Australian Journal of Basic and Applied Sciences, 6(7): 15-25, 2012
ISSN 1991-8178

A New Method For Efficient Subspace Tracking In Array Signal Processing

[1]Omid Mahdiyar, [2]Ali Sadeghzadeh Sheikhan, [1]Babak Gholami

[1]Kazerun Branch, Islamic Azad University, Marvdasht, Iran.
[2]Electrical Engineering Department, Khajeh-Nasir Toosi University of Technology, Iran.

Abstract: Eigen-decomposition of data into signal and noise subspaces plays an important role in subspace based methods. Conventional methods of eigen-decomposition have high computational complexity so that they are not applicable for many real-time applications such as radar and sonar. In this paper, we review available algorithms for reducing computational complexity, and then we present a new algorithm for reducing computational complexity of signal subspace tracking.

Key words: Tracking**,** Array signal processing.

INTRODUCTION

Eigen-decomposition of data into signal and noise subspaces plays an important role in subspace based methods in signal processing. One of the most important applications of eigen-decomposition is in target tracking and direction of arrival (DOA) of targets. High resolution methods for DOA estimation such as the Multiple Signal Classification (MUSIC) algorithm, the minimum norm algorithm and many other algorithms are some of applications of Eigen-decomposition. Subspace estimation is a costly numerical task, especially in time variant applications such as sonar signal processing.

Estimation of the signal subspace is commonly based on the traditional Eigen Value Decomposition (EVD) or Singular Value Decomposition (SVD). However, the main drawback of these kinds of decompositions is their inherent complexity.

In this paper, we review the present algorithms available for reduction of computational complexity. Then, we present a new algorithm for calculation complexity reduction of signal subspace tracking.

The following section of this paper contains the mathematical model of signal and a brief introduction to DOA estimation algorithms. In the section III, a review on subspace tracking methods is presented. In section IV, the new method for subspace tracking is presented. In the section V, the efficiency of the new algorithm is compared with other algorithms using simulations.

The mathematical model of signal and brief introduction on DOA algorithms:

Mathematical Model of Signal:

In mathematical model of signal, we use samples of $\mathbf{x}(t)$ recorded during the observation of outputs of an *n* sensor array, satisfying the following model

$$\mathbf{x}(t) = \mathbf{A}(\theta)\mathbf{s}(t) + \boldsymbol{n}(t) \tag{1}$$

Where $\mathbf{x} \in C^n$ is the vector of sensor outputs, $\mathbf{s} \in C^n$ is the vector of complex signal amplitudes, $\mathbf{n} \in C^n$ is an additive noise vector, $\mathbf{A}(\theta) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_r)] \in C^{n\times r}$ is the matrix of steering vectors $\mathbf{a}(\theta_j)$, and θ_j $j = 1, \dots, r$ is the parameter of the j^{th} source, for example its DOA. It is assumed that $\mathbf{a}(\theta_j)$ is a smooth function of θ_j and the array is calibrated. Also we assume that the elements of $\mathbf{s}(t)$ are stationary random processes and the elements of $\boldsymbol{n}(t)$ are zero mean stationary random processes which are uncorrelated with the elements of $\mathbf{s}(t)$.

The covariance matrix of outputs of sensors can be written as $\mathbf{R} = E\{\mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t)\} = \mathbf{ASA}^{\mathrm{H}} + \mathbf{R}_n$, where $\mathbf{S} = E\{\mathbf{s}(t)\boldsymbol{s}^H(t)\}$ is the signal covariance matrix assumed to be nonsingular and $\mathbf{R}_n$ is the noise covariance matrix.

In practice, we have to estimate the covariance matrix using the observed data. If we have **K** snapshots of observed data, the covariance matrix can be estimated as following:

$$\hat{\mathbf{R}} = \frac{1}{K}\sum_{t=1}^{K} x(t)x^H(t) \tag{2}$$

The covariance matrix (**R**) is positive definite. Therefore, it can be decomposed to:

Corresponding Author: Omid Mahdiyar, Kazerun Branch, Islamic Azad University, Marvdasht, Iran.

$$\mathbf{R} = \mathbf{U}\boldsymbol{\Lambda}\ \mathbf{U}^H \tag{3}$$

where **U** is a unitary matrix whose columns are eigenvectors of **R**, and $\boldsymbol{\Lambda} = diag\{\lambda_1, \lambda_2, ..., \lambda_M\}$ is a diagonal matrix of sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M$. Note that the columns of **R** correspond to the sorted eigenvalues.

