Robust adaptive array beamforming with random error in cycle frequency

Y.-T.Lee and J.-H.Lee

Abstract: By exploiting the cyclostationary properties, the SCORE algorithms presented by Agee *et al.* (1990) have been shown to be effective in performing adaptive beamforming without requiring the direction vector of the desired signal. However, these algorithms suffer from severe performance degradation in the presence of a random error in cycle frequency. The authors first establish a statistical model of the cyclic correlation matrix when the random error exists. According to this statistical model, two robust methods based on the SCORE algorithms are developed to achieve robust adaptive beamforming against random error. Analytical formulas are then derived for evaluating the performance of the proposed methods. Several simulation examples are also presented for confirming the theoretical analysis and showing the effectiveness of the proposed methods.

1 Introduction

Cyclostationarity [1], which is a statistical property exhibited by most man-made communication signals, corresponds to the underlying periodicity arising from carrier frequencies or baud rates. Recently, a class of spectral self-coherence restoral (SCORE) algorithms has been presented in [2] to deal with blind adaptive beamforming for cyclostationary signals. However, these algorithms suffer from server performance degradation in the presence of cycle frequency error (CFE). Some previous work on the problem has been reported in [4–8]. A statistical analysis presented in [4, 5] shows the cycle leakage through a sinc function due to finite data. On the other hand, two approaches, namely multi-cyclic MUSIC and adaptive-α cyclic MUSIC, have been presented in [6] to deal with performance degradation due to CFE. However, they cannot tackle the problem efficiently. Variations of the adaptive-α cyclic MUSIC are explored in [7]. Based on the concept of subintervals, only the cyclic correlation matrices of subintervals are required to reduce the memory requirements and the sensitivity of cyclic MUSIC to CFE. Recently, an efficient technique for coping with the problem of nonrandom CFE in cyclostationarity-exploiting adaptive array beamforming was developed [8]. Robust adaptive beamforming is achieved by appropriately estimating the actual cycle frequency.

In this paper, two of the SCORE algorithms presented in [2], namely the least-square SCORE (LS-SCORE) and cross-SCORE algorithms, are considered in the presence of random CFE. Based on the central limit theorem, the cyclic correlation matrix with random CFE is equivalent to the cyclic correlation matrix computed by using the actual cycle frequency plus a Gaussian random error matrix when the random CFE is an independent sequence. By making use of the ideas originally developed in [9], where two approaches for curing the performance degradation of conventional adaptive beamforming with random steering vector error were presented, two methods based on the LS-SCORE and cross-SCORE algorithms are developed to alleviate the performance degradation due to random CFE. Then, analytic formulas are derived for evaluating the performance of the proposed methods.

IEE Proceedings online no. 20010425

DOI: 10.1049/ip-rsn:20010425

Paper first received 21st February 2000 and in final revised form 19th March 2001

The authors are with the Department of Electrical Engineering, Building 2, National Taiwan University, Taipei 106, Taiwan, Republic of China

2 Adaptive beamforming using cyclostationarity

2.1 Signal cyclostationarity

For a signal $s(t)$, the cyclic correlation function and conjugate cyclic correlation function are defined as the following infinite-time averages:

$$r_{ss}(f,\tau) = \langle s(t+\tau/2)s^*(t-\tau/2)e^{-j2\pi ft}\rangle_\infty \qquad (1)$$

and

$$r_{ss^*}(f,\tau) = \langle s(t+\tau/2)s(t-\tau/2)e^{-j2\pi ft}\rangle_\infty \qquad (2)$$

respectively, where the superscript '*' denotes the complex conjugate. $s(t)$ is then said to be cyclostationary if $r_{ss}(f,\tau)$ or $r_{ss^*}(f,\tau)$ does not equal zero at some time delay τ and cycle frequency (CF) $f \neq 0$. Let $\boldsymbol{x}(t)$ denote the data vector received by an array. The cyclic correlation matrix and conjugate cyclic correlation matrix are defined by

$$\boldsymbol{R}_{xx}(f,\tau) = \langle \boldsymbol{x}(t+\tau/2)\boldsymbol{x}^H(t-\tau/2)e^{-j2\pi ft}\rangle_\infty \qquad (3)$$

and

$$\boldsymbol{R}_{xx^*}(f,\tau) = \langle \boldsymbol{x}(t+\tau/2)\boldsymbol{x}^T(t-\tau/2)e^{-j2\pi ft}\rangle_\infty \qquad (4)$$

respectively, where the superscript 'H' denotes the conjugate transpose and 'T' the transpose.

2.2 SCORE algorithms

Consider an M-element array excited by a SOI (signal of interest), J interferers, and spatially white noise. The received array data vector $\boldsymbol{x}(t)$ is then given by

$$\boldsymbol{x}(t) = s(t)\boldsymbol{a}_d + \sum_{j=1}^{J} s_j(t)\boldsymbol{a}_j + \boldsymbol{n}(t) = s(t)\boldsymbol{a}_d + \boldsymbol{i}(t) \quad (5)$$

where $s(t)$ and $s_j(t)$ denote the waveforms of SOI and the jth interferer, $\boldsymbol{a}_d$ and $\boldsymbol{a}_j$ are the direction vectors of the SOI and the jth interferer, respectively, and $\boldsymbol{n}(t)$ is the noise vector. The array output is then given by $y(t) = \boldsymbol{w}^H\boldsymbol{x}(t)$, where $\boldsymbol{w}$ denotes the weight vector.

Assume that $s(t)$ is cyclostationary and has a CF equal to α, but $\boldsymbol{i}(t)$ includes all signals not of interest (SNOIs) and noise and is temporally uncorrelated with $s(t)$. Based on the LS-SCORE algorithm of [2], a cost function is defined as follows:

$$F_{ls}(\boldsymbol{w}; \boldsymbol{c}) = \langle |y(t) - r(t)|^2 \rangle_T \quad (6)$$

where the reference signal $r(t)$ is given by $r(t) = \boldsymbol{c}^H\boldsymbol{x}(t-\tau)e^{j2\pi\alpha t}$ and $\langle\cdot\rangle_T$ denotes the average over the time interval $[0, T]$. $\boldsymbol{c}$ is a control vector and is fixed for the LS-SCORE algorithm. The optimal weight vector $\boldsymbol{w}_l$ minimising eqn. 6 is given by

$$\hat{\boldsymbol{w}}_l = \hat{\boldsymbol{R}}_{xx}^{-1}\hat{\boldsymbol{r}}_{xr}(\alpha) \quad (7)$$

where $\hat{\boldsymbol{R}}_{xx} = \langle \boldsymbol{x}(t)\boldsymbol{x}^H(t)\rangle_T$ and $\hat{\boldsymbol{r}}_{xr}(\alpha) = \langle \boldsymbol{x}(t)r^*(t)\rangle_T$ are the sample autocorrelation matrix of $\boldsymbol{x}(t)$ and the cross-correlation vector of $\boldsymbol{x}(t)$ and $r(t)$ computed over $[0, T]$, respectively.

