IEEE TRANSACTIONS ON

SIGNAL PROCESSING

A PUBLICATION OF THE IEEE SIGNAL PROCESSING SOCIETY

IEEE Signal Processing Society

www.signalprocessingsociety.org

Indexed in PubMed® and MEDLINE®, products of the United States National Library of Medicine

PubMed MEDLINE U.S. National Library of Medicine

JANUARY 2011 VOLUME 59 NUMBER 1 ITPRED (ISSN 1053-587X)

REGULAR PAPERS

(Contents Continued on Back Cover)

IEEE

(Contents Continued from Front Cover)

Subspace-Based Adaptive Method for Estimating Direction-of-Arrival With Luenberger Observer

Jingmin Xin, *Senior Member, IEEE*, Nanning Zheng, *Fellow, IEEE*, and Akira Sano, *Member, IEEE*

***Abstract*—In this paper, we propose a computationally simple and efficient subspace-based adaptive method for estimating directions-of-arrival (AMEND) for multiple coherent narrowband signals impinging on a uniform linear array (ULA), where the previously proposed QR-based method is modified for the number determination, a new recursive least-squares (RLS) algorithm is proposed for null space updating, and a dynamic model and the Luenberger state observer are employed to solve the estimate association of directions automatically. The statistical performance of the RLS algorithm in stationary environment is analyzed in the mean and mean-squares senses, and the mean-square-error (MSE) and mean-square derivation (MSD) learning curves are derived explicitly. Furthermore, an analytical study of the RLS algorithm is carried out to quantitatively compare the performance between the RLS and least-mean-square (LMS) algorithms in the steady-state. The theoretical analyses and effectiveness of the proposed RLS algorithm are substantiated through numerical examples.**

***Index Terms*—Adaptive filtering algorithm, direction-of-arrival (DOA) estimation, learning curve, Luenberger observer, state estimation, transient analysis.**

I. Introduction

SUBSPACE-BASED direction-of-arrival (DOA) estimation methods including the traditional methods with eigendecomposition and the computationally efficient methods without eigendecomposition have been well studied owing to their relative simplicity and high-resolution capability (see, e.g., [1], [2] and references therein). However, these methods usually fail or degenerate in some applications, where the directions of incident signals are time-varying with crossover points on their trajectories, and/or some incident signals appear and disappear sometimes (see, e.g., [40]). For recent three decades, the problem of estimating the DOAs of multiple moving incident signals has attracted much attention [3]–[15].

By making use of the stochastic dynamic models with several kinematic parameters such as the angular velocity and acceleration, some Kalman state estimation based algorithms were developed for the tracking of the crossing directions (e.g., [5], [10]–[13], [15], and [35]), where the crux is the estimate association, which indicates the correct association between the DOA estimates of different incident signals at two successive time instants (i.e., the so-called "data association" [4], [47]). In these algorithms, the directions are generally assumed to be slowly time-varying, while the additive process and measurement noises are assumed to be white Gaussian noise and the direction tracking problem is usually divided into two sequential procedures [13]. First, in the localization procedure, the initial estimates of the directions are obtained from the time-averaged statistics of array data during a limited interval by using a traditional batch method such as the computationally cumbersome maximum-likelihood (ML) methods [12], [13], [15], [22] or the subspace-based MUSIC with eigendecomposition [11], [33]. Then in the tracking procedure, these initial estimates are used as "measurements" of directions and refined by employing the Kalman filtering [23], [26]. By utilizing the recursive least-squares (RLS) based projection approximation subspace tracking with deflation (PASTd) [16] to accomplish the DOA estimation in the aforementioned localization procedure, where the computationally burdensome and time-consuming eigendecomposition is avoided, a relatively efficient subspace and state estimation based algorithm was proposed for DOA tracking of uncorrelated signals [17].

In fact, there are two important issues in state estimation based direction tracking: choice of dynamic direction model and computation of filter gain [5]. Unfortunately, a mathematical model representing the direction trajectory is usually not exact (even in the statistical sense) [5]. In attempt to compensate for modeling errors and measurement excursions, some simplified or sophisticated models for direction dynamics with different dimension are used in the literature (cf. [4], [13], and [14]). When the additive process and measurement noises in the stochastic dynamic model of direction have the Gaussian probability distribution, the Kalman filtering can provide the optimal and time-varying filter gain for state estimation [26], [27]. Nevertheless, the Kalman filtering depends on the statistical property of the additive process and measurement noises, which are usually unknown in direction tracking, and further these noises are hardly the strictly "white" noise [5]. Although several elaborate procedures (such as the asymptotic Cramér-Rao lower bound (CRB) of DOA estimation [22]) were suggested for approximate estimation of the noise variances (e.g., [4], [11], [15], [17], [20], [21], [35]), but they may increase the computational burden and result in performance degeneration because of the inherent sensitivity of Kalman filtering to the modeling errors [14].

Manuscript received March 13, 2010; accepted September 28, 2010. Date of publication October 07, 2010; date of current version December 17, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Maria Sabrina Greco. This work was supported in part by the National Natural Science Foundation of China by Grant 60772096, the State Key Development Program for Basic Research of China by Grant 2007CB311005, and the Doctoral Program Foundation of Institutions of Higher Education of China by Grant 20070698018.

J. Xin and N. Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: jxin@mail.xjtu.edu.cn; nnzheng@mail.xjtu.edu.cn).

A. Sano is with the Department of System Design Engineering, Keio University, Yokohama 223-8522, Japan (e-mail: sano@sd.keio.ac.jp).

