, 13], the recursive updating of the inverse of the matrix $\mathbf{Z}(t)$ is

$$\mathbf{Z}(t) = \frac{1}{\mu}(\mathbf{Z}(t-1) - \mathbf{g}(t)\mathbf{h}(t)^H) \tag{15}$$

which suggests that $\mathbf{Z}(t-1) = \mathbf{Z}^H(t-1)$. This is theorically true. However, in a stochastic updating as it is done in the PAST or OPAST algorithm, $\mathbf{Z}(t-1)$ and $\mathbf{Z}^H(t-1)$ may differ due to rounding errors causing the divergence of the algorithms. To overcome this proplem, Yang [7] had forced $\mathbf{Z}(t)$ to be Hermitian. This constraint was not considered in the OPAST [8]. The proposed version in Algorithms 1 and 2 does not diverge for all simulations.

5. FAST APPROXIMATE POWER ITERATION ALGORITHM

The approximate power iteration algorithm (API) consists of a less restrictive approximation in implementing (11) than it is done in PAST [7]. Indeed, in [6] it is assumed that

$$\prod_{\mathbf{W}(t)} \approx \prod_{\mathbf{W}(t-1)}, \tag{16}$$

where $\prod_{\mathbf{W}(t)} = \mathbf{W}(t)\mathbf{W}^H(t)$. Equation (16) is equivalent to writing

$$\mathbf{W}(t) \approx \mathbf{W}(t-1)\Theta(t) \tag{17}$$

with $\boldsymbol{\Theta}(t) \triangleq \mathbf{W}^H(t-1)\mathbf{W}(t)$.

Using this modification, the normalization matrix $\mathbf{Z}(t)$ and the subspace matrix $\mathbf{W}(t)$ (see Algorithm 1) can be estimated recursively as follows (for more details, the reader can refer to [6]):

$$\mathbf{Z}(t) = \frac{1}{\mu}\boldsymbol{\Theta}^H(t)(\mathbf{I}_r - \mathbf{g}(t)^H\mathbf{y}(t))\mathbf{Z}(t-1)\boldsymbol{\Theta}^{-H}(t), \tag{18}$$

$$\mathbf{W}(t) = (\mathbf{W}(t-1) + \mathbf{e}(t)\mathbf{g}^H(t))\boldsymbol{\Theta}^H(t), \tag{19}$$

where $\mathbf{I}_r$ denotes the $r \times r$ identity matrix, $\mathbf{e}(t)$, $\mathbf{g}(t)$ and $\mathbf{y}(t)$ are as defined in the PAST algorithm (Algorithm 1).

By using (19) and keeping in mind that $\mathbf{W}(t-1)$ and its update $\mathbf{W}(t)$ are orthonormal matrices in addition to the fact that the error vector $\mathbf{e}(t)$ is orthogonal to $\mathbf{W}(t-1)$, then $\boldsymbol{\Theta}(t)$ can be written as

$$\boldsymbol{\Theta}(t) = \left(\mathbf{I}_r + \|\mathbf{e}(t)\|^2\mathbf{g}(t)\mathbf{g}^H(t)\right)^{-1/2}. \tag{20}$$

We can note that by setting $\boldsymbol{\Theta}(t) = \mathbf{I}_r$, both matrices $\mathbf{Z}(t)$ and $\mathbf{W}(t)$ as defined in (18) and (19) reduce to those of PAST algorithm. However, for the OPAST algorithm, the $\boldsymbol{\Theta}(t)$ matrix is kept for $\mathbf{W}(t)$.

FAPI is a fast implementation of API ($O(NMr)$) which consists in substituting $\boldsymbol{\Theta}(t)$ by a faster computation of the inverse square root. The pseudocode of the algorithm is given in Algorithm 3.

[2] One way to get the inverse square root is via the following identity: $(\mathbf{I} + \mathbf{x}\mathbf{x}^H)^{1/2} = \mathbf{I} + (1/\|\mathbf{x}\|^2)(1/\sqrt{1+\|\mathbf{x}\|^2}) - 1\mathbf{x}\mathbf{x}^H$.