To improve the convergence speed of the LS-SCORE algorithm, the cross-SCORE algorithm is proposed by [2], based on maximising the correlation coefficient, which is given by

$$\tilde{F}_{cs}(\boldsymbol{w}; \boldsymbol{c}) = \frac{|\boldsymbol{w}^H\hat{\boldsymbol{R}}_{xu}\boldsymbol{c}|^2}{(\boldsymbol{w}^H\hat{\boldsymbol{R}}_{xx}\boldsymbol{w})(\boldsymbol{c}^H\hat{\boldsymbol{R}}_{uu}\boldsymbol{c})} \quad (8)$$

between $y(t)$ and $r(t)$, where $\boldsymbol{u}(t) = \boldsymbol{x}(t-\tau)e^{j2\pi\alpha t}$ is the control signal. It is easy to show that the optimal weight and control vectors maximising eqn. 8 are the dominant eigenvectors of

$$\lambda\hat{\boldsymbol{R}}_{xx}\hat{\boldsymbol{w}}_c = \hat{\boldsymbol{R}}_{xu}\hat{\boldsymbol{R}}_{uu}^{-1}\hat{\boldsymbol{R}}_{ux}\hat{\boldsymbol{w}}_c \quad \text{and} \quad \lambda\hat{\boldsymbol{R}}_{uu}\hat{\boldsymbol{c}}_c = \hat{\boldsymbol{R}}_{ux}\hat{\boldsymbol{R}}_{xx}^{-1}\hat{\boldsymbol{R}}_{xu}\hat{\boldsymbol{c}}_c \quad (9)$$

respectively.

3 Proposed methods

3.1 Statistical model

Assume that $f = \alpha + \Delta\alpha_n$, where f and α are the presumed CF and the actual CF, respectively, $\Delta\alpha_n$ represents the random cycle frequency error (CFE) at the time instant nT_s, and T_s is the sampling interval. Substituting $t = nT_s$ into eqn. 3, we have the discrete-time version of the cyclic correlation matrix as

$$\begin{aligned}\boldsymbol{R}_{xx}(f, \tau) &= \lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}(n)\boldsymbol{x}^H(n-\tau)e^{-j2\pi(\alpha+\Delta\alpha_n)nT_s} \\ &= \boldsymbol{R}_{xx}(\alpha, \tau) + \boldsymbol{R}_e(f, \tau) \end{aligned} \quad (10)$$

where $\boldsymbol{R}_e(f, \tau)$ represents the corresponding error matrix equal to $\lim_{N\to\infty}(1/N)\sum_{n=1}^{N}(e^{-j2\pi\Delta\alpha_n nT_s} - 1)\boldsymbol{D}_x(n, \tau)$ and $\boldsymbol{D}_x(n, \tau) = \boldsymbol{x}(n)\boldsymbol{x}^H(n-\tau)e^{-j2\pi\alpha nT_s}$ is independent of $\Delta\alpha_n$. Let $\{\Delta\alpha_n, n = 1, 2, \ldots\}$ be an independent random sequence; the random sequence $\{(e^{-j2\pi\Delta\alpha_n nT_s} - 1), n = 1, 2, \ldots\}$ is thus independent. Then, $\boldsymbol{R}_e(f, \tau)$ equals the sum of infinite independent random matrices and becomes a Gaussian random matrix according to the central limit theorem shown by ([10], p. 15). As a result, we conclude that when CFE is an independent random sequence, the resulting cyclic correlation matrix is equal to the cyclic correlation matrix computed by using the actual CF plus a Gaussian random error matrix.

3.2 Robust methods based on LS-SCORE algorithm

From eqn. 10, the weight vector shown by eqn. 7 can be expressed as

$$\boldsymbol{w}_l = \boldsymbol{R}_{xx}^{-1}\boldsymbol{R}_{xx}(f, \tau)\boldsymbol{c} = \boldsymbol{R}_{xx}^{-1}\{\boldsymbol{R}_{xx}(\alpha, \tau)\boldsymbol{c} + \boldsymbol{R}_e(f, \tau)\boldsymbol{c}\} \quad (11)$$

when T approaches infinity. Let $\boldsymbol{R}_e(f, \tau)$ have zero mean. Then, the vector $\boldsymbol{a}_e = \boldsymbol{R}_e(f, \tau)\boldsymbol{c}$ is a zero-mean Gaussian random vector since $\boldsymbol{c}$ is a fixed vector. Hence, the log-likelihood function corresponding to $\boldsymbol{a}_e$ is given by

$$\begin{aligned} f_1 = -\boldsymbol{a}_e^H\boldsymbol{C}^{-1}\boldsymbol{a}_e = -\boldsymbol{c}^H(&\boldsymbol{R}_{xx}(\alpha, \tau) \\ &- \boldsymbol{R}_{xx}(f, \tau))\boldsymbol{C}^{-1}(\boldsymbol{R}_{xx}(\alpha, \tau) - \boldsymbol{R}_{xx}(f, \tau))\boldsymbol{c} \end{aligned} \quad (12)$$

where $\boldsymbol{C}$ denotes the covariance matrix of $\boldsymbol{a}_e$.

Accordingly, the role of $\boldsymbol{a}_e$ in eqn. 11 can be viewed as the resulting steering error vector due to the random CFE. Using the first method of [9], we define a cost function related to $\boldsymbol{a}_e$ as

$$J(\boldsymbol{a}) = \boldsymbol{a}^H\boldsymbol{R}_{xx}^{-1}\boldsymbol{a} + k(\boldsymbol{a} - \boldsymbol{R}_{xx}(f, \tau)\boldsymbol{c})^H\boldsymbol{C}^{-1}(\boldsymbol{a} - \boldsymbol{R}_{xx}(f, \tau)\boldsymbol{c}) \quad (13)$$

where the first term is the inverse of array output power corresponding to the constraint vector $\boldsymbol{a}$, $\boldsymbol{R}_{xx}$ is the ensemble autocorrelation matrix of $\boldsymbol{x}(t)$, and k is a positive parameter. We obtain the optimal $\boldsymbol{a}_o$ and $\boldsymbol{w}_{l1}$ for minimising eqn. 13 as

$$\boldsymbol{a}_o = \left(\boldsymbol{I} + \frac{1}{k}\boldsymbol{C}\boldsymbol{R}_{xx}^{-1}\right)^{-1}\boldsymbol{R}_{xx}(f, \tau)\boldsymbol{c} \quad (14)$$

and

$$\boldsymbol{w}_{l1} = \left(\boldsymbol{R}_{xx} + \frac{1}{k}\boldsymbol{C}\right)^{-1}\boldsymbol{R}_{xx}(f, \tau)\boldsymbol{c} \quad (15)$$

