ELSEVIER

Signal Processing 48 (1996) 111–121

SIGNAL PROCESSING

Fast algorithms for direction-of-arrival finding using large ESPRIT arrays

Hongyuan Zha*

Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA

Received 27 January 1995; revised 31 May 1995 and 4 October 1995

Abstract

To explore the redundancy in large ESPRIT arrays, we propose fast algorithms for direction-of-arrival (DOA) finding without computing (partial) eigendecompositions. For batch processing, the algorithms have computational complexity $O(M^2d)$ for the covariance matrix case and $O(MNd)$ for the data matrix case, where M is the number of sensors, d the number of signals and N the number of snapshots. For a real-time implementation of the proposed algorithms, the computational complexity is $O(Md) + O(d^3)$ per data vector update. We also present numerical simulation results to illustrate the efficiency of the proposed algorithms.

Zusammenfassung

Wir untersuchen die Rendundanz in großen ESPRIT-Arrays und schlagen schnelle Algorithmen zur Direction-of-Arrival (DOA) Schätzung ohne Berechnung von (partiellen) Eigenzerlegungen vor. Unter Annahme einer Blockverarbeitung haben die Algorithmen einen Berechnungsaufwand von $O(M^2d)$ im Kovarianzmatrix-Fall und $O(MNd)$ im Datenmatrix-Fall, wobei M die Anzahl der Sensoren, d die Anzahl der Signale und N die Anzahl der Snapshots bedeuten. Unter Annahme einer Echtzeit-Implementierung der vorgeschlagenen Algorithmen ist der Berechnungsaufwand pro Datenvektor-Update durch $O(Md) + O(d^3)$ gegeben. Die Effizienz der vorgeschlagenen Algorithmen wird durch Ergebnisse numerischer Simulationen illustriert.

Résumé

Explorant la redondance dans les grands réseaux ESPRIT, nous proposons des algorithmes rapides d'estimation de la direction d'arrivée (DDA) sans calculer les décompositions en valeurs propres (partielles). Pour le traitement en temps différé, l'algorithme a une complexité de calcul de $O(M^2d)$ pour le cas de la matrice de covariance, et de $O(MNd)$ pour le cas de la matrice de donnée, où M est le nombre de senseurs, d le nombre de signaux et N le nombre de prises du signal. Pour une implantation en temps réel de l'algorithme proposé, la complexité de calcul est de $O(Md) + O(d^3)$ par mise à jour du vecteur du donnée. Nous présentons également des résultats de simulations numériques afin d'illustrer l'efficacité des agorithmes proposés.

Keywords: Array signal processing; Direction-of-arrival; ESPRIT; Eigenvalue decomposition; Singular value decomposition; QR decomposition; Fast algorithms; Updating

---

* E-mail: zha@cse.psu.edu. The work of this author was supported in part by NSF grant CCR-9308399.

SSDI 0165-1684(95)00128-X

1. Introduction

In the past decade, subspace-based methods for direction-of-arrival (DOA) finding have attracted considerable attention [5–7]. The concepts, techniques and algorithms from numerical linear algebra have played a fundamental role in the development of subspace-based algorithms, especially in the computational and implementational aspects. A common process in those subspace-based algorithms is the computation of the so-called signal subspace which is usually done through eigenvalue decomposition (EVD) of the covariance matrix of the sensor array output or the singular value decomposition (SVD) of the data matrix that consists of the snapshots of the sensor array output. EVD and SVD are rather computationally costly matrix operations, especially when the dimensions of the underlying matrices are large. Many efforts have been directed towards deriving fast algorithms for computing EDV, SVD and their approximate variations that are less computationally demanding. We only mention the *fast subspace decomposition* method that employs the low-rank plus multiple-of-identity structure of the sensor array output covariance matrix [12–14] and the subspace tracking algorithm using updatable two-sided orthogonal transformations [8, 9]. For a more complete list of various fast algorithms, the reader is referred to [1, 2]. Recently, several other algorithms have been proposed that use the redundancy in large sensor arrays to reduce the computational cost in DOA finding. Those algorithms require much simpler matrix operations than EVD or SVD and can be easily implemented in real-time [3, 11, 15]. In this paper, we will explore this line of ideas in the context of large ESPRIT arrays. The proposed fast algorithms rely on simpler matrix operations such as (partial) QR decomposition and can handle certain classes of more general unknown colored noise. The greatest advantage is the low computational complexity of on-line implementations for tracking slowly-varying DOAs. The rest of the paper is organized as follows. We begin in Section 2 with a brief discussion of the signal model and the concept of signal subspaces. Then we review the total least squares (TLS) version of the ESPRIT algorithm in Section 3. The major part of the paper (Section 4) is devoted to the presentation of the new fast algorithms and some implementation details. An error analysis for the TLS step in the ESPRIT algorithm is given in Section 5 and finally numerical simulations are presented in Section 6.

2. The data model and the signal subspace

We consider a widely-used signal model in narrowband sensor array signal processing [5],

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

where $x(t) \in \mathscr{C}^M$ is the sensor array output vector, $s(t) \in \mathscr{C}^d$ is the signal vector such that $M \geqslant 2d$. The matrix $A(\theta) = [a(\theta_1), \ldots, a(\theta_d)]$ contains the array response vectors and θ_i, $i = 1, \ldots, d$, are the DOAs of the signals. Assume the additive sensor noise vector $n(t)$ is white with covariance matrix $\sigma_n^2 I_M$ and uncorrelated with the signal vector. Then, the covariance matrix of $x(t)$ is

$$R_{xx} = A(\theta)R_{ss}A(\theta)^{\mathrm{H}} + \sigma_n^2 I_M, \tag{2}$$

where R_{ss} is the covariance matrix of the signals. The objectives of sensor array signal processing are to estimate the DOAs $\{\theta_i\}_{i=1}^d$ and/or the signal waveforms $s(\cdot)$ using estimates of R_{xx} based on the sensor array outputs. The key concept in the subspace-based methods is the so-called *signal subspace* which is defined as

