Shrinkage Linear and Widely Linear Complex-Valued Least Mean Squares Algorithms for Adaptive Beamforming

Yun-Mei Shi, Lei Huang, *Senior Member, IEEE*, Cheng Qian, and H. C. So, *Senior Member, IEEE*

***Abstract*—In this paper, shrinkage linear complex-valued least mean squares (SL-CLMS) and shrinkage widely linear complex-valued least mean squares (SWL-CLMS) algorithms are devised for adaptive beamforming. By exploiting the relationship between the noise-free *a posteriori* and *a priori* error signals, the SL-CLMS method is able to provide a variable step size to update the weight vector for the adaptive beamformer, significantly enhancing the convergence speed and decreasing the steady-state misadjustment. On the other hand, besides adopting a variable step size determined by minimizing the square of the augmented noise-free *a posteriori* errors, the SWL-CLMS approach exploits the noncircular properties of the signal of interest, which considerably improves the steady-state performance. Simulation results are presented to illustrate their superiority over the CLMS, complex-valued normalized LMS, variable step size, recursive least squares (RLS) algorithms and their corresponding widely linear-based schemes. Additionally, our proposed algorithms are more computationally efficient than the RLS solutions though they may have a slightly slower convergence rate.**

***Index Terms*—Complex-valued least mean squares (CLMS), convergence speed, shrinkage, steady-state, variable step size, widely linear.**

I. Introduction

In adaptive filtering applications for modeling, equalization, control, echo cancellation and beamforming, the complex-valued least mean squares (CLMS) algorithm is a well-known adaptive estimation and prediction technique which is capable of converging to the optimal Wiener solution [1]. The application of the CLMS algorithm to the beamforming and its analysis have been extensively studied [1]–[4]. The weight vector of the adaptive beamformer can be computed based on different kinds of design criteria. The most promising criteria include the minimum mean-squared error (MMSE) [3], minimum variance [5] and constant modulus [6]. In this paper, we focus on the scenario where the MMSE criterion is applied to the adaptive beamforming system because it only requires the training sequence of the desired signal.

The incoming signals are usually assumed to be circular in the conventional adaptive array and a linear and time invariant complex filter $\boldsymbol{w}$ is determined, such that its output $y(k) \triangleq \boldsymbol{w}^H \boldsymbol{x}(k)$ can optimize a second-order criterion, e.g., the MMSE [3] or minimum variance distortionless response (MVDR) [5], under certain constraints. Here, the superscript $(\cdot)^H$ denotes the conjugate transpose operator and $\boldsymbol{x}(k)$ is the observation vector at the output of the array. But in practice, noncircular (NC) signals, such as BPSK, offset-QPSK, PAM and ASK-modulated signals, have been widely used in many modern communication systems. Although the conventional adaptive beamformer has been shown to be optimal for circular sources [7], [8], it becomes suboptimal for NC signals [9]–[13] because the properties of the NC signals have not yet been completely exploited. Optimal widely-linear filters are of great value in many practical situations if the signals share the second-order NC properties. In order to make full use of the second-order NC properties, a widely-linear complex-valued LMS (WL-CLMS) algorithm has been proposed in [14] which uses the augmented vector $\tilde{\boldsymbol{x}}(k) = [\boldsymbol{x}^T(k), \boldsymbol{x}^H(k)]^T$ to replace $\boldsymbol{x}(k)$. Here, $(\cdot)^T$ is the transpose operator. It is illustrated in [14] that the WL-CLMS algorithm can yield much less mean square error (MSE) than the CLMS algorithm. Moreover, the performance analysis of the CLMS and WL-CLMS algorithms for smart antennas has been conducted in [15]. It is shown that the CLMS algorithm is suboptimal for the second-order NC signals whereas the WL-CLMS scheme can potentially achieve a lower steady-state MSE [16], [17] and provide a more accurate prediction of the mean and mean-squares behaviors [18].

The CLMS and WL-CLMS algorithms in their basic form directly give a solution to update the weight vector at each iteration. The constant step size (CSS) μ controls the convergence rate of the weight vector and also determines the excess MSE (EMSE) [1]. The CSS which is suitable for certain environments may result in poor performance when the incoming signals are nonstationary. This motivates us to choose the step size according to the incoming signals. One possibility of accomplishing this is the complex-valued normalized LMS (CNLMS) [19], [20] or its widely-linear variant, namely, the widely-linear complex-valued normalized LMS (WL-CNLMS) [21]. Although the CNLMS and WL-CNLMS algorithms potentially have faster convergence speed than the CLMS-type schemes, they still rely on the CSS which

Manuscript received April 23, 2014; revised August 14, 2014 and October 13, 2014; accepted October 23, 2014. Date of publication November 20, 2014; date of current version December 02, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Min Dong. This work was supported in part by the National Natural Science Foundation of Chinaunder Grants 61222106 and 61171187 and by the Shenzhen Kongqie talent programunder Grant KQC201109020061A.

Y. M. Shi, L. Huang, and C. Qian are with the Department of Electronic and Information Engineering, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China (e-mail: dr.lei.huang@ieee.org).

H. C. So is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon 852, Hong Kong.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2014.2367452

affects the convergence rate and degrades the performance of the beamformer. Apart from the CNLMS-type algorithm, another well-known solution is the complex-valued recursive-least-square (RLS) or its widely-linear variant, namely, the complex-valued widely-linear RLS (WL-RLS) algorithm which updates the weight vector with fast convergence speed and small steady-state misadjustment. However, the RLS and WL-RLS algorithms are much more computationally demanding than the CLMS-type algorithms [22], [23]. In order to reduce the complexity while keeping fast convergence speed, the variable step size (VSS) and widely-linear VSS (WL-VSS) algorithms have been introduced in [24] and [25], respectively. The VSS-type method can provide faster convergence speed and require less computational cost per iteration than the RLS-type algorithms. Several variants of the VSS algorithms have been proposed in [26]–[29]. However, they cannot enjoy both fast tracking as well as small misadjustment with simple implementation.

In this paper, we devise shrinkage linear complex-valued LMS (SL-CLMS) and shrinkage widely-linear complex-valued LMS (SWL-CLMS) algorithms. The SL-CLMS algorithm provides an approximately optimal VSS by minimizing the instantaneous square of the noise-free *a posteriori* error. In contrast with the SL-CLMS algorithm, the SWL-CLMS method not only offers a VSS but also exploits the NC properties of the signal-of-interest (SOI), resulting in a more accurate weight vector. Simulations under different parameter settings demonstrate that this VSS can improve the convergence speed of the weight vector and yield much smaller steady-state errors. To calculate the VSS, we need to estimate the noise-free *a priori* error which can be obtained by the methods described in [30]–[32]. Furthermore, it is illustrated that the proposed solutions provide larger output signal-to-interference-plus-noise ratio (SINR) and smaller MSE than the state-of-the-art methods. Note that the variance of the additive noise of the array sensors is assumed *a priori* known for our proposed methods [32].

The rest of the paper is organized as follows. In Section II, we present the data model and review the CLMS and WL-CLMS algorithms. In Section III, the SL-CLMS and SWL-CLMS algorithms are derived. In Section IV, numerical examples are provided for performance comparison. Finally, Section V draws the conclusion.

The following notations are used throughout the paper. Matrices and vectors are represented by bold upper-case and bold lower-case characters, respectively. Superscripts $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^*$ and $(\cdot)^{-1}$ stand for the transpose, conjugate transpose, conjugate and matrix inverse, respectively. The $\boldsymbol{I}_M$ and $\boldsymbol{0}_{M\times K}$ denote the $M \times M$ identity matrix and $M \times K$ zero matrix, respectively. The $E[\cdot]$ stands for the mathematical expectation. The $\|\cdot\|$, $|\cdot|$, $\|\cdot\|_1$ and $\|\cdot\|_0$ denote the Euclidean norm, absolute value, ℓ_1-norm and ℓ_0-norm, respectively.

II. Preliminaries

Let us consider an array of M sensors receiving a far-field narrowband signal $s_0(k)$ with the direction-of-arrival (DOA) θ_d. The signal is assumed to be zero-mean and potentially second-order NC. The array measurement vector $\boldsymbol{x}(k)$ can be written as

$$\boldsymbol{x}(k) = \boldsymbol{a}(\theta_d)s_0(k) + \boldsymbol{n}(k) \tag{1}$$

where $\boldsymbol{a}(\theta_d) = [1, e^{j\omega\tau_1}, \cdots, e^{j\omega\tau_{M-1}}]^T$ is the steering vector of the SOI with ω denoting the center frequency of the narrowband signal and $\tau_i, i = 1\cdots M-1$, which is related to θ_d, being the time delay of the i-th sensor relative to the first sensor, $\boldsymbol{n}(k) = [n_1(k), \ldots, n_M(k)]^T$ is the additive noise vector consisting of the interferences and background noise, which can be expressed as

$$\boldsymbol{n}(k) = \sum_{i=1}^{P} \boldsymbol{a}(\theta_i)s_i(k) + \boldsymbol{\eta}(k). \tag{2}$$

Here, we assume that there are P statistically uncorrelated NC interferences whose complex envelopes are $s_i(k), i = 1, 2, \ldots, P$, and their steering vectors are $\boldsymbol{a}(\theta_i), i = 1, 2, \ldots, P$, $\boldsymbol{\eta}(k)$ denotes the background noise which is uncorrelated with the SOI and interferences.

Given the weight vector $\boldsymbol{w} = [w_1, \ldots, w_M]^T$, the output of the beamformer is

$$y(k) = \boldsymbol{w}^H\boldsymbol{x}(k). \tag{3}$$

In the conventional adaptive CLMS algorithm, the optimal weight vector $\boldsymbol{w}_{\rm opt}$ is obtained by minimizing the MSE between the output of the beamformer and desired signal $s_d(k)$, i.e.,

$$\begin{aligned}\boldsymbol{w}_{\rm opt} &= \arg\min_{\boldsymbol{w}} E[|e(k)|^2] \\ &= \arg\min_{\boldsymbol{w}} E[|s_d(k) - y(k)|^2] \\ &= \arg\min_{\boldsymbol{w}} E[|s_0(k) - y(k)|^2].\end{aligned} \tag{4}$$

Here, we set $s_d(k) = s_0(k)$. After some straightforward calculations, the optimal weight vector is

$$\boldsymbol{w}_{\rm opt} = \boldsymbol{R}_x^{-1}\boldsymbol{p}_x \tag{5}$$

where $\boldsymbol{R}_x \triangleq E[\boldsymbol{x}(k)\boldsymbol{x}^H(k)]$ and $\boldsymbol{p}_x \triangleq E[\boldsymbol{x}(k)s_0^*(k)]$. Similar to [1], by minimizing the power of the instantaneous error $J[\boldsymbol{w}(k)]$:

$$J[\boldsymbol{w}(k)] \triangleq |s_0(k) - y(k)|^2 = |s_0(k) - \boldsymbol{w}^H(k)\boldsymbol{x}(k)|^2, \tag{6}$$

we get the update equation for the weight vector

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu e^*(k)\boldsymbol{x}(k) \tag{7}$$

with μ denoting the step size.