It can be shown that the eigenvectors of **R** that are perpendicular to **A** have the eigenvalues equal to the noise variance (σ^2). There are *M-r* independent eigenvectors where *r* is the number of targets. All of the corresponding eigenvalues of these *r* eigenvectors are greater than σ^2. So we can divide **U** into two matrices. The first one is $\mathbf{U_s}$ which contains the eigenvectors corresponding to the eigenvalues greater than σ^2 ($\lambda_1 \geq ... \geq \lambda_r > \sigma^2$) and the second one is $\mathbf{U_n}$ corresponding to the $\lambda_{r+1} = ... = \lambda_M = \sigma^2$. So the covariance matrix can be rewritten as:

$$\mathbf{R} = \mathbf{U}_s \boldsymbol{\Lambda}_s \mathbf{U}_s^H + \mathbf{U}_n \boldsymbol{\Lambda}_n \mathbf{U}_n^H \tag{4}$$

where $\boldsymbol{\Lambda}_s$ is a diagonal matrix whose *r* eigenvalues are greater than σ^2. Also $\boldsymbol{\Lambda}_n$ is a diagonal matrix with diagonal elements equal to σ^2. The columns of $\mathbf{U_n}$ span the noise subspace and the columns of $\mathbf{U_s}$ span the signal subspace.

We assume that *r* is known but in general case *r* must be estimated. There are several methods for estimating *r* (H. Krim, 1994; B.D. Van Veen and K.M. Buckley, 1988).

1.1. A Brief Introduction on DOA Estimation Algorithms:

The DOA estimation algorithms are used in sonar signal processing, radar signal processing and communication. The high resolution DOA estimation algorithms rely on the covariance matrix of array's outputs and its subspace decomposition. The followings are the most famous algorithms of high resolution DOA estimation.

1.1.1. MUltiple SIgnal Classification (MUSIC):

MUSIC is one of the most famous DOA estimation algorithms based on covariance matrix decomposition. In 1-D DOA estimation, we are looking for the angle of target (θ). The estimator is:

$$\hat{\theta}_{MUSIC} = \arg\max\{\frac{1}{a^H(\theta)\mathbf{U}_n\mathbf{U}_n^H a(\theta)}\} \tag{5}$$

where $\mathbf{U}_n$ is the noise subspace and $a(\theta)$ is the direction vector for each angle. The MUSIC is applicable for every arrangement of sensors. The computational complexity of MUSIC is reasonable.

1.1.2. Eigenvector Algorithm:

This algorithm is very similar to MUSIC. In this algorithm, each eignvector is multiplied by its corresponding eigenvalue. Therefore, the fake peaks (fake targets) in this method are less than MUSIC.

The estimator equation of this algorithm is as following:

$$\hat{\theta}_{EV} = \arg\max\{\frac{1}{a^H(\theta)\mathbf{V}_n a(\theta)}\} \tag{6}$$

where $\mathbf{V}_n = \sum_{I=r+1}^{M} \frac{1}{\lambda_i} \mathbf{U}_i \mathbf{U}_i^H$ and $\mathbf{U}_i$ is the *i*-th eigenvector of **R**.

2. Review of Subspace Tracking Methods:

As mentioned in the previous section, there are many DOA estimation algorithms relied on subspace decomposition.

The most common methods for subspace estimation are eigenvalue decomposition (EVD) and singular value decomposition (SVD). The main weak point of these conventional methods is their heavy computational complexity. Assume we have an *n*-sensor array and we are interested to estimate the DOA of *r* targets which $n >> r$. The EVD and SVD have $O(n^3)$ computations in each subspace estimation. Thus, these algorithms are too time-consuming and they are not suitable for real-time applications.

In this paper, we present a new algorithm for efficiently estimating of subspaces.

2.1. Karasalo Method (I. Karasalo, 1986):

Assume that the x is the output of sensors so the data covariance matrix is:

$$\mathbf{R}_{xx} = E\{xx^H\} \tag{7}$$

If the data contains the signal of r sources uncorrelated with noise, we can rewrite the covariance matrix as:

$$\mathbf{R}_{xx} = \mathbf{R}_{ss} + \mathbf{R}_{nn} \tag{8}$$

Where $\mathbf{R}_{ss}$ and $\mathbf{R}_{nn}$ are the signal and the noise covariance matrices, respectively. $\mathbf{R}_{ss}$ is semi-positive definite and $rank(\mathbf{R}_{ss}) \le r$. Moreover, we can rewrite the noise covariance matrix as $\mathbf{R}_{nn} = P_n \overline{\mathbf{R}}_{nn}$ where P_n is an unknown scalar, and $\overline{\mathbf{R}}_{nn}$ is a semi-positive definite matrix (for the uncorrelated noise $\overline{\mathbf{R}}_{nn} = \mathbf{I}$). So we have:

$$\mathbf{R}_{xxm} = a_m x_m x_m^H + b_m \mathbf{R}_{xx(m-1)} \quad ; m = 1,2,... \tag{9}$$

where a_m and b_m are specific positive scalars. Our aim is renewing P_n and $\mathbf{R}_{ss}$ at each stage. If we assume that P_n and $\mathbf{R}_{ss}$ are known at m=0, we have to renew these parameters for new observations:

$$\mathbf{R}_{xx1} = a_1 x_1 x_1^H + b_1 \mathbf{R}_{xx0} \tag{10}$$

In Karasalo method, the parameters P_n and $\mathbf{R}_{ss}$ are the solution of following minimization problem:

$$\{\hat{P}_{n1}, \hat{\mathbf{R}}_{ss1}\} = \arg \min_{P_{n1}, \mathbf{R}_{ss1}} \{\left\| \mathbf{L}_n^{-1}(\mathbf{R}_{xx1} - \mathbf{R}_{ss} - P_n \overline{\mathbf{R}}_{nn})\mathbf{L}_n^{-H} \right\|_E\} \tag{11}$$

where $\mathbf{L}_n$ is the Cholesky factor for $\overline{\mathbf{R}}_{nn}$, i.e. $\overline{\mathbf{R}}_{nn} = \mathbf{L}_n \mathbf{L}_n^H$. Also $\|.\|_E$ stands for Euclidian norm of matrix.