Next, performing the eigendecomposition of $\boldsymbol{R}_{xx}$, we have $\boldsymbol{R}_{xx} = \sum_{i=1}^{M}\lambda_i\boldsymbol{e}_i\boldsymbol{e}_i^H$, where $\lambda_1 \geq \cdots \geq \lambda_{J+1} > \lambda_{J+2} = \cdots = \lambda_M = \sigma_n^2$ are the eigenvalues of $\boldsymbol{R}_{xx}$ in descending order, σ_n^2 is the variance of the noise, and $\boldsymbol{e}_i$, $i = 1, \ldots, M$ are the corresponding eigenvectors. The subspaces spanned by $\boldsymbol{E}_s = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{J+1}\}$ and $\boldsymbol{E}_n = \{\boldsymbol{e}_{J+2}, \ldots, \boldsymbol{e}_M\}$ are called the signal subspace and noise subspace, respectively. Hence, $\boldsymbol{R}_{xx}$ can be rewritten as

$$\boldsymbol{R}_{xx} = \boldsymbol{E}_s\boldsymbol{\Lambda}_s\boldsymbol{E}_s^H + \boldsymbol{E}_n\boldsymbol{\Lambda}_n\boldsymbol{E}_n^H \quad (16)$$

where $\boldsymbol{\Lambda}_s = \mathrm{diag}\{\lambda_1, \ldots, \lambda_{J+1}\}$ and $\boldsymbol{\Lambda}_n = \sigma_n^2\boldsymbol{I}$. Since $\boldsymbol{R}_{xx}(\alpha, \tau)\boldsymbol{c} = r_{ss}(\alpha, \tau)(\boldsymbol{a}_d^H\boldsymbol{c})\boldsymbol{a}_d$ lies in the signal subspace, we use the second method of [9] and define a cost function as

$$J(\boldsymbol{a}) = \boldsymbol{a}^H\boldsymbol{E}_n\boldsymbol{E}_n^H\boldsymbol{a} + k(\boldsymbol{a} - \boldsymbol{R}_{xx}(f, \tau)\boldsymbol{c})^H\boldsymbol{C}^{-1}(\boldsymbol{a} - \boldsymbol{R}_{xx}(f, \tau)\boldsymbol{c}) \quad (17)$$

The optimal $\boldsymbol{a}_o$ and $\boldsymbol{w}_{l2}$ minimising eqn. 17 can be easily obtained as

$$\boldsymbol{a}_o = \left(\boldsymbol{I} + \frac{1}{k}\boldsymbol{C}\boldsymbol{E}_n\boldsymbol{E}_n^H\right)^{-1}\boldsymbol{R}_{xx}(f, \tau)\boldsymbol{c} \quad (18)$$

and

$$w_{l2} = \left(E_s\Lambda_s E_s^H + \left(I + \frac{1}{k}C\right)E_n\Lambda_n E_n^H\right)^{-1} R_{xx}(f, \tau)c \tag{19}$$

3.3 Robust methods based on cross-SCORE algorithm

Since $R_{xu} = R_{xx}(f, \tau)e^{-j\pi f\tau}$ and $R_{uu}^{-1} = R_{xx}^{-1}$, we obtain from eqns. 9 and 10 that the weight vector w_c of the original cross-SCORE algorithm equals the dominant eigenvector of

$$\begin{aligned}\lambda w_c &= R_{xx}^{-1}R_{xx}(f, \tau)R_{xx}^{-1}R_{xx}^H(f, \tau)w_c \\ &= R_{xx}^{-1}R_{xx}(\alpha, \tau)R_{xx}^{-1}R_{xx}^H(\alpha, \tau)w_c \\ &\quad + R_{xx}^{-1}R_{xx}(\alpha, \tau)R_{xx}^{-1}R_e^H(f, \tau)w_c \\ &\quad + R_{xx}^{-1}R_e(f, \tau)R_{xx}^{-1}R_{xx}(\alpha, \tau)w_c \\ &\quad + R_{xx}^{-1}R_e(f, \tau)R_{xx}^{-1}R_e^H(f, \tau)w_c\end{aligned} \tag{20}$$

when T approaches infinity. Substituting $R_{xx}(\alpha, \tau) = r_{ss}(\alpha, \tau)a_d a_d^H$ into eqn. 20 and performing some algebraic manipulations, we have

$$R_{xx}^{-1}R_{xx}(\alpha, \tau)R_{xx}^{-1}R_{xx}^H(\alpha, \tau)w_c = \underbrace{\{|r_{ss}(\alpha, \tau)|^2(a_d^H R_{xx}^{-1}a_d)(a_d^H w_c)\}}_{\gamma_1} R_{xx}^{-1}a_d \tag{21}$$

$$R_{xx}^{-1}R_{xx}(\alpha, \tau)R_{xx}^{-1}R_e^H(f, \tau)w_c = \underbrace{\{r_{ss}(\alpha, \tau)(a_d^H R_{xx}^{-1}R_e(f, \tau)w_c)\}}_{\gamma_2} R_{xx}^{-1}a_d \tag{22}$$

$$R_{xx}^{-1}R_e(f, \tau)R_{xx}^{-1}R_{xx}^H(\alpha, \tau)w_c = \underbrace{\{r_{ss}^*(\alpha, \tau)(a_d^H w_c)\}}_{\gamma_3} R_{xx}^{-1}a_{e1} \tag{23}$$

$$R_{xx}^{-1}R_e(f, \tau)R_{xx}^{-1}R_e^H(f, \tau)w_c = R_{xx}^{-1}a_{e2} \tag{24}$$

where $a_{e1} = R_e(f, \tau)R_{xx}^{-1}a_d$ is a Gaussian random vector, and $a_{e2} = R_e(f, \tau)R_{xx}^{-1}R_e^H(f, \tau)w_c$ is not a Gaussian random vector due to the nonlinear operation of $R_e(f, \tau)$. Let the elements of $R_e(f, \tau)$ be independent identically distributed (i.i.d.) white Gaussian random variables with zero mean and variance σ_e^2. Then, eqn. 23 becomes a white Gaussian random vector. Assume that σ_e^2 is small enough. The term of eqn. 24 is negligible in comparison with the other terms of eqns. 21, 22 and 23. Accordingly, we can rewrite eqn. 20 as

$$w_c \approx R_{xx}^{-1}\left\{\frac{\gamma_1 + \gamma_2}{\lambda}a_d + \frac{\gamma_3}{\lambda}a_{e1}\right\} = \mu R_{xx}^{-1}(a_d + a_e) \tag{25}$$

where

$$\mu = \frac{\gamma_1 + \gamma_2}{\lambda} \quad \text{and} \quad a_e = \frac{\gamma_3}{\gamma_1 + \gamma_2}a_{e1}$$

is a zero-mean white Gaussian random vector. We observe from eqns. 11 and 25 that the weight vectors obtained by using the LS-SCORE and the cross-SCORE algorithms have the similar form in the presence of random CFE. Letting the covariance matrix of a_e in eqn. 25 be $C = \sigma_c^2 I$, we note from eqns. 15 and 19 that based on the cross-SCORE algorithm it is appropriate to set the optimal weight vector equal to the dominant eigenvector of

$$\lambda w_c = R_{zz}^{-1}R_{xx}(f, \tau)R_{zz}^{-1}R_{xx}^H(f, \tau)w_c \tag{26}$$

where

$$R_{zz}^{-1} = \begin{cases} \left(R_{xx} + \dfrac{\sigma_c^2}{k}I\right)^{-1} & \text{for the first method} \\ E_s\Lambda_s^{-1}E_s^H + \dfrac{k}{k + \sigma_c^2}E_n\Lambda_n^{-1}E_n^H & \text{for the second method}\end{cases} \tag{27}$$

