Computationally Efficient Direction of Arrival Estimation without Subspace Decomposition

Jia-jia Rong, Feng-gang Yan, Shuai Liu
School of Information and Electrical Engineering，Harbin Institute of Technology, Weihai，Weihai, China
Email: 18363121609@163.com, yfglion@163.com, 13721929862@163.com

***Abstract*—In this paper, a computationally efficient direction of arrival (DOA) estimation algorithm without subspace decomposition is presented with a uniform linear array (ULA). The ULA is divided into two sub-arrays and two noise-free cross-correlation matrices are computed using the data vectors received by the two sub-arrays. By rearranging the elements of a vector formed from the new constructed cross-correlation matrix, an equivalent noise subspace is obtained and the expensive subspace decomposition process is efficiently avoided. Numerical simulations are conducted to substantiate the effectiveness of the proposed method, and it is shown that the new technique can provide good performances with a low signal-to-noise ratio (SNR) and with a small number of snapshots.**

Keywords—array signal processing; direction of arrival estimation; subspace decomposition; uniform linear array.

I. INTRODUCTION

The high resolution direction of arrival (DOA) estimation is one of the most important branches of array signal processing, which is widely used in radar, communication, sonar, earthquake, chronometer and other aspects of science and technology [1]-[5]. Among a large number of DOA estimation algorithms, the methods based on the subspace play an important role owing to its high performance, which have drawn considerable interests over recent years [6], [7]. The key principle of subspace based algorithm is to obtain the signal subspace or the noise subspace by conducting either the eigenvalue decomposition (EVD) or singular value decomposition (SVD) on the array covariance matrix.

Unfortunately, the subspace decomposition process is computationally intensive and time-consuming, and therefore, subspace-based methods are usually unacceptable for real environments, especially when the number of array sensors is large. To overcome this drawback, many techniques such as the propagator method (PM) for sources bearing estimation obtaining the noise subspace through the linear operation of the correlation matrix without EVD or SVD [8], simple computation of projection matrix for bearing estimation computing a projection matrix for bearing estimations by using any M rows of the covariance matrix [9] have been developed.

Recently, X. Nie *et al*. presented a computationally efficient subspace algorithm (noted as CESA in this paper) for two-dimensional (2-D) DOA estimation with L-shaped array [10]. The CESA method reduces the computational burden by dealing with three vectors composed of the first column-, the first row- and diagonal entry- of the correlation matrix. Although CESA is implemented for two-dimensional DOA estimation with L-shaped array structured, this technique can be also exploited for one-dimensional DOA estimation with uniform linear array (ULA).

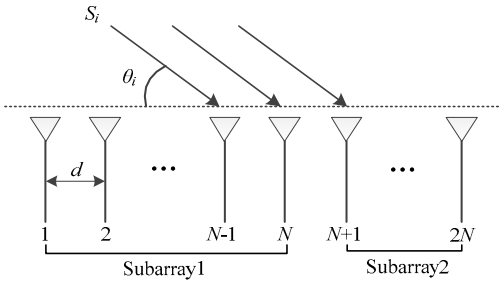

Fig.1. Uniform linear array model

Concerning the problem that poor estimation performance is got when CESA is used for one-dimensional (1-D) DOA estimation with ULA straightforwardly, in this paper, we propose a new method for fast 1-D DOA estimation using a uniform linear array (ULA) based on the idea of CESA, in which the ULA is divided into two sub-arrays and both the forward and backward cross-correlation matrices of array received data are simultaneously to improve the accuracy of DOA estimation. Numerical simulations are provided to demonstrate the efficiency of our method.