It can be shown that the solution of equation (11) leads to the solution of following eigenvalue problem:

$$\mathbf{L}_n^{-1} \mathbf{R}_{xx1} \mathbf{L}_n^{-H} u = s^2 u \tag{12}$$

where u_i and s_i^2 ; i=1,2,..,n are the eigenvectors and eigen values, respectively. The optimum solution of (11) is:

$$P_{n1} = (1 - \frac{1}{M-r}) s_{r+2}^2 + \frac{1}{M-r} s_{r+1}^2$$
$$\mathrm{R}_{ss1} = L_n (\sum_{i=1}^{r} (s_i^2 - P_{n1}) u_i u_i^H) L_n^H \tag{13}$$

The computational complexity of this method is of order $O(nr^2)$.

2.2. The Approximated Power Iteration (API) Algorithm:

The power iteration method tracks the signal covariance (C). In this method we have:

$$\mathbf{C}_{xy}(t) = \mathbf{C}(t)\mathbf{W}(t-1) \tag{14}$$

$$\mathbf{W}(t) = f(\mathbf{C}_{xy}(t)) \tag{15}$$

where $f(.)$ represents the orthogonal function of bases. Note that the calculation of $\mathbf{C}_{xy}(t)$ needs n^2r computations and the estimation of **W** has $O(nr^2)$ computations. Therefore, this method is suitable for real time applications. Some suggestions are presented in (R. Badeau, *et al.,* 2005) to reduce the computational complexity of API.

The renewing steps of signal subspace using API are shown in Table.1. The initial values for **W** and **Z** are:

$$W(0) = \begin{bmatrix} I_r \\ 0_{(n-r)\times r} \end{bmatrix} \tag{16}$$

$$Z(0) = I_r \tag{17}$$

Table 1: The API Algorithm

For each time step do
$y(t) = W^H(t-1)x(t)$
$h(t) = Z(t-1)y(t)$
$g(t) = \frac{h(t)}{\beta + y^H(t)h(t)}$
$e(t) = x(t) - W(t-1)y(t)$
$\Theta(t) = (I_r + \Vert e(t)\Vert^2 g(t)g^H(t))^{\frac{-1}{2}}$
$Z(t) = \frac{1}{\beta}\Theta^H(t)(I_r - g(t)y^H(t))Z(t-1)\Theta^{-H}(t)$
$W(t) = (W(t-1) + e(t)g^H(t))\Theta^H(t)$

2.3Projection Approximation Subspace Tracking (PAST):

Before 1995, the computational order of proposed algorithms were $O(nr^2)$ or higher. But the era of very fast algorithms began by introducing the PAST.

If x is the output of sensors, its covariance matrix will be $\mathbf{C} = E\{xx^H\}$. Now assume the following cost function:

$$J(\mathbf{W}) = E\{\Vert x - \mathbf{W}\mathbf{W}^H x\Vert^2\} \tag{18}$$

where **W** is a $M \times r$ matrix and r is the number of sources. It can be shown that equation (18) has a global minimum and the signal subspace minimize this cost function (B. Yang, 1995). The gradient method can be used for renewing the **W**:

$$\mathbf{W}(t) = \mathbf{W}(t-1) + \mu[x(t) - \mathbf{W}(t-1)y(t)]y^H(t) \tag{19}$$

where $y(t) = \mathbf{W}^H(t-1)x(t)$, an estimation of J(**W**) is:

$$\hat{J}(\mathbf{W}(t)) = \sum_{i=1}^{t} \beta^{t-i} \Vert x(i) - \mathbf{W}(t)\mathbf{W}^H(t)x(i)\Vert^2 \tag{20}$$

where β is the forgetting factor. The core idea of PAST algorithm is to replace $\mathbf{W}^H(t)x(i)$ by $y(i) = \mathbf{W}^H(i-1)x(i)$. Therefore, the cost function is:

$$J'(\mathbf{W}(t)) = \sum_{i=1}^{t} \beta^{t-i} \Vert x(i) - \mathbf{W}(t)y(i)\Vert^2 \tag{21}$$

This approximation converts the problem to a weighted quadratic minimization problem. It can be shown that the solution of (21) is:

$$\mathbf{W}(t) = \mathbf{C}_{xy}(t)\mathbf{C}_{yy}^{-1}(t) \tag{22}$$

where,

$$\mathbf{C}_{xy}(t) = \beta\mathbf{C}_{xy}(t-1) + x(t)y^H(t)$$

$$\mathbf{C}_{yy}(t) = \beta\mathbf{C}_{yy}(t-1) + y(t)y^H(t) \tag{23}$$

We can use RLS to implement PAST. The PAST algorithm is shown in Table.2.

