

# Robust Adaptive Array Beamforming for Cyclostationary Signals Under Cycle Frequency Error

Ju-Hong Lee and Yung-Ting Lee

**Abstract**—This paper deals with the problem of robust adaptive array beamforming for cyclostationary signals. By exploiting the signal cyclostationarity, the SCORE algorithms presented in [6] 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 even if there is a small mismatch in the cycle frequency of the desired signal. In this paper, we first evaluate the performance of the SCORE algorithms in the presence of cycle frequency error (CFE). An analytical formula is derived to show the behavior of the performance degradation due to CFE. Based on the theoretical result, we then develop an efficient method in conjunction with the SCORE algorithms to achieve robust adaptive beamforming against the CFE. Several simulation examples for confirming the theoretical analysis and showing the effectiveness of the proposed method are also presented.

**Index Terms**—Adaptive arrays, beamforming, cyclostationary signals.

## I. INTRODUCTION

ARRAYS of sensors such as radio antennas are useful in the process of detecting the presence of the desired signal, estimating their directions of arrival (DOA) and other parameters, and estimating the signal waveforms themselves. For conventional array beamforming, the *priori* information required for adapting the weights is either the direction of arrival or the waveform of the desired signal [1]. For a steered-beam adaptive beamformer, the adaptive weights are calculated by minimizing the beamformer's output power subject to the constraint that forces the array to make a constant response at the steering direction. Hence, the performance of the beamformer is very sensitive to the accuracy of the steering vector. It has been shown that the mismatch between the direction vector of the desired signal and the steering vector causes severe performance degradation even if a small mismatch arises. In practice, the problem of the mismatch possibly exists in many applications such as the application in mobile communication systems. This leads to many restrictions on the development of the conventional array beamforming.

A signal with cyclostationarity has the statistical property of correlating with either a frequency-shift or complex-conjugate version of itself. By restoring this property at a known value of frequency separation, it is possible to favor the desired signal

and to discriminate against the interference and noise [2], [5]. In addition, many cyclostationarity-exploiting algorithms avoid the need for training signals, the knowledge of array manifold, and noise characteristics. Moreover, adaptive beamforming can suppress not only Gaussian but also non-Gaussian interferers by utilizing the signal cyclostationarity. As a result, adaptive beamforming utilizing signal cyclostationarity has been widely considered. Recently, a class of spectral self-coherent restoral (SCORE) algorithms has been presented in [6] to deal with the problem of blind adaptive signal extraction. When infinite time-averaging interval is available, it has been shown in [6] that the solutions of the SCORE algorithms are the same as that of the conventional approaches developed based on maximizing the output signal-to-interference plus noise ratio (SINR) for adaptive beamforming. The *priori* information that the SCORE algorithms require to work is only the cycle frequencies of the desired signal. Hence, its performance is sensitive to the accuracy of the presumed cycle frequencies. However, the actual cycle frequencies may not be known very well in some applications due to for example the phenomenon of Doppler shift. Therefore, it is worth evaluating the performance of the SCORE algorithms in the presence of cycle frequency error (CFE) and developing techniques to solve the problem. Some previous work considering the related problem in direction finding has been reported in [7]–[10]. A statistical analysis of the usual estimator of the cycle autocorrelation is included in [7] and [8]. This analysis shows the effects of cycle leakage through a sinc window due to finite data samples. On the other hand, two approaches have been presented by [9] for dealing with the direction-finding problems due to CFE. The first one, called multicyclic MUSIC, uses a multicyclic correlation matrix to alleviate the performance degradation due to CFE. However, it is observed that this approach cannot tackle the problem effectively. The other one, called adaptive- $\alpha$  cyclic MUSIC, estimates the cycle frequency by using the fast Fourier transform (FFT) frequency estimator to compute the cyclic autocorrelation of the data received from a single sensor. The drawback of this approach is that huge memory is required for data storage to obtain satisfactory results. Variations of the adaptive- $\alpha$  cyclic MUSIC are explored in [10]. Based on the concept of subintervals, only the cyclic correlation matrices of the subintervals are required to reduce the memory requirements and the sensitivity of cyclic MUSIC to CFE. However, the techniques presented by [10] limits the cycle frequency coverage due to using subintervals.

Manuscript received July 8, 1997; revised July 13, 1998. This work was supported by the National Science Council under Grant NSC86-2213-E002-054.

The authors are with the Department of Electrical Engineering, National Taiwan University, Taipei, 106, Taiwan, R.O.C.

Publisher Item Identifier S 0018-926X(99)03728-X.

In the literature, two of the SCORE algorithms presented by [6], namely the least-square SCORE (LS-SCORE) and the cross-SCORE algorithms have been extensively considered [7]. In this paper, we first evaluate the performance of these two SCORE algorithms when the CFE regarding the desired signal exists. Using the theory of Fourier transform, the problem formulation for each of these two SCORE algorithms under CFE is presented. This results in an analytical formula which demonstrates the behavior of the performance degradation for these two SCORE algorithms. It is shown that the output SINR of an adaptive array beamformer using either the LS-SCORE or the cross-SCORE algorithm deteriorates like a sinc function as the number of data snapshots increases. Based on this theoretical result, an efficient method is developed to alleviate the effect of CFE on array performance. Based on the proposed method, the cycle frequency of the desired signal is estimated according to an iterative procedure. This estimate is then used to form an appropriate reference signal and control vector for these two SCORE algorithms. It is shown that adaptive array beamforming using the proposed method in conjunction with the SCORE algorithms can effectively cure the problem of blind signal extraction in the presence of CFE.

This paper is organized as follows. In Section II, we briefly describe the original LS-SCORE and cross-SCORE algorithms presented by [6]. The performance analyses of these two SCORE algorithms in the presence of CFE are presented in Section III. An efficient method is then proposed to alleviate the performance degradation caused by the CFE in Section IV. Modification of the proposed method for reducing the computational burden and storage requirement is presented in Section V. Several computer simulation examples for confirming the theoretical results and showing the effectiveness of the proposed method are provided in Section VI. Finally, we conclude the paper in Section VII.

## II. ADAPTIVE BEAMFORMING FOR CYCLOSTATIONARY SIGNALS

### A. Signal Cyclostationarity

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

$$R_s(\alpha, \tau) = \langle s(t + \tau/2)s^*(t - \tau/2)e^{-j2\pi\alpha t} \rangle_\infty \quad (1)$$

and

$$R_{s^*}(\alpha, \tau) = \langle s(t + \tau/2)s(t - \tau/2)e^{-j2\pi\alpha t} \rangle_\infty \quad (2)$$

respectively, where the superscript “\*” denotes the complex conjugate.  $s(t)$  is then said to be cyclostationary if  $R_s(\alpha, \tau)$  or  $R_{s^*}(\alpha, \tau)$  does not equal zero at cycle frequency  $\alpha$  for some time delay  $\tau$ . Many man-made communication signals exhibit cyclostationarity with cycle frequency equal to the twice of the carrier frequency or multiples of the baud rate or combinations of these [3], [4].

Let the data vector received by an array beamformer be designated as  $\mathbf{x}(t)$ . Then its cyclic autocorrelation matrix and cyclic conjugate autocorrelation matrix are given by

$$\mathbf{R}_{xx}(\alpha, \tau) = \langle \mathbf{x}(t + \tau/2)\mathbf{x}^H(t - \tau/2)e^{-j2\pi\alpha t} \rangle_\infty \quad (3)$$

and

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

respectively, where the superscript “ $H$ ” denotes the conjugate transpose and “ $T$ ” the transpose. Next, we briefly describe adaptive beamforming utilizing signal cyclostationarity.

### B. The SCORE Algorithms

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

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

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

Assume that  $s(t)$  is cyclostationary and has a cycle frequency  $\alpha$ , but  $\mathbf{i}(t)$  is not cyclostationary at  $\alpha$  and is temporally uncorrelated with  $s(t)$ . Based on the LS-SCORE algorithm of [6], a cost function is defined as follows:

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

where the reference signal  $r(t)$  is given by

$$r(t) = \mathbf{c}^H \mathbf{x}(t - \tau)e^{j2\pi\alpha t} \quad (7)$$

and  $\langle \cdot \rangle_T$  denotes the average over the time interval  $[0, T]$ .  $\mathbf{c}$  is a control vector and is fixed for the LS-SCORE algorithm. The optimal weight vector  $\mathbf{w}_{ls}$  minimizing (6) is given by

$$\mathbf{w}_{ls} = \hat{\mathbf{R}}_{xx}^{-1} \hat{\mathbf{r}}_{xr}(\alpha) \quad (8)$$

where  $\hat{\mathbf{R}}_{xx} = \langle \mathbf{x}(t)\mathbf{x}^H(t) \rangle_T$  and  $\hat{\mathbf{r}}_{xr}(\alpha) = \langle \mathbf{x}(t)r^*(t) \rangle_T$  are the sample autocorrelation matrix of  $\mathbf{x}(t)$  and the cross-correlation vector of  $\mathbf{x}(t)$  and  $r(t)$  computed over  $[0, T]$ , respectively. For any control vector  $\mathbf{c}$  as long as  $\mathbf{c}^H \mathbf{a}_d \neq 0$ , it is shown in [6] that (8) converges to the solution of conventional adaptive array beamforming based on the maximum output SINR criterion when  $T$  approaches infinite.