When the signal $s_0(k)$ is NC, the covariance matrix $\boldsymbol{R}_x$ can no longer completely describe the second-order properties of the random vector $\boldsymbol{x}(k)$. Suppose that the SOI has the strictly NC property which may correspond to the BPSK, offset QPSK and PAM signals. Consequently, the SOI symbol snapshot vector can be decomposed as

$$s_0(k) = \tilde{s}(k)e^{j\varphi_s} \tag{8}$$

where $e^{j\varphi_s}$ contains the initial complex phase shift of the SOI and $\tilde{s}(k)$ is the real-valued symbol snapshots. According to (8), the received signal and its conjugated version are correlated. That is to say, the conjugated signal also contains useful information of the SOI, i.e., the pseudo-covariance matrix $\boldsymbol{C}_x = E[\boldsymbol{x}(k)\boldsymbol{x}^T(k)] \neq \boldsymbol{0}_{M\times M}$. In order to take advantage of the benefits associated with the NC properties, we apply the widely-

linear processing and define the augmented measurement vector $\tilde{\boldsymbol{x}}(k)$ as

$$\begin{aligned}\tilde{\boldsymbol{x}}(k) &\triangleq \begin{bmatrix}\boldsymbol{x}(k)\\ \boldsymbol{x}^*(k)\end{bmatrix}\\ &= \begin{bmatrix}\boldsymbol{a}(\theta_d)\\ \boldsymbol{a}^*(\theta_d)e^{-2j\varphi_s}\end{bmatrix}s_0(k)+\begin{bmatrix}\boldsymbol{n}(k)\\ \boldsymbol{n}^*(k)\end{bmatrix}\\ &= \tilde{\boldsymbol{a}}(\theta_d)s_0(k)+\tilde{\boldsymbol{n}}(k)\end{aligned} \tag{9}$$

where $\tilde{\boldsymbol{a}}(\theta_d)$ and $\tilde{\boldsymbol{n}}(k)$ are the augmented steering vector and augmented noise vector, respectively. Moreover, in analogy to the CLMS, the cost function for the WL-CLMS is written as

$$\begin{aligned}J(\tilde{\boldsymbol{w}}(k)) &\triangleq |\tilde{e}(k)|^2\\ &= |s_0(k)-\tilde{y}(k)|^2\\ &= |s_0(k)-\tilde{\boldsymbol{w}}^H(k)\tilde{\boldsymbol{x}}(k)|^2\end{aligned} \tag{10}$$

where $\tilde{e}(k) = s_0(k)-\tilde{y}(k)$ is the augmented instantaneous error and $\tilde{y}(k)=\tilde{\boldsymbol{w}}^H(k)\tilde{\boldsymbol{x}}(k)$ is the output of the augmented beamformer. Here, $\tilde{\boldsymbol{w}}(k) = [\boldsymbol{w}_1^T(k), \boldsymbol{w}_2^T(k)]^T$ is the widely-linear weight vector whose update equation is given by

$$\boldsymbol{w}_1(k+1) = \boldsymbol{w}_1(k)+\mu\tilde{e}^*(k)\boldsymbol{x}(k), \tag{11}$$

$$\boldsymbol{w}_2(k+1) = \boldsymbol{w}_2(k)+\mu\tilde{e}^*(k)\boldsymbol{x}^*(k). \tag{12}$$

Thus, minimizing $E[|\tilde{e}(k)|^2]$ leads to the optimal widely-linear weight vector, given as

$$\tilde{\boldsymbol{w}}_{\rm opt} = \tilde{\boldsymbol{R}}_x^{-1}\tilde{\boldsymbol{p}}_x \tag{13}$$

where

$$\tilde{\boldsymbol{R}}_x = \begin{bmatrix}\boldsymbol{R}_x & \boldsymbol{C}_x\\ \boldsymbol{C}_x^* & \boldsymbol{R}_x^*\end{bmatrix} \tag{14}$$

and $\tilde{\boldsymbol{p}}_x \triangleq E[\tilde{\boldsymbol{x}}(k)s_0^*(k)]$.

III. Proposed Algorithms

In the standard CLMS and WL-CLMS algorithms, μ is a constant. In order to obtain such a weight vector $\boldsymbol{w}(k)$ that is as close as possible to the Wiener solution, the VSS and its variants have been developed in [24]–[29]. These VSS-type and WL-VSS-type algorithms, however, cannot guarantee that the VSS is approximately optimal for each weight vector update as they have not well exploited the characteristics of the additive noise and input signals. Usually, the characteristics of the additive noise and input signals considerably affect the behavior of the VSS. In this section, we derive the SL-CLMS and SWL-CLMS algorithms which can provide an approximately optimal step size for each weight vector update, leading to a superior performance of the adaptive beamformer.

A. SL-CLMS Algorithm

Replacing the CSS μ in (7) with a VSS μ_k, the weight vector is updated as

$$\begin{aligned}\boldsymbol{w}(k+1) &= \boldsymbol{w}(k)+\mu_k[s_0(k)-y]^*\boldsymbol{x}(k)\\ &= [\boldsymbol{I}_M-\mu_k\boldsymbol{x}(k)\boldsymbol{x}^H(k)]\boldsymbol{w}(k)+\mu_k s_0^*(k)\boldsymbol{x}(k).\end{aligned} \tag{15}$$

Assume that the paired sequence $\{\boldsymbol{x}(k), s_0(k)\}$ is wide-sense stationary. Consequently, the optimal weight vector $\boldsymbol{w}_{\rm opt}(k)$ is invariant in time, i.e., $\boldsymbol{w}_{\rm opt}(k) = \boldsymbol{w}_{\rm opt}$ [33], [34]. Setting the weight error vector as

$$\boldsymbol{v}(k) = \boldsymbol{w}(k)-\boldsymbol{w}_{\rm opt} \tag{16}$$

we can express the update equation for $\boldsymbol{w}(k)$ in terms of $\boldsymbol{v}(k)$:

$$\boldsymbol{v}(k+1) = [\boldsymbol{I}_M-\mu_k\boldsymbol{x}(k)\boldsymbol{x}^H(k)]\boldsymbol{v}(k)+\mu_k\epsilon_{\rm opt}^*(k)\boldsymbol{x}(k) \tag{17}$$

where $\epsilon_{\rm opt}(k) = s_0(k)-\boldsymbol{w}_{\rm opt}^H\boldsymbol{x}(k)$. The error between the output of the beamformer and SOI $s_0(k)$ at the time instant k is

$$\begin{aligned}e(k) &= s_0(k)-\boldsymbol{w}^H(k)\boldsymbol{x}(k)\\ &= \epsilon_{\rm opt}(k)+\boldsymbol{w}_{\rm opt}^H\boldsymbol{x}(k)-\boldsymbol{w}^H(k)\boldsymbol{x}(k)\\ &= \epsilon_{\rm opt}(k)+e_f(k)\end{aligned} \tag{18}$$

where

$$e_f(k) = \boldsymbol{w}_{\rm opt}^H\boldsymbol{x}(k)-\boldsymbol{w}^H(k)\boldsymbol{x}(k) = -\boldsymbol{v}^H(k)\boldsymbol{x}(k) \tag{19}$$

is the noise-free *a priori* error. Moreover, the *a posteriori* error can be expressed as

$$\varepsilon(k) = \epsilon_{\rm opt}(k)+\varepsilon_f(k) \tag{20}$$

with

$$\begin{aligned}\varepsilon_f(k) &= \boldsymbol{w}_{\rm opt}^H\boldsymbol{x}(k)-\boldsymbol{w}^H(k+1)\boldsymbol{x}(k)\\ &= -\boldsymbol{v}^H(k+1)\boldsymbol{x}(k)\end{aligned} \tag{21}$$

being the noise-free *a posteriori* error. Taking the conjugate transpose of (17) and post-multiplying by $\boldsymbol{x}(k)$ on both sides, we get

$$\varepsilon_f(k) = (1-\mu_k\|\boldsymbol{x}(k)\|^2)e_f(k)-\mu_k\epsilon_{\rm opt}(k)\|\boldsymbol{x}(k)\|^2. \tag{22}$$

The power of $\varepsilon_f(k)$ is expressed as

$$\begin{aligned}|\varepsilon_f(k)|^2 = |e_f(k)|^2 &-\mu_k\left[2\|\boldsymbol{x}(k)\|^2|e_f(k)|^2\right.\\ &\left.+\epsilon_{\rm opt}^*(k)e_f(k)\|\boldsymbol{x}(k)\|^2+\epsilon_{\rm opt}(k)e_f^*(k)\|\boldsymbol{x}(k)\|^2\right]\\ &+\mu_k^2\left[\|\boldsymbol{x}(k)\|^4|e_f(k)|^2+\|\boldsymbol{x}(k)\|^4e_f(k)\epsilon_{\rm opt}^*(k)\right.\\ &\left.+\|\boldsymbol{x}(k)\|^4e_f^*(k)\epsilon_{\rm opt}(k)+\|\boldsymbol{x}(k)\|^4|\epsilon_{\rm opt}(k)|^2\right].\end{aligned} \tag{23}$$

Taking the derivative of $|\varepsilon_f(k)|^2$ in (23) with respect to μ_k and setting the resultant expression to 0 yield

$$\begin{aligned}2|e_f(k)|^2+\epsilon_{\rm opt}^*(k)e_f(k)+\epsilon_{\rm opt}(k)e_f^*(k)&\\ = 2\mu_k\|\boldsymbol{x}(k)\|^2[|e_f(k)+\epsilon_{\rm opt}(k)|^2].&\end{aligned} \tag{24}$$

Substituting (18) into (24), we have

$$\begin{aligned}2|e_f(k)|^2+\epsilon_{\rm opt}^*(k)e_f(k)+\epsilon_{\rm opt}(k)e_f^*(k)&\\ = 2\mu_k\|\boldsymbol{x}(k)\|^2|e(k)|^2.&\end{aligned} \tag{25}$$

It is enlightening to take the expectation of (25) on both sides to study the mean behavior of μ_k. Note that

$$\begin{aligned} E[\epsilon^*_{\rm opt}(k)\boldsymbol{x}(k)] &= E[s^*_0(k)\boldsymbol{x}(k)] - E[\boldsymbol{x}(k)\boldsymbol{x}^H(k)]\boldsymbol{w}_{\rm opt} \\ &= \boldsymbol{p}_x - \boldsymbol{R}_x\boldsymbol{R}_x^{-1}\boldsymbol{p}_x \\ &= \boldsymbol{0}_{M\times 1}. \end{aligned} \tag{26}$$

Therefore, the input signal $\boldsymbol{x}(k)$ is stochastically orthogonal to $\epsilon_{\rm opt}(k)$. Indeed, $e_f(k)$ is approximately uncorrelated with $\boldsymbol{x}(k)$ for transient as well as steady states, as will be verified in the simulation results. This is because the amplitude of $e_f(k)$ becomes quite small and changes slightly compared with $\boldsymbol{x}(k)$ [32]. As a result, we can neglect the dependence of $e_f(k)$ on $\boldsymbol{x}(k)$. Due to the fact that $e(k) = e_f(k) + \epsilon_{\rm opt}(k)$, we obtain

$$E[\|\boldsymbol{x}(k)\|^2|e(k)|^2] \approx E[\|\boldsymbol{x}(k)\|^2]E[|e(k)|^2]. \tag{27}$$

Next, $\boldsymbol{v}(k)$ depends only on $\{\boldsymbol{x}(j), s_0(j)\}$ for $j < k$ and is independent of the current input signal $\boldsymbol{x}(k)$ as we assume that the input sequences are independent. Then we have

$$E\left[\epsilon^*_{\rm opt}(k)e_f(k)\right] = -E\left[\epsilon^*_{\rm opt}(k)\boldsymbol{v}^H(k)\boldsymbol{x}(k)\right] = 0. \tag{28}$$

Taking the expectation of (25) and combining the results in (26)–(28), we get

$$E[\mu_k] = \frac{E[|e_f(k)|^2]}{E[\|\boldsymbol{x}(k)\|^2]E[|e(k)|^2]} \tag{29}$$

where it is assumed that

$$E[\mu_k\|\boldsymbol{x}(k)\|^2|e(k)|^2] = E[\mu_k]E[\|\boldsymbol{x}(k)\|^2|e(k)|^2]. \tag{30}$$

Here (30) is based on the assumption that μ_k is uncorrelated with $\boldsymbol{x}(k)$ and $e(k)$. Moreover, the rationale of this assumption will be explained at the end of Section III-A.