II. DATE MODEL

Suppose that there are K incoherent narrowband far-field signals $s_1(t), s_2(t), \ldots, s_K(t)$ from distinct directions represented as $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_K]$ implying on a ULA composed of M omnidirectional sensors, where the left first sensor be the reference. Without loss of generality, assume that M is an even number, i.e., M=2N. The ULA is divided into two sub-arrays of identical array elements, which are named subarray1 and subarray2 respectively, the output of subarray1 and subarray2 can be expressed respectively as

$$\begin{aligned} \boldsymbol{X}_a(t) &= [x_1(t) \quad x_2(t) \quad \cdots \quad x_N(t)]^{\mathrm{T}} \\ &= \boldsymbol{A}(\theta)\boldsymbol{S}(t) + \boldsymbol{N}_a(t) \end{aligned} \tag{1}$$

$$\begin{aligned}\boldsymbol{X}_b(t) &= [x_{N+1}(t) \quad x_{N+2}(t) \quad \cdots \quad x_{2N}(t)]^{\mathrm{T}} \\ &== \boldsymbol{A}(\theta)\boldsymbol{\Phi}\boldsymbol{S}(t) + \boldsymbol{N}_b(t)\end{aligned} \tag{2}$$

where $\boldsymbol{A}(\theta) \triangleq [\boldsymbol{a}(\theta_1) \quad \boldsymbol{a}(\theta_2) \quad \cdots \quad \boldsymbol{a}(\theta_K)]$ is the $N \times K$ array manifold matrix, $\boldsymbol{a}(\theta_i) \triangleq [1 \quad e^{j\alpha_i} \quad \cdots \quad e^{j(N-1)\alpha_i}]^{\mathrm{T}}$ is the $N \times 1$ steering vector, $\alpha_i \triangleq 2\pi d \cos\theta_i / \lambda$, λ is the signal wavelength with d denoting the distance between adjacent sensors. $\boldsymbol{S}(t) \triangleq [s_1(t) \quad s_2(t) \quad \cdots \quad s_K(t)]^{\mathrm{T}}$ is the $K \times 1$ signal vector, $\boldsymbol{\Phi} \triangleq diag(e^{jN\alpha_1} \quad e^{jN\alpha_2} \quad \cdots \quad e^{jN\alpha_K})$ is a $K \times K$ diagonal matrix, $\boldsymbol{N}_a(t) \triangleq [n_1(t) \quad n_2(t) \quad \cdots \quad n_N(t)]^{\mathrm{T}}$ is the $N \times 1$ additive white Gaussian noise (AWGN) vector of subarray1 with zero-mean and variance σ_n^2 and $\boldsymbol{N}_b(t) \triangleq [n_{N+1}(t) \quad n_{N+2}(t) \quad \cdots \quad n_{2N}(t)]^{\mathrm{T}}$ is the $N \times 1$ AWGN vector of subarray2 with zero-mean and variance σ_n^2. Note that the noise is statistically independent of signal samples.

III. PROPOSED ALGORITHM

To suppress the additive noise component, we define a forward cross-correlation matrix $\boldsymbol{R}_{ab}$ and a backward cross-correlation matrix $\boldsymbol{R}_{ba}$, which can be respectively written as

$$\begin{aligned}\boldsymbol{R}_{ab} &= E\{\boldsymbol{X}_a(t)\boldsymbol{X}_b^{\mathrm{H}}(t)\} \\ &= \boldsymbol{A}(\theta)E\{\boldsymbol{S}(t)\boldsymbol{S}^{\mathrm{H}}(t)\}\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}(\theta) + E\{\boldsymbol{N}_a(t)\boldsymbol{N}_b^{\mathrm{H}}(t)\} \quad (3) \\ &= \boldsymbol{A}(\theta)\boldsymbol{R}_s\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}(\theta)\end{aligned}$$

$$\begin{aligned}\boldsymbol{R}_{ba} &= E\{\boldsymbol{X}_b(t)\boldsymbol{X}_a^{\mathrm{H}}(t)\} \\ &= \boldsymbol{A}(\theta)\boldsymbol{\Phi}E\{\boldsymbol{S}(t)\boldsymbol{S}^{\mathrm{H}}(t)\}\boldsymbol{A}^{\mathrm{H}}(\theta) + E\{\boldsymbol{N}_b(t)\boldsymbol{N}_a^{\mathrm{H}}(t)\} \quad (4) \\ &= \boldsymbol{A}(\theta)\boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{A}^{\mathrm{H}}(\theta)\end{aligned}$$