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.2010.2084998

Furthermore in the algorithms aforementioned, the coherent (i.e., fully correlated) signals are not considered, and the number of incident signals is assumed to be fixed and known *a priori*. Inspired by the subspace-based method without eigendecomposition (SUMWE) for direction estimation of the coherent narrowband signals impinging on a uniform linear array (ULA) [2], we proposed an adaptive bearing estimation and tracking (ABEST) algorithm [18], where the least-mean-square (LMS) or normalized LMS (NLMS) algorithm is used to obtain the null space. Unfortunately, the tracking of the number of incident signals and that of crossing directions were not considered.

Therefore, the objective of this paper is to investigate the problem of tracking the crossing directions in a computationally efficient way, when the number of incident signals is unknown. On the bases of the previously proposed batch SUMWE [2] and the method for estimating the number of signals without eigendecomposition (MENSE) [34], a new subspace-based adaptive method for estimating DOAs (AMEND) is proposed for the coherent narrowband signals impinging on a ULA, where the number of signals is detected by using the QR-based MENSE, the null space is updated by using a RLS-based algorithm and is introduced into the approximate Newton iteration to obtain the "measurements" of directions in the localization procedure, and then the Luenberger observer [24], [25] with the three-dimensional deterministic dynamic direction model is employed to refine the estimated directions in the tracking procedure. In the proposed method, the Luenberger observer gain is designed through the pole placement by considering a compromise between the rapidity of the observer response and the sensitivity to the unmodelled disturbances and measurement noises in the dynamic direction model, and computation formula of the observer gain is clarified, while the estimation of noise variances required by the ordinary Kalman filtering and the estimate association of directions are avoided. Moreover we give a variation of the AMEND algorithm with "self-initialization" for the scenario, where some incident signals appear and disappear abruptly. By adopting the averaging principle (e.g., [26], [36], [37]), the statistical performance of the proposed RLS algorithm in stationary environment is analyzed in the mean and mean-squares senses, and the mean-square-error (MSE) and mean-square derivation (MSD) learning curves are derived explicitly. Furthermore an analytical study is carried out for the quantitative performance comparison between the RLS algorithm and the previously proposed LMS algorithm [18] in the steady state. Finally, the theoretical analyses and effectiveness of the proposed AMEND algorithm are verified and substantiated through numerical examples.

II. Problem Statement

A. Array Data Model and Assumptions

We consider a ULA of M identical and omnidirectional sensors with adjacent spacing d and assume that p narrowband signals $\{s_i(t)\}$ impinge on the array along distinct directions $\{\theta_i(t)\}$ from far-field. The received noisy signal $y_m(t)$ at the mth sensor can be expressed as

$$\begin{aligned} y_m(t) &= \sum_{i=1}^{p} s_i(t)\, e^{j\omega_0(m-1)\tau(\theta_i(t))} + w_m(t) \\ &= \boldsymbol{b}_m^T(\theta)\boldsymbol{s}(t) + w_m(t) \end{aligned} \tag{1}$$

where $\boldsymbol{b}_m(\theta) \triangleq [e^{j\omega_0(m-1)\tau(\theta_1(t))}, e^{j\omega_0(m-1)\tau(\theta_2(t))}, \ldots, e^{j\omega_0(m-1)\tau(\theta_p(t))}]^T$, $\boldsymbol{s}(t) \triangleq [s_1(t), s_2(t), \ldots, s_p(t)]^T$, $\omega_0 \triangleq 2\pi f_0$, $\tau(\theta_i(t)) \triangleq (d/c)\sin\theta_i(t)$, c and f_0 are the propagation speed and the center frequency, and $w_m(t)$ is the additive noise, while d satisfies $0 < d \leq \lambda/2$ for avoiding spatial aliasing, and λ is the carrier wavelength given by $\lambda = c/f_0$.

In this paper, we make the following basic assumptions on the data model.

1) The incident signals $\{s_i(t)\}$ are temporally complex white Gaussian random processes with zero-mean and variance given by $E\{s_i(n)s_i^*(t)\} = r_{s_i}\delta_{n,t}$ and $E\{s_i(n)s_i(t)\} = 0\,\forall n, t$, where $E\{\cdot\}$, $(\cdot)^*$, and $\delta_{n,t}$ denote the statistical expectation, the complex conjugate, and the Kronecker delta function which is equal to unity when $n = t$ and zero otherwise. The source signal covariance matrix $\boldsymbol{R}_s$ is defined by $\boldsymbol{R}_s \triangleq E\{\boldsymbol{s}(t)\boldsymbol{s}^H(t)\}$, where $(\cdot)^H$ denotes the Hermitian transpose, and the rank of $\boldsymbol{R}_s$ is given by $\mathrm{rank}(\boldsymbol{R}_s) = 1$ when the signals $\{s_i(t)\}$ are coherent.
2) The additive noises $\{w_m(t)\}$ are temporally and spatially complex white Gaussian random process with zero-mean and variance given by $E\{w_m(n)w_k^*(t)\} = \sigma^2\delta_{m,k}\delta_{n,t}$ and $E\{w_m(n)w_k(t)\} = 0\,\forall m, k$ and $\forall n, t$, and they are uncorrelated with the incident signals $\{s_i(t)\}$.
3) For tracking the time-varying directions, we assume that $\theta_i(t)$ is slowly varying (relative to the sampling rate $1/T_s$) so that $\theta_i(t) \approx \theta_i(nT)$ for $t \in (nT, (n+1)T]$ and $n = 0, 1, \ldots$ and that N_s snapshots of array data are available over an interval T of direction updating, i.e., $T = N_sT_s$ (e.g., [11], [17]).
4) The number of incident signals p satisfies the inequality that $p < M/2$ for an array of M sensors.