$$\mathscr{S} = \operatorname{span}\{a(\theta_1), \ldots, a(\theta_d)\}.$$

All subspace-based algorithms have essentially two key steps: (1) estimate the signal subspace from the noisy sensor array outputs; (2) estimate the DOAs from the estimated signal subspace [6, 7]. In the next section, we discuss how to accomplish the second step using the TLS-ESPRIT algorithm. The first step, the extraction of the signal subspace can be accomplished by computing the EVD of the sensor array output covariance matrix R_{xx} or the SVD of the data matrix [7]. In fact, let us write R_{xx} in terms of its eigenvectors:

$$R_{xx} = \sum_{i=1}^{M} \lambda_i u_i u_i^{\mathrm{H}}.$$

Assuming that the signals are non-coherent, i.e., R_{ss} is non-singular, and $A(\theta)$ is of full column rank, it is easy to see that R_{xx} has exactly $M-d$ smallest eigenvalues which are equal to σ_n^2. Now

$$R_x = \sum_{i=1}^{d} \lambda_i u_i u_i^{\mathrm{H}} + \sigma_n^2 \sum_{i=d+1}^{M} u_i u_i^{\mathrm{H}}$$
$$= \sum_{i=1}^{d} (\lambda_i - \sigma_n^2) u_i u_i^{\mathrm{H}} + \sigma_n^2 I_M .$$

Comparing with (2), we conclude that $\mathscr{S} = \mathrm{span}\{a(\theta_1), \dots, a(\theta_d)\} = \mathrm{span}\{u_1, \dots, u_d\}$. Thus, the number of signals d can be determined by the number of repeated smallest eigenvalues of R_{xx}, and the signal subspace is spanned by the eigenvectors corresponding to the d largest eigenvectors of R_{xx}. In practice, the covariance R_{xx} is estimated using the output vectors

$$R_{xx} \approx \frac{1}{N} \sum_{i=0}^{N-1} x(t_i) x(t_i)^{\mathrm{H}} = \frac{1}{N} X X^{\mathrm{H}}.$$

Therefore instead of computing the EVD of R_{xx}, we may also compute the SVD of the data matrix $X = [x(t_0), \dots, x(t_{N-1})]$.

The computation of the EVD of R_{xx} requires $\mathrm{O}(M^3)$ arithmetic operations, and the SVD of X requires $\mathrm{O}(NM^2 + M^3)$ arithmetic operations. Hence, the computational complexity is very high when M and N are large. This motivates considerable research and development in the fast algorithms of EVD, SVD and their approximate variations.

3. A brief review of ESPRIT algorithm

In this section we briefly review the ESPRIT method [6] which will be modified in Section 4 to provide more computationally efficient algorithms. The basic assumption about the array configuration in the ESPRIT method is the existence of two identical sub-arrays that are displaced by a common vector in space. The sensor outputs can be collected in two vectors (cf. (1)):

$$x(t) = A(\theta) s(t) + n_x(t),$$
$$y(t) = A(\theta) \Phi s(t) + n_y(t),$$

where $\Phi = \mathrm{diag}(e^{j\omega_0 \Delta \sin\theta_1 / c}, \dots, e^{j\omega_0 \Delta \sin\theta_d / c})$. Here Δ is the length of the displacement vector, ω_0 is the center frequency of the narrow-band signals and c is the propagation speed. We notice that the DOAs can be obtained from the diagonal elements of Φ. Write

$$z(t) = A_c s(t) + n_z(t), \tag{3}$$

where

$$z(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}, \quad A_c = \begin{pmatrix} A(\theta) \\ A(\theta)\Phi \end{pmatrix}, \quad n_z(t) = \begin{pmatrix} n_x(t) \\ n_y(t) \end{pmatrix},$$

and compute the EVD of $R_{zz} = A_c R_{ss} A_c^{\mathrm{H}} + \sigma_n^2 I$,

$$R_{zz} = E_s \,\mathrm{diag}(\lambda_1, \dots, \lambda_d) E_s^{\mathrm{H}} + \sigma^2 E_n E_n^{\mathrm{H}},$$

where the columns of E_s are the eigenvectors of R_{zz} corresponding to the d largest eigenvalues while those of E_n are associated with the remaining $M-d$ eigenvalues which turn out to be all equal to σ_n^2. Since E_s spans the signal subspace, there is a non-singular T such that

$$E_s = \begin{pmatrix} E_x \\ E_y \end{pmatrix} = \begin{pmatrix} A(\theta) \\ A(\theta)\Phi \end{pmatrix} T.$$

Hence we obtain

$$E_y = A(\theta)\Phi T = A(\theta) T T^{-1} \Phi T = E_x \psi,$$
$$\text{where } \psi = T^{-1} \Phi T, \tag{4}$$

i.e., the diagonal elements of Φ are the eigenvalues of Ψ. The above relation is sometimes called the invariance principle of the ESPRIT method. When an estimate $\hat{R}_{zz} \approx R_{zz}$ is used, we only have

$$\mathrm{span}\{\hat{E}_x\} \approx \mathrm{span}\{\hat{E}_y\},$$

where quantities with "^" are computed versions using the estimate $\hat{R}_{zz}$. To obtain an estimate of Ψ, we can use the following TLS method [4]. Compute the SVD of

$$(E_x, E_y) = U \Sigma V^{\mathrm{H}}.$$

Write the orthogonal matrix V into a block matrix

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}, \qquad V_{ij} \in \mathscr{C}^{d \times d}, \quad i, j = 1, 2.$$

Then an estimate of Ψ can be obtained as

$$\Psi \approx \hat{\Psi} = -V_{12}V_{22}^{-1}.$$

Compute the eigenvalues of $\hat{\Psi}$. Then

$$\hat{\theta}_i = \sin^{-1}(c\arg(\lambda_i(\hat{\Psi}))/(\omega_0 \Delta)), \quad i = 1, \dots, d.$$

4. New algorithms

Our new algorithms can be considered as extensions of those in [3,11,15] to the case of large ESPRIT arrays. As in the above-mentioned references, we also need to assume that the number of signals d is known beforehand. The key observation is that it is not necessary to compute the entire covariance matrix which results in reduction in computational complexity. We first consider the covariance matrix case and then the data matrix case. We also show how the algorithms can be simplified in the case of uniform linear arrays. Finally, we present an on-line implementation that has a very low computational complexity.