Since $E[|e_f(k)|^2]$ is the EMSE which measures the mismatch between the weight vector $\boldsymbol{w}(k)$ and optimal weight vector $\boldsymbol{w}_{\rm opt}$, setting

$$\mu_k = \frac{E[|e_f(k)|^2]}{E[\|\boldsymbol{x}(k)\|^2]E[|e(k)|^2]} \tag{31}$$

and replacing μ by μ_k in (7), we obtain the CLMS variant with VSS μ_k. In many practical applications, $E[\|\boldsymbol{x}(k)\|^2]$ is known and $E[|e(k)|^2]$ is estimated via

$$\sigma_e^2(k) = \lambda\sigma_e^2(k-1) + (1-\lambda)|e(k)|^2 \tag{32}$$

where λ is the forgetting factor and satisfies $0 \ll \lambda \leq 1$. However, the noise-free *a priori* error $e_f(k)$ is unknown and the main difficulty to solve (31) is to calculate $E[|e_f(k)|^2]$. According to the shrinkage denoising (soft-thresholding) method described in [30], [31], the noise-free *a priori* error $e_f(k)$ can be recovered from the *a priori* error $e(k)$. A noisy SOI $\boldsymbol{s} \in \mathbb{C}^n$ can be expressed as

$$\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{b} \tag{33}$$

where $\boldsymbol{y}$ is the observed data vector and $\boldsymbol{b}$ is a zero-mean independent and identically distributed (i.i.d.) Gaussian vector. The task is to recover the SOI $\boldsymbol{s}$ from $\boldsymbol{y}$. According to [31], the cost function of this denoising problem is

$$f[\boldsymbol{z}] = 0.5\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|^2 + \alpha\|\boldsymbol{z}\|_1 \tag{34}$$

where $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ is a prespecified and redundant dictionary and α governs the tradeoff between the representation error and its sparsity, namely, the first and second terms on the right hand side of (34). The SOI $\boldsymbol{s}$ can be represented as

$$\boldsymbol{s} = \boldsymbol{A}\boldsymbol{z} \tag{35}$$

with $\boldsymbol{z}$ being expected to be sparse, that is to say, $\|\boldsymbol{z}\|_0 \ll n$. As a result, by minimizing the cost function in (34) with respect to $\boldsymbol{z}$, the optimal estimates of SOI is computed as

$$\hat{\boldsymbol{s}} = \boldsymbol{A}\hat{\boldsymbol{z}} \tag{36}$$

where

$$\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} f[\boldsymbol{z}]. \tag{37}$$

From (18), we see that the noise-free *a priori* error $e_f(k)$ can be recovered from the *a priori* error $e(k)$ using the shrinkage denoising method. The cost function in (34) now becomes

$$f[e_f(k)] = 0.5|e(k) - e_f(k)|^2 + \alpha|e_f(k)|. \tag{38}$$

Due to the fact that n reduces to 1 and $\boldsymbol{A}$ becomes 1 in this case, we obtain $\hat{\boldsymbol{s}} = \hat{\boldsymbol{z}}$. In other words, $\hat{e}_f(k)$ is the optimal estimate of $e_f(k)$. Minimizing $f[e_f(k)]$ with respect to $e_f(k)$ yields

$$\hat{e}_f(k) = \mathrm{sign}[e(k)]\max(|e(k)| - \alpha, 0). \tag{39}$$

Note that the choice of the threshold parameter α is very important. Assume that the background noise is zero-mean white Gaussian distributed with covariance matrix $\sigma_\eta^2\boldsymbol{I}_{\rm M}$ and the interferences are far away from the main lobe of the SOI. As a result, the optimal weight vector $\boldsymbol{w}_{\rm opt}$ maintains the most energy of the SOI while suppressing the energy of the interferences, i.e.,

$$\boldsymbol{w}_{\rm opt}^H\boldsymbol{a}(\theta_d) \approx 1, \tag{40a}$$

$$\boldsymbol{w}_{\rm opt}^H\boldsymbol{a}(\theta_i) \approx 0, \quad i = 1, 2, \ldots, P. \tag{40b}$$

Form (18) and (40), we obtain

$$\begin{aligned} E[|\epsilon_{\rm opt}(k)|^2] &= E[|e_f(k) - e(k)|^2] \\ &= E\left[\left|s_0(k) - \boldsymbol{w}_{\rm opt}^H\boldsymbol{x}(k)\right|^2\right] \\ &\approx E\left[\left|s_0(k) - s_0(k) - \boldsymbol{w}_{\rm opt}^H\boldsymbol{\eta}(k)\right|^2\right] \\ &= \sigma_\eta^2\|\boldsymbol{w}_{\rm opt}\|^2. \end{aligned} \tag{41}$$

As the background noise $\boldsymbol{\eta}(k)$ is a spatially and temporally white Gaussian process, we set $\|\boldsymbol{w}_{\rm opt}\|^2 \geq \frac{1}{M}$ to guarantee that the beampattern equals 1 at the DOA of the SOI and the array gain is maximized [35]. As a result, from (41) we obtain

$$E[|e_f(k) - e(k)|^2] \geq \frac{\sigma_\eta^2}{M}. \tag{42}$$

TABLE I
PSEUDO CODE OF SL-CLMS ALGORITHM

Step 1 initialization of $\sigma_{ef}^2(0)$, $\sigma_e^2(0)$, Q, λ and $\boldsymbol{w}(1)$.
Step 2 set $\alpha = \sqrt{Q\sigma_\eta^2/M}$.
Step 3 for time index $k = 1, 2, \ldots, N$, do

$$e(k) = s_0(k) - \boldsymbol{w}^H(k)\boldsymbol{x}(k) \tag{44a}$$
$$\sigma_e^2(k) = \lambda\sigma_e^2(k-1) + (1-\lambda)|e(k)|^2 \tag{44b}$$
$$\hat{e}_f(k) = \text{sign}[e(k)]\max(|e(k)| - \alpha, 0) \tag{44c}$$
$$\sigma_{ef}^2(k) = \lambda\sigma_{ef}^2(k-1) + (1-\lambda)|\hat{e}_f(k)|^2 \tag{44d}$$
$$\mu_k = \frac{\sigma_{ef}^2(k)}{E[||\boldsymbol{x}(k)||^2]\sigma_e^2(k)} \tag{44e}$$
$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu_k e^*(k)\boldsymbol{x}(k) \tag{44f}$$

end

Note that the dependence of $E[|\epsilon_{\text{opt}}(k)|^2]$ on the interferences has been neglected in the derivation of (41). As a result, fine tuning of the threshold parameter α around $\sqrt{\sigma_\eta^2/M}$ could yield improved performance. Here, α is chosen as $\alpha = \sqrt{Q\sigma_\eta^2/M}$ where Q is the parameter to compensate the approximation we have made above. In analogy to $E[|e(k)|^2]$,

$$\sigma_{ef}^2(k) = \lambda\sigma_{ef}^2(k-1) + (1-\lambda)|\hat{e}_f(k)|^2 \tag{43}$$

can be used to estimate the numerator $E[|e_f(k)|^2]$ in (31). The proposed SL-CLMS algorithm is summarized as in Table I. From the update equations, namely, Step 3 in the SL-CLMS algorithm, we can conclude that the assumption of (30) is valid. It follows from (44b) and (44d) that the $\sigma_e^2(k)$ and $\sigma_{ef}^2(k)$ vary slowly compared with $\boldsymbol{x}(k)$ and $e(k)$ when λ is close to 1. As a result, according to (44e), μ_k is approximately uncorrelated with $\boldsymbol{x}(k)$ and $e(k)$.

B. SWL-CLMS Algorithm

Similar to the analysis in Section III-A, subtracting the corresponding optimal weight vectors $\boldsymbol{w}_{1,\text{opt}}$ of $\boldsymbol{w}_1(k)$ on both sides of (11), we get the update equation of the weight error vector $\boldsymbol{v}_1(k) = \boldsymbol{w}_1(k) - \boldsymbol{w}_{1,\text{opt}}$ as

$$\boldsymbol{v}_1(k+1) = \boldsymbol{v}_1(k) - \mu_k\tilde{e}^*(k)\boldsymbol{x}(k). \tag{45}$$

Moreover, subtracting $\boldsymbol{w}_{2,\text{opt}}$ on both sides of (12) yields

$$\boldsymbol{v}_2(k+1) = \boldsymbol{v}_2(k) - \mu_k\tilde{e}^*(k)\boldsymbol{x}^*(k) \tag{46}$$

with $\boldsymbol{v}_2(k) = \boldsymbol{w}_2(k) - \boldsymbol{w}_{2,\text{opt}}$. Here, the VSS μ_k is used to replace the CSS μ in (11) and (12). According to (10), (45) and (46), the vector-matrix form for the weight vector errors becomes

$$\begin{bmatrix}\boldsymbol{v}_1(k+1)\\ \boldsymbol{v}_2(k+1)\end{bmatrix} = \begin{bmatrix}\boldsymbol{I}_{\text{M}} - \mu_k\boldsymbol{x}(k)\boldsymbol{x}^H(k) & -\mu_k\boldsymbol{x}(k)\boldsymbol{x}^T(k)\\ -\mu_k\boldsymbol{x}^*(k)\boldsymbol{x}^H(k) & \boldsymbol{I}_{\text{M}} - \mu_k\boldsymbol{x}^*(k)\boldsymbol{x}^T(k)\end{bmatrix} \times \begin{bmatrix}\boldsymbol{v}_1(k)\\ \boldsymbol{v}_2(k)\end{bmatrix} - \mu_k\begin{bmatrix}\boldsymbol{x}(k)\\ \boldsymbol{x}^*(k)\end{bmatrix}\tilde{\epsilon}_{\text{opt}}^*(k) \tag{47}$$

where $\tilde{\epsilon}_{\text{opt}}(k) \triangleq s_0(k) - \boldsymbol{w}_{1,\text{opt}}^H(k)\boldsymbol{x}(k) - \boldsymbol{w}_{2,\text{opt}}^H\boldsymbol{x}^*(k)$ is the error obtained by utilizing the optimal augmented weight vector $\tilde{\boldsymbol{w}}_{\text{opt}} = [\boldsymbol{w}_{1,\text{opt}}^T, \boldsymbol{w}_{2,\text{opt}}^T]^T$ on the augmented input signal $\tilde{\boldsymbol{x}}(k) = [\boldsymbol{x}^T(k), \boldsymbol{x}^H(k)]^T$. Taking the conjugate transpose of (47) and post-multiplying both sides with the augmented input signal $\tilde{\boldsymbol{x}}(k) = [\boldsymbol{x}^T(k), \boldsymbol{x}^H(k)]^T$, we get

$$\tilde{\varepsilon}_f(k) = (1 - 2\mu_k\|\boldsymbol{x}(k)\|^2)\tilde{e}_f(k) - 2\mu_k\tilde{\epsilon}_{\text{opt}}\|\boldsymbol{x}(k)\|^2 \tag{48}$$

where

$$\begin{aligned}\tilde{\varepsilon}_f(k) &= [\boldsymbol{w}_{1,\text{opt}} - \boldsymbol{w}_1(k+1)]^H\boldsymbol{x}(k)\\ &\quad + [\boldsymbol{w}_{2,\text{opt}} - \boldsymbol{w}_2(k+1)]^H\boldsymbol{x}^*(k)\\ &= -\boldsymbol{v}_1^H(k+1)\boldsymbol{x}(k) - \boldsymbol{v}_2^H(k+1)\boldsymbol{x}^*(k)\end{aligned} \tag{49}$$

is the augmented noise-free *a posteriori* error and

$$\begin{aligned}\tilde{e}_f(k) &= [\boldsymbol{w}_{1,\text{opt}} - \boldsymbol{w}_1(k)]^H\boldsymbol{x}(k)\\ &\quad + [\boldsymbol{w}_{2,\text{opt}} - \boldsymbol{w}_2(k)]^H\boldsymbol{x}^*(k)\\ &= -\boldsymbol{v}_1^H(k)\boldsymbol{x}(k) - \boldsymbol{v}_2^H(k)\boldsymbol{x}^*(k)\end{aligned} \tag{50}$$

is the augmented noise-free *a priori* error. In analogy to the SL-CLMS, the square of the instantaneous augmented noise-free *a posteriori* error can be written as

$$\begin{aligned}|\tilde{\varepsilon}_f(k)|^2 &= |\tilde{e}_f(k)|^2 - \mu_k[4\|\boldsymbol{x}(k)\|^2|\tilde{e}_f(k)|^2\\ &\quad + 2\tilde{\epsilon}_{\text{opt}}^*(k)\tilde{e}_f(k)\|\boldsymbol{x}(k)\|^2 + 2\tilde{\epsilon}_{\text{opt}}(k)\tilde{e}_f^*(k)\|\boldsymbol{x}(k)\|^2]\\ &\quad + \mu_k^2[4\|\boldsymbol{x}(k)\|^4|\tilde{e}_f(k)|^2 + 4\|\boldsymbol{x}(k)\|^4\tilde{e}_f(k)\tilde{\epsilon}_{\text{opt}}^*(k)\\ &\quad + 4\|\boldsymbol{x}(k)\|^4\tilde{e}_f^*(k)\tilde{\epsilon}_{\text{opt}}(k) + 4\|\boldsymbol{x}(k)\|^4|\tilde{\epsilon}_{\text{opt}}(k)|^2].\end{aligned} \tag{51}$$