The solution of ***J*(W)** has orthogonal bases but the solution of ***J'*(W)** are not orthogonal. The computational complexity order is *O(nr).*

3.The Proposed Algorithm for Subspace Tracking:

The non-orthogonality of bases of signal subspace in PAST is one of its disadvantages. We propose a new algorithm called Constrained Projection Approximation Subspace Tracking (CPAST) to overcome this disadvantage.

In CPAST we are looking for the solution of the following optimization problem:

$$\underset{\mathbf{W}}{\text{minimize}} \quad J'(\mathbf{W}(t)) = \sum_{i=1}^{t} \beta^{t-i} \left\| \mathbf{x}(i) - \mathbf{W}(t)\mathbf{y}(i) \right\|^2$$

$$\textit{subject to} \quad \mathbf{W}^H(t)\mathbf{W}(t) = \mathbf{I}_r \qquad (24)$$

Table 2: the steps of PAST algorithm

For each time step do
$y(t) = W^H(t-1)x(t)$
$h(t) = P(t-1)y(t)$
$g(t) = \dfrac{h(t)}{\beta + y^H(t)h(t)}$
$P(t) = \dfrac{1}{\beta}[P(t-1) - g(t)h^H(t)]$
$e(t) = x(t) - W(t-1)y(t)$
$W(t) = W(t-1) + e(t)g^H(t)$

The constraint in (24) guarantees the orthogonality of bases. The optimization problem in (24) can be solved by Lagrange multipliers method:

$$\underset{\mathbf{W}}{\text{minimize}} \quad h(\mathbf{W}) = tr(\mathbf{C}) - 2tr(\sum_{i=1}^{t} \beta^{t-i}\mathbf{x}(i)\mathbf{y}^H(i)\mathbf{W}^H(t)) +$$
$$tr(\sum_{i=1}^{t} \beta^{t-i}\mathbf{y}(i)\mathbf{y}^H(i)\mathbf{W}^H(t)\mathbf{W}(t)) + \lambda\left\|\mathbf{W}^H\mathbf{W} - \mathbf{I}_r\right\|_F^2 \qquad (25)$$

where *tr(.)* and $\|.\|_F$ represent the matrix trace and Frobenius norm of matrix, respectively. For solving (25), the gradient of (25) must be equal to zero, so:

$$-\sum_{i=1}^{t} \beta^{t-i}\mathbf{x}(i)\mathbf{y}^H(t) + \sum_{i=1}^{t} \beta^{t-i}\mathbf{W}(t)\mathbf{y}(i)\mathbf{y}^H(t) +$$
$$\lambda[-2\mathbf{W}(t) + 2\mathbf{W}(t)\mathbf{W}^H(t)\mathbf{W}(t)] = \mathbf{0} \qquad (26)$$

This equation can be rewritten as:

$$\mathbf{W}(t) = (\sum_{i=1}^{t} \beta^{t-i}\mathbf{x}(i)\mathbf{y}^H(i)) \times$$
$$[\sum_{i=1}^{t} \beta^{t-i}\mathbf{y}(i)\mathbf{y}^H(i) - 2\lambda\mathbf{I}_r + 2\lambda\mathbf{W}^H(t)\mathbf{W}(t)]^{-1} \qquad (27)$$

If we substitute (27) in the constraint equation $\mathbf{W}^H\mathbf{W} = \mathbf{I}_r$, we have:

$$[(\sum_{i=1}^{t} \beta^{t-i}\mathbf{y}(i)\mathbf{x}^H(i))(\sum_{i=1}^{t} \beta^{t-i}\mathbf{x}(i)\mathbf{y}^H(i))] =$$
$$[\sum_{i=1}^{t} \beta^{t-i}\mathbf{y}(i)\mathbf{y}^H(i) - 2\lambda\mathbf{I}_r + 2\lambda\mathbf{W}^H(t)\mathbf{W}(t)]^2 \qquad (28)$$

We define the matrix **L** as following:

$$\mathbf{L} = \sum_{i=1}^{t} \beta^{t-i} \mathbf{y}(i)\mathbf{y}^H(i) - 2\lambda \mathbf{I}_r + 2\lambda \mathbf{W}^H(t)\mathbf{W}(t) \tag{29}$$

By substituting **L** in (28):

$$\mathbf{L} = [(\sum_{i=1}^{t} \beta^{t-i} \mathbf{y}(i)\mathbf{x}^H(i))(\sum_{i=1}^{t} \beta^{t-i} \mathbf{x}(i)\mathbf{y}^H(i))]^{\frac{1}{2}}$$
$$= [\mathbf{C}_{\mathbf{xy}}^H(t)\mathbf{C}_{\mathbf{xy}}(t)]^{\frac{1}{2}} \tag{30}$$

where $[.]^{1/2}$ represents the square root of matrix. It can be shown that:

$$\mathbf{W}(t) = \mathbf{C}_{\mathbf{xy}}(t)(\mathbf{C}_{\mathbf{xy}}^H(t)\mathbf{C}_{\mathbf{xy}}(t))^{\frac{-1}{2}} \tag{31}$$

The equation (31) guarantees the orthogonally of subspace bases.