4.2. Covariance matrix case

To proceed, partition the matrix $A(\theta)$ as

$$A(\theta) = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}, \quad A_1 \in C^{d\times d}.$$

Assume that the array is *strongly unambiguous* such that any d rows of $A(\theta)$ are linearly dependent [3]. Then A_1 is non-singular. Partition the covariance matrix R_{zz} accordingly, $R_{zz} = (R_{ij})_{i,j=1}^{4}$. It is easy to see that the first block row of R_{zz} is

$$(A_1R_{ss}A_1^{\mathrm{H}}, \underbrace{A_1R_{ss}A_2^{\mathrm{H}}}_{R_{12}}, A_1R_{ss}\Phi^{\mathrm{H}}A_1^{\mathrm{H}}, \underbrace{A_1R_{ss}\Phi^{\mathrm{H}}A_2^{\mathrm{H}}}_{R_{14}}),$$

and R_{14} and R_{12} satisfy the relation

$$R_{14}^{\mathrm{H}} = R_{12}^{\mathrm{H}}(R_{ss}A_1^{\mathrm{H}})^{-1}\Phi(R_{ss}A_1^{\mathrm{H}}). \tag{5}$$

This relation is very much similar to the relation in (4) in the sense that it represents two bases of a linear subspace. The linear transformation that maps one basis to the other provides us with information needed to compute the DOAs. It is easy to see that the diagonal elements of Φ are the eigenvalues of

$$\Xi \equiv (R_{ss}A_1^{\mathrm{H}})^{-1}\Phi(R_{ss}A_1^{\mathrm{H}}).$$

Therefore, the invariance principle of ESPRIT can also be applied to $(R_{12}^{\mathrm{H}}, R_{14}^{\mathrm{H}})$ to obtain an estimate of Ξ. We summarize the above in the following algorithm.

Algorithm 1 (*Covariance matrix case*).

(1) Compute $\hat{R}_{12}$ and $\hat{R}_{14}$, estimates of R_{12} and R_{14}, respectively. Each of the matrices is $d\times(M-d)$, and can be computed by forming the product of two matrices of dimensions $d\times N$ and $N\times(M-d)$. The number of flops for this step is $2N(M-d)d$.[1]

(2) Compute the SVD of $(\hat{R}_{12}^{\mathrm{H}}, \hat{R}_{14}^{\mathrm{H}})$

$$(\hat{R}_{12}^{\mathrm{H}}, \hat{R}_{14}^{\mathrm{H}}) = U\Sigma V^{\mathrm{H}}.$$

The dimension of the matrix is $(M-d)\times 2d$. In this step, we only need to compute the singular values and the right singular vectors which cost $\min\{8(M-d)d^2 + 32d^3,\ 4(M-d)d^2 + 40d^3\}$ flops.

(3) Partition the right singular vector matrix V as

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}, \qquad V_{ij} \in \mathscr{C}^{d\times d}, \quad i,j = 1,2,$$

and an estimate of Ξ can be obtained as

$$\Xi \approx \hat{\Xi} \equiv -V_{12}V_{22}^{-1}.$$

(4) Compute the eigenvalues of $\hat{\Xi}$ and estimate the DOAs as

$$\hat{\theta}_i = \sin^{-1}(c\arg(\lambda_i(\hat{\Xi}))/(\omega_0 \Delta)), \quad i = 1, \dots, d.$$

The combined cost of steps (3) and (4) is $11d^3$ flops.

In summary, the total cost of Algorithm 1 is $2N(M-d)d + \min\{8(M-d)d^2 + 32d^3, 4(M-d)d^2 + 40d^3\} + 11d^3$. In contrast, for the TLS-ESPRIT, the computation of the covariance matrix R_{zz} alone costs $4NM^2$ flops. The computation of the EVD of R_{zz} will add an extra $40M^3$ flops. Even if using FSD of [13], which

[1] We count each complex multiplication as one *flop*, and ignore the additions.

reduces EVD part of the computation to $4M^2(d+1) + 2M(d^2 + 7d + 5)$. The dominant cost of $4NM^2$ is still an order of magnitude higher.

4.2. Data matrix case

We can actually operate directly on the data matrix without forming the covariance matrix. Let the data matrix be $\tilde{Z} = (z(t_0), \dots, z(t_{N-1}))$ (compare Eq. (3)). An estimate of the covariance matrix R_{zz} can be computed as

$$\hat{R}_{zz} = \frac{1}{N}\sum_{i=0}^{N-1} z(t(i))z(t(i))^{\mathrm{H}} = \frac{1}{N}\tilde{Z}\tilde{Z}^{\mathrm{H}}.$$

Let $Z = \tilde{Z}/\sqrt{N}$. Compute the QR decomposition of the first d columns of Z^{H} and propagate the transformation to the rest of the columns such that

$$Z^{\mathrm{H}} = Q\begin{pmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ 0 & W_{22} & W_{23} & W_{24} \end{pmatrix},$$

where $W_{11} \in \mathscr{C}^{d\times d}$, $W_{12} \in \mathscr{C}^{d\times(M-d)}$, $W_{13} \in \mathscr{C}^{d\times d}$, $W_{14} \in \mathscr{C}^{d\times(M-d)}$, and Q is a unitary matrix. It can be easily verified that $\hat{R}_{12}$ and $\hat{R}_{14}$, the (1, 2) and (1, 4) blocks of $\hat{R}_{zz}$, can be represented as

$$\hat{R}_{12} = W_{11}^{\mathrm{T}} W_{12}, \qquad \hat{R}_{14} = W_{11}^{\mathrm{T}} W_{14}.$$