Due to the fact that  $r(t)$  contains the interference, the LS-SCORE algorithm converges slowly. To tackle this drawback, the Cross-SCORE algorithm is proposed by [6] based on maximizing the correlation coefficient which is given by

$$\tilde{F}_{sc}(\mathbf{w}; \mathbf{c}) = \frac{|\mathbf{w}^H \hat{\mathbf{R}}_{xu} \mathbf{c}|^2}{(\mathbf{w}^H \hat{\mathbf{R}}_{xx} \mathbf{w})(\mathbf{c}^H \hat{\mathbf{R}}_{uu} \mathbf{c})} \quad (9)$$

between  $y(t)$  and  $r(t)$ , where  $\mathbf{u}(t) = \mathbf{x}(t - \tau)e^{j2\pi\alpha t}$  is the control signal. From the Cauchy-Schwartz inequality, the optimal weight and control vectors maximizing (9) are given by

$$\mathbf{w}_{opt} = \eta_1 \hat{\mathbf{R}}_{xx}^{-1} \hat{\mathbf{R}}_{xu} \mathbf{c}_{opt}, \quad \text{for a fixed } \mathbf{c}_{opt} \quad (10)$$

and

$$\mathbf{c}_{opt} = \eta_2 \hat{\mathbf{R}}_{uu}^{-1} \hat{\mathbf{R}}_{ux} \mathbf{w}_{opt}, \quad \text{for a fixed } \mathbf{w}_{opt} \quad (11)$$

respectively, where  $\eta_1$  and  $\eta_2$  represent the normalization constants. It has been shown in [6] that  $\mathbf{w}_{\text{opt}}$  is also the solution that maximizes the output SINR when  $T$  is infinite. Moreover, the simulation results presented in [6] show that the cross-SCORE algorithm possesses the advantage of faster convergence speed over the LS-SCORE algorithm, especially in the presence of strong interference. However, the cross-SCORE algorithm requires to solve a generalized eigenvalue problem or to adapt the weight and control vectors simultaneously. Hence, it requires more computational complexity than the LS-SCORE algorithm.

### III. PERFORMANCE ANALYSIS IN THE PRESENCE OF CFE

Here, the performances of the LS-SCORE and the cross-SCORE algorithms in the presence of CFE are evaluated. We only present the results using the cyclic autocorrelation matrix for (8) and (10), respectively. Based on the presented results in this section, it is very straightforward to obtain the results using the cyclic conjugate autocorrelation matrix. From (3), the cyclic autocorrelation matrix at an arbitrary frequency  $f$  can be written as

$$\begin{aligned} \mathbf{R}_{xx}(f, \tau) &= \left\langle \mathbf{x}\left(t + \frac{\tau}{2}\right) \mathbf{x}^H\left(t - \frac{\tau}{2}\right) e^{-j2\pi f t} \right\rangle_{\infty} \\ &= R_s(f, \tau) \mathbf{a}_d \mathbf{a}_d^H + \sum_{j=1}^J R_{s_j}(f, \tau) \mathbf{a}_j \mathbf{a}_j^H \\ &\quad + \mathbf{R}_n(f, \tau) \end{aligned} \quad (12)$$

where

$$R_s(f, \tau) = \left\langle s\left(t + \frac{\tau}{2}\right) s^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f t} \right\rangle_{\infty} \quad (13)$$

$$R_{s_j}(f, \tau) = \left\langle s_j\left(t + \frac{\tau}{2}\right) s_j^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f t} \right\rangle_{\infty} \quad (14)$$

and

$$\mathbf{R}_n(f, \tau) = \left\langle \mathbf{n}\left(t + \frac{\tau}{2}\right) \mathbf{n}^H\left(t - \frac{\tau}{2}\right) e^{-j2\pi f t} \right\rangle_{\infty}. \quad (15)$$

Due to the fact that the cyclic spectrum of a cyclostationary signal is discrete in the cycle frequency, the cyclic autocorrelation function of the SOI can be written as

$$R_s(f, \tau) = \sum_n d_n(\tau) \delta(f - \alpha_n) \quad (16)$$

where  $\alpha_n$  denotes the cycle frequencies of the SOI and  $d_n(\tau)$  the corresponding strengths at  $\alpha_n$ . Next, we can rewrite the sample cross-correlation vector of  $\mathbf{x}(t)$  and  $r(t)$  at  $f$  as follows:

$$\hat{\mathbf{r}}_{xr}(f) = \langle \mathbf{x}(t) r^*(t) \rangle_T = \left\langle \frac{1}{T} \text{rect}\left(\frac{t}{T}\right) \mathbf{x}(t) r^*(t) \right\rangle_{\infty} \quad (17)$$

where

$$\text{rect}\left(\frac{t}{T}\right) = \begin{cases} 1, & \text{for } -\frac{T}{2} \leq t \leq \frac{T}{2}, \\ 0, & \text{elsewhere.} \end{cases} \quad (18)$$

Substituting (7) into (17) yields

$$\hat{\mathbf{r}}_{xr}(f) = \left\langle \frac{1}{T} \text{rect}\left(\frac{t}{T}\right) \mathbf{x}(t) \mathbf{x}^H(t - \tau) e^{-j2\pi f t} \right\rangle_{\infty} \mathbf{c}. \quad (19)$$

Utilizing the property of Fourier transform, we obtain

$$\begin{aligned} \hat{\mathbf{r}}_{xr}(f) &= \left\langle \frac{1}{T} \text{rect}\left(\frac{t}{T}\right) e^{-j2\pi f t} \right\rangle_{\infty} \\ &\quad \star \langle \mathbf{x}(t) \mathbf{x}^H(t - \tau) e^{-j2\pi f t} \rangle_{\infty} \mathbf{c} \\ &= \text{SINC}(fT) \star \mathbf{r}_{xr}(f) \end{aligned} \quad (20)$$

where

$$\text{SINC}(fT) = \frac{\sin(\pi fT)}{(\pi fT)} \quad (21)$$

and “ $\star$ ” denotes the convolution operation. From (7) and (12), the cross-correlation vector  $\mathbf{r}_{xr}(f)$  can be further expressed by

$$\begin{aligned} \mathbf{r}_{xr}(f) &= \mathbf{R}_{xx}(f, \tau) \mathbf{c} e^{-j\pi f \tau} \\ &= k_d R_s(f, \tau) \mathbf{a}_d + \sum_{j=1}^J k_j R_{s_j}(f, \tau) \mathbf{a}_j + \mathbf{k}_n \end{aligned} \quad (22)$$

where  $k_d = \mathbf{a}_d^H \mathbf{c} e^{-j\pi f \tau}$ ,  $k_j = \mathbf{a}_j^H \mathbf{c} e^{-j\pi f \tau}$ , and  $\mathbf{k}_n = \mathbf{R}_n(f, \tau) \mathbf{c} e^{-j\pi f \tau}$ . By substituting (22) into (20) and using (16), the sample cross-correlation vector  $\hat{\mathbf{r}}_{xr}(f)$  of (20) becomes

$$\begin{aligned} \hat{\mathbf{r}}_{xr}(f) &= k_d \left\{ \sum_n d_n(\tau) \text{SINC}((f - \alpha_n)T) \right\} \mathbf{a}_d \\ &\quad + \sum_{j,m} k_j \{ \rho_{j,m}(\tau) \text{SINC}((f - \beta_{j,m})T) \} \mathbf{a}_j \\ &\quad + \tilde{\mathbf{k}}_{nn} \star \text{SINC}(fT) \end{aligned} \quad (23)$$

where  $\beta_{j,m}$ 's are the cycle frequencies of the  $j$ th cyclostationary interferer and  $\rho_{j,m}$ 's are the corresponding strengths. Moreover, the interferers without cyclostationarity and noise are contained in  $\tilde{\mathbf{k}}_{nn}$ . In fact,  $\hat{\mathbf{r}}_{xr}(f)$  also includes the cyclic cross correlations between the SOI and the interference, the SOI and noise, and the interference and noise. However, they are negligible when  $T$  is large enough. Equation (23) shows the effects of cycle leakage through a sinc window due to finite data samples. These effects are also derived in [7] and [8] by using a different manner.