where $\boldsymbol{R}_s \triangleq E\{\boldsymbol{S}(t)\boldsymbol{S}^{\mathrm{H}}(t)\}$ denotes the autocorrelation matrix of the signal sources. Note that $\boldsymbol{N}_a(t)$ and $\boldsymbol{N}_b(t)$ are independent with each other, i.e., $E\{\boldsymbol{N}_a(t)\boldsymbol{N}_b^{\mathrm{H}}(t)\} = \boldsymbol{0}_{N\times N}$. Since $s_i(t)$, $i = 1,2,\ldots,K$ are uncorrelated with each other, $\boldsymbol{R}_s$ is a diagonal matrix with each diagonal element being a positive real number p_i, $i = 1,2,\ldots,K$, which stands for the power of the i-th signal source. Hence, $\boldsymbol{R}_s$ can be denoted as

$$\boldsymbol{R}_s = diag(p_1 \quad p_2 \quad \ldots \quad p_K) \tag{5}$$

In order to make full use of the information included in the two cross-correlation matrixes $\boldsymbol{R}_{ab}$ and $\boldsymbol{R}_{ba}$, we define a new cross-correlation matrix $\boldsymbol{R}_{new}$ expressed as

$$\begin{aligned}\boldsymbol{R}_{new} &\triangleq \boldsymbol{R}_{ab} + \boldsymbol{R}_{ba} \\ &= \boldsymbol{A}(\theta)\boldsymbol{R}_s\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}(\theta) + \boldsymbol{A}(\theta)\boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{A}^{\mathrm{H}}(\theta) \\ &= \boldsymbol{A}(\theta)(\boldsymbol{R}_s\boldsymbol{\Phi}^{\mathrm{H}} + \boldsymbol{\Phi}\boldsymbol{R}_s)\boldsymbol{A}^{\mathrm{H}}(\theta) \\ &= \boldsymbol{A}(\theta)\boldsymbol{\Lambda}\boldsymbol{A}^{\mathrm{H}}(\theta)\end{aligned} \tag{6}$$

where $\boldsymbol{\Lambda} \triangleq \boldsymbol{R}_s\boldsymbol{\Phi}^{\mathrm{H}} + \boldsymbol{\Phi}\boldsymbol{R}_s$. Since both $\boldsymbol{R}_s$ and $\boldsymbol{\Phi}$ are diagonal matrices, $\boldsymbol{\Lambda}$ is also a diagonal matrix, which can be further expressed as

$$\begin{aligned}\boldsymbol{\Lambda} &= \boldsymbol{R}_s\boldsymbol{\Phi}^{\mathrm{H}} + \boldsymbol{\Phi}\boldsymbol{R}_s = \boldsymbol{R}_s\boldsymbol{\Phi}^{\mathrm{H}} + \boldsymbol{R}_s\boldsymbol{\Phi} \\ &= \boldsymbol{R}_s(\boldsymbol{\Phi}^{\mathrm{H}} + \boldsymbol{\Phi}) \\ &= \begin{bmatrix} \xi_1 & 0 & \cdots & 0 \\ 0 & \xi_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \xi_K \end{bmatrix}\end{aligned} \tag{7}$$

where $\xi_i \triangleq p_i(e^{jN\alpha_i} + e^{-jN\alpha_i})$, $i = 1,2,\ldots,K$. Clearly, the diagonal elements of $\boldsymbol{\Lambda}$ are also positive real numbers, therefore, we have $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^*$.

Following the idea in [10], we utilize $\boldsymbol{r}_{new}^{(:,1)}$ to represent the first column of $\boldsymbol{R}_{new}$. Based on (6), since the first column of $\boldsymbol{A}^{\mathrm{H}}(\theta)$ is a vector of ones, $\boldsymbol{r}_{new}^{(:,1)}$ can be specifically written as

$$\boldsymbol{r}_{new}^{(:,1)} = \boldsymbol{A}(\theta)\boldsymbol{\Lambda}\boldsymbol{1} \tag{8}$$

where $\boldsymbol{1}$ denotes the $K \times 1$ vector, whose elements are all ones. Define a new $N \times 1$ vector expressed as