Combining with (5), we can deduce that

$$W_{41}^{\mathrm{H}} \approx W_{12}^{\mathrm{H}} W_{11} \Xi W_{11}^{-1} \equiv W_{12}^{\mathrm{H}} \Theta,$$

where $\Theta = W_{11}\Xi W_{11}^{-1}$. The above relation again specifies two bases for a single linear subspace. Therefore, we can apply the invariance principle to the matrix pair $(W_{12}^{\mathrm{H}}, W_{14}^{\mathrm{H}})$ just as we did in the covariance matrix case. Notice that the diagonal elements of Φ can be computed as the eigenvalues of Θ. It can be verified that the computational cost of the above procedure is $\mathrm{O}(MNd)$. We summarize the above in the following algorithm.

Algorithm 2 (*Data matrix case*).

(1) Compute a partial QR decomposition of Z^{H} such that

$$Z^{\mathrm{H}} = Q\begin{pmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ 0 & W_{22} & W_{23} & W_{24} \end{pmatrix},$$

where $W_{11} \in \mathscr{C}^{d\times d}$ is upper triangular. The computation of the QR decomposition of the first d columns of Z^{H} costs $Nd^2 - d^3/3$ flops. Notice that the Q matrix needs not be explicitly formed: it can be kept in product form with each factor of a Householder matrix. Then the matrices W_{12} and W_{14} can be computed by $4N(M-d)d$ flops.

(2) Compute the SVD of $(W_{12}^{\mathrm{H}}, W_{14}^{\mathrm{H}})$

$$(W_{12}^{\mathrm{H}}, W_{14}^{\mathrm{H}}) = U\Sigma V^{\mathrm{H}},$$

(3) Partition the right singular vector matrix V as

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}, \qquad V_{ij} \in \mathscr{C}^{d\times d}, \quad i,j = 1,2,$$

and an estimate of Θ can be obtained as

$$\Theta \approx \hat{\Theta} \equiv -V_{12}V_{22}^{-1}.$$

(4) Compute the eigenvalues of $\hat{\Theta}$ and estimate the DOAs as

$$\hat{\theta}_i = \sin^{-1}(c\arg(\lambda_i(\hat{\Xi}))/(\omega_0\Delta)), \quad i = 1, \dots, d.$$

The costs of steps (2)–(4) are the same as those in Algorithm 1. Hence the total cost of Algorithm 2 is $4N(M-d)d + \min\{8(M-d)d^2 + 32d^3, 4(M-d)d^2 + 40d^3\} + 11d^3$. In comparison with TLS-ESPRIT, the SVD of Z costs $\min\{8NM^3 + 32M^3, 4NM^3 + 40M^3\}$. If the FSD of [13] is used, the signal subspace can be computed in $8NMd$. Algorithm 2 and FSD are of the same order, but Algorithm 2 is about twice as fast.

Remark. The partial QR decomposition in Step (1) of Algorithm 2 can be computed using matrix–matrix operations, i.e., level-3 BLAS as in [4]. First House-holder transformations $H_1, \dots, H_d$ are computed to triangularize the first d columns of Z^{H}. Then we compute a block representation of the product $H_1, \dots, H_d = I + WY^{\mathrm{H}}$, where W and Y are $N \times d$ matrices. Then we propagate the transformations by computing

$$\begin{aligned}(R_{12}, R_{14}) &= [Z^{\mathrm{H}}(:, d+1:M), Z^{\mathrm{H}}(:, M+d+1:2M)] \\ &\quad + W[1:d, :)Y^{\mathrm{H}}[Z^{\mathrm{H}}(:, d+1:M), \\ &\quad Z^{\mathrm{H}}(:, M+d+1:2M)],\end{aligned}$$

which is a level-3 BLAS update. Here we used MATLAB colon notation to denote the selected rows and columns of a matrix. Algorithms rich in matrix–matrix operations are more efficient in high performance computers because they reduce the amount of data movement and memory reference.

4.3. *Tracking slowly-varying DOAs*

In this section, we consider tracking slowly-varying DOAs. We will use exponential windowing with a forgetting factor $\lambda < 1$. At the end of step N, we have a matrix

$$(W_{11}, W_{12}, W_{14})$$

that comes from the partial QR decomposition of Z^{H} (compare step (1) of Algorithm 2). Notice that W_{11} is $d \times d$ upper triangular. At the next time step, a new sensor array output vector $z(t_N)$,

$$z(t_N)^{\mathrm{H}} = (z_1^{\mathrm{H}}, z_2^{\mathrm{H}}, z_3^{\mathrm{H}}, z_4^{\mathrm{H}}), \quad z_1 \in \mathscr{C}^d,$$

partitioned conformally with the partitioning of Z^{H}, is available. We need to construct a sequence of d Givens rotations G to reduce z_1^{H} to zero such that

$$G^{\mathrm{H}}\begin{pmatrix} \lambda W_{11} & \lambda W_{12} & \lambda W_{14} \\ z_1^{\mathrm{H}} & z_2^{\mathrm{H}} & z_4^{\mathrm{H}} \end{pmatrix} = \begin{pmatrix} \tilde{W}_{11} & \tilde{W}_{12} & \tilde{W}_{14} \\ 0 & \tilde{z}_2^{\mathrm{H}} & \tilde{z}_4^{\mathrm{H}} \end{pmatrix}, \tag{6}$$

where $\tilde{W}_{11}$ is upper triangular. This is a standard rank-one QR updating problem and can be computed with $4(2M - d)d$ arithmetic operations. Notice that we also propagate the Givens rotation to the rest of the columns. Then a new estimate of Θ can be obtained by applying the ESPRIT invariance principle to the matrix pair $(\tilde{W}_{12}^{\mathrm{H}}, \tilde{W}_{14}^{\mathrm{H}})$ as is outlined in Algorithm 2. Unfortunately, the computational cost of computing the SVD of $(\tilde{W}_{12}^{\mathrm{H}}, \tilde{W}_{14}^{\mathrm{H}})$ in the TLS step is about $\mathrm{O}(Md^2)$ arithmetic operations even without computing the left singular vector matrix U.