6. SIMULATION RESULTS

6.1. Example 1: simulated data

The simulation model consists of a uniform linear array of $N = 12$ elements with $M = 10$ delay taps at each element. The clutter is Gaussian distributed with clutter-to-noise ratio CNR = 20 dB at each element (no jammer and internal clutter motion ICM are present). The target is assumed in the main beam direction 0° with SNR = 0 dB. As a measure of performance, we use the SINR loss as defined below:

$$\mathrm{SINR}_{\mathrm{Loss}} = \frac{\sigma^2}{NM}\,\frac{|\mathbf{w}^H\mathbf{v}_t|^2}{\mathbf{w}^H\mathbf{R}_i\mathbf{w}}. \tag{21}$$

$\mathbf{w}$ denotes the STAP weight vector using the subspace estimates. All the simulations were carried out over 20 Monte Carlo runs, the forgetting factor is $\mu = 0.99$ for all the algorithms. The rank of the interference subspace is approximated via Brennan's rule [10] ($r = N + \beta(M-1)$).

Figures 6 and 7 show the $\mathrm{SINR}_{\mathrm{Loss}}$ as a function of the sample support. In this example, both PAST and PASTd require an orthonormalization step to accelerate their convergence.[3] This is performed with the Gram-Schmidt procedure which needs extra computation of $O(NMr^2)$. The PASTd algorithm outperforms PAST algorithm because unlike PAST, PASTd is based on the sequential estimation of the principal components which mitigate the effect of the eigenvalue spread. FAPI converges much faster than all the algorithms and it exhibits a very close performance to the EC. This can be justified by the less restrictive approximation in FAPI $\mathbf{W}(t)\mathbf{W}(t)^H \simeq \mathbf{W}(t-1)\mathbf{W}(t-1)^H$ rather than $\mathbf{W}(t) \simeq \mathbf{W}(t-1)$ in the projection approximation algorithms. As expected, the OPAST algorithm performance is in between PAST and FAPI algorithms. In Figure 7, we show a cut of the SINR loss at the main beam look angle as a function of the normalized Doppler frequency using $K = 50$ snapshots. We can note that the FAPI algorithm notch filter has a very close performance to that of the EC which is near the optimum filter (for which the interference-plus-noise covariance matrix is known) compared to the loaded SMI [15].

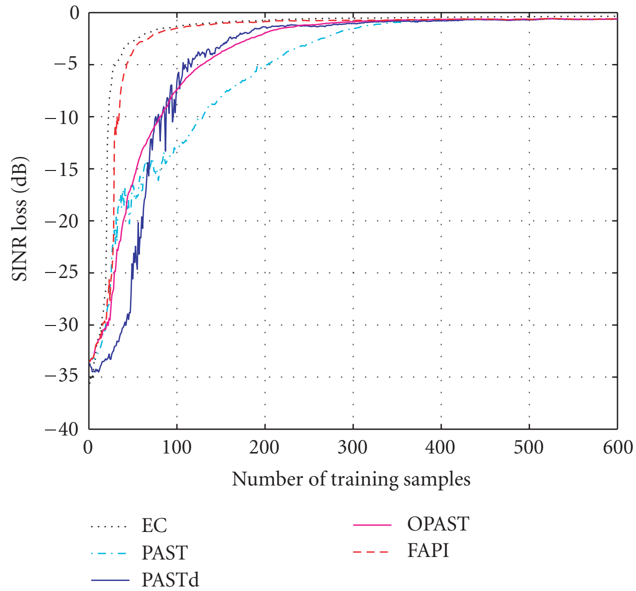

FIGURE 6: SINR loss as a function of the training data support K. Forgetting factor $\mu = 0.99$.