Taking $|\tilde{\varepsilon}_f(k)|^2$ as the cost function and setting its derivative with respect to μ_k to 0, we obtain

$$\begin{aligned}&2|\tilde{e}_f(k)|^2 + \tilde{\epsilon}_{\text{opt}}^*(k)\tilde{e}_f(k) + \tilde{\epsilon}_{\text{opt}}(k)\tilde{e}_f^*(k)\\ &\qquad = 4\mu_k\|\boldsymbol{x}(k)\|^2[|\tilde{e}_f(k) + \tilde{\epsilon}_{\text{opt}}(k)|^2].\end{aligned} \tag{52}$$

The instantaneous error at the time instant k is

$$\begin{aligned}\tilde{e}(k) &= s_0(k) - \tilde{w}^H(k)\tilde{\boldsymbol{x}}(k)\\ &= \tilde{e}_f(k) + \tilde{\epsilon}_{\text{opt}}(k).\end{aligned} \tag{53}$$

Substituting (53) into (52) yields

$$\begin{aligned}&2|\tilde{e}_f(k)|^2 + \tilde{\epsilon}_{\text{opt}}^*(k)\tilde{e}_f(k) + \tilde{\epsilon}_{\text{opt}}(k)\tilde{e}_f^*(k)\\ &\qquad = 4\mu_k\|\boldsymbol{x}(k)\|^2|\tilde{e}(k)|^2.\end{aligned} \tag{54}$$

It is observed that

$$\begin{aligned}E\left[\tilde{\epsilon}_{\text{opt}}^*(k)\tilde{\boldsymbol{x}}(k)\right] &= E\left[s_0^*(k)\tilde{\boldsymbol{x}}(k)\right] - E[\tilde{\boldsymbol{x}}(k)\tilde{\boldsymbol{x}}^H(k)]\tilde{\boldsymbol{w}}_{\text{opt}}\\ &= \tilde{\boldsymbol{p}}_x - \tilde{\boldsymbol{R}}_x\tilde{\boldsymbol{R}}_x^{-1}\tilde{\boldsymbol{p}}_x\\ &= \boldsymbol{0}_{2M\times 1}.\end{aligned} \tag{55}$$

In analogy to independence assumption of $e_f(k)$ and $\boldsymbol{x}(k)$ in Section III-A, we assume that $\tilde{e}_f(k)$ is approximately uncorrelated with $\boldsymbol{x}(k)$. Because $\tilde{\epsilon}_{\text{opt}}(k)$ is orthogonal to the augmented input signal $\tilde{x}(k)$ and $\tilde{e}(k) = \tilde{e}_f(k) + \tilde{\epsilon}_{\text{opt}}(k)$, we conclude that

TABLE II
PSEUDO CODE OF SWL-CLMS ALGORITHM

Step 1 initialization of $\tilde{\sigma}_{ef}^2(0)$, $\tilde{\sigma}_e^2(0)$, Q, λ, $\boldsymbol{w}_1(1)$ and $\boldsymbol{w}_2(1)$.
Step 2 set $\alpha = \sqrt{Q\sigma_\eta^2/(2M)}$.
Step 3 for time index $k = 1, 2, \ldots, N$, do

$$\tilde{e}(k) = s_0(k) - \boldsymbol{w}_1^H(k)\boldsymbol{x}(k) - \boldsymbol{w}_2^H\boldsymbol{x}^*(k) \quad (69a)$$
$$\tilde{\sigma}_e^2(k) = \lambda\tilde{\sigma}_e^2(k-1) + (1-\lambda)|\tilde{e}(k)|^2 \quad (69b)$$
$$\hat{\tilde{e}}_f(k) = \text{sign}[\tilde{e}(k)]\max(|\tilde{e}(k)| - \alpha, 0) \quad (69c)$$
$$\tilde{\sigma}_{ef}^2(k) = \lambda\tilde{\sigma}_{ef}^2(k-1) + (1-\lambda)|\hat{\tilde{e}}_f(k)|^2 \quad (69d)$$
$$\mu_k = \frac{\tilde{\sigma}_{ef}^2(k)}{2E[\|\boldsymbol{x}(k)\|^2]\tilde{\sigma}_e^2(k)} \quad (69e)$$
$$\boldsymbol{w}_1(k+1) = \boldsymbol{w}_1(k) + \mu_k\tilde{e}^*(k)\boldsymbol{x}(k) \quad (69f)$$
$$\boldsymbol{w}_2(k+1) = \boldsymbol{w}_2(k) + \mu_k\tilde{e}^*(k)\boldsymbol{x}^*(k) \quad (69g)$$

end

the *a priori* error $\tilde{e}(k)$ is approximately uncorrelated with the augmented input signal $\tilde{x}(k)$, i.e.,

$$E[\|\boldsymbol{x}(k)\|^2|\tilde{e}(k)|^2] \approx E[\|\boldsymbol{x}(k)\|^2]E[|\tilde{e}(k)|^2]. \quad (56)$$

According to (45) and (46), we conclude that $\boldsymbol{v}_1(k)$ is uncorrelated with $\boldsymbol{x}(k)$ and $\boldsymbol{v}_2(k)$ is uncorrelated with $\boldsymbol{x}^*(k)$. Setting $\tilde{\boldsymbol{v}}(k) = [\boldsymbol{v}_1^T(k), \boldsymbol{v}_2^T(k)]^T$ and making use of the result in (55), we obtain

$$\begin{aligned} E[\tilde{\epsilon}_{\text{opt}}^*(k)\tilde{e}_f(k)] &= E\left\{\left[-\boldsymbol{v}_1^H(k)\boldsymbol{x}(k) - \boldsymbol{v}_2^H(k)\boldsymbol{x}^*(k)\right]\tilde{\epsilon}_{\text{opt}}^*(k)\right\} \\ &= E\left[-\tilde{\boldsymbol{v}}^H(k)\tilde{\boldsymbol{x}}(k)\tilde{\epsilon}_{\text{opt}}^*(k)\right] \\ &= 0. \end{aligned} \quad (57)$$

Taking expectations on both sides of (52) and employing (56) and (57), we have

$$E[\mu_k] = \frac{E[|\tilde{e}_f(k)|^2]}{2E[\|\boldsymbol{x}(k)\|^2]E[|\tilde{e}(k)|^2]} \quad (58)$$

which is based on the assumption

$$E[\mu_k\|\boldsymbol{x}(k)\|^2|\tilde{e}(k)|^2] = E[\mu_k]E[\|\boldsymbol{x}(k)\|^2|\tilde{e}(k)|^2] \quad (59)$$

and the rationale of this assumption will be explained at the end of Section III-B. Taking $E[\mu_k]$ as the estimate of μ_k and substituting it into (47), we get the WL-CLMS variant with VSS μ_k.

Following the discussion in Section III-A, the MSE of the error signals $E[|\tilde{e}(k)|^2]$ in (58) is estimated via

$$\tilde{\sigma}_e^2(k) = \lambda\tilde{\sigma}_e^2(k-1) + (1-\lambda)|\tilde{e}(k)|^2, \quad (60)$$

and the time average of the square of the $\tilde{e}_f(k)$, i.e.,

$$\tilde{\sigma}_{ef}^2(k) = \lambda\tilde{\sigma}_{ef}^2(k-1) + (1-\lambda)|\tilde{e}_f(k)|^2 \quad (61)$$

can be used to replace $E[|\tilde{e}_f(k)|^2]$ in (58). The estimate of $\tilde{e}_f(k)$ is recovered from $\tilde{e}(k)$ by using (39) as

$$\hat{\tilde{e}}_f(k) = \text{sign}[\tilde{e}(k)]\max(|\tilde{e}(k)| - \alpha, 0). \quad (62)$$

Next, we will discuss how to choose the threshold parameter α which is associated with $E[|\tilde{\epsilon}_{\text{opt}}(k)|^2]$. As the interferences and background noise are uncorrelated with the SOI, $\tilde{\boldsymbol{p}}_x$ can be expressed as

$$\begin{aligned} \tilde{\boldsymbol{p}}_x &= E\left[\tilde{\boldsymbol{x}}(k)s_0^*(k)\right] \\ &= E\left[\tilde{\boldsymbol{a}}(\theta_d)\sigma_s^2 + \boldsymbol{n}(k)s_0^*(k)\right] \\ &= \sigma_s^2\tilde{\boldsymbol{a}}(\theta_d) \end{aligned} \quad (63)$$

where σ_s^2 is the power of the SOI. As a result, (13) can be rewritten as

$$\tilde{\boldsymbol{w}}_{\text{opt}} = \sigma_s^2\tilde{\boldsymbol{R}}_x^{-1}\tilde{\boldsymbol{a}}(\theta_d) \quad (64)$$

which is similar to the optimal widely-linear MVDR beamformer [35]

$$\tilde{\boldsymbol{w}}_{\text{MVDR}} = \kappa\tilde{\boldsymbol{R}}_x^{-1}\tilde{\boldsymbol{a}}(\theta_d) \quad (65)$$

with $\kappa = 1/[\tilde{\boldsymbol{a}}^H(\theta_d)\tilde{\boldsymbol{R}}_x^{-1}\tilde{\boldsymbol{a}}(\theta_d)]$. The $\tilde{\boldsymbol{w}}_{\text{opt}}$ and $\tilde{\boldsymbol{w}}_{\text{MVDR}}$ only have a difference in the constant factor. As a result, when the DOAs of the interferences are far away from the DOA of the SOI, the following conclusions can be drawn

$$\tilde{\boldsymbol{w}}_{\text{opt}}^H\tilde{\boldsymbol{a}}(\theta_d) \approx 1, \quad (66a)$$
$$\tilde{\boldsymbol{w}}_{\text{opt}}^H\tilde{\boldsymbol{a}}(\theta_i) \approx 0, \quad i = 1, 2, \ldots, P \quad (66b)$$

where $\tilde{\boldsymbol{a}}(\theta_i) = [\boldsymbol{a}^T(\theta_i), e^{-2j\varphi_i}\boldsymbol{a}^H(\theta_i)]^T$, $i = 1, 2, \ldots, P$, with φ_i being the initial phase of the ith interference. The approximate estimate of $E[|\tilde{\epsilon}_{\text{opt}}(k)|^2]$ is given as

$$\begin{aligned} E[|\tilde{\epsilon}_{\text{opt}}(k)|^2] &= E\left[\left|s_0(k) - \tilde{\boldsymbol{w}}_{\text{opt}}^H\tilde{\boldsymbol{x}}(k)\right|^2\right] \\ &\approx E\left[\left|s_0(k) - s_0(k) - \tilde{\boldsymbol{w}}_{\text{opt}}^H\tilde{\boldsymbol{\eta}}(k)\right|^2\right] \\ &= \sigma_\eta^2\|\tilde{\boldsymbol{w}}_{\text{opt}}\|^2 \end{aligned} \quad (67)$$

with $\tilde{\boldsymbol{\eta}}(k) = [\boldsymbol{\eta}^T(k), \boldsymbol{\eta}^H(k)]^T$. According to [35], we always have $\|\tilde{\boldsymbol{w}}_{\text{opt}}\|^2 \geq \frac{1}{2M}$, which, when substituted into (67), leads to

$$E[|\tilde{\epsilon}_{\text{opt}}(k)|^2] \geq \frac{\sigma_\eta^2}{2M}. \quad (68)$$

Moreover, the threshold parameter α can be chosen such that $\alpha = \sqrt{Q\sigma_\eta^2/(2M)}$ with Q varying from 1 to 4, which is the regulation parameter to compensate the approximation we have made in the derivation of α. The SWL-CLMS algorithm is summarized in Table II. It follows from (69b) and (69d) in Step 3, we can conclude that $\tilde{\sigma}_e^2(k)$ and $\tilde{\sigma}_{ef}^2(k)$ vary slowly when the parameter λ is close to 1. Consequently, according to (69e), the VSS μ_k is approximately uncorrelated with $\boldsymbol{x}(k)$ and $\tilde{e}(k)$, and thus the assumption of (59) is valid.