$$\overline{\boldsymbol{r}}_{new}^{(:,1)} \triangleq \boldsymbol{J}_N(\boldsymbol{r}_{new}^{(:,1)})^* = \boldsymbol{J}_N(\boldsymbol{A}(\theta)\boldsymbol{\Lambda}\boldsymbol{1})^* = \boldsymbol{J}_N\boldsymbol{A}^*(\theta)\boldsymbol{\Lambda}\boldsymbol{1} \tag{9}$$

where $\boldsymbol{J}_N$ is an $N \times N$ exchange matrix with ones on its anti-diagonal and zeros elsewhere. According to (8) and (9), by combining $\boldsymbol{r}_{new}^{(:,1)}$ with $\overline{\boldsymbol{r}}_{new}^{(:,1)}$, construct a $2N \times 1$ vector as follow

$$\overline{\boldsymbol{r}} \triangleq \begin{bmatrix} \overline{\boldsymbol{r}}_{new}^{(:,1)} \\ \boldsymbol{r}_{new}^{(:,1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_N\boldsymbol{A}^*(\theta) \\ \boldsymbol{A}(\theta) \end{bmatrix}\boldsymbol{\Lambda}\boldsymbol{1} \tag{10}$$

After a simple deduction, we can find that $\boldsymbol{J}_N\boldsymbol{A}^*(\theta)$ is the conjugate anti-symmetry of $\boldsymbol{A}(\theta)$, therefore the last row of $\boldsymbol{J}_N\boldsymbol{A}^*(\theta)$ equals to the first row of $\boldsymbol{A}(\theta)$. Thus we can remove the identical rows of $\overline{\boldsymbol{r}}$ and obtain a new vector $\boldsymbol{r}$ as follows

$$\boldsymbol{r} = \begin{bmatrix} e^{-j(N-1)\alpha_1} & \cdots & e^{-j(N-1)\alpha_K} \\ \vdots & \cdots & \vdots \\ 1 & \cdots & 1 \\ \vdots & \cdots & \vdots \\ e^{j(N-1)\alpha_1} & \cdots & e^{j(N-1)\alpha_K} \end{bmatrix} \times \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_K \end{bmatrix} = \overline{\boldsymbol{A}}(\theta)\boldsymbol{\xi} \tag{11}$$

where $\overline{\boldsymbol{A}}(\theta) = \begin{bmatrix} e^{-j(N-1)\alpha_1} & \cdots & e^{-j(N-1)\alpha_K} \\ \vdots & \cdots & \vdots \\ 1 & \cdots & 1 \\ \vdots & \cdots & \vdots \\ e^{j(N-1)\alpha_1} & \cdots & e^{j(N-1)\alpha_K} \end{bmatrix}$, $\boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_K \end{bmatrix}$.

As can be seen from (11), the expanded form of $\boldsymbol{r}$ include a special $(2N-1) \times K$ array manifold matrix $\overline{\boldsymbol{A}}(\theta)$, which means that the number of array elements increases from N to $2N-1$. As a result, the array aperture lost in the process of sub-array partition can be reduced by splicing $\boldsymbol{A}(\theta)$ with $\boldsymbol{J}_N\boldsymbol{A}^*(\theta)$, and the accuracy of DOA estimation can be improved as well.

Based on the above analysis, we can utilize $\boldsymbol{r}$ to form a $(2N-K) \times K$ matrix $\boldsymbol{G}$ to stand for the signal subspace, which is corresponds to the array manifold $\overline{\boldsymbol{A}}(\theta)$ given above

$$\boldsymbol{G} = [\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \cdots \quad \boldsymbol{r}_K] \tag{12}$$

where $\boldsymbol{r}_i = \boldsymbol{r}(i:2N-K-1+i)$, $i=1,2,\ldots,K$ is a $(2N-K)\times 1$ column vector, and $\boldsymbol{r}(m:n)$ denotes the vector consisting of the m-th to the n-th elements of $\boldsymbol{r}$. Based on (11), (12) can be rewritten in a matrix product form as

$$\boldsymbol{G} = \boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{D} \tag{13}$$