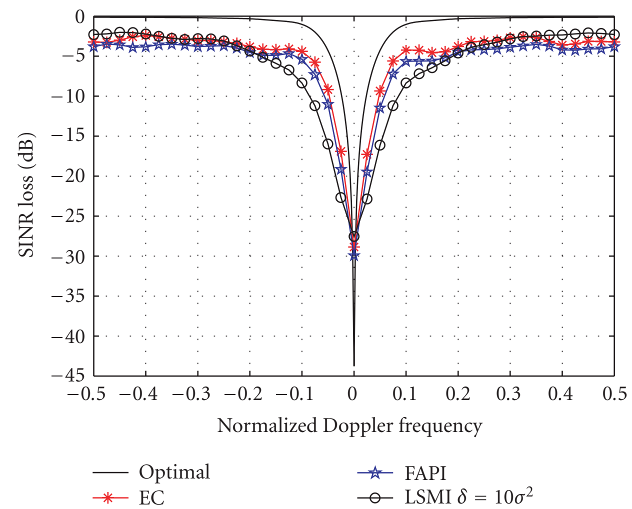

FIGURE 7: SINR loss as a function of the normalized Doppler frequency.

6.2. Example 2: experimental data

In this example, we briefly evaluate the performance of FAPI algorithm using data from the multichannel airborne radar measurements (MCARM). The sensor is an L-band phased array antenna using 22 elements arranged in a 2×11 grid ($N = 22$). Each CPI comprises 128 pulses ($M = 128$). There are 630 independent range samples available for the training-data support. The subject data comes from flight 5, acquisition 575. Figure 8 shows the 2D spectrum of the MCARM radar data [9] at range cell 200, we can clearly note the monostatic STAP clutter ridge. Its slope is approximately $\beta \simeq 1$. Figure 9 depicts power versus range at pulse 1, the first range cells represent transmit leakage [16] and are not used for our simulations. In this example, only range cells from 200 to 600, $N = 14$ channels, and $M = 16$ pulses are used for the performance evaluation. We inject artificially a target at range cell 290 at angle bin 65 corresponding to 0° (broadside) and normalized Doppler frequency of 0.3 with amplitude 0.00003. Because of the nonavailability of the noise level of the antenna in the MCARM data, we prefered to use the modified sample matrix inversion (MSMI) as defined below:

$$\eta = \frac{|\mathbf{w}^H\mathbf{x}|^2}{\mathbf{w}^H\hat{\mathbf{R}}\mathbf{w}}. \tag{22}$$

[3] The slow convergence is mainly due to the high-input eigenvalue spread of the STAP covariance matrix which is a problem inherent in adaptive algorithms [14] ($\lambda_{\max} \simeq$ CNR $+ 10\log(NM)$(dB)), see Figure 4.

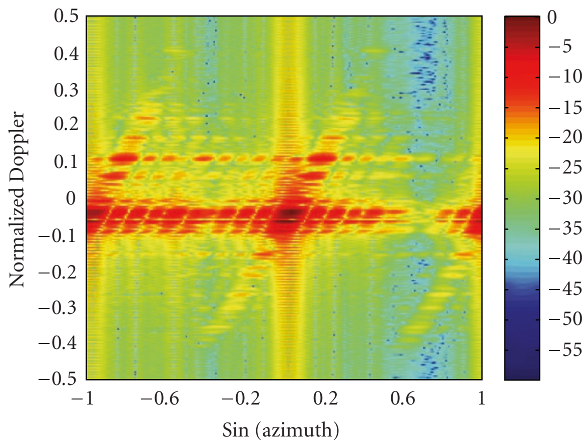

FIGURE 8: Power spectrum for MCARM radar data range cell 200.