C. Complexity Analysis

Here, we evaluate the computational complexity of the proposed SL-CLMS and SWL-CLMS algorithms. The computational cost is determined in the form of the number of complex arithmetic operations, i.e., additions and multiplications, per iteration per symbol for each approach as a function of the number of sensors M. The results are depicted in Table III. It is seen that the complexities of the SL-CLMS and SWL-CLMS algorithms are $\mathcal{O}(M)$ while those of the RLS and WL-RLS solutions are $\mathcal{O}(M^2)$. As a result, the proposed CLMS-based

TABLE III
COMPUTATIONAL COMPLEXITIES OF DIFFERENT ALGORITHMS

Algorithm	Additions	Multiplications
RLS [22]	$2M^2+M+1$	$2M^2+4M+2$
WL-RLS [23]	$4M^2+3M+1$	$4M^2+6M+4$
SL-CLMS	$2M+4$	$2M+9$
SWL-CLMS	$4M+4$	$4M+10$
SL-CLMS (no approx.)	$3M+5$	$3M+13$
SWL-CLMS (no approx.)	$5M+3$	$5M+10$

approaches enjoy a simpler computational complexity than the RLS-based schemes especially when M is large. For the SL-CLMS and SWL-CLMS without the approximations in (27) and (56), they have slightly higher computational loads than their variants with the approximations as the former need to calculate $\|\boldsymbol{x}(k)\|^2$ per iteration. Moreover, it is observed that the WL-based algorithms are more computationally demanding than the conventional methods because the calculations of larger matrices result in higher complexity requirement.

In order to validate the complexities of these methods, we provide their computational times in Section IV. The computational times are determined by the execution time in MATLAB with the version of 7.11.0.584 (R2010b). Note that the execution time is measured by the average CPU time required to compute each algorithm on a personal computer with an Intel i3–3220 3.3 GHz processor and 4 GB RAM.

IV. SIMULATION RESULTS

In this section, we evaluate the output SINRs, MSEs and computational times of the proposed algorithms for adaptive beamforming. In particular, the SL-CLMS algorithm is compared with the CLMS, CNLMS, RLS and VSS methods. Moreover, the SWL-CLMS algorithm is compared with the WL-CLMS, WL-CNLMS, WL-RLS and WL-VSS solutions. Furthermore, the effects of approximations in (27) and (56), are numerically investigated. In our simulations, a uniform linear array consisting of $M=4$ omnidirectional sensors with an interelement spacing of half wavelength is considered. We assume that there are four equipowered BPSK signals with noncircularity coefficient of 1 and their initial phases are all fixed at 0°. Among the four BPSK signals impinging on the array, the SOI is presumed to arrive at $\theta_d=-45°$ while three interferences impinge on the array with DOAs of $\theta_1=8°, \theta_2=-13°$ and $\theta_3=30°$. Unless stated otherwise, the signal-to-noise ratio (SNR) and the interference-to-noise ratio are all fixed at 10 dB. In all examples, the numerical results are averaged over 500 independent Monte Carlo simulations. For the SL-CLMS algorithm, the initial values of the parameters are set as $\sigma_{ef}^2(0)=0$, $\sigma_e^2(0)=0$ and $\boldsymbol{w}(1)=\boldsymbol{0}_{M\times 1}$. Moreover, the initial values of the parameters for the SWL-CLMS algorithm are $\tilde{\sigma}_{ef}^2(0)=0$, $\tilde{\sigma}_e^2(0)=0$, $\boldsymbol{w}_1(1)=\boldsymbol{0}_{M\times 1}$ and $\boldsymbol{w}_2(1)=\boldsymbol{0}_{M\times 1}$. The forgetting factor λ is fixed at $\lambda=0.95$ for both of them.

A. Output SINR Performance Versus Number of Iterations

In the first example, the effect of Q on the output SINR performance is examined. For comparison, the numerical results of the CLMS and WL-CLMS algorithms are provided as well. The step size of the CLMS and WL-CLMS algorithms are fixed

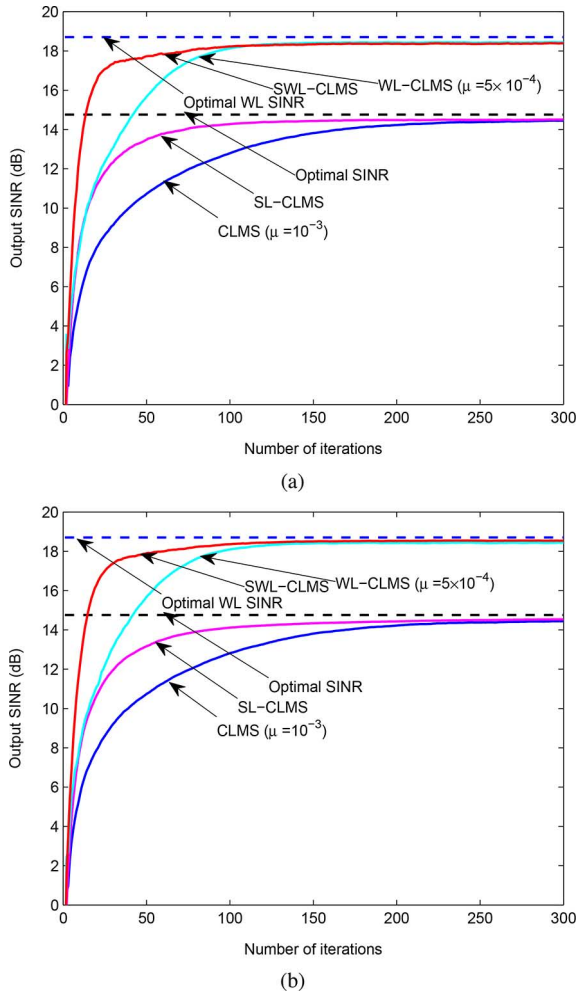
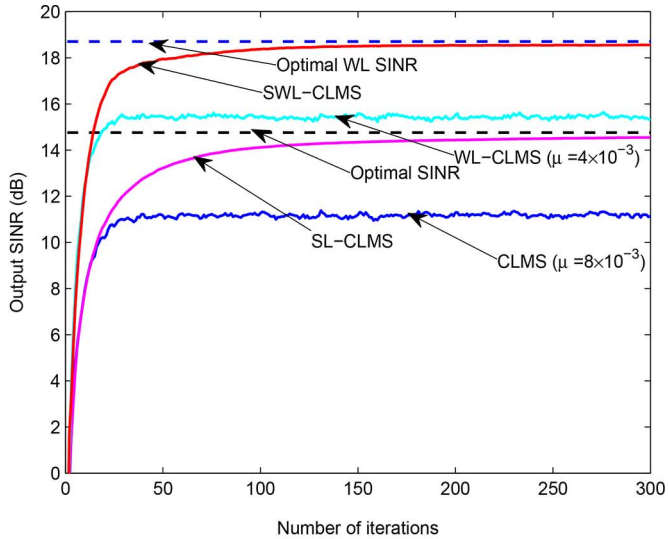


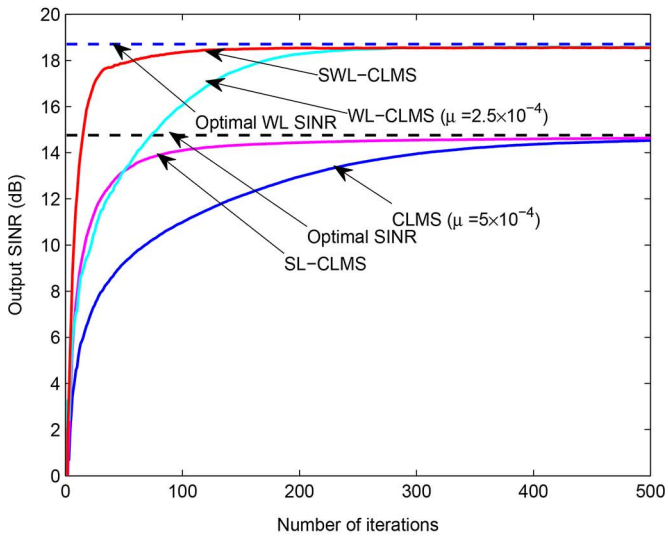
Fig. 1. Output SINRs of SL-CLMS, SWL-CLMS, CLMS and WL-CLMS algorithms. (a) $Q=1$. (b) $Q=2$.

at 10^{-3} and 5×10^{-4}, respectively. It is observed in Fig. 1 that the SL-CLMS and SWL-CLMS algorithms converge faster than their counterparts in both cases. Moreover, due to the fact that the WL-CLMS and SWL-CLMS algorithms take advantage of the NC properties of the SOI, they yield larger output SINR than their counterparts. From Fig. 1, we see that the value of Q has little influence on the steady-state performance of the SL-CLMS algorithm. Furthermore, the SWL-CLMS algorithm with $Q=2$ has a slightly performance improvement compared with the scenario of $Q=1$.

Fig. 2 shows the output SINR performance versus the number of iterations with Q being fixed at 2. In Fig. 2(a), the CSS μ is fixed at 4×10^{-3} and 8×10^{-3} for the WL-CLMS and CLMS algorithms, respectively. In Fig. 2(b), the CSS μ is fixed at 2.5×10^{-4} and 5×10^{-4} for the WL-CLMS and CLMS algorithms, respectively. It is seen in Fig. 2(a) that the WL-CLMS and CLMS algorithms have a faster convergence speed compared with those in Fig. 1(b). That is due to the fact that a large CSS leads to a fast convergence speed. However, the WL-CLMS and CLMS algorithms can only arrive at a small output SINR in the steady-state. If we fix the step size as half of the value of μ in Fig. 1(b), from Fig. 2(b), we observe that the WL-CLMS and CLMS algorithms arrive at the same output SINR in the steady-state as the proposed methods. However,



(a)



(b)

Fig. 2. Output SINRs of SL-CLMS, SWL-CLMS, CLMS, WL-CLMS algorithms for different μ.

it is seen in Fig. 2(b) that the WL-CLMS and CLMS algorithms approach the steady-state at about the 200th and 400th iterations, respectively. Moreover, the WL-CLMS and CLMS algorithms arrive at the steady-state at around the 100th and 200th iterations, respectively, as shown in Fig. 1(b). That is due to the fact that a small CSS leads to a slow convergence speed. On the contrary, it is observed in Fig. 2 that the SWL-CLMS and SL-CLMS methods reach the steady-state only at around the 60th and 100th iterations, respectively. Furthermore, the output SINRs of the proposed SWL-CLMS and SL-CLMS algorithms can converge to the optimal WL SINR and SINR, respectively. As a result, by utilizing the VSS, our proposed algorithms can provide a faster convergence speed and a larger output SINR than the WL-CLMS and CLMS solutions.

The empirical results of the output SINRs of the SL-CLMS, SWL-CLMS, RLS and WL-RLS algorithms are shown in Fig. 3. It is seen that the SL-CLMS method approximately provides the same output SINR as the RLS method while the SWL-CLMS scheme has the same output SINR as the

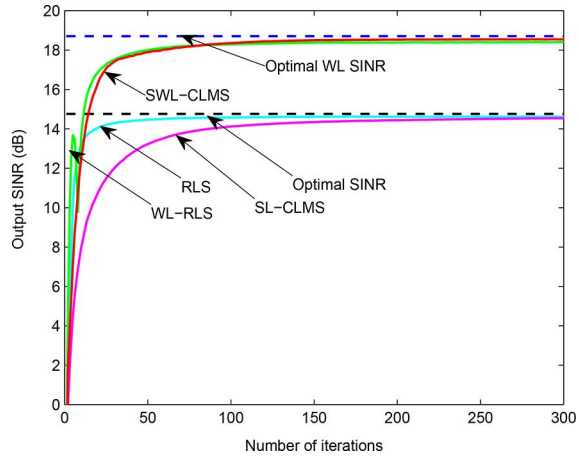


Fig. 3. Output SINRs of SL-CLMS, SWL-CLMS, RLS and WL-RLS algorithms.

WL-RLS solution in the steady-state. Moreover, the output SINRs of the SL-CLMS and RLS algorithms are smaller than the WL-based schemes because they do not exploit the NC properties of the signals. However, Fig. 3 shows that the RLS and WL-RLS algorithms can converge faster than the SL-CLMS and SWL-CLMS methods, respectively. Although the RLS-based approaches converge faster, they require much more computational cost than the proposed methods, as will be verified in Fig. 11.