Then the array data model in (1) can be compactly reexpressed as

$$\boldsymbol{y}(k) = \boldsymbol{A}(\theta(n))\boldsymbol{s}(k) + \boldsymbol{w}(k) \tag{2}$$

for $k = nN_s + 1, nN_s + 2, \ldots, (n+1)N_s$ and $n = 0, 1, 2, \ldots$, where $\boldsymbol{y}(k) \triangleq [y_1(k), y_2(k), \ldots, y_M(k)]^T$, $\boldsymbol{w}(k) \triangleq [w_1(k), w_2(k), \ldots, w_M(k)]^T$, and $\boldsymbol{A}(\theta(n))$ is the array response matrix given by $\boldsymbol{A}(\theta(n)) \triangleq [\boldsymbol{a}(\theta_1(n)), \boldsymbol{a}(\theta_2(n)), \ldots, \boldsymbol{a}(\theta_p(n))]$ with $\boldsymbol{a}(\theta_i(n)) \triangleq [1, e^{j\omega_0\tau(\theta_i(n))}, \ldots, e^{j\omega_0(M-1)\tau(\theta_i(n))}]^T$. Hence the direction tracking is formulated as the adaptively estimating the directions $\{\theta_i(n)\}$ of the incident signals for $n = 0, 1, 2, \ldots$ from N_s snapshots of $\{\boldsymbol{y}(k)\}$ measured at $k = nN_s + 1, nN_s + 2, \ldots, (n+1)N_s$ while maintaining the correct association between the current estimate $\hat{\theta}_i(n)$ and the previous estimate $\hat{\theta}_i(n-1)$ of the same incident signal.

B. Deterministic Dynamic Model of Direction Trajectory

By letting the angular velocity and acceleration of the direction $\theta_i(n)$ at the instant n be $\dot{\theta}_i(n)$ and $\ddot{\theta}_i(n)$ and denoting the corresponding state vector as $\boldsymbol{x}_i(n) \triangleq [\theta_i(n), \dot{\theta}_i(n), \ddot{\theta}_i(n)]^T$, the slowly time-varying trajectory of direction $\theta_i(n)$ can be approximately expressed by a deterministic state model with constant acceleration in the absence of process and measurement noises as

$$\boldsymbol{x}_i(n+1) = \boldsymbol{F}\boldsymbol{x}_i(n) \tag{3}$$

and the direction $\theta_i(n)$ can be measured from the state vector $\boldsymbol{x}_i(n)$ by

$$\theta_i(n) = \boldsymbol{c}^T \boldsymbol{x}_i(n) \tag{4}$$

where $\boldsymbol{F}$ and $\boldsymbol{c}$ are the transition matrix and measurement vector given by [4], [11]

$$\boldsymbol{F} \triangleq \begin{bmatrix} 1 & T & 0.5T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

and $\boldsymbol{c} = [1,\ 0,\ 0]^T$.

In this paper, we focus our attention on the RLS-based subspace updating for direction estimation by using (2) and on the Luenberger state observer based tracking of directions with crossing by using (3) and (4). Furthermore the statistical analysis of proposed RLS algorithm will be studied in details.

III. RLS-Based Subspace Updating

Since the LMS algorithm is an approximation method based on the gradient descent technique, its convergence is relatively slow though it has the simplicity and computational efficiency, while the RLS algorithm offers an exact solution of the least-squares (LS) solution at each instant by using the exponentially weighted criterion and provides considerable improvement in convergence behavior and tracking capability [26], [27]. Here we investigate a new RLS-based algorithm for subspace updating, where the number of signals p is assumed to be known in this Section.

A. Principle of On-Line Direction Estimation

From (2), by dividing the array into L overlapping subarrays with p sensors forwards and backwards and defining the signal vectors of the lth forward/backward subarrays as $\boldsymbol{y}_{fl}(k) \triangleq [y_l(k), y_{l+1}(k), \ldots, y_{l+p-1}(k)]^T$ and $\boldsymbol{y}_{bl}(k) \triangleq [y_{M-l+1}(k), y_{M-l}(k), \ldots, y_{L-l+1}(k)]^H$, we can obtain four Hankel instantaneous correlation matrices at the instant k as [18]

$$\boldsymbol{\Phi}_f(k) = \boldsymbol{Y}_f(k) y_M^*(k),\ \bar{\boldsymbol{\Phi}}_f(k) = \bar{\boldsymbol{Y}}_f(k) y_1^*(k) \tag{6}$$

$$\boldsymbol{\Phi}_b(k) = \boldsymbol{Y}_b(k) y_1(k),\ \bar{\boldsymbol{\Phi}}_b(k) = \bar{\boldsymbol{Y}}_b(k) y_M(k) \tag{7}$$