4 Theoretical analysis

Here, we evaluate the performance in terms of the array output signal-to-interference plus noise power ratio (SINR) for each of the methods proposed above. Consider an M-element uniform linear array (ULA) excited by a SOI, an interferer, and additive white Gaussian noise (AWGN). Substituting $R_{xx}(\alpha, \tau) = r_{ss}(\alpha, \tau)a_d a_d^H$ and $C = \sigma_c^2 I$ into eqns. 15 and 19, we obtain that the optimal weight vector w_l equals $R_{zz}^{-1}R_{xx}(f, \tau)c = \mu R_{zz}^{-1}(a_d + (1/\mu)R_e(f, \tau)c)$ for the proposed methods based on the LS-SCORE algorithm, where $\mu = r_{ss}(\alpha, \tau)(a_d^H c)$ and R_{zz}^{-1} is given by eqn. 27. In contrast, we have from eqns. 25 and 26 that $w_c \approx \mu R_{zz}^{-1}\{a_d + [\gamma_3/(\gamma_1 + \gamma_2)]a_{e1}\}$ when using the cross-SCORE algorithm, where $\mu = (\gamma_1 + \gamma_2)/\lambda$. Moreover, γ_1, γ_2 and γ_3 are given by eqns. 21, 22 and 23, respectively, except that R_{zz}^{-1} is used instead of R_{xx}^{-1}. As a result, under the situation where the resulting steering error vector is white Gaussian, the optimal weight vector obtained by using the proposed methods developed based on either the LS-SCORE or cross-SCORE algorithms has a general expression

$$w_{eq} = \mu R_{zz}^{-1}(a_d + \sigma_c a_e) \tag{28}$$

where $|a_d|^2 = M$, and $\sigma_c a_e$ denotes the corresponding white Gaussian steering error vector with mean zero and covariance matrix $\sigma_c^2 I$. We note that $\sigma_c a_e$ is given by $(1/\mu)R_e(f, \tau)c$ and $[\gamma_3/(\gamma_1 + \gamma_2)]a_{e1}$ for the proposed methods based on the LS-SCORE algorithm and cross-SCORE algorithm, respectively. Let σ_s^2, σ_j^2, and σ_n^2 represent the powers of SOI, interference and noise, respectively. From eqns. 16 and 27, we can rewrite R_{zz}^{-1} as

$$R_{zz}^{-1} = E_s\tilde{\Lambda}_s^{-1}E_s^H + E_n\tilde{\Lambda}_n^{-1}E_n^H \tag{29}$$

where

$$[\tilde{\Lambda}_s, \tilde{\Lambda}_n] = \begin{cases} \left[\Lambda_s + \dfrac{\sigma_c^2}{k}I, \left(\sigma_n^2 + \dfrac{\sigma_c^2}{k}\right)I\right] & \text{for the first method} \\ \left[\Lambda_s, \dfrac{k + \sigma_c^2}{k}\Lambda_n\right] & \text{for the second method}\end{cases}$$

For deriving the output SINR for the proposed methods, we first rewrite $a_e = c_s a_d + c_j a_j + E_n E_n^H a_e$. Due to the fact that $a_d^H E_n E_n^H a_e = a_j^H E_n E_n^H a_e = 0$, we have

$$\begin{cases} a_d^H a_e = Mc_s + (a_d^H a_j)c_j \\ a_j^H a_e = (a_j^H a_d)c_s + Mc_j\end{cases} \tag{30}$$

and hence

$$c_s = \frac{Ma_d^H a_e - (a_d^H a_j)a_j^H a_e}{M^2 - |a_j^H a_d|^2} \quad \text{and} \quad c_j = \frac{Ma_j^H a_e - (a_j^H a_d)a_d^H a_e}{M^2 - |a_j^H a_d|^2} \tag{31}$$

Moreover, because $\boldsymbol{a}_j^H \boldsymbol{R}_{xx}^{-1} \boldsymbol{a}_d \approx 0$, we have $\boldsymbol{a}_j^H \boldsymbol{E}_s \boldsymbol{\Lambda}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_d \approx 0$ and $\boldsymbol{a}_j^H \{\boldsymbol{R}_{xx} + (\sigma_c^2/k)\boldsymbol{I}\}^{-1} \boldsymbol{a}_d \approx 0$ when σ_c^2/k is small enough. Therefore, $\boldsymbol{a}_j^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_d \approx 0$ for the first and second methods. On the other hand, $\boldsymbol{a}_e^H \boldsymbol{E}_n \boldsymbol{E}_n^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_d = 0$ and $\boldsymbol{a}_e^H \boldsymbol{E}_n \tilde{\boldsymbol{\Lambda}}_n^{-1} \boldsymbol{E}_n^H \boldsymbol{a}_d = 0$ due to the orthogonality of $\boldsymbol{E}_s$ and $\boldsymbol{E}_n$. Accordingly, we have

$$\boldsymbol{a}_e^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_d \approx c_s^* \boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_d \tag{32}$$

and the output power gain of SOI is

$$\begin{aligned} PG_s &= |\boldsymbol{w}_{eq}^H \boldsymbol{a}_d|^2 \\ &= |\mu|^2 |(\boldsymbol{a}_d + \sigma_c \boldsymbol{a}_e)^H \{\boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H + \boldsymbol{E}_n \tilde{\boldsymbol{\Lambda}}_n^{-1} \boldsymbol{E}_n^H\} \boldsymbol{a}_d|^2 \\ &= |\mu|^2 |(\boldsymbol{a}_d + \sigma_c \boldsymbol{a}_e)^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_d|^2 \\ &\approx |\mu|^2 |1 + \sigma_c c_s|^2 (\boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_d)^2 \end{aligned} \tag{33}$$

Similarly, because $\boldsymbol{a}_e^H \boldsymbol{E}_n \boldsymbol{E}_n^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_j = 0$ and $\boldsymbol{a}_e^H \boldsymbol{E}_n \tilde{\boldsymbol{\Lambda}}_n^{-1} \boldsymbol{E}_n^H \boldsymbol{a}_j = 0$, the output power gain of interference can be expressed by

$$\begin{aligned} PG_j &= |\boldsymbol{w}_{eq}^H \boldsymbol{a}_j|^2 = |\mu|^2 |(\boldsymbol{a}_d + \sigma_c \boldsymbol{a}_e)^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_j|^2 \\ &= |\mu \sigma_c|^2 |\boldsymbol{a}_e^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_j|^2 \\ &\approx |\mu \sigma_c c_j|^2 (\boldsymbol{a}_j^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_j)^2 \end{aligned} \tag{34}$$