Let us now compare the SL-CLMS algorithm with the CNLMS and VSS [24] methods and compare the SWL-CLMS algorithm with the WL-CNLMS and WL-VSS [25] approaches. According to [20], [21], the update equation for the CNLMS is

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \frac{\bar{\mu}}{\boldsymbol{x}^H(k)\boldsymbol{x}(k)} e^*(k)\boldsymbol{x}(k) \tag{70}$$

and the update equations for the WL-CNLMS are

$$\boldsymbol{w}_1(k+1) = \boldsymbol{w}_1(k) + \frac{\tilde{\bar{\mu}}}{\boldsymbol{x}^H(k)\boldsymbol{x}(k)} \tilde{e}^*(k)\boldsymbol{x}(k), \tag{71}$$

$$\boldsymbol{w}_2(k+1) = \boldsymbol{w}_2(k) + \frac{\tilde{\bar{\mu}}}{\boldsymbol{x}^H(k)\boldsymbol{x}(k)} \tilde{e}^*(k)\boldsymbol{x}^*(k) \tag{72}$$

with $\bar{\mu} = 0.2$ and $\tilde{\bar{\mu}} = 0.1$. The VSS method derived in [24] has the form of

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu_k e^*(k)\boldsymbol{x}(k) \tag{73}$$

where the updating rule for the VSS μ_k is

$$\grave{\mu}_{k+1} = \bar{\alpha}\mu_k + \bar{\gamma}\|e(k)\|^2 \tag{74}$$

with $\bar{\alpha} = 0.98$, $\bar{\gamma} = 10^{-3}$ and

$$\mu_{k+1} = \begin{cases} \mu_{\max} & \text{if } \grave{\mu} > \mu_{\max} \\ \mu_{\min} & \text{if } \grave{\mu} < \mu_{\min} \\ \grave{\mu}_{k+1} & \text{otherwise.} \end{cases} \tag{75}$$

Here, $\mu_{\min} = 10^{-6}$ and $\mu_{\max} = 3 \times 10^{-3}$. Similar to the VSS method, the weight vector of the WL-VSS algorithm is adjusted according to

$$\boldsymbol{w}_1(k+1) = \boldsymbol{w}_1(k) + \mu_k \tilde{e}^*(k)\boldsymbol{x}(k), \tag{76}$$

$$\boldsymbol{w}_2(k+1) = \boldsymbol{w}_2(k) + \mu_k \tilde{e}^*(k)\boldsymbol{x}^*(k) \tag{77}$$

where μ_k is determined by the same rule as the VSS method except that $\mu_{\min}$ and $\mu_{\max}$ are fixed at 5×10^{-7} and 1.5×10^{-3},

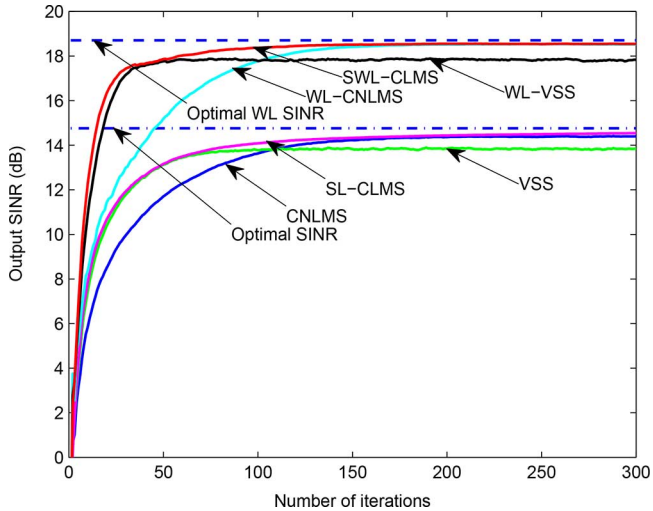


Fig. 4. Output SINRs of SL-CLMS, SWL-CLMS, CNLMS, WL-CNLMS, VSS and WL-VSS algorithms.

respectively. It can be seen in Fig. 4 that the proposed methods yield faster convergence speeds than the CNLMS and WL-CNLMS algorithms and larger output SINRs than the VSS and WL-VSS solutions. Furthermore, the SWL-CLMS, WL-CNLMS and WL-VSS algorithms outperforms their counterparts, i.e., the SL-CLMS, CNLMS and VSS methods because they exploit the NC properties of the SOI.

In order to investigate the effect of approximations in (27) and (56), similar to the estimates of $E[|e(k)|^2]$ and $E[|\tilde{e}(k)|^2]$ in (32) and (60), we estimate $E[\|\boldsymbol{x}(k)\|^2|e(k)|^2]$ and $E[\|\boldsymbol{x}(k)\|^2|\tilde{e}(k)|^2]$ by using the following equations:

$$\sigma_{xe}^2(k) = \lambda\sigma_{xe}^2(k-1) + (1-\lambda)|e(k)|^2\|\boldsymbol{x}(k)\|^2 \quad (78)$$

and

$$\tilde{\sigma}_{xe}^2(k) = \lambda\tilde{\sigma}_{xe}^2(k-1) + (1-\lambda)|\tilde{e}(k)|^2\|\boldsymbol{x}(k)\|^2, \quad (79)$$

respectively. According to (31) and (58), when there exist no approximations in (27) and (56), the VSS μ_k for the SL-CLMS and SWL-CLMS algorithms can be rewritten as

$$\mu_k = \frac{E[|e_f(k)|^2]}{E[\|\boldsymbol{x}(k)\|^2|e(k)|^2]} \quad (80)$$

and

$$\mu_k = \frac{E[|\tilde{e}_f(k)|^2]}{E[2\|\boldsymbol{x}(k)\|^2|\tilde{e}(k)|^2]} \quad (81)$$

respectively. As a result, based on (78) and (79), the update equations of the VSS μ_k in Tables I and II can be rewritten as

$$\mu_k = \frac{\sigma_{ef}^2(k)}{\sigma_{xe}^2(k)} \quad (82)$$

and

$$\mu_k = \frac{\tilde{\sigma}_{ef}^2(k)}{2\tilde{\sigma}_{xe}^2(k)}, \quad (83)$$

respectively. Fig. 5 shows the output SINRs of the SL-CLMS and SWL-CLMS algorithms with and without approximations in (27) and (56) for $Q = 2$. The SNR is fixed at 10 dB and 0 dB in Figs. 5(a) and (b), respectively. It is seen that the SL-CLMS and SWL-CLMS algorithms yield approximately the same

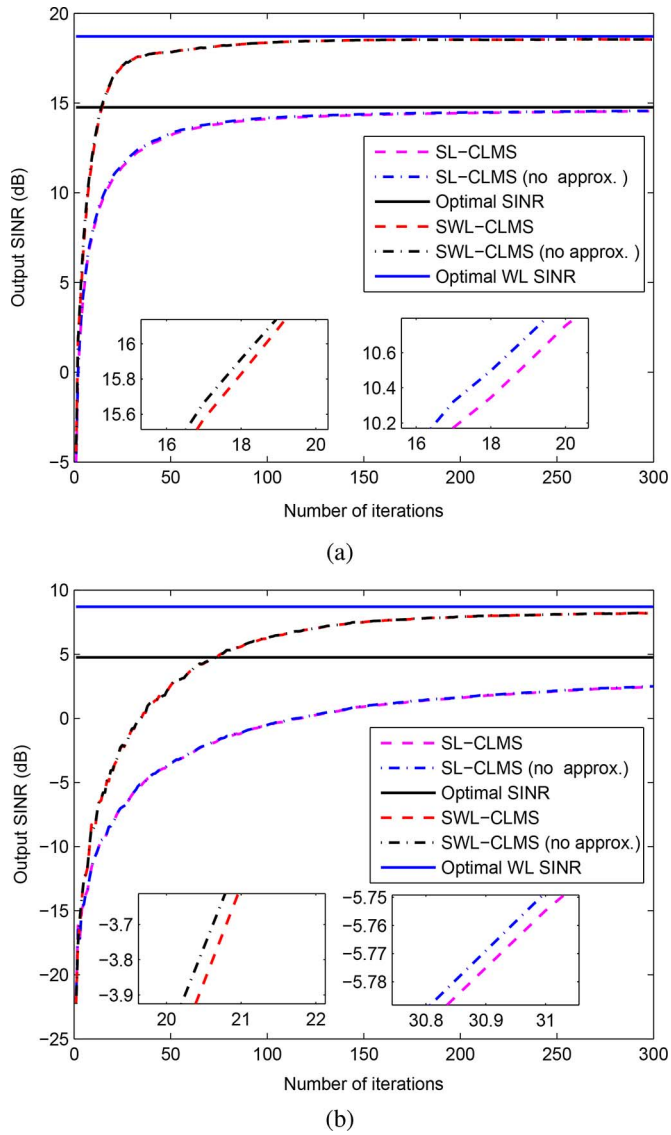


Fig. 5. Output SINRs of the SL-CLMS and SWL-CLMS algorithms with and without approximations in (27) and (56). (a) SNR = 10 dB. (b) SNR = 0 dB.

output SINRs no matter the approximations in (27) and (56) are adopted or not. That is because the noise-free *a priori* error $e_f(k)$ and the augmented noise-free *a priori* error $\tilde{e}_f(k)$ become small even prior to the steady-state and vary slightly compared to the input signal $\boldsymbol{x}(k)$. So the approximations in (27) and (56) have little influence on the performance of the proposed methods. Moreover, we can see that these approximations are still valid for different SNR settings.

B. Output MSE Performance Versus Number of Iterations

In this example, we examine the output MSEs of various approaches with the same parameter settings as those in Fig. 1. From Fig. 6, we observe that the SL-CLMS and SWL-CLMS algorithms converge faster than the CLMS and WL-CLMS schemes, respectively. Note that the SL-CLMS and CLMS algorithms have almost the same steady-state MSEs, and the SWL-CLMS and WL-CLMS schemes yield approximately the same steady-state MSEs. Furthermore, the WL-based solutions can obtain a much smaller MSE than the conventional

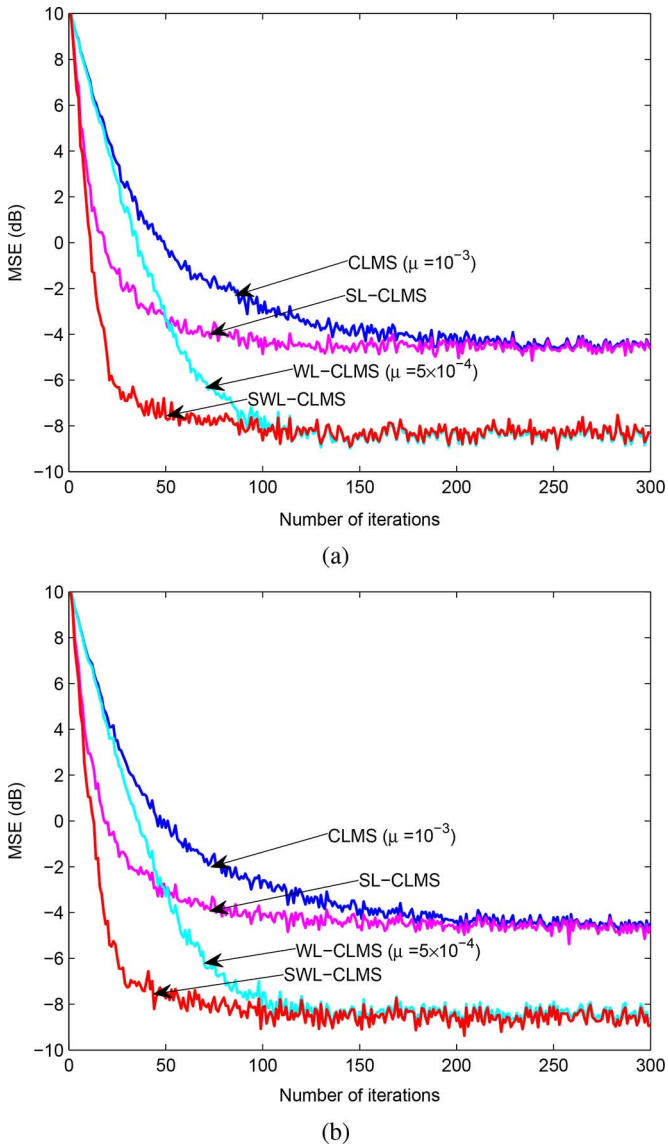


Fig. 6. Learning curves: MSEs of SL-CLMS, SWL-CLMS, CLMS and WL-CLMS algorithms. (a) $Q = 1$. (b) $Q = 2$.