Next, let the presumed cycle frequency for the SOI be denoted by  $\hat{\alpha}$ , which differs from the cycle frequency  $\alpha_1$  of the SOI by  $\Delta\alpha$ , where  $\Delta\alpha$  represents the amount of CFE. Substituting  $f = \alpha_1 + \Delta\alpha$  into (23) yields

$$\begin{aligned} \hat{\mathbf{r}}_{xr}(\alpha_1 + \Delta\alpha) &= k_d \left\{ (\Delta\alpha T) \right. \\ &\quad \left. + \sum_{n \neq 1} d_n(\tau) \text{SINC}(\tilde{\alpha}_n T) \right\} \mathbf{a}_d \\ &\quad + \sum_{j,m} k_j \{ \rho_{j,m}(\tau) \text{SINC}(\tilde{\beta}_{j,m} T) \} \mathbf{a}_j \\ &\quad + \tilde{\mathbf{k}}_{nn} \star \text{SINC}((\alpha + \Delta\alpha)T) \end{aligned} \quad (24)$$

where  $\tilde{\alpha}_n = \alpha_1 - \alpha_n + \Delta\alpha$  and  $\tilde{\beta}_{j,m} = \alpha_1 - \beta_{j,m} + \Delta\alpha$ . Consider the case where  $\alpha_1$  and  $\beta_{j,m}$ 's are well separated. Using the fact that the value of  $\text{SINC}(fT)$  is small enough for large  $fT$ , the effect of interference and noise is negligible when  $\Delta\alpha = 0$ . Consequently, we note from (24) that  $\hat{\mathbf{r}}_{xr}(f)$  is almost proportional to the aperture vector  $\mathbf{a}_d$ . On the other hand, when  $\Delta\alpha \neq 0$  due to the fact that  $\text{SINC}(\Delta\alpha T) = 0$

when  $T = T_n = (\pm n)/\Delta\alpha$ ,  $n = 1, 2, \dots$ , we note that  $\hat{\mathbf{r}}_{xr}(f)$  is approximately equal to  $k_d d_1(\tau) \text{SINC}(\Delta\alpha T) \mathbf{a}_d$  if  $T$  is far from  $T_n$ . However, when  $T$  is close to  $T_n$ , the effect due to the interference and noise is not negligible. Thus,  $\hat{\mathbf{r}}_{xr}(f)$  is not proportional to  $\mathbf{a}_d$  any more. This leads to performance degradation for the LS-Score algorithm. Owing to  $T_n = (\pm n)/\Delta\alpha$ , we would expect that there exist periodic nulls in the curves of the output SINR versus the number of snapshots for adaptive beamforming based on the LS-Score algorithm in the presence of CFE.

Considering the performance analysis for the cross-Score algorithm, we note from (10) and (11) that the only difference between the LS-Score and cross-Score algorithms is that the cross-Score algorithm adapts the control vector  $\mathbf{c}$ . From (10), the factor  $\hat{\mathbf{R}}_{xu} \mathbf{c}_{\text{opt}}$  can be written as

$$\begin{aligned} \mathbf{R}_{xu} \mathbf{c}_{\text{opt}} &= \{\mathbf{R}_{xx}(f, \tau) e^{-j\pi f \tau}\} \mathbf{c}_{\text{opt}} \\ &= \bar{k}_d R_s(f, \tau) \mathbf{a}_d + \sum_{j=1}^J \bar{k}_j R_{s_j}(f, \tau) \mathbf{a}_j + \bar{\mathbf{k}}_n \end{aligned} \quad (25)$$

when the number of data snapshots is infinite, where  $\bar{k}_d = \mathbf{a}_d^H \mathbf{c}_{\text{opt}} e^{-j\pi f \tau}$ ,  $\bar{k}_j = \mathbf{a}_j^H \mathbf{c}_{\text{opt}} e^{-j\pi f \tau}$ , and  $\bar{\mathbf{k}}_n = \mathbf{R}_n(f, \tau) \mathbf{c}_{\text{opt}} e^{-j\pi f \tau}$ . Comparing (22) and (25), we note that they have the same form except that the control vector  $\mathbf{c}$  of (22) is replaced by the control vector  $\mathbf{c}_{\text{opt}}$  in order to obtain (25). Hence, the performance degradation similar to that shown above for the LS-Score algorithm happens to the cross-Score algorithm when the CFE arises. Simulation results showing the performance degradation due to CFE are presented in Section VI for confirmation.

#### IV. THE PROPOSED METHOD

A main property of a signal that exhibits cyclostationarity is that spectrum lines are regenerated by quadratic nonlinearities only at discrete frequencies  $\alpha_n$  as can be seen from (16). Based on (24), both of the  $\text{SINC}(\Delta\alpha T)$  and  $\text{SINC}(\tilde{\alpha}_n T)$  approach zero as  $T$  increases. Hence, the performance degradation for these two Score algorithms becomes more severe as the number of snapshots increases. To alleviate this difficulty, we present an efficient method in conjunction with the Score algorithms as follows.

First, define a sample autocorrelation-related matrix of the received data vector  $\mathbf{x}(t)$  as follows:

$$\hat{\mathbf{R}}_{xx}(\hat{\alpha}, \tau, \kappa) = \langle e^{j\pi\kappa t} \mathbf{x}(t) \mathbf{x}^H(t - \tau) e^{-j2\pi\hat{\alpha}t} \rangle_T \quad (26)$$

where  $\kappa$  denotes a frequency parameter. Based on (26), we further construct the following matrix product:

$$\begin{aligned} \hat{\mathbf{R}}_{xx}(\hat{\alpha}, \tau, \kappa) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, \tau, \kappa) \\ = \langle \mathbf{x}(t) \mathbf{x}^H(t - \tau) e^{-j2\pi(\hat{\alpha} - \kappa/2)t} \rangle_T \langle \mathbf{x}(t) \mathbf{x}^H(t - \tau) \\ \cdot e^{-j2\pi(\hat{\alpha} - \kappa/2)t} \rangle_T^H. \end{aligned} \quad (27)$$

It is easy to show from (12) that (27) becomes

$$\begin{aligned} \mathbf{R}_{xx}(\hat{\alpha}, \tau, \kappa) \mathbf{R}_{xx}^H(\hat{\alpha}, \tau, \kappa) \\ \propto (\mathbf{a}_d^H \mathbf{a}_d) \left| R_s\left(\hat{\alpha} - \frac{\kappa}{2}, \tau\right) \right|^2 \mathbf{a}_d \mathbf{a}_d^H \\ + \sum_j (\mathbf{a}_j^H \mathbf{a}_j) \left| R_{s_j}\left(\hat{\alpha} - \frac{\kappa}{2}, \tau\right) \right|^2 \mathbf{a}_j \mathbf{a}_j^H \end{aligned} \quad (28)$$

when the time interval  $T$  approaches infinite, where  $\propto$  denotes “proportional to.” Expression (28) reveals that the matrix product is a matrix with rank one if  $\hat{\alpha} - \kappa/2 = \alpha_1$  and a null matrix if  $\hat{\alpha} - \kappa/2$  does not equal the cycle frequencies of the SOI and interferers. When  $T$  is finite, we note that the largest eigenvalue of the matrix product increases as  $\hat{\alpha} - \kappa/2$  approaches  $\alpha_1$  and the dominance of the component  $(\mathbf{a}_d^H \mathbf{a}_d) |\hat{R}_s(\hat{\alpha} - \kappa/2, \tau)|^2 \mathbf{a}_d \mathbf{a}_d^H$  in (28) becomes more significant. Hence, the matrix product is closer to a matrix with rank one as the time interval  $T$  increases. On the other hand, when  $\hat{\alpha} - \kappa/2$  equals neither  $\alpha_1$  nor any one of the cycle frequencies of the  $J$  interferers, the matrix product is a matrix with rank  $(J + 1)$  and its largest eigenvalue decreases as  $T$  increases.