In the following we show that the TLS step can actually be combined with the Givens rotation step (6) to obtain an algorithm with $\mathrm{O}(Md) + \mathrm{O}(d^3)$ overall computational complexity per data vector update. Notice that from (6), we have with $\bar{G} = G(I_d, 0)^{\mathrm{H}}$,

$$\tilde{W}_{12}^{\mathrm{H}} = (\bar{\lambda} W_{12}^{\mathrm{H}}, z_4)\bar{G}, \qquad \tilde{W}_{14}^{\mathrm{H}} = (\bar{\lambda} W_{14}^{\mathrm{H}}, z_2)\bar{G}.$$

Let the SVD of $(W_{12}^{\mathrm{H}}, W_{14}^{\mathrm{H}}) = U\Sigma V^{\mathrm{H}}$. Then

$$\begin{aligned}(\tilde{W}_{12}^{\mathrm{H}}, \tilde{W}_{14}^{\mathrm{H}}) &= (\bar{\lambda} W_{12}^{\mathrm{H}}, z_2, \bar{\lambda} W_{14}^{\mathrm{H}}, z_4)\,\mathrm{diag}(\bar{G}, \bar{G}) \\ &= (\bar{\lambda}(W_{12}^{\mathrm{H}}, W_{14}^{\mathrm{H}}), (z_2, z_4))\tilde{G} \\ &= (U, U_\perp)\begin{pmatrix} \Sigma & F \\ 0 & H \\ 0 & 0 \end{pmatrix}\hat{G},\end{aligned}$$

where $\tilde{G}^{\mathrm{H}}$ and $\hat{G}^{\mathrm{H}}$ are orthonormal matrices, and $H \in \mathscr{C}^{2\times 2}$ is upper triangular. The matrix F can be computed as

$$F = U^{\mathrm{H}}(z_2, z_4) = V\Sigma^{-1}\begin{pmatrix} W_{12}(z_2, z_4) \\ W_{14}(z_2, z_4) \end{pmatrix},$$

which costs $\mathrm{O}(Md)$ arithmetic operations. For the computation of H, we proceed as follows. Notice that

$$\begin{pmatrix} H \\ 0 \end{pmatrix} = U_\perp^{\mathrm{H}}(z_2, z_4), \qquad U_\perp U_\perp^{\mathrm{H}} = I - UU^{\mathrm{H}}.$$

Hence

$$U_\perp \begin{pmatrix} H \\ 0 \end{pmatrix} = (I - UU^{\mathrm{H}})(z_2, z_4),$$

and it gives the QR decomposition of $(I - UU^{\mathrm{H}})(z_2, z_4)$, which can be computed as

$$\begin{aligned}(I - UU^{\mathrm{H}})(z_2, z_4) &= (z_2, z_4) - UF \\ &= (z_2, z_4) - (W_{12}^{\mathrm{H}}, W_{14}^{\mathrm{H}})\Sigma^{-1}V^{\mathrm{H}}F,\end{aligned}$$

which costs $\mathrm{O}(Md)$ arithmetic operations. It is now easy to see the overall computational cost is $\mathrm{O}(Md) + \mathrm{O}(d^3)$.

4.4. *The special case of a uniform linear array*

In this section, we consider the case of a uniform linear array. We show that Algorithms 1 and 2 can be simplified in this special case. Write the array output vector as

$$x(t) = A(\theta)s(t) + n(t).$$

Then $A(\theta)$ is a Vandermonde matrix. There are many sub-array structures that satisfy $J_1A(\theta) = J_2A(\theta)\Phi$, where J_1 and J_2 are selection matrices that specify the array elements that go into each individual sub-arrays. In the following discussion, we choose $J_1 = (0, I_{M-1})$, $J_2 = (I_{M-1}, 0)$. Let

$$A(\theta) = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}, \quad A_1 \in C^{d\times d}.$$

Then the first block row of the covariance matrix R_{xx} has the form

$$(A_1R_{ss}A_1^{\mathrm{H}}, \underbrace{A_1R_{ss}A_2^{\mathrm{H}}}_{R_{12}}).$$

Now we have with $\tilde{J}_1 = (0, I_{M-d-1})$, $\tilde{J}_2 = (I_{M-d-1}, 0)$ that

$$\tilde{J}_2R_{14}^{\mathrm{H}} = \tilde{J}_1R_{12}^{\mathrm{H}}\Phi.$$

Therefore, we can apply the invariance principle of ESPRIT to $(\tilde{J}_1R_{12}^{\mathrm{H}}, \tilde{J}_2R_{12}^{\mathrm{H}})$. It can be verified that the total computational cost is $N(M-d)d + \min\{8(M-d)d^2 + 32d^3, 4(M-d)d^2 + 40d^3\} + 11d^3$. In the data matrix case, let the data matrix be $X = (x(t_0), \ldots, x(t_{N-1}))$. Compute its partial QR decomposition,

$$X^{\mathrm{H}} = Q\begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix},$$

with $S_{11} \in \mathscr{C}^{d\times d}$ upper triangular, i.e., we only triangularize the first d columns of X^{H} and propagate the transformation to the other $M-d$ columns. Since $R_{12}^{\mathrm{H}} \approx \tilde{J}_1S_{12}^{\mathrm{H}}S_{11}$, we can apply the invariance principle of ESPRIT to the matrix pair $(\tilde{J}_1S_{12}^{\mathrm{H}}, \tilde{J}_2S_{12}^{\mathrm{H}})$. The total computational cost is $2N(M-d)d + \min\{8(M-d)d^2 + 32d^3, 4(M-d)d^2 + 40d^3\} + 11d^3$. The comparisons with TLS-ESPRIT and Fast Subspace Decomposition (FSD) are similar as those carried out in Sections 4.1 and 4.2, and will not be repeated here.