Moreover, the output power gain of noise is given by

$$\begin{aligned} PG_n &= \boldsymbol{w}_{eq}^H \boldsymbol{w}_{eq} = |\mu|^2 (\boldsymbol{a}_d + \sigma_c \boldsymbol{a}_e)^H \boldsymbol{R}_{zz}^{-2} (\boldsymbol{a}_d + \sigma_c \boldsymbol{a}_e) \\ &= |\mu|^2 \{\boldsymbol{a}_d^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_d + \sigma_c \boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_d \\ &\quad + \sigma_c \boldsymbol{a}_d^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_e + \sigma_c^2 \boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_e\} \\ &= |\mu|^2 \{\boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-2} \boldsymbol{E}_s^H \boldsymbol{a}_d + \sigma_c \boldsymbol{a}_e^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-2} \boldsymbol{E}_s^H \boldsymbol{a}_d \\ &\quad + \sigma_c \boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-2} \boldsymbol{E}_s^H \boldsymbol{a}_e + \sigma_c^2 \boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_e\} \\ &\approx |\mu|^2 \{(\boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-2} \boldsymbol{E}_s^H \boldsymbol{a}_d)(1 + \sigma_c c_s + \sigma_c c_s^*) \\ &\quad + \sigma_c^2 \boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_e\} \end{aligned} \tag{35}$$

where $\boldsymbol{R}_{zz}^{-2} = \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-2} \boldsymbol{E}_s^H + \boldsymbol{E}_n \tilde{\boldsymbol{\Lambda}}_n^{-2} \boldsymbol{E}_n^H$ and $\boldsymbol{a}_e^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-2} \boldsymbol{E}_s^H \boldsymbol{a}_d \approx c_s^* \boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-2} \boldsymbol{E}_s^H \boldsymbol{a}_d$ are obtained from eqns. 29 and 32, respectively.

Next, we evaluate $\boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-p} \boldsymbol{E}_s^H \boldsymbol{a}_d$ and $\boldsymbol{a}_j^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-p} \boldsymbol{E}_s^H \boldsymbol{a}_j$ for $p = 1$ and 2. Consider the case that there only exists a SOI and background noise; it is easy to show that $\boldsymbol{E}_s = \boldsymbol{e}_{max} \propto \boldsymbol{a}_d/\sqrt{M}$ and $\tilde{\boldsymbol{\Lambda}}_s = \xi_s = M\sigma_s^2 + \sigma_n^2 + \sigma_c^2/k$ for the first method. Hence, we have $\boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-p} \boldsymbol{E}_s^H \boldsymbol{a}_d \approx M\xi_s^{-p}$ for the first method. When there exists an additional interferer, we note that $\boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-p} \boldsymbol{E}_s^H \boldsymbol{a}_d \approx M\xi_s^{-p}$ is also valid if $|\boldsymbol{a}_d^H \boldsymbol{a}_j| \ll M$. Similarly, we have $\boldsymbol{a}_j^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-p} \boldsymbol{E}_s^H \boldsymbol{a}_j \approx M\xi_j^{-p}$, where $\xi_j = M\sigma_j^2 + \sigma_n^2 + \sigma_c^2/k$. Accordingly, the output SINR obtained by using eqn. 28 for the first method, based on either the LS-SCORE or the cross-SCORE algorithms, is given by

$$SINR_1 = \frac{PG_s \times \sigma_s^2}{PG_j \times \sigma_j^2 + PG_n \times \sigma_n^2} = \frac{M^2 \xi_s^{-2} |1 + \sigma_c c_s|^2 SNR}{\sigma_c^2 |c_j|^2 M^2 \xi_j^{-2} INR + M\xi_s^{-2}(1 + \sigma_c c_s + \sigma_c c_s^*) + \sigma_c^2 \boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_e} \tag{36}$$

where $SNR = \sigma_s^2/\sigma_n^2$ and $INR = \sigma_j^2/\sigma_n^2$. From eqn. 44 in the Appendix, the expectation of $SINR_1$ is approximately given by

$$\begin{aligned} E\{SINR_1\} &= \frac{M^2 \xi_s^{-2} E\{|1 + \sigma_c c_s|^2\} SNR}{\sigma_c^2 E\{|c_j|^2\} M^2 \xi_j^{-2} INR + M\xi_s^{-2} E\{(1 + \sigma_c c_s + \sigma_c c_s^*)\} + \sigma_c^2 E\{\boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_e\}} \\ &= M^2 \xi_s^{-2} SNR \Bigg/ \Bigg[M\xi_s^{-2} + (M^2 INR + M)\xi_j^{-2} \times \frac{M\sigma_c^2}{M^2 - |\boldsymbol{a}_j^H \boldsymbol{a}_d|^2 + M\sigma_c^2} \\ &\quad + \sigma_c^2 \frac{(M-2)[\sigma_n^2 + (\sigma_c^2/k)]^{-2} (M^2 - |\boldsymbol{a}_j^H \boldsymbol{a}_d|^2)}{M^2 - |\boldsymbol{a}_j^H \boldsymbol{a}_d|^2 + M\sigma_c^2} \Bigg] \end{aligned} \tag{37}$$

Similarly, we have $\boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-p} \boldsymbol{E}_s^H \boldsymbol{a}_d = M\rho_s^{-p}$ and $\boldsymbol{a}_j^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-p} \boldsymbol{E}_s^H \boldsymbol{a}_j = M\rho_j^{-p}$ for the second method, where $\rho_s = M\sigma_s^2 + \sigma_n^2$ and $\rho_j = M\sigma_j^2 + \sigma_n^2$. From eqn. 45, the output SINR and the expectation of the output SINR by

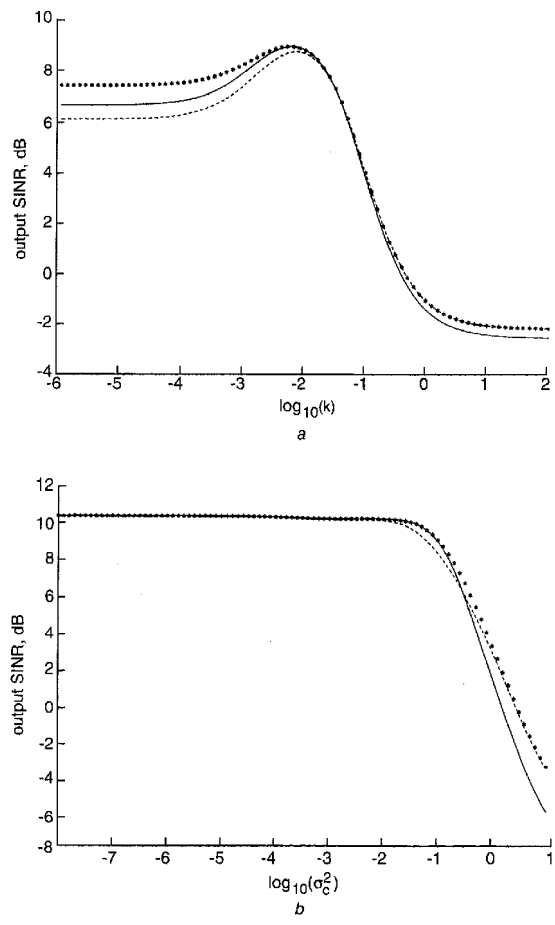

Fig. 1 *Output SINR against logarithmic values of k and σ_c^2 for example 1*

---- simulation results using first method
**** theoretical results using eqn. 36
—— theoretical results using eqn. 37
a Against log *k*
b Against log σ_c^2

using eqn. 28 for the second method, based on either the LS-SCORE or the cross-SCORE algorithms, are approximately given by