where $\boldsymbol{C}$ is a $(2N-K)\times K$ matrix and $\boldsymbol{D}$ is a $K\times K$ matrix expressed as

$$\boldsymbol{C} = \begin{bmatrix} e^{-j(N-1)\alpha_1} & \cdots & e^{-j(N-1)\alpha_K} \\ \vdots & \cdots & \vdots \\ 1 & \cdots & 1 \\ \vdots & \cdots & \vdots \\ e^{j(N-K)\alpha_1} & \cdots & e^{j(N-K)\alpha_K} \end{bmatrix} \tag{14}$$

$$\boldsymbol{D} = \begin{bmatrix} 1 & e^{j\alpha_1} & \cdots & e^{j(K-1)\alpha_1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & e^{j\alpha_K} & \cdots & e^{j(K-1)\alpha_K} \end{bmatrix} \tag{15}$$

apparently, both $\boldsymbol{C}$ and $\boldsymbol{D}$ are Vandermonde matrices. Because all the signals are incoherent with each other, matrices $\boldsymbol{C}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{D}$ are all of full rank. Hence, the space expanded by $\boldsymbol{G}$ and $\boldsymbol{C}$ are identical, i.e., $span(\boldsymbol{G}) = span(\boldsymbol{C})$. In addition, $\boldsymbol{G}$ contains the DOA information we expected. To get a complete signal subspace, a Gram-Schmidt orthogonalization process is further applied on the columns of $\boldsymbol{G}$, leading to an orthogonalization of $\bar{\boldsymbol{G}}$. Define a cost function $f(\theta)$ as

$$f(\theta) = \frac{1}{\bar{\boldsymbol{a}}^{\mathrm{H}}(\theta)(\boldsymbol{I} - \bar{\boldsymbol{G}}\bar{\boldsymbol{G}}^{\mathrm{H}})\bar{\boldsymbol{a}}(\theta)} \tag{16}$$

where $\bar{\boldsymbol{a}}(\theta)$ is a $(2N-K)\times 1$ column vector given by

$$\bar{\boldsymbol{a}}(\theta) = \begin{bmatrix} 1 \\ e^{j2\pi d\cos\theta/\lambda} \\ \vdots \\ e^{j2\pi(2N-K-1)d\cos\theta/\lambda} \end{bmatrix} \tag{17}$$

then DOA can be estimated by maximizing the cost function $f(\theta)$.

In summary, the detailed steps for our method are listed in Table Ⅰ.

IV. Analysis On Computational Complexity

Since both the proposed method and CESA contain a search step with a similar complexity. This section mainly focuses on comparing the computational complexity of our method with CESA in terms of getting the noise subspace.

The only difference between the computational complexities of two algorithms is that our method computes an extra backward cross-correlation matrix $\boldsymbol{R}_{ba}$. As analyzed in [10], CESA uses the ULA to get the noise subspace by computing the forward cross-correlation matrix $\boldsymbol{R}_{ab}$, then constructs a $N\times 1$ vector, followed by a Gram-Schmidt orthogonalization process on the $(2N-K)\times K$ matrix $\boldsymbol{G}$. The computational complexities of these three parts are N^2L , NL , and $K(K+1)(2N-K)^2/2$ flops respectively, where L denotes the snapshot number. Hence, the total computational complexity of CESA is $(N^2L+NL+K(K+1)(2N-K)^2/2)$, while the total computational complexity of our method is $(2N^2L+NL+K(K+1)(2N-K)^2/2)$ .Although that the computational burden of our method is a little bit more than that of CESA, the performance of our method improve apparently as the compensation, which will be verified in section V as follows.