The computational complexity order of (31) is nr^2 and it is greater than that of some algorithms such as PAST. In order to reduce the computational complexity of CPAST, we present an adaptive algorithm for renewing the signal subspace matrix.

We can rewrite (31) as following:

$$\mathbf{W}(t) = \mathbf{C}_{xy}(t)\mathbf{\Phi}(t) \tag{32}$$

Where:

$$\mathbf{\Phi}(t) = [\mathbf{C}_{xy}^H(t)\mathbf{C}_{xy}(t)]^{\frac{-1}{2}} \tag{33}$$

Multiplying $\mathbf{\Phi}(t)$ by (23) from right side and then using (31), we have:

$$\mathbf{W}(t) = \beta \mathbf{W}(t-1)\mathbf{\Phi}^{-1}(t-1)\mathbf{\Phi}(t) + \mathbf{x}(t)\mathbf{y}^H(t) \tag{34}$$

Substituting (32) in (23) and then multiplying by $\mathbf{W}^H(t-1)$ from left side leads to:

$$\mathbf{\Phi}^{-1}(t) = \beta \mathbf{\Phi}^{-1}(t-1) + \mathbf{y}(t)\mathbf{y}^H(t) \tag{35}$$

For reducing the computational complexity, we use Matrix Inversion Lemma (MIL) which is:

$$(\mathbf{A} + \mathbf{BCD})^{-1} =$$
$$\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1})^{-1}\mathbf{D}\mathbf{A}^{-1} \tag{36}$$

Using MIL, we have:

$$[\mathbf{\Phi}^{-1}(t)]^{-1} = \frac{1}{\beta}\mathbf{\Phi}(t-1)(\mathbf{I}_r - \mathbf{y}(t)\mathbf{g}^H(t)) \tag{37}$$

Where:

$$\mathbf{g}(t) = \frac{\mathbf{y}^H(t)\mathbf{\Phi}(t-1)}{\beta + \mathbf{y}^H(t)\mathbf{\Phi}(t-1)\mathbf{y}(t)} \tag{38}$$

Multiplying $\mathbf{\Phi}^{-1}(t)$ by (37) from left, we have:

$$\mathbf{I}_r = \frac{1}{\beta}\mathbf{\Phi}^{-1}(t)\mathbf{\Phi}(t-1)(\mathbf{I}_r - \mathbf{y}(t)\mathbf{g}^H(t)) \tag{39}$$

By inversing both sides of (39):

$$\mathbf{I}_r = \beta(\mathbf{I}_r - \mathbf{y}(t)\mathbf{g}^H(t))^{-1}\mathbf{\Phi}^{-1}(t-1)\mathbf{\Phi}(t) \tag{40}$$

So, the renewing equation for **W** is:

$$\mathbf{W}(t) = \mathbf{W}(t-1) - (\mathbf{W}(t-1)\mathbf{y}(t))\mathbf{g}^H(t) +$$
$$\mathbf{x}(t)\mathbf{y}^H(t)\mathbf{\Phi}(t) \tag{41}$$

Using (41) for renewing **W**, reduces the computational complexity order of CPAST to *nr*.

To compute $\mathbf{\Phi}(t)$ efficiently, we can use the following equations:

$$\alpha = \mathbf{x}^H(t)\mathbf{x}(t) \tag{42}$$

$$\mathbf{U}(t) = \beta\mathbf{\Psi}^H(t-1)(\mathbf{C}_{xy}^H(t-1)\mathbf{x}(t))\mathbf{y}^H(t) \tag{43}$$

$$\begin{aligned}\mathbf{\Omega}(t) &= \beta^2\mathbf{\Psi}^H(t-1)\mathbf{\Omega}(t-1)\mathbf{\Psi}(t-1) + \\ &\mathbf{U}(t) + \mathbf{U}^H(t) + \alpha\mathbf{y}(t)\mathbf{y}^H(t)\end{aligned} \tag{44}$$

$$\mathbf{\Phi}(t) = \mathbf{\Omega}^{\frac{-1}{2}}(t) \tag{45}$$

The algorithm of CPAST is shown in Table.3. All of equations are developed for exponentially window. This window is suitable for stationary signals. The sliding (truncated) window is proper for non-stationary signals.

If we use the sliding window, the covariance matrix renewing is as:

$$\begin{aligned}\mathbf{C}_{xx}(t) &= \sum_{i=t-l+1}^{t}\beta^{t-i}\mathbf{x}(i)\mathbf{x}^H(i) = \\ &\beta\mathbf{C}_{xx}(t-1) + \mathbf{x}(t)\mathbf{x}^H(t) - \beta^l\mathbf{x}(t-l)\mathbf{x}^H(t-l) \\ &= \beta\mathbf{C}_{xx}(t-1) + \mathbf{z}(t)\mathbf{G}\mathbf{z}^H(t)\end{aligned} \tag{46}$$

Where, l is the window length and z and G are defined as:

$$\mathbf{z}(t) = \left[\mathbf{x}(t) \quad \vdots \quad \mathbf{x}(t-l)\right]_{n\times 2} \tag{47}$$

$$G = \begin{bmatrix}1 & 0 \\ 0 & -\beta^l\end{bmatrix}_{2\times 2} \tag{48}$$

3.Performance Evaluation of CPAST Algorithm:

In order to evaluate the performance of CPAST and comparing it with other algorithms we need a criterion. The most common criterion is the Maximum Principal Angle (MPA) which shows the distance between subspaces. The subspaces are similar if MPA be close to zero.