$$SINR_2 = \frac{M^2\rho_s^{-2}|1+\sigma_c c_s|^2 SNR}{\sigma_c^2|c_j|^2 M^2\rho_j^{-2} INR + M\rho_s^{-2}(1+\sigma_c c_s+\sigma_c c_s^*) + \sigma_c^2 \boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2}\boldsymbol{a}_e} \tag{38}$$

and

$$\begin{aligned} E\{SINR_2\} &= \frac{M^2\rho_s^{-2}E\{|1+\sigma_c c_s|^2\}SNR}{\sigma_c^2 E\{|c_j|^2 M^2\rho_j^{-2} INR + M\rho_s^{-2}E\{(1+\sigma_c c_s+\sigma_c c_s^*)\} + \sigma_c^2 E\{\boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2}\boldsymbol{a}_e\}} \\ &= M^2\rho_s^{-2}SNR \Big/ \Big[M\rho_s^{-2} + (M^2 INR + M)\rho_j^{-2} \times \frac{M\sigma_c^2}{M^2 - |\boldsymbol{a}_j^H\boldsymbol{a}_d|^2 + M\sigma_c^2} \\ &\quad + \sigma_c^2 \frac{(M-2)\{[k^2/(k+\sigma_c^2)^2]\sigma_n^{-4}\}(M^2 - |\boldsymbol{a}_j^H\boldsymbol{a}_d|^2)}{M^2 - |\boldsymbol{a}_j^H\boldsymbol{a}_d|^2 + M\sigma_c^2} \Big] \end{aligned} \tag{39}$$

5 Computer simulations

For all simulations, we use a ULA with $M = 11$ and interelement spacing = half wavelength of the SOI. A SOI impinging on the array from 5° off broadside has SNR = 0 dB and CF = 2, while an interference with direction angle 30° off broadside has INR = 9 dB and CF = 3. Moreover, both the SOI and interference are BPSK signals with rectangular pulse shape and baud rate = 0.1. The sampling rate is set to 5. The noise is spatially white Gaussian with mean zero and variance one. All the simulation results of the first four examples are obtained by averaging 1000 independent runs and using 2000 data snapshots for each run. We observe that 2000 data snapshots are enough for computing the correlation matrices required for the first four examples. 50 independent runs are averaged for showing the simulation results of the last example.

Example 1: Here, we confirm eqns. 36 and 37 when the optimal weight vector obtained by using the first method, based on either the LS-SCORE or the cross-SCORE algorithm, has the form of eqn. 28. Fig. 1 shows the array output SINR against the logarithmic value of k with $\sigma_c^2 = 0.16$ and against the logarithmic value of σ_c^2 with $k = 0.01$. We observe that the theoretical results

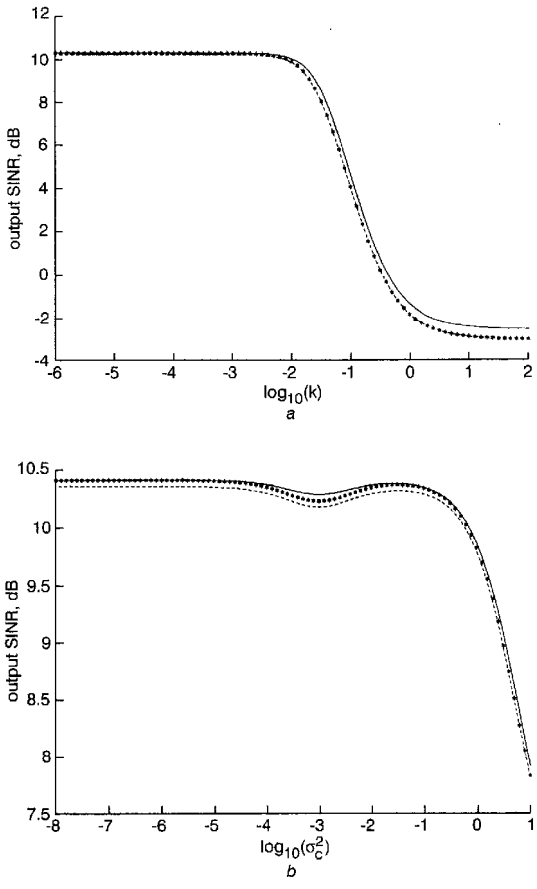

Fig. 2 *Output SINR against logarithmic values of k and σ_c^2 for example 2*

- - - - simulation results using second method
**** theoretical results using eqn. 38
—— theoretical results using eqn. 39
a Against log k
b Against log σ_c^2

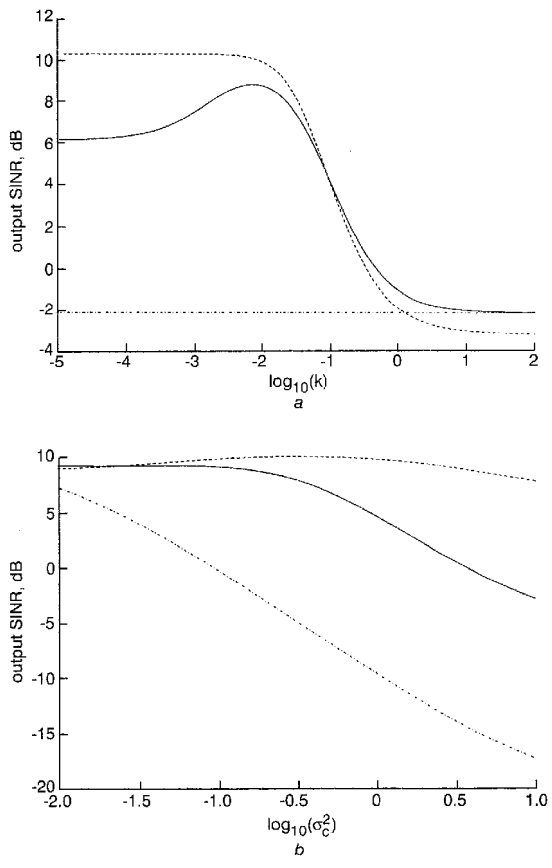

Fig. 3 *Output SINR against logarithmic values of k and σ_c^2 for example 3*

—·—·— original LS-SCORE algorithm
—— first method based on LS-SCORE algorithm
- - - - second method based on LS-SCORE algorithm
a Against log k
b Against log σ_c^2

obtained by eqns. 36 and 37 are close to the simulation results. We also note that the mismatch between the theoretical and simulation results increases as σ_c^2/k increases. This is mainly due to the fact that decreasing, σ_c^2/k improves the approximation $\boldsymbol{a}_f^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-1} \boldsymbol{E}_s^H \boldsymbol{a}_d \approx 0$ for deriving eqn. 32.

Example 2: Here, we confirm eqns. 38 and 39 when the optimal weight vector obtained by using the second method, based on either the LS-SCORE or the cross-SCORE algorithm, has the form of eqn. 28. The output SINR against the logarithmic value of k with $\sigma_c^2 = 0.16$, and against the logarithmic value of σ_c^2 with $k = 0.001$, are plotted in Fig. 2. We observe that the theoretical results obtained by eqns. 38 and 39 are close to the simulation results.