Based on the above observations, an estimation procedure for finding an appropriate estimate of the cycle frequency  $\alpha_1$  of the SOI when  $n$  data snapshots are sampled at the sampling interval  $T_s$  is summarized as follows:

*Procedure 1:*

- Step 1) Let the robust interval be  $\kappa_b(0)$  for performing the estimation and the matrix of (26) computed by using  $n$  data snapshots be designated as  $\hat{\mathbf{R}}_{xx}(\hat{\alpha}, n, \tau, \kappa)$ . Use the first  $N_o$  data snapshots to compute the initial matrix of  $\hat{\mathbf{R}}_{xx}(\hat{\alpha}, n, \tau, \kappa) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, n, \tau, \kappa)$ , where  $N_o \geq L/(2\kappa_b(0)T_s)$  and  $L$  is a preset positive integer.
- Step 2) Set the increment  $\delta\kappa(n)$  of the frequency parameter  $\kappa$  equal to  $\delta\kappa(0) = 1/(2N_oT_s)$  when the number of data snapshots  $n \leq N_o$ . That is, the increment equals a constant  $1/(2N_oT_s)$  for the first  $N_o$  data samples. Otherwise, set  $\delta\kappa(n) = 1/(2nT_s)$  which decreases as  $n$  increases to improve the accuracy of the estimation.
- Step 3) For  $n \leq N_o$ , compute the largest eigenvalue of (27) for each  $\kappa$  taking the values starting from  $-\tilde{\kappa}_b(0)$  to  $\tilde{\kappa}_b(0)$  with an increment  $\delta\kappa(0)$ , where  $\tilde{\kappa}(0) = \max\{\delta\kappa(n) \lfloor \kappa_b(0)/\delta\kappa(n) \rfloor, \kappa_b(0)\}$  and  $\lfloor a \rfloor$  denotes the nearest integer  $\geq a$ . For  $n > N_o$ , if  $|\hat{\alpha}(n-1) - \hat{\alpha}(n-2)| = |\kappa_f(n-1) - \kappa_f(n-2)|/2 \leq \delta\kappa(n-1)$ , this shows that the obtained estimate is very close to the true value. Hence, we compute the largest eigenvalue of (27) for each  $\kappa$  taking the values starting from  $[\hat{\alpha}(n-1) - 1/(2nT_s)]$  to  $[\hat{\alpha}(n-1) + 1/(2nT_s)]$  with an increment  $\delta\kappa(n) = 1/(2nT_s)$ . Instead of employing conventional eigenvalue decomposition (EVD) directly for computing the largest eigenvalues, we present an efficient approach for finding an appropriate approximate of the largest eigenvalue at each  $\kappa$  to considerably reduce the required computational complexity from  $O(M^3)$  to  $O(M^2)$  as follows.

- (3.1) Compute the average of the column vectors of  $\hat{\mathbf{R}}_{xx}(\hat{\alpha}, n, \tau, \kappa) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, n, \tau, \kappa)$  as follows:

$$\begin{aligned} \hat{\mathbf{g}}_{\max}(n, \kappa) \\ = \frac{1}{M} \hat{\mathbf{R}}_{xx}(\hat{\alpha}, n, \tau, \kappa) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, n, \tau, \kappa) \\ \cdot [1, 1, \dots, 1]^T. \end{aligned} \quad (29)$$

(3.2) Compute an appropriate approximate of the largest eigenvalue at  $\kappa$  according to the Rayleigh theory [8, pp. 216–217] as shown in (30) at the bottom of the page.

Step 4) Find the highest value of the approximates of the largest eigenvalues obtained from Step 3 and let the corresponding frequency parameter  $\kappa$  be designated as  $\kappa_f(n)$ .

Step 5) Compute the estimate of the cycle frequency  $\alpha_1$  as follows:  $\hat{\alpha}(n) = \hat{\alpha} - \kappa_f(n)/2$ .

After obtaining the appropriate estimate  $\hat{\alpha}(n)$  for the cycle frequency  $\alpha_1$  of the SOI, we then construct the following new reference and control signals for the LS-SCORE and Cross-SCORE algorithms, respectively, in (31) below, where  $\mathbf{x}(n)$ ,  $\tilde{\mathbf{u}}(n)$ , and  $\tilde{r}(n)$  denote the signals of  $\mathbf{x}(t)$ ,  $\tilde{\mathbf{u}}(t)$ , and  $\tilde{r}(t)$  sampled at the  $n$ th time instant. From (22) and the property of Fourier transform, the cross-correlation vector of  $\mathbf{x}(t)$  and the new reference signal  $\tilde{r}(t)$  is given by

$$\begin{aligned} \mathbf{r}_{x\tilde{r}}(f) &= \text{rect}\left(\frac{f}{2\delta\kappa(n)}\right) \star \mathbf{r}_{xr}(f) \\ &= k_d \left\{ \text{rect}\left(\frac{f}{2\delta\kappa(n)}\right) \star R_s(f, \tau) \right\} \mathbf{a}_d \\ &\quad + \sum_{j=1}^J k_j \left\{ \text{rect}\left(\frac{f}{2\delta\kappa(n)}\right) \star R_{s_j}(f, \tau) \right\} \mathbf{a}_j \\ &\quad + \left\{ \text{rect}\left(\frac{f}{2\delta\kappa(n)}\right) \star \mathbf{k}_n \right\}. \end{aligned} \quad (32)$$

Using the results given by (24) and (32), we obtain the corresponding sample cross-correlation vector of  $\mathbf{x}(t)$  and  $\tilde{r}(t)$  from (32) by letting  $f = \hat{\alpha}(n)$  as follows:

$$\begin{aligned} \hat{\mathbf{r}}_{x\tilde{r}}(\hat{\alpha}(n)) &= \text{rect}\left(\frac{f}{2\delta\kappa(n)}\right) \star \hat{\mathbf{r}}_{xr}(f)|_{f=\hat{\alpha}(n)} \\ &= k_d \left\{ d_1(\tau)\Omega(\Delta\alpha T) + \sum_{n \neq 1} d_n(\tau)\Omega(\ddot{\alpha}_n T) \right\} \mathbf{a}_d \\ &\quad + \sum_{j,m} k_j \{ \rho_{j,m}(\tau)\Omega(\ddot{\beta}_{j,m} T) \} \mathbf{a}_j \\ &\quad + \tilde{\mathbf{k}}_n \star \Omega\left(\left(\hat{\alpha} - \frac{\kappa_f(n)}{2}\right)T\right) \end{aligned} \quad (33)$$

where  $\Delta\alpha = \hat{\alpha}(n) - \alpha_1$ ,  $\ddot{\alpha}_n = \hat{\alpha}(n) - \alpha_n$ ,  $\ddot{\beta}_{j,m} = \hat{\alpha}(n) - \beta_{j,m}$ , and  $\Omega(fT) = \text{rect}(f/(2\delta\kappa(n))) \star \text{SINC}(fT)$ . We note from (32) that utilizing the new reference and control signals shown by (31) for the LS-SCORE and cross-SCORE

algorithms, respectively, generates a rectangular spectrum with width  $2\delta\kappa(n)$  instead of a spectral line at each of the cycle frequencies. Moreover, based on the proposed estimation procedure, the difference between  $\hat{\alpha}(n)$  and the cycle frequency  $\alpha_1$  of the SOI decreases as the increment  $\delta\kappa(n)$  decreases. From (33), we further note that the mainlobe width of  $\text{rect}(f/(2\delta\kappa(n))) \star \text{SINC}(fT) \geq 2\delta\kappa(n)$ . Accordingly, the dominance of  $\mathbf{a}_d$  in  $\hat{\mathbf{r}}_{x\tilde{r}}(\hat{\alpha}(n))$  becomes more significant as the number  $n$  of the snapshots increases when  $\delta\kappa(n)$  is small enough. Therefore, the approximation between  $\hat{\mathbf{r}}_{x\tilde{r}}(\hat{\alpha}(n))$  and  $\mathbf{a}_d$  gets better as the frequency increment  $\delta\kappa(n)$  decreases. Hence, the performance deterioration due to the CFE can be reduced as  $n$  increases. As to the effect of  $\Omega(fT)$  on (33), our experience shows that the enhancement due to  $\Omega(fT)$  on the interference and noise is negligible if  $\delta\kappa(n)$  is small enough.