where $\boldsymbol{Y}_f(k) \triangleq [\boldsymbol{y}_{f1}(k), \boldsymbol{y}_{f2}(k), \ldots, \boldsymbol{y}_{fL-1}(k)]^T, \bar{\boldsymbol{Y}}_f(k) \triangleq [\boldsymbol{y}_{f2}(k), \boldsymbol{y}_{f3}(k), \ldots, \boldsymbol{y}_{fL}(k)]^T, \boldsymbol{Y}_b(k) \triangleq [\boldsymbol{y}_{b1}(k), \boldsymbol{y}_{b2}(k), \ldots, \boldsymbol{y}_{bL-1}(k)]^T, \bar{\boldsymbol{Y}}_b(k) \triangleq [\boldsymbol{y}_{b2}(k), \boldsymbol{y}_{b3}(k), \ldots, \boldsymbol{y}_{bL}(k)]^T$, and $L = M - p + 1$. By dividing these $(M-p) \times p$ matrices in (6) and (7) into the $p \times p$ and $(M-2p) \times p$ submatrices in the downward direction

$$\boldsymbol{\Phi}_f(k) = \begin{bmatrix} \boldsymbol{\Phi}_{f1}(k) \\ \boldsymbol{\Phi}_{f2}(k) \end{bmatrix}, \quad \bar{\boldsymbol{\Phi}}_f(k) = \begin{bmatrix} \bar{\boldsymbol{\Phi}}_{f1}(k) \\ \bar{\boldsymbol{\Phi}}_{f2}(k) \end{bmatrix}$$

$$\boldsymbol{\Phi}_b(k) = \begin{bmatrix} \boldsymbol{\Phi}_{b1}(k) \\ \boldsymbol{\Phi}_{b2}(k) \end{bmatrix}, \quad \bar{\boldsymbol{\Phi}}_b(k) = \begin{bmatrix} \bar{\boldsymbol{\Phi}}_{b1}(k) \\ \bar{\boldsymbol{\Phi}}_{b2}(k) \end{bmatrix} \tag{8}$$

then we can obtain [18]

$$\boldsymbol{\Phi}_2(k) = \boldsymbol{P}^H \boldsymbol{\Phi}_1(k) + \boldsymbol{E}_o(k) \tag{9}$$

where $\boldsymbol{\Phi}_1(k) \triangleq [\boldsymbol{\Phi}_{f1}(k), \bar{\boldsymbol{\Phi}}_{f1}(k), \boldsymbol{\Phi}_{b1}(k), \bar{\boldsymbol{\Phi}}_{b1}(k)], \boldsymbol{\Phi}_2(k) \triangleq [\boldsymbol{\Phi}_{f2}(k), \bar{\boldsymbol{\Phi}}_{f2}(k), \boldsymbol{\Phi}_{b2}(k), \bar{\boldsymbol{\Phi}}_{b2}(k)]$, $\boldsymbol{E}_o(k) \triangleq -\boldsymbol{Q}^H \boldsymbol{G}(k)$, $\boldsymbol{G}(k) \triangleq [y_M^*(k)\boldsymbol{W}_f(k), y_1^*(k)\bar{\boldsymbol{W}}_f(k), y_1(k)\boldsymbol{W}_b(k), y_M(k) \cdot \bar{\boldsymbol{W}}_b(k)], \boldsymbol{W}_f(k) \triangleq [\boldsymbol{w}_{f1}(k), \boldsymbol{w}_{f2}(k), \ldots \boldsymbol{w}_{fL-1}(k)]^T$, $\bar{\boldsymbol{W}}_f(k) \triangleq [\boldsymbol{w}_{f2}(k), \boldsymbol{w}_{f3}(k), \ldots, \boldsymbol{w}_{fL}(k)]^T, \boldsymbol{W}_b(k) \triangleq [\boldsymbol{w}_{b1}(k), \boldsymbol{w}_{b2}(k), \ldots, \boldsymbol{w}_{bL-1}(k)]^T, \bar{\boldsymbol{W}}_b(k) \triangleq [\boldsymbol{w}_{b2}(k), \boldsymbol{w}_{b3}(k), \ldots, \boldsymbol{w}_{bL}(k)]^T, \boldsymbol{w}_{fl}(k) \triangleq [w_l(k), w_{l+1}(k), \ldots, w_{l+p-1}(k)]^T$, $\boldsymbol{w}_{bl}(k) \triangleq [w_{M-l+1}(k), w_{M-l}(k), \ldots, w_{L-l+1}(k)]^H$, and $\boldsymbol{P}$ is a $p \times (M-2p)$ linear operator (i.e., "weight" hereafter), which satisfies the following relation for direction estimation [18], [19]

$$\boldsymbol{Q}^H \bar{\boldsymbol{A}} = \boldsymbol{O}_{(M-2p)\times p} \tag{10}$$

where $\boldsymbol{Q} = [\boldsymbol{P}^T, -\boldsymbol{I}_{M-2p}]^T$, $\bar{\boldsymbol{A}}$ is the submatrix of $\boldsymbol{A}(\theta(k))$ consisting of its first $M-p$ rows, $\boldsymbol{O}_{m\times q}$ and $\boldsymbol{I}_m$ denote the $m \times q$ null matrix and $m \times m$ identity matrix. Clearly the columns of $\boldsymbol{Q}$ form a basis for the null space of $\bar{\boldsymbol{A}}^H$, and it implies that $\boldsymbol{\Pi}_Q \bar{\boldsymbol{a}}(\theta) = \boldsymbol{0}_{(M-p)\times 1}$ for $\theta = \theta_i(k)$, where $\boldsymbol{\Pi}_Q$ is the orthogonal projector onto this subspace, which is required for good performance of direction estimation (cf. [2], [16], [19], [43]) and given by $\boldsymbol{\Pi}_Q \triangleq \boldsymbol{Q}(\boldsymbol{Q}^H\boldsymbol{Q})^{-1}\boldsymbol{Q}^H$, and $\bar{\boldsymbol{a}}(\theta) = [1, e^{j\omega_0\tau(\theta)}, \ldots, e^{j\omega_0(L-2)\tau(\theta)}]^T$. As a result, the incident directions $\{\theta_i(k)\}$ can be estimated by minimizing the cost function [2]

$$f(\theta) = \bar{\boldsymbol{a}}^H(\theta) \boldsymbol{\Pi}_Q \bar{\boldsymbol{a}}(\theta). \tag{11}$$