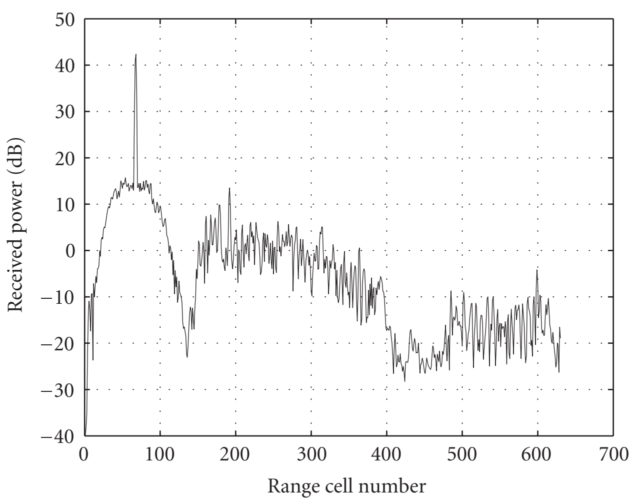

FIGURE 9: Output power of MCARM versus range at pulse 1.

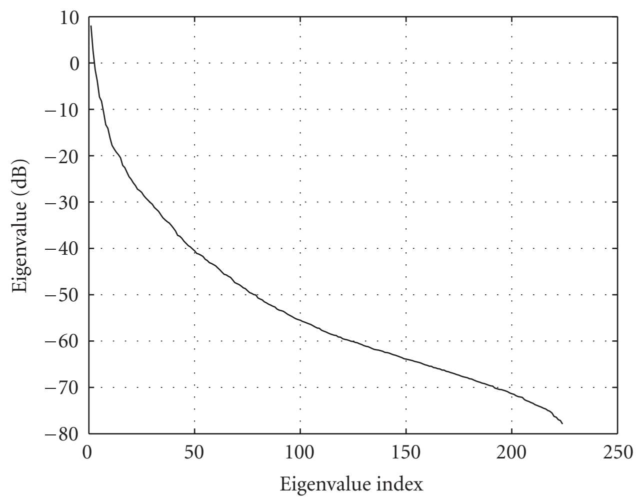

FIGURE 10: Eigenspectrum using the sample covariance matrix of the MCARM data.

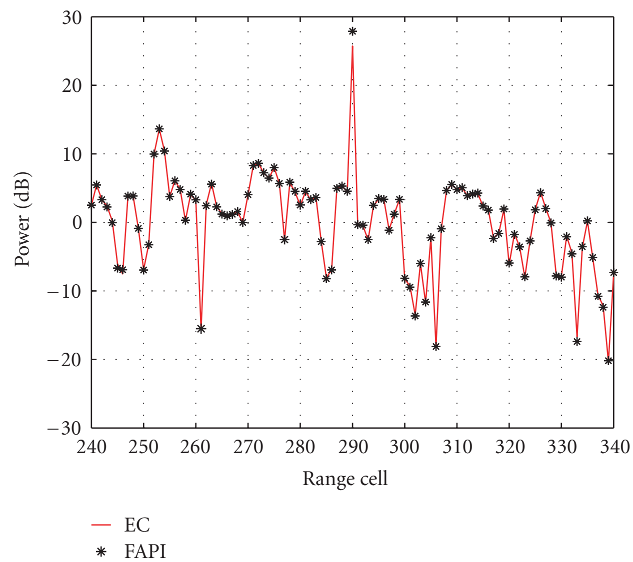

FIGURE 11: MSMI statistics over range for EC and FAPI, injected target at range cell 290.

Figure 10 shows the eigenspectrum of the sample covariance matrix, the spectrum does not show a sharp cutoff at the interference rank value as for simulated data. There is a gradual decrease, this is mainly due to the crabbing angle. We choose a rank of $r = 90$ corresponding to noise power of -55 dB[4] rather than the rank suggested by Brennan's rule which is $r = 14 - (16 - 1) = 29$.

Figure 11 shows the MSMI statistics over range. We note that the target is visible for both EC and FAPI with the same level of power.

7. CONCLUSION

In this paper, the application of fast recursive subspace algorithms for interference mitigation in STAP radar is considered. We evaluate the performance of a recently developed fast subspace algorithm known as FAPI. The results of simulation using synthetic data and measured data show that FAPI algorithm approaches the same performance of the EC with a linear computation complexity ($O(NMr)$) rather than ($O(NM)^3$) for the EC.

[4] The choice of this noise power level corresponds to realistic values [9].