TABLE Ⅰ Detailed Steps For Proposed Method

- Step 1: Calculate $\boldsymbol{R}_{ab}$ and $\boldsymbol{R}_{ba}$ by (3) and (4);
- Step 2: Obtain $\boldsymbol{R}_{new}$ by (6);
- Step 3: Take the first column of $\boldsymbol{R}_{new}$ out and note as $\boldsymbol{r}_{new}^{(:,1)}$, then get $\bar{\boldsymbol{r}}_{new}^{(:,1)}$ by (9);
- Step 4: Construct $\bar{\boldsymbol{r}}$ by (10) and then get $\boldsymbol{r}$ by removing the identical rows of $\bar{\boldsymbol{r}}$;
- Step 5: Obtain $\boldsymbol{G}$ by (12) and then compute the Gram-Schmidt orthogonalization of $\boldsymbol{G}$ noted as $\bar{\boldsymbol{G}}$.
- Step 6: Formulate the cost function $f(\theta)$ by (16) and estimate the directions of arrival by searching the spectrum peak of $f(\theta)$.

V. Simulation Results

In this section, some simulations are conducted to investigate the performance of the proposed method. To see clearly the performance of our method, we compare it with CESA used on ULA. Throughout the simulation, the array geometry is depicted in Fig.1. The number of sensor elements and that of incident signals are set as $2N=16$ and $K=2$, respectively. The sensor spacing in experiment is fixed as $d=\lambda/2$. Based on the definition of θ_i in Fig.1, the angle search range is set as $[0°,180°]$.

In the first simulation, we compare the spectrum of the proposed method with that of CESA, where the incident angles of two signal are set as $[\theta_1,\theta_2]=[30°,110°]$. The snapshot number is fixed as 500 and the SNR is set as SNR=10dB. Fig.2 plots the spectrum of two methods.

It can be seen from Fig.2 that both two methods can form the spectrum peaks near the angles θ_1=30° and θ_2=110° . But, it is clearly that the peaks of our method are much sharper than that of CESA.

In the second simulation, we examine the performance of the proposed method in terms of root-mean-square-error (RMSE), where the RMSE is defined as

$$\text{RMSE}=\sqrt{\frac{1}{L}\sum_{i=1}^{L}\left(\hat{\theta}_i-\theta\right)^2} \tag{18}$$

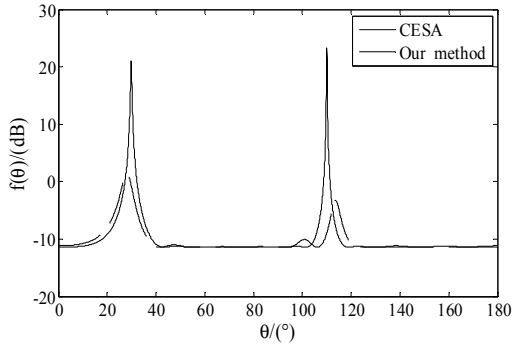

Fig.2. The spectrum of CESA and our method, where θ_1=30° , θ_2=110° , 500 snapshots, SNR = 10 dB.

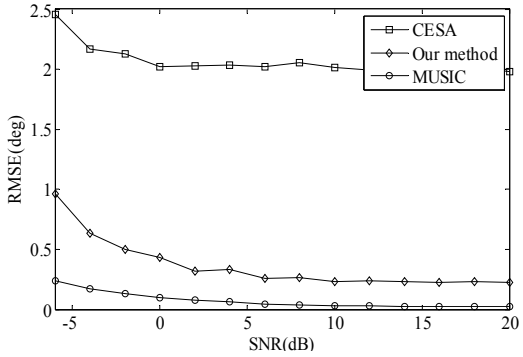

Fig.3. RMSE of CESA, our method and MUSIC versus the SNR， 500 snapshots, 200 Monte Carlo trials, $K=1$ source at $\theta_1=30°$.

where θ is the true DOA and $\hat{\theta}_i$ is the estimate of θ in the i-th independent Monte Carlo trial with L denoting the number of Monte Carlo trials.

Fig.3 plots the RMSEs as functions of the SNR, the number of snapshots is set as N=500 and the SNR varies with a step size 2 dB from -6dB to 20dB. For comparison reference, the standard MUSIC is also applied.

In the third simulation, we examine the performance of three methods against the snapshot number. The SNR is fixed at 15 dB and the snapshot number varies with a step size 50 from 50 to 1000.