Table 3: CPAST Algorithm for exponential window

$\mathbf{W}(0) = \begin{bmatrix}\mathbf{I} \\ \cdots \\ \mathbf{0}\end{bmatrix}$; $\mathbf{C}_{\mathbf{xy}}(0) = \begin{bmatrix}\mathbf{I} \\ \cdots \\ \mathbf{0}\end{bmatrix}$;
$\mathbf{\Phi}(0) = \mathbf{\Omega}(0) = \mathbf{I}$
FOR $t = 1,2,\ldots$ DO
$\mathbf{y}(t) = \mathbf{W}^H(t-1)\mathbf{x}(t)$
$\mathbf{C}_{\mathbf{xy}}(t) = \beta\mathbf{C}_{\mathbf{xy}}(t-1) + \mathbf{x}(t)\mathbf{y}^H(t)$
$\mathbf{U}(t) = \beta(\mathbf{C}_{xy}^H(t-1)\mathbf{x}(t))\mathbf{y}^H(t)$
$\mathbf{\Omega}(t) = \beta^2\mathbf{\Omega}(t-1) + \mathbf{U}(t) + \mathbf{U}^H(t) +$
$\mathbf{y}(t)(\mathbf{x}^H(t)\mathbf{x}(t))\mathbf{y}^H(t)$
$\mathbf{\Phi}(t) = \mathbf{\Omega}^{\frac{-1}{2}}(t)$ $\quad \mathbf{g}(t) = \dfrac{\mathbf{y}^H(t)\mathbf{\Phi}(t-1)}{\beta + \mathbf{y}^H(t)\mathbf{\Phi}(t-1)\mathbf{y}(t)}$
$\mathbf{W}(t) = \mathbf{W}(t-1) - (\mathbf{W}(t-1)\mathbf{y}(t))\mathbf{g}^H(t) +$
$\mathbf{x}(t)(\mathbf{y}^H(t)\mathbf{\Phi}(t))$

2.3.Simulations:

2.2.1.The First Simulation:

Assume we have a 21-sensor array and two targets with same SNR (SNR=10). One of these targets is located in -50^{o} and another one is located in +50^{o}. Also, assume that the forgetting factor equals to 0.99. The maximum principal angle for this scenario is shown in Fig.1.

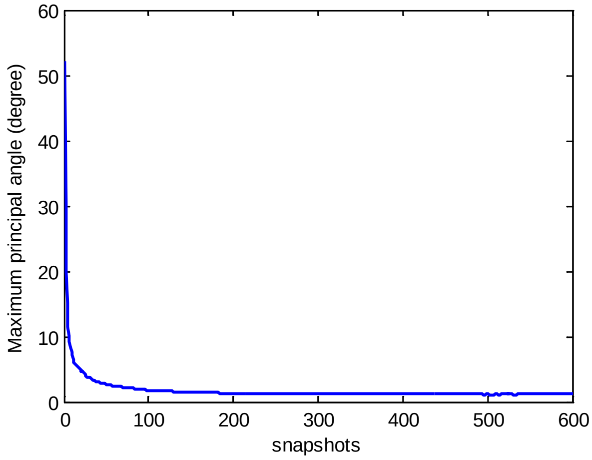

Fig. 1: estimation of MPA using CPAST

2.2.1The Second Simulation:

Now assume that there are three moving targets. The paths of targets are shown in Fig.2. The forgetting factor in this scenario is 0.75. The estimated maximum principal angle for this example is shown in Fig.3. As shown in Fig.3, the CPAST gives a good estimation of signal subspace. The maximum principal angle has a peak at angular intersection snapshot. That is, the subspace estimation is difficult at angular intersection snapshots.