Consider the determination of the robust interval  $\kappa_b(0)$ . Based on the theoretical results shown in Section III, we first use  $N_b$  data snapshots to compute the following matrix:

$$\mathbf{B}(f) = \left\{ \sum_{n=1}^{N_b} \mathbf{x}(n)\mathbf{x}^H(n)e^{-j2\pi fnT_s} \right\} \cdot \left\{ \sum_{n=1}^{N_b} \mathbf{x}(n)\mathbf{x}^H(n)e^{-j2\pi fnT_s} \right\}^H \quad (34)$$

and find the largest eigenvalues  $\lambda_{\max}(f)$  of  $\mathbf{B}(f)$  with respect to  $f$ . Let  $\vartheta(f) = \lambda_{\max}(f)/\text{Tr}[\mathbf{B}(f)]$ , where  $\text{Tr}[\mathbf{B}(f)]$  denotes the trace of  $\mathbf{B}(f)$ . Finally, we choose an appropriate robust interval  $\kappa_b(0)$  from the interval of  $[\hat{\alpha} - \psi, \min\{\hat{\alpha} - \varphi, 1/(N_b T_s)\}]$ , where  $\psi$  and  $\varphi$  represent the values nearest and next nearest to  $\hat{\alpha}$ , respectively, and make  $\vartheta(f)$  generate peaks. As to the value of  $N_b$ , an appropriate  $N_b$  leads to the result of  $|\hat{\alpha} - \psi| \leq 1/(N_b T_s)$ .

## V. MODIFICATION OF THE PROPOSED METHOD

Consider the computation load and the storage required by the proposed method presented in Section IV. We note that the matrix product shown by (27) must be recomputed from the  $n$  data snapshots for each  $\kappa$  since the frequency increment  $\delta\kappa(n)$  varies as  $n$  increases. As a result, both the computation load and the storage requirement increase as  $n$  increases. To alleviate this difficulty, we further modify the proposed method as follows. First, the frequency increment  $\delta\kappa(n)$  in Step 2 is replaced by a fixed frequency increment  $\delta\kappa$ . Next, the largest eigenvalues of (27) are calculated for each  $\kappa$  taking values starting from  $-\kappa_b(0)$  to  $\kappa_b(0)$  with an increment  $\delta\kappa$  in Step 3.

$$\hat{\lambda}_{\max}(n, \kappa) = \frac{\hat{\mathbf{g}}_{\max}^H(n, \kappa) \hat{\mathbf{R}}_{xx}(\hat{\alpha}, n, \tau, \kappa) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, n, \tau, \kappa) \hat{\mathbf{g}}_{\max}(n, \kappa)}{\hat{\mathbf{g}}_{\max}^H(n, \kappa) \hat{\mathbf{g}}_{\max}(n, \kappa)} \quad (30)$$

$$\begin{aligned} \tilde{r}(n) &= \begin{cases} 2\delta\kappa(0) \text{SINC}(2\delta\kappa(0)n) \mathbf{c}^H \mathbf{x}(n - \tau) e^{j2\pi\hat{\alpha}(n)T_s}, & \text{for } n \leq N_o \\ 2\delta\kappa(n) \text{SINC}(2\delta\kappa(n)n) \mathbf{c}^H \mathbf{x}(n - \tau) e^{j2\pi\hat{\alpha}(n)T_s}, & \text{for } n > N_o \end{cases} \\ \tilde{\mathbf{u}}(n) &= \begin{cases} 2\delta\kappa(0) \text{SINC}(2\delta\kappa(0)n) \mathbf{x}(n - \tau) e^{j2\pi\hat{\alpha}(n)T_s} & \text{for } n \leq N_o \\ 2\delta\kappa(n) \text{SINC}(2\delta\kappa(n)n) \mathbf{x}(n - \tau) e^{j2\pi\hat{\alpha}(n)T_s}, & \text{for } n > N_o \end{cases} \end{aligned} \quad (31)$$

Based on these modifications, the matrix product of (27) can be recursively computed as follows:

$$\begin{aligned}
& \hat{\mathbf{R}}_{xx}(\hat{\alpha}, n, \tau, \kappa) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, n, \tau, \kappa) \\
&= \left[ \left( \frac{n-1}{n} \right) \hat{\mathbf{R}}_{xx}(\hat{\alpha}, n-1, \tau, \kappa) + \frac{1}{n} \mathbf{x}(n) \mathbf{x}^H(n) \right. \\
&\quad \left. \times e^{-j2\pi(\hat{\alpha}-\kappa/2)nT_s} \right] \\
&\quad \times \left[ \left( \frac{n-1}{n} \right) \hat{\mathbf{R}}_{xx}(\hat{\alpha}, n-1, \tau, \kappa) \right. \\
&\quad \left. + \frac{1}{n} \mathbf{x}(n) \mathbf{x}^H(n) e^{-j2\pi(\hat{\alpha}-\kappa/2)nT_s} \right]^H \\
&= \left( \frac{n-1}{n} \right)^2 \hat{\mathbf{R}}_{xx}(\hat{\alpha}, n-1, \tau, \kappa) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, n-1, \tau, \kappa) \\
&\quad + \left( \frac{n-1}{n^2} \right) \hat{\mathbf{R}}_{xx}(\hat{\alpha}, n-1, \tau, \kappa) \mathbf{x}(n) \mathbf{x}^H(n) \\
&\quad \times e^{j2\pi(\hat{\alpha}-\kappa/2)nT_s} \\
&\quad + \left( \frac{n-1}{n^2} \right) \mathbf{x}(n) \mathbf{x}^H(n) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, n-1, \tau, \kappa) \\
&\quad \times e^{-j2\pi(\hat{\alpha}-\kappa/2)nT_s} \\
&\quad + \left( \frac{1}{n^2} \right) (\mathbf{x}(n) \mathbf{x}^H(n) \mathbf{x}(n) \mathbf{x}^H(n)). \quad (35)
\end{aligned}$$

Using (35), we can compute the corresponding largest eigenvalue recursively as follows. First, let the vector  $\mathbf{y}$  be given by

$$\mathbf{y} = \hat{\mathbf{R}}_{xx}(\hat{\alpha}, n, \tau, \kappa) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, n, \tau, \kappa) \hat{\mathbf{g}}_{\max}(n-1, \kappa). \quad (36)$$

Then, set the corresponding largest eigenvalue equal to

$$\hat{\lambda}_{\max}(n, \kappa) = \hat{\mathbf{g}}_{\max}^H(n-1, \kappa) \mathbf{y} \quad (37)$$

with the initial eigenvector  $\hat{\mathbf{g}}_{\max}(0, \kappa) = [1, 1, \dots, 1]^T / \sqrt{N}$ . Finally, update the corresponding eigenvector according to

$$\hat{\mathbf{g}}_{\max}(n, \kappa) = \frac{\mathbf{y}}{|\mathbf{y}|}. \quad (38)$$

Based on the above modifications, we present a modified procedure for estimating the frequency parameter  $\kappa$  as follows.

*Procedure 2:*

- Step 1) Let the robust interval be  $\kappa_b(0)$  and the fixed frequency increment be  $\delta\kappa$  for performing the estimation. Use the first  $N_o$  data snapshots to compute the initial matrix of  $\hat{\mathbf{R}}_{xx}(\hat{\alpha}, n, \tau, \kappa) \hat{\mathbf{R}}_{xx}^H(\hat{\alpha}, n, \tau, \kappa)$ , where  $N_o \geq L/(2\kappa_b(0)T_s)$  and  $L$  is a preset positive integer.
- Step 2) For  $n \leq N_0$ , we compute the largest eigenvalue of (27) for each  $\kappa$  taking the values starting from  $-\kappa_b(0)$  to  $\kappa_b(0)$  with an increment  $P\delta\kappa$  according to (36)–(38), where  $P$  is a preset positive integer. For  $n > N_0$ , go to Step 6.
- Step 3) Find the highest value of the approximates of the largest eigenvalues obtained from Step 2 and let the corresponding frequency parameter  $\kappa$  be designated as  $\kappa_1(n)$ .
- Step 4) Compute the largest eigenvalue of (27) for each  $\kappa$  taking the values starting from  $\kappa_1(n) - P\delta\kappa$  to

$\kappa_1(n) + P\delta\kappa$  with an increment of  $\delta\kappa$  according to (36)–(38).