To summarize, the proposed new algorithms achieve low computational costs as compared to TLS-ESPRIT or FSD methods. In comparison to TLS-ESPRIT (covariance case and data matrix case) and FSD (covariance case) the reduction can be an order of magnitude, while in comparison to FSD (data matrix case), the new algorithms run about twice as fast. The new algorithms are also much cheaper to update. The drawback of the new algorithms is that they usually have larger variances in the estimated DOAs, especially in the low SNR cases (cf. Example 1 in Section 6).

5. Error analysis of the TLS step

In this section, we will take a closer look at the TLS step in the ESPRIT Algorithm and the proposed new algorithms. The TLS step involves the SVD of a matrix of the form (F_1, F_2), where for the exact covariance matrix,

$$F_2 = F_1W, \qquad W = T\Phi T^{-1}.$$

For example, for the covariance matrix case (cf. Section 4.1) we have $F_1 = R_{12}^{\mathrm{H}}$ and $F_2 = R_{14}^{\mathrm{H}}$, and

$$F_1 = A_2SA_1^{\mathrm{H}},$$

$$F_2 = A_2\Phi SA_1^{\mathrm{H}} = F_1(R_{ss}A_1^{\mathrm{H}})^{-1}\Phi(R_{ss}A_1^{\mathrm{H}}).$$

The purpose of the TLS step is to find an estimate of W based on the estimated covariance matrix and use its eigenvalues as estimates of the diagonal elements of Φ. For an estimated covariance matrix, assume F_1 and F_2 are perturbed to

$$\tilde{F}_1 = F_1 + E_1, \qquad \tilde{F}_2 = F_2 + E_2,$$

where E_1 and E_2 are errors due to using an estimated covariance matrix. The objective of the following error analysis is to access the effect of the error (E_1, E_2) on the perturbation of the eigenvalues of W. The TLS step amounts to finding

$$\hat{F}_1 = \tilde{F}_1 + \tilde{E}_1, \qquad \hat{F}_2 = \tilde{F}_2 + \tilde{E}_2,$$

such that

$$\mathrm{span}\{\hat{F}_1\} = \mathrm{span}\{\hat{F}_2\}$$

and $\|(\tilde{E}_1, \tilde{E}_2)\|_F$ is minimized. It is easy to see that the minimal $\|(\tilde{E}_1, \tilde{E}_2)\|_F \leqslant \|(E_1, E_2)\|_F$. Here we can write

$$\hat{F}_1 = F_1 + \hat{E}_1, \qquad \hat{F}_2 = F_2 + \hat{E}_2,$$

with $\|(\hat{E}_1, \hat{E}_2)\|_F \leqslant 2\|(E_1, E_2)\|_F$. Now we consider the effect of $(\hat{E}_1, \hat{E}_2)$ on the SVD of (F_1, F_2). It can be verified that the right singular vector matrix of $(F_1, F_2) = U\Sigma V^{\mathrm{H}}$ has the following form:

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} = \begin{pmatrix} (I + WW^{\mathrm{H}})^{-1/2}P & -W(I + W^{\mathrm{H}}W)^{-1/2}P \\ W^{\mathrm{H}}(I + WW^{\mathrm{H}})^{-1/2}P & (I + W^{\mathrm{H}}W)^{-1/2}P \end{pmatrix},$$

where P is unitary. The last d columns of V form an orthonormal basis of the null space of (F_1, F_2). Using SVD perturbation analysis [10], for $(\hat{F}_1, \hat{F}_2) = (F_1, F_1 W) + (\hat{E}_1, \hat{E}_2)$, we can find a matrix G with $\|G\|_F \approx \|(\hat{E}_1, \hat{E}_2\|_F$ such that

$$\operatorname{span}\left\{\begin{pmatrix} V_{11} \\ V_{21} \end{pmatrix} G + \begin{pmatrix} V_{12} \\ V_{22} \end{pmatrix}\right\}$$

spans the approximate null space of $(\hat{F}_1, \hat{F}_2)$, i.e., it contains the right singular vectors corresponding to the d smallest singular values. Therefore, the estimated W is

$$\hat{W} = W(I - W^{-1}\tilde{G})(I + W^{\mathrm{H}}\tilde{G})^{-1}, \tag{7}$$

where $\tilde{G} = (I + W^{\mathrm{H}}W)^{-1/2}PGP^{\mathrm{H}}(I + WW^{\mathrm{H}})^{-1/2}$. The well-known Bauer–Fike theorem [4] in eigenvalue perturbation states that for each diagonal element $\Phi(i,i)$ of Φ, there is an eigenvalue of $\lambda(\hat{W})$ of $\hat{W}$ such that

$$|\Phi(i,i) - \lambda(\hat{W})| \leqslant \|T\|_2 \|T^{-1}\|_2 \|W - \hat{W}\|_2 .$$

Using the expression for $\hat{W}$ in (7), we can verify that

$$\|W - \hat{W}\|_2 \leqslant (1 + \|W\|_2^2)\|(E_1, E_2)\|_F + \mathrm{O}(\|E_1, E_2)\|_F^2).$$

Therefore, we have

$$|\Phi(i,i) - \lambda(\hat{W})| \leqslant \|T\|_2 \|T^{-1}\|_2 \times (1 + \|W\|_2^2)\|(E_1, E_2)\|_F + \mathrm{O}(\|E_1, E_2)\|_F^2).$$