Example 3: This example shows the effectiveness of the proposed methods based on the LS-SCORE algorithm. Substituting $\boldsymbol{R}_{xx}(\alpha, \tau) = r_{ss}(\alpha, \tau)\boldsymbol{a}_d\boldsymbol{a}_d^H$ into eqn. 11, we obtain $\boldsymbol{r}_{xr}(f) = \boldsymbol{R}_{xx}(f, \tau)\boldsymbol{c} = r_{ss}(\alpha, \tau)(\boldsymbol{a}_d^H\boldsymbol{c})\boldsymbol{a}_d + \boldsymbol{a}_e$ that has the form of $\boldsymbol{a}_d + \sigma_c\boldsymbol{a}_e$, where $\sigma_c = 1/[r_{ss}(\alpha, \tau)(\boldsymbol{a}_d^H\boldsymbol{c})]$. Let $\sigma_c\boldsymbol{a}_e$ be a white Gaussian random vector with mean zero and covariance matrix $\sigma_c^2\boldsymbol{I}$. Fig. 3 depicts the output SINR against the logarithmic value of k with $\sigma_c^2 = 0.16$ and against the logarithmic value of σ_c^2 with $k = 0.01$. The simulation results of the original LS-SCORE algorithm are also provided for comparison. We observe that the proposed methods based on the LS-SCORE algorithm can effectively alleviate the performance degradation due to the random CFE by choosing a suitable k. Fig. 3*b* also shows that the second method is more effective than the first method, especially for large σ_c^2.

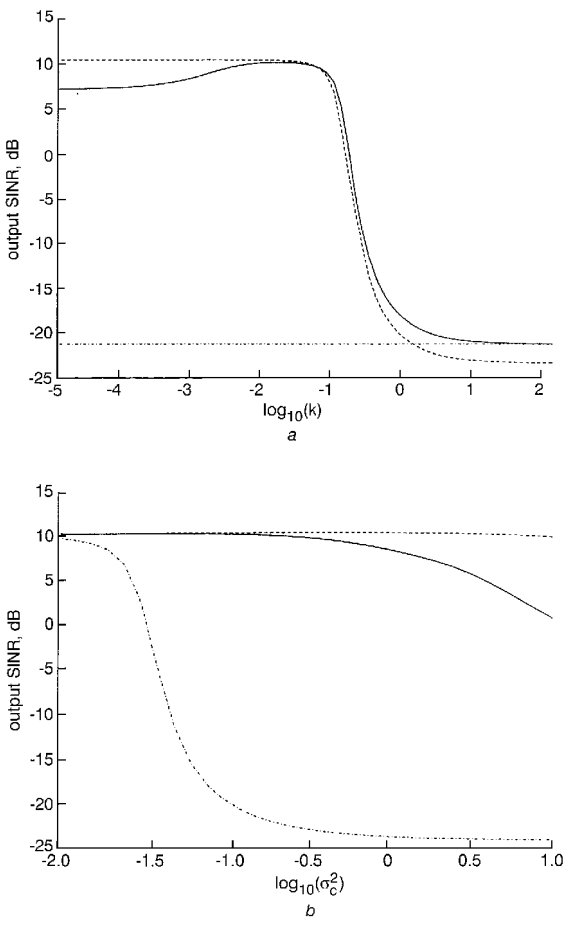

Fig. 4 *Output SINR against logarithmic values of k and σ_c^2 for example 4*

–·–·– original cross-SCORE algorithm
—— first method based on cross-SCORE algorithm
- - - - second method based on cross-SCORE algorithm
a Against log *k*
b Against log σ_c^2

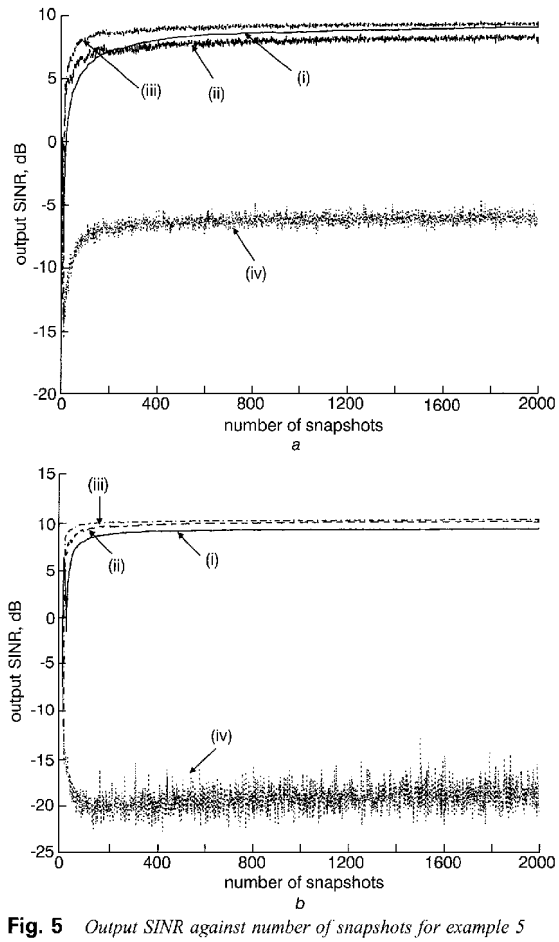

Fig. 5 *Output SINR against number of snapshots for example 5*

(i) Original algorithm without CFE
(ii) First method based on algorithm with random CFE
(iii) Second method based on algorithm with random CFE
(iv) Original algorithm with random CFE
a LS-SCORE algorithm
b Cross-SCORE algorithm

Example 4: Here, we show the effectiveness of the proposed methods based on the cross-SCORE algorithm. Substituting $\boldsymbol{R}_{xx}(\alpha, \tau) = r_{ss}(\alpha, \tau)\boldsymbol{a}_d\boldsymbol{a}_d^H$ into eqn. 10, we obtain $\boldsymbol{R}_{xx}(f, \tau) = r_{ss}(\alpha, \tau)\boldsymbol{a}_d\boldsymbol{a}_d^H + \boldsymbol{R}_e(f, \tau)$ that has the form of $\boldsymbol{a}_d\boldsymbol{a}_d^H + \sigma_c\boldsymbol{R}_e(f, \tau)$, where $\sigma_c = 1/[r_{ss}(f, \tau)]$. Let the elements of $\boldsymbol{R}_e(f, \tau)$ be i.i.d. white Gaussian random variables with mean zero and variance one. The output SINR against the logarithmic value of k with $\sigma_c^2 = 0.16$ and against the logarithmic value of σ_c^2 with $k = 0.01$ is plotted in Fig. 4. We observe that the proposed methods provide satisfactory performance when k is appropriately chosen. Fig. 4*b* also shows that the second method is more robust than the first method, especially for large σ_c^2.