- Step 5) Find the highest value of the approximates of the largest eigenvalues obtained from Step 4 and let the corresponding frequency parameter  $\kappa$  be designated as  $\kappa_2(n)$ .
- Step 6) For  $N_0 < n \leq N_1$ , where  $N_1 = 1/(2P\delta\kappa T_s)$ , if  $|\kappa_2(n-1) - \kappa_2(n-2)|/2 \leq \delta\kappa$  which shows that the current estimate is very close to the true value, then set  $\kappa_2(n) = \kappa_2(n-1)$  and go to Step 9. Otherwise, we repeat the similar process for  $n \leq N_0$  as described from Steps 2–5 to find the appropriate estimate  $\kappa_2(n)$ .
- Step 7) For  $n > N_1$ , if  $|\kappa_2(n-1) - \kappa_2(n-2)|/2 \leq \delta\kappa$  which shows that the obtained estimate is sufficiently close to the true value, then set  $\kappa_2(n) = \kappa_2(n-1)$  and go to Step 9. Otherwise, we compute the largest eigenvalue of (27) for each  $\kappa$  taking the values starting from  $-\kappa_b(0)$  to  $\kappa_b(0)$  with an increment  $\delta\kappa$  according to (36)–(38).
- Step 8) Find the highest value of the approximates of the largest eigenvalues obtained from Step 7 and let the corresponding frequency parameter  $\kappa$  be designated as  $\kappa_2(n)$ .
- Step 9) Compute the estimate of the cycle frequency  $\alpha_1$  as follows:  $\hat{\alpha}(n) = \hat{\alpha} - \kappa_2(n)/2$ .

It can be seen that two loops are employed for  $n \leq N_0$  to save the required computing cost and the storage required for  $\hat{\mathbf{R}}_{xx}(\hat{\alpha}, n, \tau, \kappa)$ . One including Steps 2 and 3 finds a frequency parameter  $\kappa_1(n)$ . The other one including Steps 4 and 5 improves the estimation accuracy by refining the estimate  $\kappa_1(n)$ . An appropriate value for  $P$  used in Steps 2 and 4 is determined as follows. Consider the total computing cost required by the modified estimation procedure. In Step 2, the number of the computations for finding the largest eigenvalues is about  $2\kappa_b(0)/(P\delta\kappa) + 1$ , while the number required by Step 4 is about  $2P - 2$ . Hence, the total number is given by

$$\begin{aligned}
Cost &= \frac{2\kappa_b(0) + 1}{P\delta\kappa} + 2P - 1 \\
&= 2 \left( \frac{\kappa_b(0)}{P\delta\kappa} + P \right) - 1 \\
&\geq 2 \left( 2 \sqrt{\frac{\kappa_b(0)}{\delta\kappa}} \right) - 1. \quad (39)
\end{aligned}$$

Expression (39) reveals that the optimal value for  $P$  which minimizes the required computing cost is given by

$$P = \sqrt{\frac{\kappa_b(0)}{\delta\kappa}}. \quad (40)$$

After obtaining  $\kappa_2(n)$  as the estimate of the frequency parameter  $\kappa$ , we again construct the new reference and control signals as shown by (31) for performing the LS-Score and cross-Score algorithms. Moreover, the sample cross-correlation vector and matrix  $\hat{\mathbf{r}}_{x\tilde{x}}(\hat{\alpha}(n))$  and  $\hat{\mathbf{R}}_{x\tilde{x}}(\hat{\alpha}(n))$  required by the LS-Score and cross-Score algorithms can

also be computed recursively as follows:

$$\begin{aligned}\hat{\mathbf{r}}_{x\tilde{\mathbf{r}}}(\hat{\alpha}(n)) &= \frac{n-1}{n} \hat{\mathbf{r}}_{x\tilde{\mathbf{r}}}(\hat{\alpha}(n-1)) + \frac{1}{n} \mathbf{x}(n) \tilde{\mathbf{r}}^H(n) \\ &= \frac{n-1}{n} \hat{\mathbf{r}}_{x\tilde{\mathbf{r}}}(\hat{\alpha}(n-1)) \\ &\quad + \frac{2\delta\kappa \text{SINC}(2\delta\kappa n T_s) e^{-j2\pi(\hat{\alpha}(n))nT_s}}{n} \\ &\quad \cdot \mathbf{x}(n) \mathbf{x}^H(n) \mathbf{c}\end{aligned}\quad (41)$$

and

$$\begin{aligned}\hat{\mathbf{R}}_{x\tilde{\mathbf{u}}}(\hat{\alpha}(n)) &= \frac{n-1}{n} \hat{\mathbf{R}}_{x\tilde{\mathbf{u}}}(\hat{\alpha}(n-1)) + \frac{1}{n} \mathbf{x}(n) \tilde{\mathbf{u}}^H(n) \\ &= \frac{n-1}{n} \hat{\mathbf{R}}_{x\tilde{\mathbf{u}}}(\hat{\alpha}(n-1)) \\ &\quad + \frac{2\delta\kappa \text{SINC}(2\delta\kappa n T_s) e^{-j2\pi(\hat{\alpha}(n))nT_s}}{n} \\ &\quad \cdot \mathbf{x}(n) \mathbf{x}^H(n).\end{aligned}\quad (42)$$

## VI. COMPUTER SIMULATION EXAMPLES

In this section, several simulation examples performed on a Pentium Pro-200 PC using MATLAB programming language are presented for confirming the theoretical results and showing the effectiveness of the proposed method. For all simulation examples, a uniform linear array (ULA) with number of elements  $M = 21$  and interelement spacing  $= \lambda/2$ , where  $\lambda$  is the wavelength of the desired signal which is impinging on the array from  $5^\circ$  off broadside with cycle frequency  $\alpha_1 = 2$  and  $\text{SNR} = 0$  dB. Moreover, the desired signal is a binary phase shift keying (BPSK) signal with rectangular pulse shape. The noise received by the array is spatially white. The sampling interval  $T_s$  for obtaining data snapshots is set to 0.2. The initial matrices  $\mathbf{R}_{xx}(0)$  and  $\mathbf{R}_{uu}(0)$  are set to  $10^{-6}\mathbf{I}$ , where  $\mathbf{I}$  denotes the identity matrix with size  $21 \times 21$ . The vector  $\mathbf{c}$  in the LS-SCORE algorithm is fixed to  $\mathbf{c} = [1, 1, \dots, 1]^T$ . The time delay  $\tau$  is set to zero for simplicity.

For finding the appropriate estimate for  $\kappa$  after receiving  $n$  data snapshots by using the proposed estimation procedure, we set  $N_0$  and  $L$  equal to 250 and 5, respectively. The value of  $\delta\kappa$  is set to 0.002 when using the proposed method based on Procedure 2 and, hence, the optimal  $P$  is given by 5 according to (40). All the simulation results are obtained by averaging 50 independent runs with independent noise samples for each run.

*Example 1:* Here, two interferers with interference-to-noise ratio (INR) = 3 dB and 0 dB are impinging on the array from  $30^\circ$  and  $40^\circ$ , respectively. Moreover, both of these interferers are BPSK signals with rectangular pulse shape. The carrier frequencies of these two interferers are 1.65 and 2.45, respectively.  $N_b$  is set to 50 and thus  $\kappa_b(0) = 0.05$ . Fig. 1 plots the simulation results in terms of the output SINR versus the number of snapshots for three different values of  $\hat{\alpha}$ , namely  $\hat{\alpha} = 2.0, 2.01$ , and  $1.9873$ . The simulation results of using the original LS-SCORE and cross-SCORE algorithms presented in [6] are also provided for comparison. From Fig. 1(a), we observe that the proposed method based on Procedure 2 and the original SCORE algorithms provide almost the same array performance when  $\hat{\alpha} = \alpha_1$ . However, the proposed method