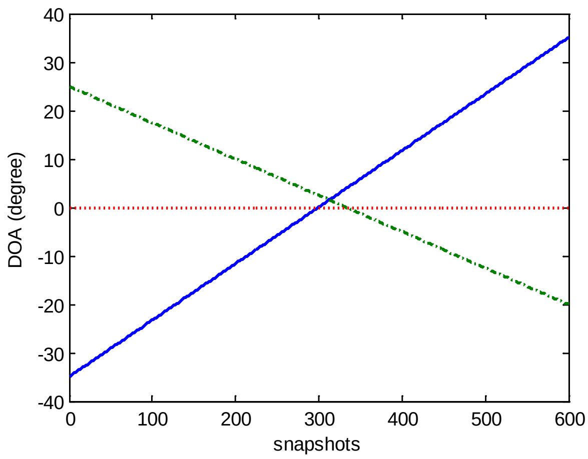

Fig. 2: paths of 3 simulated moving targets

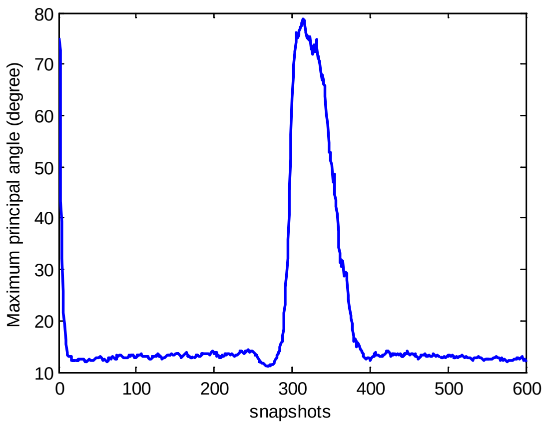

Fig. 3: maximum principal angle for three moving targets.

2.3Comparison of Various Subspace Tracking with CPAST Based on MPA Ratio:

There are various algorithms for subspace tracking and different criteria for comparing these algorithms. We will compare CPAST with PAST (B. Yang, 1995), BISVD1 (P. Strobach, 1997), PROTEUS1 (B. Champagne, 1994) and API (R. Badeau, et al., 2005) in this sub-section. The quantity we use for comparing CPAST with each of the other algorithms is the ratio of CPAST's MPA to the MPA of each algorithm (in dB) , as following:

$$20\log(\frac{\theta_{CPAST2}}{\theta_{a\lg}}) \quad (49)$$

Where, θ_{CPAST2} and $\theta_{a\lg}$ are the MPA of CPAST and MPA of that algorithm, respectively.

In Fig.4, the ratio of MPA of CPAST and PAST is shown.

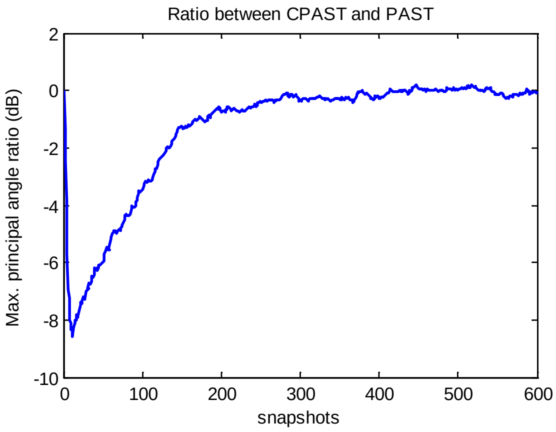

Fig. 4: The ratio between the MPA of CPAST and PAST

In Fig.5, the ratio of MPA of CPAST and BISVD1 is shown.

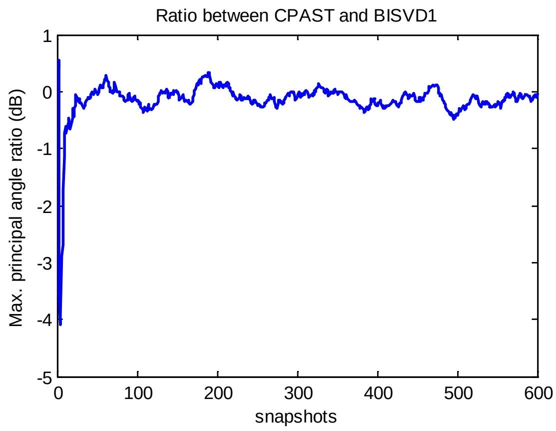

Fig. 5: The ratio between the MPA of CPAST and BISVD1

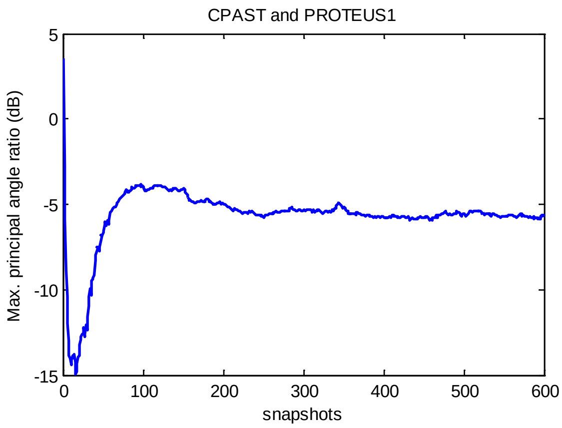

Fig. 6: The ratio between the MPA of CPAST and PROTEUS

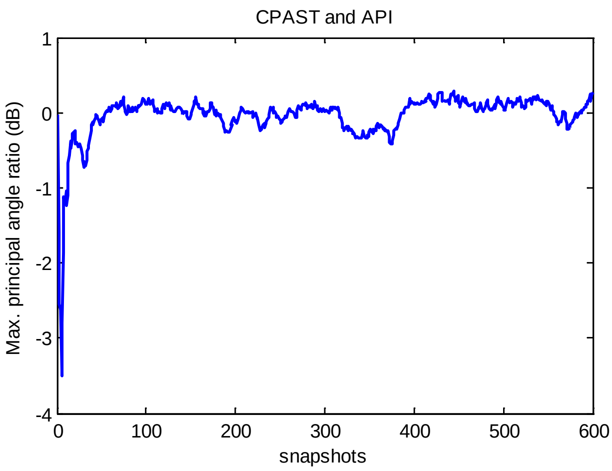

Fig. 7: The ratio between the MPA of CPAST and API

As it is shown in above figures, the CPAST converged more rapidly than PAST, BISVD1, PROTEUS1 and API. Moreover, the performance of CPAST is better than PAST and PROTEUS1.