As an application of the above analysis, let us compare the covariance matrix case in Section 4.1 and the data matrix case in Section 4.2. For the covariance matrix case $W = T^{-1}\Phi T$ with $T = R_{ss}A_1^{\mathrm{H}}$. If the signals are non-coherent, then R_{ss} will be a well-conditioned matrix. Thus T will be an ill-conditioned matrix if A_1 is ill-conditioned. This happens when the DOAs θ_i are very close to each other. On the other hand, for the data matrix case,

$$W_{\mathrm{data}} \approx W_{11}(R_{ss}A_1^{\mathrm{H}})^{-1}\Phi(R_{ss}A_1^{\mathrm{H}})W_{11}^{-1}.$$

Since $W_{11}W_{11}^{\mathrm{H}} \approx A_1 R_{ss} A_1^{\mathrm{H}}$, we have

$$W_{\mathrm{data}} \approx P^{\mathrm{H}} R_{ss}^{-1/2} \Phi R_{ss}^{-1/2} P,$$

where P is some unitary matrix. W_{data} is almost independent of the spacing of θ_i. Therefore, from the point of view of sensitivity to perturbation, Algorithm 2 is a better choice over Algorithm 1.

6. Simulation results

In this section, we present some examples of numerical simulations to illustrate our proposed fast algorithms. In all the examples, we consider a uniform linear array with M elements spaced $\lambda/2$ apart, where λ is the wavelength of signals. The two identical sub-arrays are constructed by taking the first $M - 1$ elements as the first sub-array, and the last $M - 1$ elements as the second sub-array. Thus the displacement vector between the two sub-arrays is of length $\lambda/2$, the sensor spacing in the original uniform linear array. The direction vector is

$$a(\theta)^{\mathrm{T}} = (1, \exp(j\sin\theta), \ldots, \exp(j(M-1)\sin\theta)).$$

The SNR is defined as $R_{ss}(i,i)/\sigma^2$.

Example 1. In this example, we will examine the effect of the SNR on the empirical means and standard deviations of the estimated DOAs. We have two uncorrelated source signals with equal power impinging on an array with $M = 20$ at angles 4° and 12°. In Table 1, we compare three algorithms: TLS-ESPRIT, Algorithm 1 (covariance matrix case) and Algorithm 2 (data matrix case). The number of samples in each run is 50, and the means and standard deviation are computed using 500 runs. It is clear that in the range of moderate to high SNR, TLS-ESPRIT and Algorithm 2 have rather comparable performance while Algorithm 1 has certain bias in the computed empirical means. For the case of low SNR, TLS-ESPRIT outperforms the proposed new algorithms.

Table 1
Comparison of TLS-ESPRIT, Algorithm 1 and Algorithm 2

SNR (dB)	TLS-ESPRIT	Data matrix	Covariance matrix
5	12.0017 ± 0.0327	11.9948 ± 0.2097	12.2578 ± 0.2770
	3.9975 ± 0.0621	3.9769 ± 0.2039	3.7182 ± 0.3118
10	12.0004 ± 0.0153	11.9972 ± 0.0737	12.0832 ± 0.0888
	3.9995 ± 0.0344	3.9986 ± 0.0675	3.9139 ± 0.0894
15	12.0000 ± 0.0080	11.9992 ± 0.0312	12.0265 ± 0.0346
	4.0000 ± 0.0192	4.0007 ± 0.0285	3.9736 ± 0.0328
20	11.9999 ± 0.0040	11.9998 ± 0.0153	12.0084 ± 0.0159
	4.0001 ± 0.0108	4.0006 ± 0.0141	3.9920 ± 0.0149
25	11.9999 ± 0.0025	12.0000 ± 0.0081	12.0027 ± 0.0082
	4.0001 ± 0.0061	4.0004 ± 0.0076	3.9976 ± 0.0077

Table 2
Comparison of TLS-ESPRIT, Algorithm 1 and Algorithm 2 for $M = 10$, 20 and 30

M	TLS-ESPRIT	Data matrix	Covariance matrix
10	12.0024 ± 0.0891	11.9860 ± 0.3882	12.0710 ± 0.4109
	4.0105 ± 0.1643	4.0146 ± 0.3867	3.9320 ± 0.4036
20	12.0004 ± 0.0153	11.9972 ± 0.0737	12.0832 ± 0.0888
	3.9995 ± 0.0344	3.9986 ± 0.0675	3.9139 ± 0.0894
30	12.0006 ± 0.0153	12.0019 ± 0.0534	12.0788 ± 0.0689
	3.9997 ± 0.0034	3.9982 ± 0.0501	3.9221 ± 0.0675

Example 2. We consider the effect of the number of sensors of the uniform linear array on the empirical means and standard deviations of the estimated DOAs. We have two uncorrelated source signals with equal power impinging on an array with M elements at angles 4° and 12°. The SNR is set at 10 dB. We compare three cases when $M = 10$, 20 and 30. Each run uses 50 samples and altogether 500 runs are carried out. The simulation results are tabulated in Table 2.

Example 3. We consider the case when the noises are correlated. One of the advantages of the new algorithms proposed in this paper is that they do not need the assumption that the noise is white. We choose the noise covariance matrix to be of the form BB^{H}, where B is a bidiagonal matrix with all non-zero elements equal to one except for $B(1,2) = 13$ and $B(2,3) = 0$. We see that in the low to moderate SNR range the new algorithms outperform the TLS-ESPRIT, while in the high SNR range they have about the same performance. The simulation results are tabulated in Table 3.