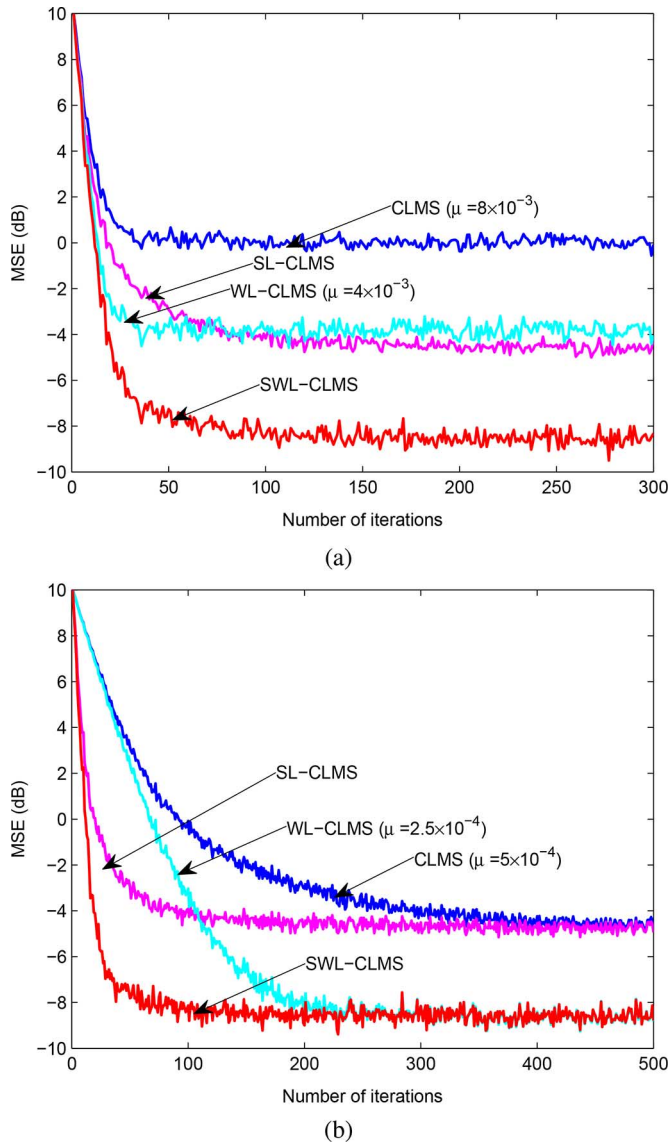


Fig. 7. Learning curves: MSEs of SL-CLMS, SWL-CLMS, CLMS and WL-CLMS algorithms for different μ.

algorithms since the NC properties have been taken into consideration. Because the augmented array aperture makes the SWL-CLMS algorithm more sensitive to the choice of α, we observe in Fig. 6(a) that the SWL-CLMS method is slightly inferior to the WL-CLMS algorithm in the steady-state at $Q = 1$. However, Fig. 6(b) shows that the SWL-CLMS algorithm slightly outperforms the WL-CLMS scheme at $Q = 2$. Moreover, unlike the WL-based algorithms, the value of Q has little influence on the performance of the SL-CLMS and CLMS approaches.

The learning curves of the SL-CLMS, SWL-CLMS, CLMS and WL-CLMS algorithms for different values of μ are shown in Fig. 7. Compared with Fig. 6(b), the CLMS and WL-CLMS schemes have much faster convergence speeds in Fig. 7(a), because their step sizes are fixed at 8×10^{-3} and 4×10^{-3}, respectively, which are eight times larger than their corresponding step sizes in Fig. 6(b). It is observed in Fig. 7(a) that the learning curves of the WL-CLMS and CLMS algorithms converge at -4 dB and 0 dB, respectively. However, with $\mu = 10^{-3}$ in Fig. 6(b), they converge at -8 dB and -4 dB, respectively. Fig. 7(b) shows the MSEs of the proposed methods versus the WL-CLMS and CLMS algorithms with $\mu = 2.5\times10^{-4}$ and $\mu = 5 \times 10^{-4}$, respectively. Although the CLMS and WL-CLMS algorithms can approach the same steady-state as the proposed methods, they need a larger number of iterations to reach the steady-state compared with the scenario in Fig. 6(b). That is, the CLMS and WL-CLMS require 200 and 400 iterations, respectively, while the SL-CLMS and SWL-CLMS only need 150 and 250 iterations, respectively. Consequently, the proposed algorithms outperform their counterparts no matter in convergence speed or MSE performance.

Fig. 8 shows the output MSEs of the SL-CLMS, SWL-CLMS, RLS and WL-RLS methods. It is observed that the RLS and WL-RLS algorithms need around 30 and 60 iterations, respectively, to arrive at the steady-state. On the other hand, the SL-CLMS and SWL-CLMS algorithms require about 120 and 100 iterations, respectively. Hence, the RLS and WL-RLS solutions yield faster convergence speeds than the

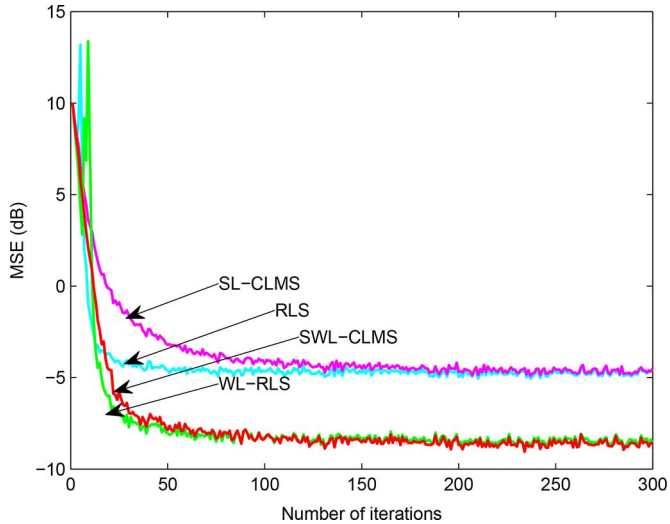


Fig. 8. Learning curves: MSEs of SL-CLMS, SWL-CLMS, RLS and WL-RLS algorithms.

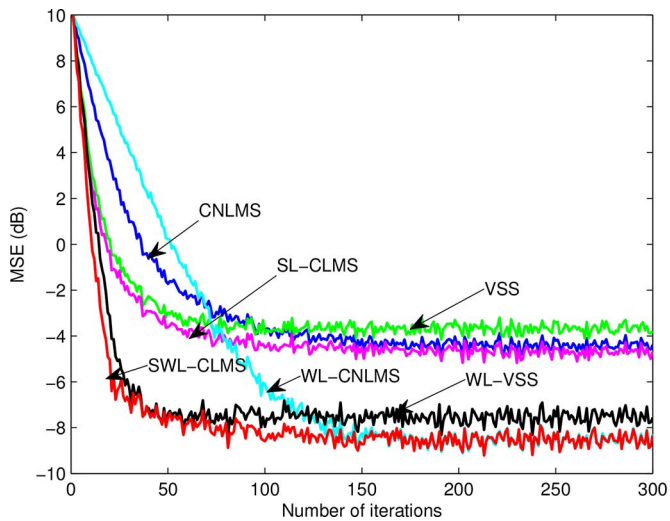


Fig. 9. Learning curves: MSEs of SL-CLMS, SWL-CLMS, CNLMS, WL-CNLMS, VSS and WL-VSS algorithms.

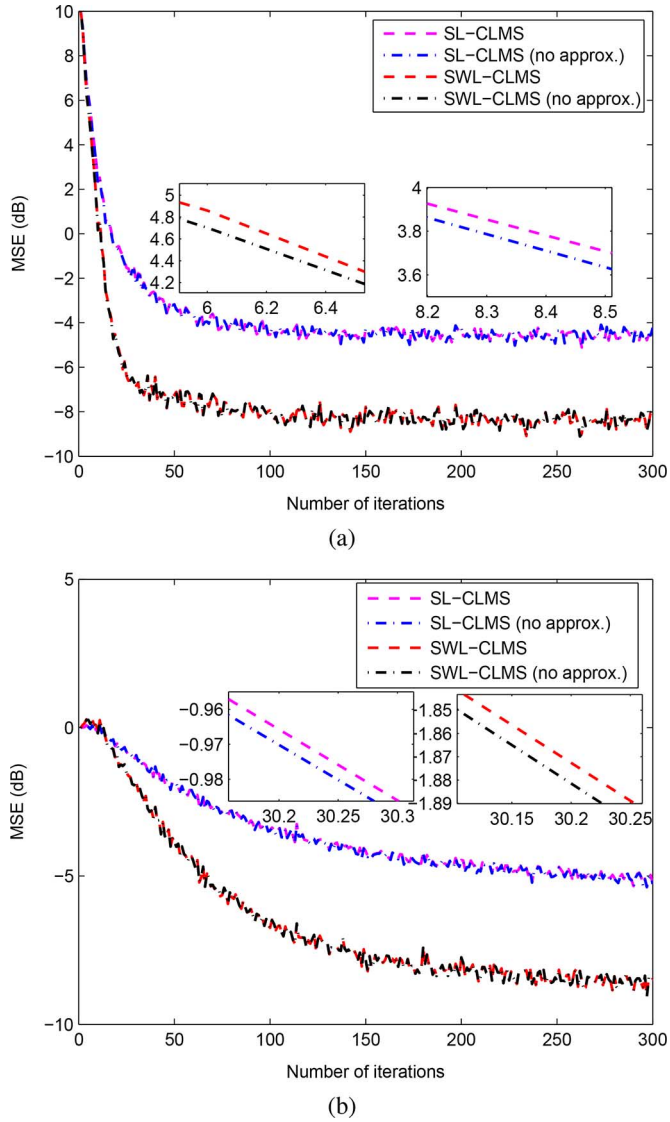


Fig. 10. Learning curves: MSEs of SL-CLMS and SWL-CLMS algorithms with and without approximations in (27) and (56). (a) $\mathrm{SNR} = 10$ dB. (b) $\mathrm{SNR} = 0$ dB.

SL-CLMS and SWL-CLMS algorithms. However, the faster convergence speeds of the former are achieved at the cost of higher computational load which will be demonstrated in Fig. 11. From Fig. 8, we can also see that the RLS and WL-RLS algorithms have larger steady-state misadjustments compared with the proposed approaches when the number of iterations is small. Furthermore, the SWL-CLMS and WL-RLS methods arrive at the same steady-state while the SL-CLMS and RLS algorithms have the same steady-state behavior. However, the SWL-CLMS and SL-CLMS can yield smaller output MSEs as they utilize the NC properties of the signals.

In Fig. 9, we examine the learning curves of the SL-CLMS and SWL-CLMS methods. For comparison, the results of the CNLMS, WL-CNLMS, VSS and WL-VSS algorithms are also presented. The parameter settings of the CNLMS, WL-CNLMS, VSS and WL-VSS algorithms are the same as those in Fig. 4. It can be seen that the SL-CLMS and SWL-CLMS methods arrive at the steady-state at about the 60th and 50th iterations, respectively. However, the CNLMS and WL-CNLMS algorithms need 100 and 150 iterations, respectively. Hence, the SL-CLMS and SWL-CLMS algorithms converge faster than the CNLMS and WL-CNLMS algorithms, respectively. Although the VSS and WL-VSS algorithms approximately have the same convergence speed as the SL-CLMS and SWL-CLMS methods, they yield larger steady-state misadjustments. We conclude that the proposed methods yield fast convergence speeds and small MSEs.

The output MSEs of the SL-CLMS and SWL-CLMS methods with and without approximations in (27) and (56) are displayed in Fig. 10. Figs. 10(a) and (b) correspond to the scenarios with $\mathrm{SNR} = 10$ dB and $\mathrm{SNR} = 0$ dB, respectively. The other parameter settings are the same as those in Fig. 5. We see that the approximations have little influence on the output MSEs of our proposed approaches. Furthermore, these approximations always hold for different SNR settings.