Remark 1: Under the assumptions on data model, by taking expectation on the both sides of (9), we obtain

$$\boldsymbol{\Phi}_2 = \boldsymbol{P}^H \boldsymbol{\Phi}_1$$

where $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are the $p \times 4p$ and $(M-2p) \times 4p$ submatrices of the correlation matrix $\boldsymbol{\Phi}$ consisting of its first p and last $M-2p$ rows, respectively, $\boldsymbol{\Phi} \triangleq [\boldsymbol{\Phi}_f, \bar{\boldsymbol{\Phi}}_f, \boldsymbol{\Phi}_b, \bar{\boldsymbol{\Phi}}_b]$, $\boldsymbol{\Phi}_f = \bar{\boldsymbol{A}}\mathrm{diag}(\boldsymbol{R}_s \boldsymbol{b}_M^*(\theta))\boldsymbol{A}_1^T$, $\bar{\boldsymbol{\Phi}}_f = \bar{\boldsymbol{A}}\boldsymbol{D}\mathrm{diag}(\boldsymbol{R}_s \boldsymbol{b}_1^*(\theta))\boldsymbol{A}_1^T$, $\boldsymbol{\Phi}_b = \bar{\boldsymbol{A}}\boldsymbol{D}^{-(M-1)}(\mathrm{diag}(\boldsymbol{R}_s \boldsymbol{b}_1^*(\theta)))^* \boldsymbol{A}_1^T, \bar{\boldsymbol{\Phi}}_b = \bar{\boldsymbol{A}}\boldsymbol{D}^{-(M-2)}(\mathrm{diag}(\boldsymbol{R}_s \boldsymbol{b}_M^*(\theta)))^* \boldsymbol{A}_1^T$, $\bar{\boldsymbol{A}} \triangleq [\boldsymbol{b}_1(\theta), \boldsymbol{b}_2(\theta), \ldots, \boldsymbol{b}_{M-p}(\theta)]^T$, $\boldsymbol{A}_1 \triangleq [\boldsymbol{b}_1(\theta), \boldsymbol{b}_2(\theta), \ldots, \boldsymbol{b}_p(\theta)]^T$, $\boldsymbol{D} \triangleq \mathrm{diag}(\boldsymbol{b}_2(\theta))$, and $\mathrm{diag}(\cdot)$ denotes the diagonal operation which extracts the diagonal of a matrix as a vector or constructs a diagonal matrix with the elements of a vector. It is worthy to note that the ith element of vector $\boldsymbol{R}_s \boldsymbol{b}_m^*$ in above can be expressed as $(\boldsymbol{R}_s \boldsymbol{b}_m^*)_i = \sum_{l=1}^p r_{s_{il}} e^{-j\omega_0(m-1)\tau(\theta_l)} \neq 0$ in view of the facts that $r_{s_{il}} \neq 0$ for $i = 1, 2, \ldots, p$, where $r_{s_{il}}$ is the cross-correlation between the signals $s_i(k)$ and $s_l(k)$ defined by $r_{s_{il}} \triangleq E\{s_i(k) s_l^*(k)\}$. Clearly the ranks of two Vandermonde matrices $\bar{\boldsymbol{A}}$ and $\boldsymbol{A}_1$ are given by $\mathrm{rank}(\bar{\boldsymbol{A}}) = \min(M-p, p) = p$ and $\mathrm{rank}(\boldsymbol{A}_1) = \min(p, p) = p$ iff $p < M/2$, while the $p \times p$ diagonal matrices $\mathrm{diag}(\boldsymbol{R}_s \boldsymbol{b}_M^*(\theta))$, $\mathrm{diag}(\boldsymbol{R}_s \boldsymbol{b}_1^*(\theta))$, and $\boldsymbol{D}$ have full rank no matter if the incident signals are coherent or not. Hence we find that the ranks (i.e., the dimension of signal subspace) of these $(M-p) \times p$ matrices $\boldsymbol{\Phi}_f$, $\bar{\boldsymbol{\Phi}}_f$, $\boldsymbol{\Phi}_b$, and $\bar{\boldsymbol{\Phi}}_b$ equal the number of signals irrespective of the statistical correlation between the incident signals, and correspondingly the cost function in (11) and the following AMEND algorithm are still applicable when there exists uncorrelated incident signal(s). □