Example 4. In this example, we consider the problem of tracking slowly moving DOAs using the tracking algorithm in Section 4.3. Notice that the computational complexity is $\mathrm{O}(Md) + \mathrm{O}(d^3)$ per update. We consider the same example in [3, Example 3]. The SNR = 10 dB for all sensors. The forgetting factor $\lambda = 0.99$. The true DOAs of two sources of equal power impinging on uniform linear array of 40 elements are

$$\theta_1 = 25^\circ,$$

$$\theta_2 = 30^\circ + 2^\circ \cos(\pi n/4000), \quad n = 1, \ldots, 4000.$$

Table 3
Comparison of TLS-ESPRIT, Algorithm 1 and Algorithm 2 with colored noise

M	TLS-ESPRIT	Data matrix	Covariance matrix
10	10.0560 ± 1.6668 4.8714 ± 1.5232	11.6532 ± 0.5476 3.7524 ± 0.5319	11.3120 ± 0.9315 3.6241 ± 1.1121
15	11.9516 ± 0.2166 3.9957 ± 0.1941	11.9278 ± 0.1567 3.9457 ± 0.1418	11.9291 ± 0.1852 3.9437 ± 0.1654
20	11.9930 ± 0.0404 3.9952 ± 0.0371	11.9895 ± 0.0496 3.9898 ± 0.0477	11.9996 ± 0.0494 3.9860 ± 0.0480

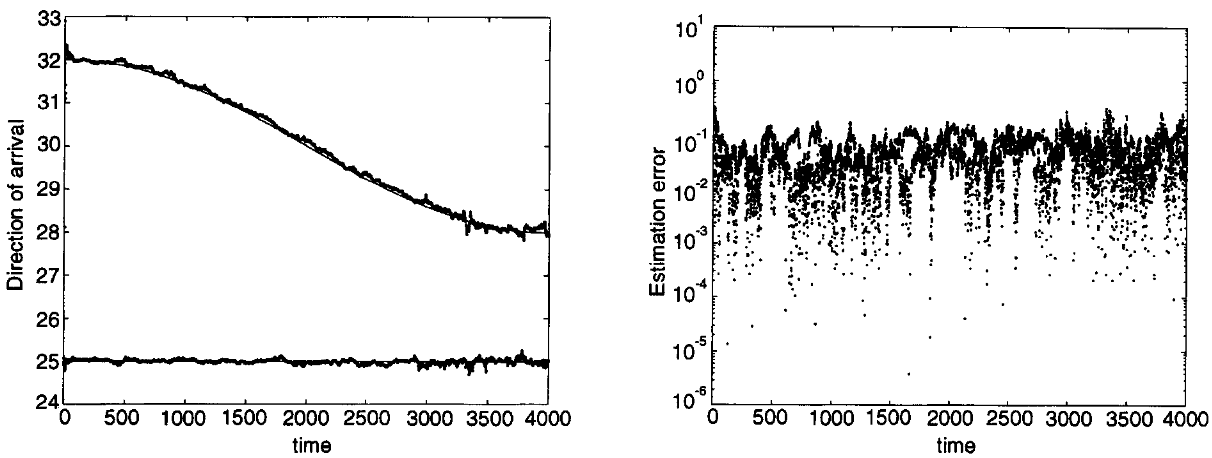

Fig. 1. DOA estimates and estimation errors. $M = 40$, SNR = 10 dB, $\lambda = 0.99$

In Fig. 1, the DOA estimates and the estimation errors $\theta_i(t) - \hat{\theta}_i(t)$, $t = 1, \ldots, 4000$ are depicted.

7. Concluding remarks

Exploring the redundancy in large ESPRIT arrays, we proposed new fast algorithms for DOA finding that avoid the computational intensive EVD or SVD. In the range of moderate to high SNR, the performance of the new algorithms are comparable to that of TLS-ESPRIT but with lower computational cost. The greatest strength of the new algorithms is their efficient on-line implementation: the computational complexity is about $O(Md) + O(d^3)$ per data vector update.

Acknowledgements

The author thanks the referees for the helpful comments and suggestions which improved the presentation of the paper.

References

[1] P. Comon and G. Golub, "Tracking a few extreme singular values and vectors in signal processing", *Proc. IEEE*, Vol. 78, 1994, pp. 1327–1343.

[2] E.M. Dowling, L.P. Ammmann and D. GeGroat, "A TQR-iteration based adaptive svd for real time angle and frequency tracking", *IEEE Trans. Signal Process.*, Vol. 42, 1994, pp. 914–926.

[3] A. Eriksson, P. Stoica and T. Söderström, "On-line subspace algorithms for tracking moving sources", *IEEE Trans. Signal Process.*, Vol. 42, 1994, pp. 2319–2330.

[4] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 1989, 2nd Edition.

[5] D.H. Johnson and D.E. Dudgeon, *Array Signal Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[6] R. Roy and T. Kailath, "ESPRIT – Estimation of signal parameters via rotational invariance techniques", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 37, 1989, pp. 984–993.

[7] R.O. Schmidt, A signal subspace approach to multiple emitter location and spectral estimation, PhD thesis, Department of Electrical Engineering, Stanford University, Stanford, 1981.

[8] G.W. Stewart, Updating a rank-revealing ULV decomposition, Technical Report CS-TR 2627, Department of Computer Science, University of Maryland, 1991.

[9] G.W. Stewart, "An updating algorithm for subspace tracking", *IEEE Trans. on Signal Process.*, Vol. 40, 1992, pp. 1535–1541.

[10] G.W. Stewart and G.-J. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[11] P. Stoica and T. Söderström, "Statistical analysis of subspace method for bearing estimation without eigendecomposition", *Proc. IEE*, Part F, Vol. 139, 1992, pp. 301–305.

[12] G. Xu and T. Kailath, "Fast estimation of principal eigenspace using lanczos algorithm", *SIAM J. Matrix Anal. Appl.*, Vol. 15, 1994, pp. 974–994.

[13] G. Xu and T. Kailath, "Fast subspace decomposition", *IEEE Trans. Signal Process.*, Vol. 42, 1994, pp. 539–551.

[14] G. Xu, H. Zha, G. Golub and T. Kailath, "Fast algorithms for updating signal subspaces", *IEEE Trans. Circuits Systems*, Vol. 41, 1994, pp. 537–549,

[15] C. Yeh, Simple computation of projection matrix for bearing estimation, *Proc. IEE*, Part F, Vol. 134, 1987, pp. 146–150.