It can be observed from Fig.3 and Fig.4 that as the SNR and the snapshot number increases, the RMSE curve of our method locates much lower and declines more apparently than that of CESA, which keeps in the vicinity of a large error value and changes slowly. It is clearly that the performance of our method is superior to CESA in different SNR and the snapshot number. This is because our method utilizes both the forward cross-correlation matrix $\boldsymbol{R}_{ab}$ and the backward cross-correlation matrix $\boldsymbol{R}_{ba}$ to construct $\boldsymbol{R}_{new}$ while CESA only uses the forward cross-correlation matrix $\boldsymbol{R}_{ab}$. Therefore, more information behind the two sub-arrays is utilized in our method compared while a part of that is used in CESA. As a result, the proposed method improves the estimation accuracy significantly as compared to CESA.

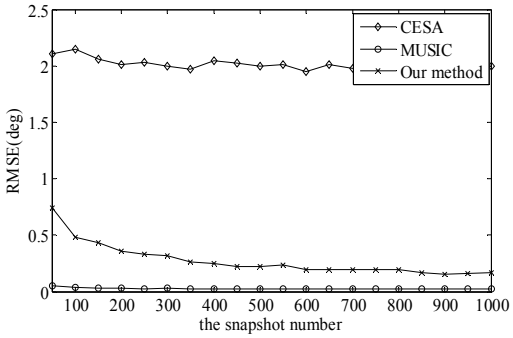

Fig.4. RMSE of CESA, our method and MUSIC versus the snapshot number，where SNR = 15, 200 Monte Carlo trials, $K=1$ source at $\theta_1=30°$.

VI. CONCLUSION

We have proposed a new computationally efficient direction of arrival estimation algorithm with a uniform linear array in this letter. In order to avoid the process of subspace decomposition, we divide the ULA into two sub-arrays and make full use of the information of two sub-arrays by computing the forward and backward cross-correlation matrices simultaneously. Simulation results indicate that the proposed method has much superior estimation performance to CESA used on a ULA.

VII. ACKNOWLEGEMETH

This work is supported by National Natural Science Foundation of China (61501142), Shandong Provincial Natural Science Foundation (ZR2014FQ003), Special Foundation of China Postdoctoral Science (2016T90289) and China Postdoctoral Science Foundation (2015M571414).

VIII. REFERENCES

[1] S. Haykin, Ed., Array Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[2] S. U. Pillai, Array Signal Processing. New York: Springer-Verlag,1989.

[3] S. Haykin, Ed., Advances in Spectrum Analysis and Array Processing. Englewood Cliffs, NJ: Prentice-Hall, 1991, 1995. Volumes I–III.

[4] H. L. Van Trees, Optimum Array Processing, Part IV of Detection, Estimation, and Modulation Theory. New York: Wiley, 2002.

[5] L. C. Godara, "Applications of antenna arrays to mobile communications, part II: Beam-forming and direction-of-arrival considerations,"Proc. IEEE, vol. 85, pp. 1195–1245, Aug. 1997.

[6] F. G. Yan, M. Jin, S. Liu, and X. L. Qiao, "Real-valued MUSIC for efficient direction estimation with arbitrary array geometries, IEEE Trans.Signal Process., vol. 62, no. 6, pp. 1548-1560, 2014.

[7] A. Liu, G. Liao, C. Zeng, Z. Yang, and Q. Xu, "An Eigenstructure Method for Estimating DOA and Sensor Gain-Phase Errors," IEEE Trans. Signal Process., vol. 59, no. 12, pp. 5944-5956, Dec. 2011.

[8] S. Marcos, A. Marsal, and M. Benidir, "The propagator method for source bearing estimation," Signal Process., vol. 42, no. 2, pp. 121–138,1995.

[9] C.-C. Yeh, "Simple computation of projection matrix for bearing estimations," Proc. Inst. Elect. Eng. F, vol. 134, no. 2, pp. 146–150, 1987.

[10] X. Nie, L.P. Li, "A Computationally Efficient Subspace Algorithm for 2-D DOA Estimation with L-shaped Array," IEEE Signal Processing Letters. Year: 2014, Volume: 21, Issue: 8. Pages: 971 – 974.