2.3Comparison of Various Subspace Tracking with CPAST Based on the Orthonormality Error of Bases:

For comparing the orthonormality of bases we use the following criterion:

$$20\log\left(\left\|\mathbf{W}^H(t)\mathbf{W}(t)-\mathbf{I}_r\right\|_F\right) \qquad (50)$$

Now assume two targets at angles -50° and 50°.

The deviation from orthonormality of signal subspace bases for CPAST and PAST are shown in Fig. 8. The CPAST have the better performance than PAST based on orthonormality. The deviation from orthonormality for other algorithms is presented in Table. 4. The CPAST has the better performance in orthonormality aspect.

2.3.Comparison of Various Subspace Tracking with CPAST Based on Computational Complexity:

The comparison of the computational complexity of various algorithms is shown in Table. 5. Also, the capability of these algorithms in estimating the number of targets is shown in Table. 6.

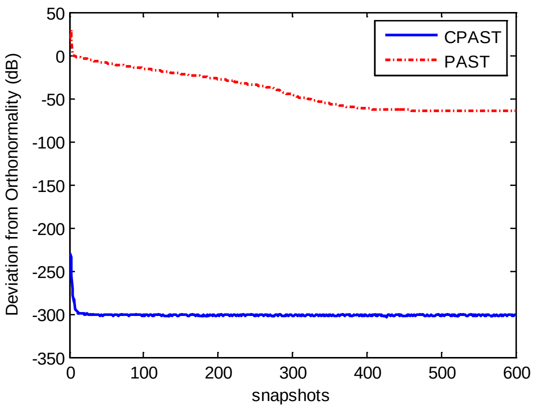

Fig. 8: The deviation from orthonormality of signal subspace bases for CPAST and PAST

Table 4: The orthonormality error for different algorithms

Algorithm	Orthonormality error
CPAST, BISVD1	about -300 dB
API	about -285 dB
PROTEUS1	about -265 dB
PAST	about -30 dB

Table 5: The computational complexity for different algorithms

Algorithm	Cost (MAC count)
CPAST2	$7nr+O(r^2)+O(r^3)$
KARASALO	$nr^2+3nr+2n+O(r^2)+O(r^3)$
PAST	$3nr+2r^2+O(r)$
BISVD1	$nr^2+3nr+2n+O(r^2)+O(r^3)$
PROTEUS1	$(3/4)nr^2+(15/4)nr+O(n)+O(r)+O(r^2)$
API	$nr^2+3nr+n+O(r^2)+O(r^3)$

Table 6: The capability of algorithms in estimating

Algorithm	Estimation of number of sources
CPAST2	*Yes*
KARASALO	*No*
PAST	*No*
BISVD1	*Yes*
PROTEUS1	*No*
API	*No*

Conclusion:

Estimation of covariance matrix and its decomposing into signal and noise subspaces are most interested in many fields of array signal processing. In many real-time applications we need fast algorithms to update the covariance matrix or track the signal subspace but the main issues of some algorithms are computational complexity, convergence speed and orthonormality of subspace bases. In this paper we reviewed some algorithms and then proposed a new algorithm for signal subspace tracking. In section 5 we simulated some scenario and compared our new algorithm (CPAST) with other algorithms.

CPAST has the better convergence speed and its computational complexity is less than other algorithms. One of the main advantages of CPAST is its guaranteed orthonormality of subspace bases. In addition to the above advantages, CPAST has the capability of estimation of number of targets.

REFERENCES

Badeau, R., B. David, and G. Richard, 2005. "Fast approximated power iteration subspace tracking," *IEEE Trans. Signal Process*., 53(8): 2931-2941.

Champagne, B., 1994. "Adaptive eigendecomposition of data covariance matrices based on first-order perturbations," *IEEE Trans. Signal Process*., 42(10): 2758-2771.

Karasalo, I., 1986. "Estimating the covariance matrix by signal subspace averaging," *IEEE Trans. Acoust., Speech, Signal Process*., ASSP-34(1): 8-12.

Krim, H., 1994. "A Data-Based Enumeration of Fully Correlated Signals," I*EEE Trans. Acoust., Speech, Signal Processing*.

Strobach, P., 1997."Bi-iteration SVD subspace tracking algorithms," *IEEE Trans. Signal Process*., 45(5): 1222-1240.

Van Veen B.D. and K.M. Buckley, 1988. "Beamforming: A Versatile Approach to Spatial Filtering," *IEEE ASSP Magazine*, pp: 4-24.

Yang, B., 1995. "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, 44(1): 95-107.

Yang, B., 1995. "Projection approximation subspace tracking," *IEEE Trans. Signal Process*., 43(1): 95-107.