B. RLS Algorithm for Null Space Updating

From (9), by introducing an $(M-2p)\times 4p$ estimation error $\bar{\boldsymbol{E}}(i)$ as

$$\bar{\boldsymbol{E}}(i) \triangleq \boldsymbol{\Phi}_2(i) - \boldsymbol{P}^H(k)\boldsymbol{\Phi}_1(i) \tag{12}$$

for $1 \le i \le k$, we can define a cost function $J_{\rm RLS}(k)$ for estimating the instantaneous linear operator $\boldsymbol{P}(k)$ as

$$J_{\rm RLS}(k) \triangleq \sum_{i=1}^{k} \gamma^{k-i} \left\|\bar{\boldsymbol{E}}(i)\right\|_F^2 + \gamma^k {\rm tr}\{\boldsymbol{P}^H(k)\boldsymbol{\Psi}_o\boldsymbol{P}(k)\} \tag{13}$$

where γ is the forgetting factor given by $0 \ll \gamma \le 1$, which is used to exponentially weight the influence of past data less heavily than that of more recent data, $\boldsymbol{\Psi}_o$ is a $p\times p$ positive-definite regularization matrix to stabilize the estimation at each instant k, which can be interpreted as a soft constrained initialization in the LS criterion, while $\|\cdot\|_F^2$ and ${\rm tr}\{\cdot\}$ denote the square of the Frobenius norm and the trace operation. By letting the instantaneous gradient matrix of $J_{\rm RLS}(k)$ in (13) with respect to (w.r.t.) the liner operator $\boldsymbol{P}(k)$ be zero, i.e., $\nabla J_{\rm RLS}(k) = 2\partial J_{\rm RLS}(k)/\partial \boldsymbol{P}^*(k) = \boldsymbol{O}_{p\times(M-2p)}$, we can get the regularized LS estimate of $\boldsymbol{P}(k)$ at the instant k with the available received array data $\{\boldsymbol{y}(i)\}_{i=1}^k$ as

$$\hat{\boldsymbol{P}}(k) = \boldsymbol{\Psi}_1^{-1}(k)\boldsymbol{\Psi}_2(k) \tag{14}$$

where the sample exponentially weighted auto-correlation and cross-correlation matrices $\boldsymbol{\Psi}_1(k)$ and $\boldsymbol{\Psi}_2(k)$ are given by

$$\begin{aligned}\boldsymbol{\Psi}_1(k) &\triangleq \sum_{i=1}^{k}\gamma^{k-i}\boldsymbol{\Phi}_1(i)\boldsymbol{\Phi}_1^H(i) + \gamma^k\boldsymbol{\Psi}_o \\ &= \gamma\boldsymbol{\Psi}_1(k-1) + \boldsymbol{\Phi}_1(k)\boldsymbol{\Phi}_1^H(k) \qquad (15)\\ \boldsymbol{\Psi}_2(k) &\triangleq \sum_{i=1}^{k}\gamma^{k-i}\boldsymbol{\Phi}_1(i)\boldsymbol{\Phi}_2^H(i) \\ &= \gamma\boldsymbol{\Psi}_2(k-1) + \boldsymbol{\Phi}_1(k)\boldsymbol{\Phi}_2^H(k). \qquad (16)\end{aligned}$$

By taking use of the well-known matrix inversion lemma (e.g., [26] and [27]) to $\boldsymbol{\Psi}_1^{-1}(k)$ of $\boldsymbol{\Psi}_1(k)$ in (15), we obtain

$$\begin{aligned}\tilde{\boldsymbol{\Psi}}_1(k) &\triangleq \boldsymbol{\Psi}_1^{-1}(k) \\ &= \gamma^{-1}(\boldsymbol{I}_p - \bar{\boldsymbol{G}}(k)\boldsymbol{\Phi}_1^H(k))\tilde{\boldsymbol{\Psi}}_1(k-1) \qquad (17)\end{aligned}$$

where $\bar{\boldsymbol{G}}(k)$ is referred as the $p\times 4p$ gain matrix given by

$$\begin{aligned}\bar{\boldsymbol{G}}(k) &\triangleq \gamma^{-1}\tilde{\boldsymbol{\Psi}}_1(k-1)\boldsymbol{\Phi}_1(k) \\ &\quad\cdot(\boldsymbol{I}_{4p} + \gamma^{-1}\boldsymbol{\Phi}_1^H(k)\tilde{\boldsymbol{\Psi}}_1(k-1)\boldsymbol{\Phi}_1(k))^{-1} \\ &= \tilde{\boldsymbol{\Psi}}_1(k)\boldsymbol{\Phi}_1(k). \qquad (18)\end{aligned}$$

Then by substituting the recursions (15) and (16) into (14) with (17) and (18), the LS estimate $\hat{\boldsymbol{P}}(k)$ can be rewritten in recursive form as

$$\begin{aligned}\hat{\boldsymbol{P}}(k) &= \gamma\tilde{\boldsymbol{\Psi}}_1(k)\boldsymbol{\Psi}_2(k-1) + \bar{\boldsymbol{G}}(k)\boldsymbol{\Psi}_2(k) \\ &= (\boldsymbol{I}_{4p} - \bar{\boldsymbol{G}}(k)\boldsymbol{\Phi}_1^H(k))\hat{\boldsymbol{P}}(k-1) + \bar{\boldsymbol{G}}(k)\boldsymbol{\Phi}_2^H(k) \\ &= \hat{\boldsymbol{P}}(k-1) + \bar{\boldsymbol{G}}(k)\boldsymbol{E}^H(k) \qquad (19)\end{aligned}$$

TABLE I
SUMMARY OF RLS ALGORITHM FOR NULL SPACE ESTIMATION AND COMPUTATIONAL COMPLEXITY IN MATLAB FLOPS

Initialization:

$$\hat{\boldsymbol{P}}(0) = \boldsymbol{O}_{p\times(M-2p)} \quad \text{and} \quad \boldsymbol{\Psi}_1(0) = \boldsymbol{\Psi}_o$$