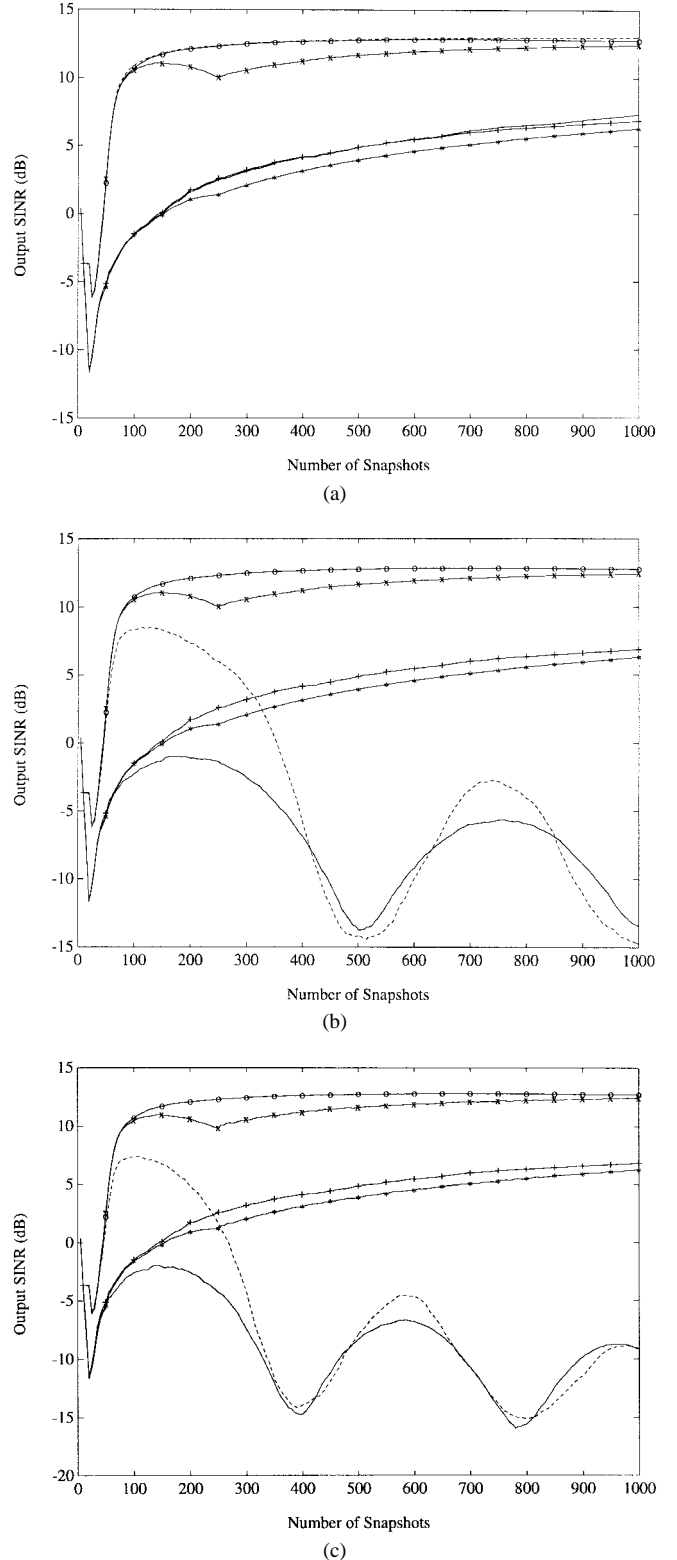


Fig. 1. The output SINR versus the number of snapshots for Example 1. (a)  $\hat{\alpha} = 2.0$ . (b)  $\hat{\alpha} = 2.01$ . (c)  $\hat{\alpha} = 1.9873$ .  $-*$ : the proposed method based on LS-SCORE algorithm using Procedure 1,  $-x-$ : the proposed method based on cross-SCORE algorithm using Procedure 1,  $-+-$ : the proposed method based on LS-SCORE algorithm using Procedure 2,  $-o-$ : the proposed method based on cross-SCORE algorithm using Procedure 2,  $---$ : the original LS-SCORE algorithm,  $---$ : the original cross-SCORE algorithm.

can effectively cure the performance degradation due the CFE. Moreover, utilizing Procedure 2 provides better convergence speed than utilizing Procedure 1 in this case.

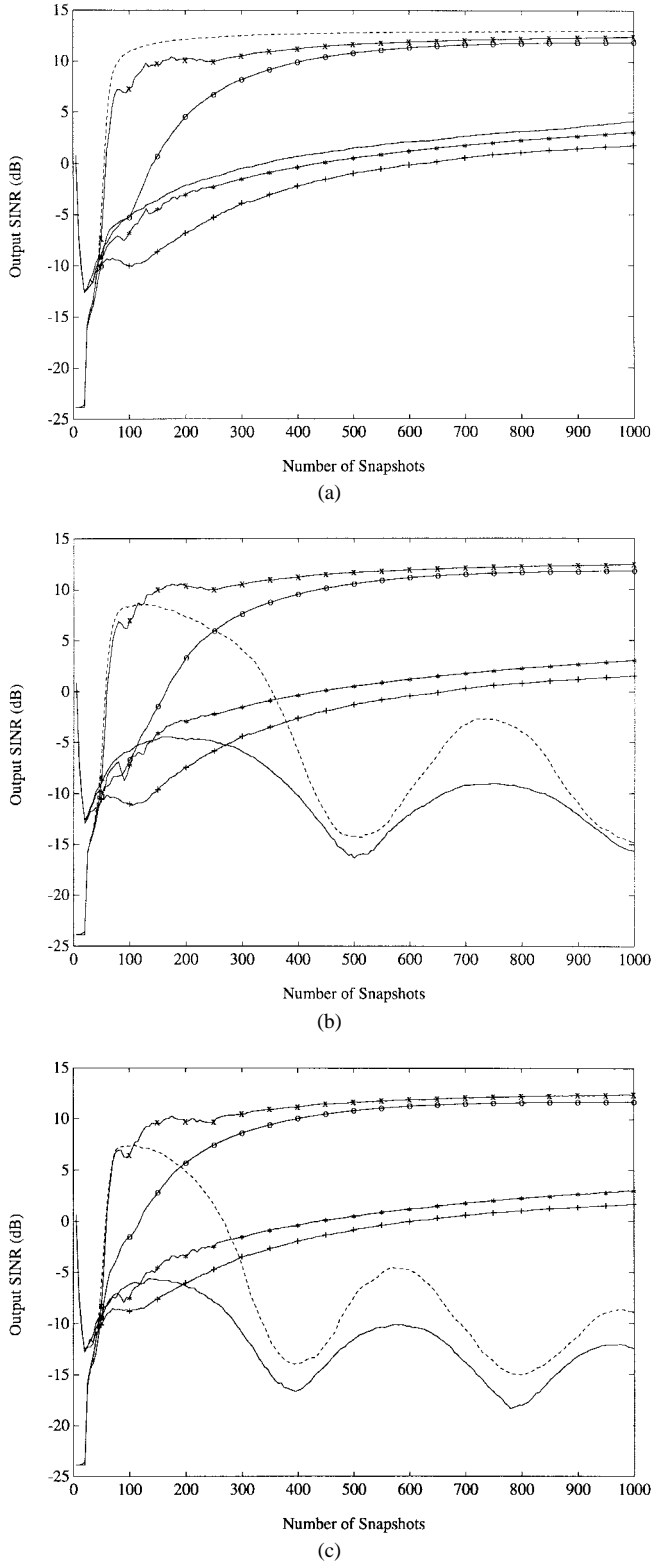


Fig. 2. The output SINR versus the number of snapshots for Example 2. (a)  $\hat{\alpha} = 2.0$ . (b)  $\hat{\alpha} = 2.01$ . (c)  $\hat{\alpha} = 1.9873$ .  $-\ast-$ : the proposed method based on LS-Score algorithm using Procedure 1,  $-\times-$ : the proposed method based on cross-Score algorithm using Procedure 1,  $-\ast-$ : the proposed method based on LS-Score algorithm using Procedure 2,  $-\circ-$ : the proposed method based on cross-Score algorithm using Procedure 2,  $---$ : the original LS-Score algorithm,  $---$ : the original cross-Score algorithm.

*Example 2:* In this case, the interferers impinging on the array are the same as those used in Example 1 except that they have INR equal to 13 and 15 dB, respectively.  $N_b$  is