C. Computational Times Versus Number of Sensors

We now investigate the computational times of the SL-CLMS, SWL-CLMS, RLS and WL-RLS methods. It

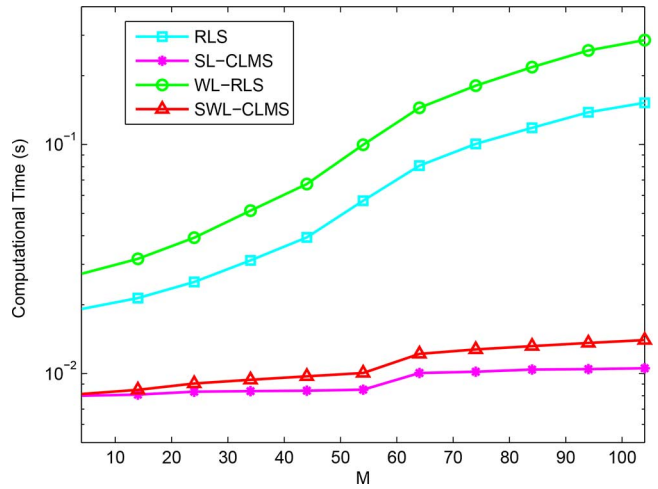


Fig. 11. Computational times of SL-CLMS, SWL-CLMS, RLS and WL-RLS algorithms versus number of sensors.

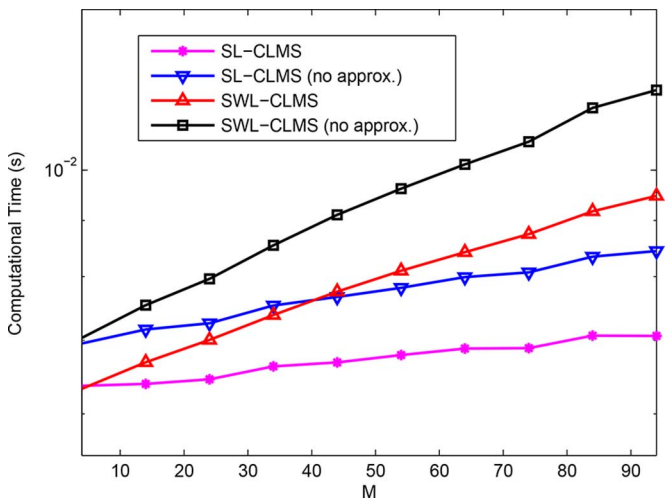


Fig. 12. Computational times of SL-CLMS and SWL-CLMS algorithms with and without approximations in (27) and (56) versus number of sensors.

is observed in Fig. 11 that the CLMS-based algorithms always have much lower computational costs than the RLS-based schemes especially when the number of sensors becomes large. The proposed SWL-CLMS algorithm is slightly more computationally expensive than the SL-CLMS method because the augmented signal structure increases the aperture of the array. Similarly, the WL-RLS algorithm has a higher computational load than the RLS approach.

Fig. 12 shows the computational times of the SL-CLMS and SWL-CLMS algorithms with and without approximations in (27) and (56). It is observed that the SL-CLMS has the lowest complexity among all the methods. The SL-CLMS algorithm without approximation requires more computational cost than its variant with approximation as the former needs to calculate $\|\boldsymbol{x}(k)\|^2$ per iteration. Similarly, the exact SWL-CLMS has a lower computational load than its variant without approximation. Moreover, the WL-based algorithms are more computationally demanding than the conventional methods.

V. Conclusion

The SL-CLMS and SWL-CLMS algorithms for adaptive beamforming have been devised in this paper. Apart from having faster convergence speeds of the weight vectors, the proposed solutions can also provide much larger output SINRs and smaller MSEs. Moreover, they always need lower computational complexities than the RLS-based algorithms. The step size of the SL-CLMS algorithm is adjusted according to the relationship between the noise-free *a posteriori* and *a priori* errors. As a result, the algorithm can adaptively adjust the step size to provide faster convergence rate and less misadjustment than the CLMS algorithm. On the other hand, besides adopting a variable step size determined by minimizing the square of the augmented noise-free *a posteriori* errors, the SWL-CLMS algorithm takes advantage of the NC properties of the SOI, eventually leading to considerable improvements in the output SINR for the adaptive beamformer.

References

[1] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *J. Acoust. Soc. Amer.*, vol. 42, no. 5, pp. 1175–1176, Dec. 2005.
[2] L. C. Godara, "Improved LMS algorithm for adaptive beamforming," *IEEE Trans. Antennas Progag.*, vol. 38, no. 10, pp. 1631–1635, Mar. 1990.
[3] S. Hossain, M. T. Islam, and S. Serikawa, "Adaptive beamforming algorithms for smart antenna systems," in *Proc. Int. Conf. Control, Autom., Syst. (ICCAS)*, Oct. 2008, pp. 412–416.
[4] F. G. Khodaei, J. Nourinia, and C. Ghobadi, "Adaptive beamforming algorithm with increased speed and improved reliability for smart antennas," *Comput. Electr. Eng.*, vol. 36, no. 6, pp. 1140–1146, Jun. 2010.
[5] M. Honig, U. Madhow, and S. Verdu, "Blind adaptive multiuser detection," *IEEE Trans. Inf. Theory*, vol. 41, no. 4, pp. 944–960, Feb. 1995.
[6] J. Treichler and B. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 2, pp. 459–472, June 1983.
[7] T. Adali, P. J. Schreier, and L. L. Scharf, "Complex-valued signal processing: The proper way to deal with impropriety," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5101–5125, Nov. 2011.
[8] B. Picinbono, "On circularity," *IEEE Trans. Signal Process.*, vol. 42, no. 12, pp. 3473–3482, Dec. 1994.
[9] P. Chevalier, "Optimal array processing for nonstationary signals," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 1996, pp. 2868–2871.
[10] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Trans. Signal Process.*, vol. 43, no. 8, pp. 2030–2033, Aug. 1995.
[11] P. Chevalier and A. Blin, "Widely linear MVDR beamformers for the reception of an unknown signal corrupted by noncircular interferences," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5323–5336, Nov. 2007.
[12] P. Chevalier, J. P. Delmas, and A. Oukaci, "Optimal widely linear MVDR beamforming for noncircular signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2009, pp. 3573–3576.
[13] P. Chevalier and F. Pipon, "New insights into optimal widely linear array receivers for the demodulation of BPSK, MSK and GMSK signals corrupted by noncircular interferences-application to SAIC," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 870–883, Mar. 2006.
[14] Q. J. You, Z. Jianyun, and Z. Xinan, "A widely-linear LMS algorithm for adaptive beamformer," in *Proc. IEEE Int. Symp. Microw., Antenna, Propag., EMC Technol. Wireless Commun.*, Apr. 2007, pp. 1060–1063.
[15] Y. R. Krishna, P. K. Prasad, P. V. Subbaiah, and B. P. Rao, "A performance analysis of CLMS and augmented CLMS algorithms for smart antennas," *Comput. Sci. Inf. Technol.*, pp. 9–19, Sept. 2012.
[16] D. P. Mandic, Y. Xia, and S. C. Douglas, "Steady state analysis of the CLMS and augmented CLMS algorithms for noncircular complex signals," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Aug. 2010, pp. 1635–1639.
[17] S. C. Douglas and D. P. Mandic, "Performance analysis of the conventional complex LMS and augmented complex LMS algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, June 2010, pp. 3794–3797.
[18] S. C. Douglas and D. P. Mandic, "Mean and mean-square analysis of the complex LMS algorithm for non-circular Gaussian signals," in *Proc. Digit. Signal Process. Workshop and 5th IEEE Signal Process. Educ. Workshop (DSP/SPE)*, Apr. 2009, pp. 101–106.

[19] J. I. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Autom. Control*, vol. 12, no. 3, pp. 282–287, June 1967.
[20] M. Tarrab and A. Feuer, "Convergence and performance analysis of the normalized LMS algorithm with uncorrelated Gaussian data," *IEEE Trans. Inf. Theory*, vol. 34, no. 4, pp. 680–691, July 1988.
[21] A. C. Hakkarainen, J. Werner, and M. Valkama, "RF imperfections in antenna arrays: Response analysis and widely-linear digital beamforming," in *Proc. IEEE Radio Wireless Symp. (RWS)*, Mar. 2013, pp. 187–189.
[22] E. Eweda, "Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels," *IEEE Trans. Signal Process.*, vol. 42, no. 11, pp. 2937–2944, Nov. 1994.
[23] S. C. Douglas, "Widely-linear recursive least-squares algorithm for adaptive beamforming," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Sept. 2009, pp. 2041–2044.
[24] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 7, pp. 1633–1642, Jul. 1992.
[25] F. J. A. Aquino, C. A. F. Rocha, and L. S. Resende, "Accelerating the convergence of the widely linear LMS algorithm for channel equalization," in *Proc. Int. Telecommun. Symp. (ITS)*, Sep. 2006, vol. 1, pp. 57–61.
[26] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-type algorithm: Analysis and simulations," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 631–639, Mar. 1997.
[27] J. J. Apolinrio, M. L. R. Campos, and P. S. R. Diniz, "Convergence analysis of the binormalized data-reusing LMS algorithm," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3235–3242, Nov. 2000.
[28] P. S. R. Diniz and S. Werner, "Set-membership binormalized data-reusing LMS algorithms," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 124–134, Mar. 2003.
[29] L. Wang, R. C. de Lamare, and Y. C. Cai, "Low-complexity adaptive step size constrained constant modulus SG algorithms for adaptive beamforming," *Signal Process.*, vol. 89, no. 12, pp. 2503–2513, Apr. 2009.
[30] I. Daubechies, M. Defrise, and C. De-Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, June 2004.
[31] M. Zibulevsky and M. Elad, "l_1-l_2 optimization in signal and image processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 76–88, May 2010.
[32] M. Z. A. Bhotto and A. Antoniou, "A family of shrinkage adaptive-filtering algorithms," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1689–1697, Apr. 2013.
[33] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151–1162, Aug. 1976.
[34] L. L. Horowitz and K. D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Trans. Circuits Syst. I*, vol. 28, no. 6, pp. 562–576, June 1981.
[35] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Optimum Array Processing*. New York, NY, USA: Wiley, 2003, pp. 60–64.

Yunmei Shi was born in Shandong, China. She received the B.Sc. degree in electronic and information engineering from Harbin Institute of Technology, Weihai, Weihai, China, in 2012, and the M.Sc. degree in electronic and communication engineering in 2014 from Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China, where she is currently pursuing the Ph.D. degree.

Her research interests include sensor array processing, statistical signal processing, convex optimization, and sparse signal representation.

Lei Huang (M'07–SM'14) was born in Guangdong, China. He received the B.Sc., M.Sc., and Ph.D. degrees in electronic engineering from Xidian University, Xi'an, China, in 2000, 2003, and 2005, respectively.

From 2005 to 2006, he was a Research Associate with the Department of Electrical and Computer Engineering, Duke University, Durham, NC. From 2009 to 2010, he was a Research Fellow with the Department of Electronic Engineering, City University of Hong Kong and a Research Associate with the Department of Electronic Engineering, The Chinese University of Hong Kong. Since 2011, he has joined the Department of Electronic and Information Engineering, Harbin Institute of Technology Shenzhen Graduate School, where he is currently a Professor. His research interests include spectral estimation, array signal processing, statistical signal processing, and their applications in radar and wireless communication systems.

Dr. Huang is currently is an editorial board member of Digital Signal Processing.

Cheng Qian was born in Zhejiang on November 27, 1988. He received the B.E. degree in communication engineering from Hangzhou Dianzi University, Hangzhou, China, in 2011, and the M.E. degree in information and communication engineering from Harbin Institute of Technology (HIT), Shenzhen, China, in 2013.

He is currently pursuing the Ph.D. degree in the field of information and communication engineering at HIT. His research interests are in array signal processing and MIMO radar.

Hing Cheung So (S'90–M'95–SM'07) was born in Hong Kong. He received the B.Eng. degree from the City University of Hong Kong and the Ph.D. degree from The Chinese University of Hong Kong, both in electronic engineering, in 1990 and 1995, respectively.

From 1990 to 1991, he was an Electronic Engineer with the Research and Development Division, Everex Systems Engineering Ltd., Hong Kong. During 1995–1996, he was a Postdoctoral Fellow with The Chinese University of Hong Kong. From 1996 to 1999, he was a Research Assistant Professor with the Department of Electronic Engineering, City University of Hong Kong, where he is currently an Associate Professor. His research interests include statistical signal processing, fast and adaptive algorithms, signal detection, parameter estimation, and source localization.

Dr. So was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING during 2010–2014. Currently, he is on the editorial boards of the IEEE SIGNAL PROCESSING MAGAZINE, *Signal Processing*, and *Digital Signal Processing*, as well as a member of the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society.