Step 1: Updating of matrix $\boldsymbol{\Psi}_1(k)$ as

$$\boldsymbol{\Psi}_1(k) = \gamma\boldsymbol{\Psi}_1(k-1) + \boldsymbol{\Phi}_1(k)\boldsymbol{\Phi}_1^H(k)$$

$\ldots\ldots 32p^3 + 6p^2$ flops

Step 2: Performing QR decomposition of $\boldsymbol{\Psi}_1(k)$ as

$$\boldsymbol{\Psi}_1(k) = \tilde{\boldsymbol{Q}}(k)\tilde{\boldsymbol{R}}(k)$$

$\ldots\ldots 24p^3 + 4p^2 + 22p - 50$ flops

Step 3: Updating of gain matrix $\bar{\boldsymbol{G}}(k)$ as

$$\bar{\boldsymbol{G}}(k) = {\rm inv}\{\tilde{\boldsymbol{R}}(k)\}\tilde{\boldsymbol{Q}}^H(k)\boldsymbol{\Phi}_1(k)$$

$\ldots\ldots 40p^3 + p(p+1)(p+2)$ flops

Step 4: Filtering of estimation error $\boldsymbol{E}(k)$ as

$$\boldsymbol{E}(k) = \boldsymbol{\Phi}_2(k) - \hat{\boldsymbol{P}}^H(k-1)\boldsymbol{\Phi}_1(k)$$

$\ldots\ldots 32p^2(M-2p) + 8p(M-2p)$ flops

Step 5: Adaptation of "weight" $\hat{\boldsymbol{P}}(k)$ as

$$\hat{\boldsymbol{P}}(k) = \hat{\boldsymbol{P}}(k-1) + \bar{\boldsymbol{G}}(k)\boldsymbol{E}^H(k)$$

$\ldots\ldots 32p^2(M-2p) + 2p(M-2p)$ flops

where $\boldsymbol{E}(k)$ is *a priori* estimation error defined by

$$\boldsymbol{E}(k) \triangleq \boldsymbol{\Phi}_2(k) - \hat{\boldsymbol{P}}^H(k-1)\boldsymbol{\Phi}_1(k) \tag{20}$$

which is "tentative" value of *a posteriori* estimation error $\bar{\boldsymbol{E}}(k) = \boldsymbol{\Phi}_2(k) - \hat{\boldsymbol{P}}^H(k)\boldsymbol{\Phi}_1(k)$ before updating, where $\bar{\boldsymbol{E}}(k)$ is equal to $\bar{\boldsymbol{E}}(i)$ in (12) for $i = k$, and $\bar{\boldsymbol{E}}(k) = \boldsymbol{E}(k)(\boldsymbol{I}_{4p} - \bar{\boldsymbol{G}}^H(k)\boldsymbol{\Phi}_1(k))$. Herein we make an assumption that the current weight matrix $\hat{\boldsymbol{P}}(k-1)$ is statistically independent of the current correlation matrices $\boldsymbol{\Phi}_1(k)$ and $\boldsymbol{\Phi}_2(k)$ as usually assumed in the adaptive filtering literature (cf. [18], [26], [27], and references therein).

Thus by performing the QR decomposition on $\boldsymbol{\Psi}_1(k)$ in (15) to alleviate the computational complexity in the calculation of gain matrix in (18) due to the matrix inversion, the RLS algorithm for updating liner operator $\hat{\boldsymbol{P}}(k)$ and the computational complexity in MATLAB flops can be summarized in Table I, where $\tilde{\boldsymbol{Q}}(k)$ and $\tilde{\boldsymbol{R}}(k)$ are the $p\times p$ unitary matrix and upper-triangular matrix, ${\rm inv}\{\cdot\}$ denotes the inversion operation of an upper-triangular matrix with back-substitution (cf. [18]), and a flop is defined as a floating-point addition or multiplication operation as adopted by MATLAB software.

Remark 2: The forgetting factor γ should be chosen with a compromise in terms of convergence rate, tracking, misadjustment and stability, and $L_{\rm eff} \triangleq 1/(1-\gamma)$ is referred to as the effective window length, which measures the memory of the algorithm, and γ is usually chosen close to one for achieving good tracking performance and reducing the sensitivity to additive noise (e.g., [27], [28], [31]). □

Remark 3: Choosing $\bar{\boldsymbol{\Psi}}_o$ has the effect of weighting the initial matrix $\hat{\boldsymbol{P}}(0)$, where larger value of $\boldsymbol{\Psi}_o$ implies more confidence in $\hat{\boldsymbol{P}}(0)$ (i.e., forcing $\hat{\boldsymbol{P}}(k)$ closer to $\boldsymbol{O}_{p\times(M-2p)}$) and slows down the convergence of algorithm [32]. □