Example 5: We use the same SOI and interference as those used in the above four examples, except that an additional

BPSK interferer with rectangular pulse shape, direction angle $= 40°$ off broadside, INR $= 6$ dB, CF $= 4$, and baud rate $= 0.1$ is added. $k = 0.01$, $\sigma_c^2 = 0.16$, and $\boldsymbol{C} = \boldsymbol{I}$ are used by the proposed methods. Fig. 5 depict the output SINR against the number of data snapshots used. In comparison with the results of using the original SCORE algorithms, we observe that the proposed methods may provide better performance than the original SCORE algorithms, even in the presence of CFE. This is due to the fact that the term $(\sigma_c^2/k)\boldsymbol{I}$ in eqn. 27 for the first method can reduce the finite sample effect if σ_c^2 is small and k is appropriately set. The second method uses the eigenstructure of the autocorrelation matrix, and hence provides better performance under a small σ_c^2 and suitable k.

6 Conclusions

This paper has shown that when the random CFE is an independent random sequence, the resulting cyclic correlation matrix equals the cyclic correlation matrix computed by using the actual cycle frequency plus a Gaussian random error matrix. Based on the theoretical result, two robust methods in conjunction with the SCORE algorithms have been developed to alleviate the performance deterioration due to random CFE. Moreover, we have evaluated the performance for each of the proposed methods. The validity of the theoretical work and the effectiveness of the proposed methods has also been demonstrated by simulation results.

7 Acknowledgments

This work was supported by the National Science Council under grant NSC88-2218-E002-027.

8 References

1 GARDNER, W.A.: 'Cyclostationarity in communications and signal processing' (New York, 1994)
2 AGEE, B.G., SCHELL, S.V., and GARDNER, W.A.: 'Spectral self-coherence restoral: A new approach to blind adaptive signal extraction using antenna arrays', *Proc. IEEE*, 1990, **78**, pp. 753–767
3 GARDNER, W.A.: 'Simplification of MUSIC and ESPRIT by exploitation of cyclostationarity', *Proc. IEEE*, 1988, **76**, pp. 845–847
4 SCHELL, S.V.: 'Performance analysis of cyclic MUSIC method of direction estimation for cyclostationary signals', *IEEE Trans. Signal Process.*, 1994, **42**, (11), pp. 3043–3050
5 SCHELL, S.V.: 'Asymptotic moments of estimated cyclic correlation matrices', *IEEE Trans. Signal Process.*, 1995, **43**, (1), pp. 173–180
6 SCHELL, S.V., and GARDNER, W.A.: 'Progress on signal-selective direction finding'. Proceedings of 5th ASSP workshop on *Spectrum estimation and modelling*, 1990, Rochester, NY, pp. 144–148
7 BIEDKA, T.E., and AGEE, B.G.: 'Subinterval cyclic MUSIC – robust DF with error in cycle frequency knowledge'. Proceedings of 25th Asilomar Conference on *Signals, systems and computers*, 1991, Pacific Grove, CA, pp. 262–266
8 LEE, J.-H., and LEE, Y.-T.: 'Robust adaptive array beamforming for cyclostationary signals under cycle frequency error', *IEEE Trans. Antennas Propag.*, 1999, **47**, pp. 233–241
9 LEE, C.-C., and LEE, J.-H.: 'Robust adaptive array beamforming under steering vector errors', *IEEE Trans. Antennas Propag.*, 1997, **45**, pp. 168–175
10 GIBRA, I.N.: 'Probability and statistical inference for scientists and engineers' (Prentice Hall, 1973)

9 Appendix

Since the elements of $\boldsymbol{a}_e$ are white Gaussian random variables with mean zero and variance one, we have

$$E\{c_s\} = E\{c_j\} = 0 \tag{40}$$

because $E\{\boldsymbol{a}_d^H \boldsymbol{a}_e\} = E\{\boldsymbol{a}_j^H \boldsymbol{a}_e\} = 0$. Consequently:

$$\begin{aligned} E\{|c_s|^2\} &= \frac{1}{(M^2 - |\boldsymbol{a}_d^H \boldsymbol{a}_j|^2)^2} \\ &\quad \times \{M^2 \boldsymbol{a}_d^H E\{\boldsymbol{a}_e \boldsymbol{a}_e^H\} \boldsymbol{a}_d - M(\boldsymbol{a}_j^H \boldsymbol{a}_d) \boldsymbol{a}_d^H E\{\boldsymbol{a}_e \boldsymbol{a}_e^H\} \boldsymbol{a}_j \\ &\quad - M(\boldsymbol{a}_d^H \boldsymbol{a}_j) \boldsymbol{a}_j^H E\{\boldsymbol{a}_e \boldsymbol{a}_e^H\} \boldsymbol{a}_d + |\boldsymbol{a}_d^H \boldsymbol{a}_j|^2 \boldsymbol{a}_j^H E\{\boldsymbol{a}_e \boldsymbol{a}_e^H\} \boldsymbol{a}_j\} \\ &= \frac{M}{M^2 - |\boldsymbol{a}_d^H \boldsymbol{a}_j|^2} \end{aligned} \tag{41}$$

where $E\{\boldsymbol{a}_e \boldsymbol{a}_e^H\} = \boldsymbol{I}$. Similarly, we have

$$E\{|c_j|^2\} = \frac{M}{M^2 - |\boldsymbol{a}_d^H \boldsymbol{a}_j|^2} \tag{42}$$

Next, we substitute $\boldsymbol{a}_e = c_s \boldsymbol{a}_d + c_j \boldsymbol{a}_j + \boldsymbol{E}_n \boldsymbol{E}_n^H \boldsymbol{a}_e$ into $E\{\boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_e\}$ and obtain

$$\begin{aligned} E\{\boldsymbol{a}_e^H \boldsymbol{R}_{zz}^{-2} \boldsymbol{a}_e\} &\approx E\{|c_s|^2\} \boldsymbol{a}_d^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-2} \boldsymbol{E}_s^H \boldsymbol{a}_d \\ &\quad + E\{|c_j|^2\} \boldsymbol{a}_j^H \boldsymbol{E}_s \tilde{\boldsymbol{\Lambda}}_s^{-2} \boldsymbol{E}_s^H \boldsymbol{a}_j \\ &\quad + E\{\boldsymbol{a}_e^H \boldsymbol{E}_n \tilde{\boldsymbol{\Lambda}}_n^{-2} \boldsymbol{E}_n^H \boldsymbol{a}_e\} \end{aligned} \tag{43}$$

It is easy to show that

$$E\{\boldsymbol{a}_e^H \boldsymbol{E}_n \tilde{\boldsymbol{\Lambda}}_n^{-2} \boldsymbol{E}_n^H \boldsymbol{a}_e\} = \text{trace}\{\tilde{\boldsymbol{\Lambda}}_n^{-2}\} = (M-2)\left(\sigma_n^2 + \frac{\sigma_c^2}{k}\right)^{-2} \tag{44}$$

when using the first method, and

$$E\{\boldsymbol{a}_e^H \boldsymbol{E}_n \tilde{\boldsymbol{\Lambda}}_n^{-2} \boldsymbol{E}_n^H \boldsymbol{a}_e\} = \text{trace}\{\tilde{\boldsymbol{\Lambda}}_n^{-2}\} = (M-2)\frac{k^2}{(k+\sigma_c^2)^2}\sigma_n^{-4} \tag{45}$$

when using the second method.