REFERENCES

[1] R. Klemm, *Space-Time Adaptive Processing: Principles and Applications*, vol. 9 of *IEE Radar, Sonar, Navigation and Avionics*, IEE Press, London, UK, 2000.

[2] M. I. Skolnik, *Radar Handbook*, McGraw-Hill, New York, NY, USA, 1990.

[3] L. E. Brennan and L. S. Reed, "Theory of adaptive radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 9, no. 2, pp. 237–252, 1973.

[4] A. Haimovich, "Eigencanceler: adaptive radar by eigenanalysis methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 2, pp. 532–542, 1996.

[5] 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*, vol. 9, no. 4, pp. 297–314, 1999.

[6] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2931–2941, 2005.

[7] B. Yang, "Projection approximation subspace tracking," *IEEE Transactions on Signal Processing*, vol. 43, no. 1, pp. 95–107, 1995.

[8] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," *IEEE Signal Processing Letters*, vol. 7, no. 3, pp. 60–62, 2000.

[9] B. Himed, "MCARM/STAP data analysis," Tech. Rep. AFRL-SN-RS-TR-1999-48, Air Force Research Laboratory, Dayton, Ohio, USA, May 1999, Vol II.

[10] J. Ward, "Space-time adaptive processing for airborne radar," Tech. Rep. 1015, MIT Lincoln Laboratory, Lexington, Mass, USA, December 1994.

[11] P. G. Richardson, "STAP covariance matrix structure and its impact on clutter plus jamming suppression solutions," *Electronics Letters*, vol. 37, no. 2, pp. 118–119, 2001.

[12] Y. Hua, "Asymptotical orthonormalization of subspace matrices without square root," *IEEE Signal Processing Magazine*, vol. 21, no. 4, pp. 56–61, 2004.

[13] Y. Jung-Lang, "A novel subspace tracking using correlation-based projection approximation," *Signal Processing*, vol. 80, no. 12, pp. 2517–2525, 2000.

[14] M. Kamenetsky and B. Widrow, "A variable leaky LMS adaptive algorithm," in *Proceedings of the IEEE conference of the 38th Asilomar on Signals, Systems and Computers*, vol. 1, pp. 125–128, Pacific Grove, Calif, USA, November 2004.

[15] Y. L. Kim, S. U. Pillai, and J. R. Guerci, "Optimal loading factor for minimal sample support space-time adaptive radar," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '98)*, vol. 4, pp. 2505–2508, Seattler, Wash, USA, May 1998.

[16] J. S. Goldstein, I. S. Reed, and P. A. Zulch, "Multistage partially adaptive stap cfar detection algorithm," *IEEE Transaction on Aerospace and Electronic Systems*, vol. 35, no. 2, pp. 645–662, 1999.

Hocine Belkacemi was born in Biskra, Algeria. He received the Engineering degree in electronics from the Institut National d'Electricité et d'Electronique (INELEC), Boumerdes, Algeria, in 1996, the Magistére degree in electronic systems from École Militaire Polytechnique, Bordj El Bahri, Algeria, in 2000, and the M.S. (DEA) degree in control and signal processing, Université de Paris-Sud XI, Orsay, France, in 2002. He is currently pursuing the Ph.D. degree in the field of signal processing at the Signals and Systems Laboratory (LSS) at Supélec, Gif-sur-Yvette. His research interests include array signal processing with application to radar, detection and estimation, and model selection.

Sylvie Marcos received the Engineer degree from the Ecole Centrale de Paris (1984) and both the Doctorate (1987) and the Habilitation (1995) degrees from the Paris-Sud XI University, Orsay, France. She is Director of Research at the National Center for Scientific Research (CNRS) and works in the Signals and Systems Laboratory (LSS) at Supélec, Gif-sur-Yvette, France. Her main research interests are presently array processing, spatiotemporal signal processing (STAP) with applications in radar and radio communications, adaptive filtering, linear and nonlinear equalizations, and multiuser detection for CDMA systems.