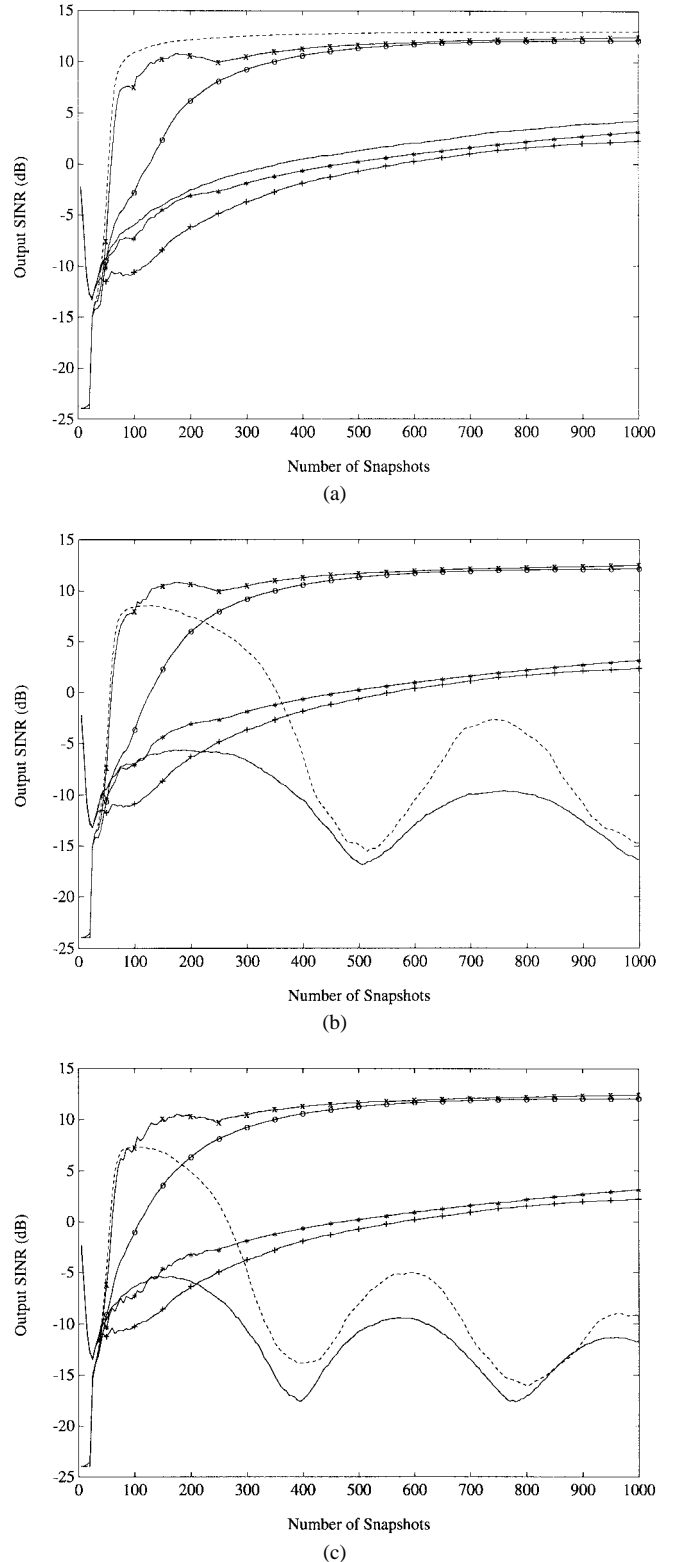


Fig. 3. The output SINR versus the number of snapshots for Example 3. (a)  $\hat{\alpha} = 2.0$ . (b)  $\hat{\alpha} = 2.01$ . (c)  $\hat{\alpha} = 1.9873$ .  $-\ast-$ : the proposed method based on LS-Score algorithm using Procedure 1,  $-\times-$ : the proposed method based on cross-Score algorithm using Procedure 1,  $-\ast-$ : the proposed method based on LS-Score algorithm using Procedure 2,  $-\circ-$ : the proposed method based on cross-Score algorithm using Procedure 2,  $---$ : the original LS-Score algorithm,  $---$ : the original cross-Score algorithm.

set to 200 and  $\kappa_b(0) = 0.05$  in this case. Fig. 2 shows the output SINR versus the number of snapshots for  $\hat{\alpha} = 2.0, 2.01$ , and 1.9873. From Fig. 2, we note that although the original



SCORE algorithms demonstrate better convergence rate for  $\alpha_1 = \hat{\alpha}$ , the proposed method provides satisfactory robust capability against the CFE. Moreover, in the presence of CFE, Procedure 1 possesses faster convergence rate than Procedure 2 in this high INR case.

*Example 3:* This example is performed to show the effect of reducing the angle separation between the desired signal and interference. The interferers used are the same as those used in Example 2 except that they are impinging on the array from  $10.5^\circ$  and  $40^\circ$ , respectively. That is, the interferer with direction angle  $10.5^\circ$  is impinging on the array at the edge of the array mainlobe. Again,  $N_b$  is set to 200 and  $\kappa_b(0) = 0.05$  for this example. The simulation results for  $\hat{\alpha} = 2.0, 2.01$ , and  $1.9873$  are depicted in Fig. 3. Again, we observe that the array performances of utilizing the proposed method and the original SCORE algorithms are very similar to those shown in Example 2 as long as all of the interferers are impinging on the array outside the array mainlobe.

Finally, we see that the theoretical results presented in Section III for performance analysis in the presence of CFE are also confirmed by Figs. 1–3.

## VII. CONCLUSION

This paper has presented an efficient method for blind adaptive beamforming using the LS-SCORE and cross-SCORE algorithms in the presence of cycle frequency error (CFE). The performance degradation of these two SCORE algorithms due to CFE have been analyzed. Based on the theoretical results, an efficient method in conjunction with the SCORE algorithms has been developed to cope with the performance deterioration of the original SCORE algorithms due to CFE. According to the proposed method, an appropriate estimate of the cycle frequency of the desired signal is first found. Using the estimate, we then construct new reference and control signals required by the LS-SCORE and cross-SCORE algorithms, respectively. Modifications of the proposed method for reducing the required computational complexity and storage requirement have also been presented. The validity of the theoretical works for performance analysis has been confirmed by simulation results. The effectiveness of the proposed method in dealing with adaptive beamforming for cyclostationary signals in the presence of CFE has also been demonstrated by simulation results.

## REFERENCES

- [1] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Array*. New York: Wiley, 1980.
- [2] W. A. Gardner, "Exploitation of spectral redundancy in cyclostationary signals," *IEEE Signal Processing Mag.*, vol. 8, pp. 14–36, Apr. 1991.
- [3] W. A. Gardner, "Spectral correlation of modulated signals: Part I—Analog modulation," *IEEE Trans. Commun.*, vol. COM-35, pp. 584–594, June 1987.

- [4] W. A. Gardner, W. A. Brown, III, and C.-K. Chen, "Spectral correlation of modulated signals: Part II—Digital modulation," *IEEE Trans. Commun.*, vol. COM-35, pp. 595–601, June 1987.
- [5] W. A. Gardner, *Cyclostationarity in Communications and Signal Processing*. New York: IEEE Press, 1994.
- [6] B. G. Agee, S. V. Schell, and W. A. Gardner, "Spectral self-coherence restoral: A new approach to blind adaptive signal extraction using antenna arrays," *Proc. IEEE*, vol. 78, pp. 753–767, Apr. 1990.
- [7] S. V. Schell, "Performance analysis of the Cyclic MUSIC method of direction estimation for cyclostationary signals," *IEEE Trans. Signal Processing*, vol. 42, pp. 3043–3050, Nov. 1994.
- [8] S. V. Schell, "Asymptotic moments of estimated cyclic correlation matrices," *IEEE Trans. Signal Processing*, vol. 43, pp. 173–180, Jan. 1995.
- [9] S. V. Schell and W. A. Gardner, "Progress on signal-selective direction finding," in *Proc. 5th Acoust. Speech, Signal Processing Workshop Spectrum Estim. Model.*, Rochester, NY, Oct. 1990, pp. 144–148.
- [10] T. E. Biedka and B. G. Agee, "Subinterval Cyclic MUSIC-robust DF with error in cycle frequency knowledge," in *Proc. 25th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, pp. 262–266, Nov. 1991.
- [11] S.-J. Yu and J.-H. Lee, "Adaptive array beamforming for cyclostationary signals," *IEEE Trans. Antennas Propagat.*, vol. 44, pp. 943–953, July 1996.
- [12] D. W. Lewis, *Matrix Theory*. Singapore: World Scientific, 1991.



**Ju-Hong Lee** was born in I-Lan, Taiwan, on December 7, 1952. He received the B.S. degree from the National Cheng-Kung University, Tainan, Taiwan, in 1975, the M.S. degree from the National Taiwan University, Taipei, Taiwan, in 1977, and the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, in 1984, all in electrical engineering.

From September 1980 to July 1984, he was a Research Assistant and was involved in research on multidimensional recursive digital filtering in the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute. From August 1984 to July 1986 he was a Visiting Associate Professor and later in August 1986 became an Associate Professor in the Department of Electrical Engineering, National Taiwan University. Since August 1989 he has been a Professor at the same university. He was appointed Visiting Professor in the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore, during a sabbatical leave in 1996. His current research interests include multidimensional digital signal processing, image processing, detection and estimation theory, analysis and processing of joint vibration signals for the diagnosis of cartilage pathology, and adaptive signal processing and its applications in communications.

Dr. Lee received Outstanding Research Awards from the National Science Council in the academic years of 1988, 1989, and 1991–1994.



**Yung-Ting Lee** was born in Tainan, Taiwan, on March 31, 1971. He received the B.S. degree from the National Cheng-Kung University, Tainan, Taiwan, in 1993, and the M.S. degree from the National Taiwan University, Taipei, Taiwan, in 1995, both in electrical engineering. He is currently working toward the Ph.D. degree in electrical engineering at the National Taiwan University.

His current research interests include adaptive signal processing, array signal processing, and the theory and application of cyclostationarity.