As a result of the above RLS algorithm, we obtain the instantaneous orthogonal projector $\overline{\boldsymbol{\Pi}}(k)$ at the instant k of $\overline{\boldsymbol{\Pi}}_Q$ in (11)

$$\boldsymbol{\Pi}(k) = \boldsymbol{Q}(k)(\boldsymbol{I}_{M-2p} - \hat{\boldsymbol{P}}^H(k)\mathrm{inv}\{\bar{\boldsymbol{R}}\}\bar{\boldsymbol{Q}}^H\hat{\boldsymbol{P}}(k))\boldsymbol{Q}^H(k) \quad (21)$$

where $\bar{\boldsymbol{Q}}$ and $\bar{\boldsymbol{R}}$ are the QR decomposition factors of $\hat{\boldsymbol{P}}(k)\hat{\boldsymbol{P}}^H(k) + \boldsymbol{I}_p$. Hence from (11), we can update the directions with the approximate Newton iteration [18]

$$\hat{\theta}_i(k) = \hat{\theta}_i(k-1) - \left.\frac{\mathrm{Re}\{\bar{\boldsymbol{d}}^H(\theta)\boldsymbol{\Pi}(k)\bar{\boldsymbol{a}}(\theta)\}}{\bar{\boldsymbol{d}}^H(\theta)\boldsymbol{\Pi}(k)\bar{\boldsymbol{d}}(\theta)}\right|_{\theta=\hat{\theta}_i(k-1)} \quad (22)$$

where $\mathrm{Re}\{\cdot\}$ denotes the real part of the quantity, and $\bar{\boldsymbol{d}}(\theta) = j\omega_0(d/c)\cos\theta[0, e^{j\omega_0\tau(\theta)}, \ldots, (L-2)e^{j\omega_0(L-2)\tau(\theta)}]^T$.

Remark 4: The total number of MATLAB flops required by the proposed RLS algorithm shown in Table I is roughly $64(M-2p)p^2+10(M-2p)p+97p^3+13p^2+24p-50$. Clearly the RLS algorithm needs $97p^3 - 11p^2 + 24p - 50$ more MATLAB flops than the LMS algorithm proposed in [18], and the computational complexity of the RLS algorithm is approximately equivalent to that of the NLMS algorithm [18]. □

IV. Direction Tracking Algorithm

In general, a key issue in direction tracking of multiple signals is the estimate association, which implies the correct association of estimated DOAs of different signals at two successive time instants (e.g., [3]), and some Kalman state estimation based parametric tracking methods were proposed for the directions with crossings [10]–[13], [15], [17], [35]. In these methods, the preliminary state estimates including the estimated DOAs (i.e., "measurements") are refined by Kalman filtering, which incorporates dynamic models of parameters of interest such as the angular velocity and acceleration to automatically paired with the estimated DOAs at different time instants and to avoid the tremendous computational burden of traditional estimate association (cf. [3], [4]). However the Kalman filtering involves the estimation of the variances of additive processes noise and measurement noise, which are usually unknown in direction tracking.

In this paper, we propose a new state estimation based direction tracking algorithm by introducing a deterministic dynamic state model of the direction trajectory and utilizing the Luenberger observer [24], [25], where the state vector consisting of the direction, angular velocity and acceleration of each signal is predicted from the existing estimated state vector by using the Luenberger observer, and the measurement of direction is obtained from the most recent updated subspace and the predicted direction by using the approximate Newton iteration. Then the predicted state vector (including the predicted DOA) is refined by the Luenberger observer with the estimated measurement of DOA, where the computationally burdensome estimate association is avoided. Similarly to the Kalman filtering in DOA tracking, the Luenberger observer has a double role: filtering the measurements and providing the predicted directions to the Newton iteration, and the association between the estimated DOAs at two successive instants of direction updating is embedded in the DOA estimation with the Luenberger observer, however the estimation of the variances of additive process and measurement noises required by the Kalman filtering is eliminated.

A. Luenberger Observer Based State Estimation

Obviously some of the state variables of $\boldsymbol{x}(n)$ in (3) are not accessible, and it is necessary to estimate them for direction tracking. Since we easily find that the dynamical system of direction trajectory $(\boldsymbol{F}, \boldsymbol{c}^T)$ in (3) and (4) is observable (cf. [29] and [30]), by using the measurement $\theta_i(n)$, the Luenberger state observer (i.e., prediction estimator) of this dynamic system is obtained

$$\hat{\boldsymbol{x}}_i(n+1) = \boldsymbol{F}\hat{\boldsymbol{x}}_i(n) + \boldsymbol{g}_i(\theta_i(n) - \boldsymbol{c}^T\hat{\boldsymbol{x}}_i(n)) \quad (23)$$

$$\hat{\theta}_i(n) = \boldsymbol{c}^T\hat{\boldsymbol{x}}_i(n) \quad (24)$$

where $\boldsymbol{g}_i$ is called the observer gain, and the difference between the measured direction $\theta_i(n)$ and the estimated one $\boldsymbol{c}^T\hat{\boldsymbol{x}}_i(n)$ (i.e., the correction term) in (23) is used to reduce the discrepancies between the deterministic dynamic model in (3) and the actual model of the direction trajectory. By defining the state estimation error as $\boldsymbol{e}_i(n) \triangleq \boldsymbol{x}_i(n) - \hat{\boldsymbol{x}}_i(n)$, from (23) and (3), we can get

$$\boldsymbol{e}_i(n+1) = (\boldsymbol{F} - \boldsymbol{g}_i\boldsymbol{c}^T)\boldsymbol{e}_i(n). \quad (25)$$

Clearly if and only if the magnitudes of all eigenvalues of the matrix $\boldsymbol{F} - \boldsymbol{g}_i\boldsymbol{c}^T$ are strictly less than one, the observer in (23) and (24) is asymptotically stable, i.e., the estimation error $\boldsymbol{e}_i(n)$ will converge to zero for any initial value of $\boldsymbol{x}_i(0)$.

Hence the observer gain $\boldsymbol{g}_i$ serves as a weighting vector and must be designed so that the observer is stable and the estimation error is acceptably small even though the initial estimate $\hat{\boldsymbol{x}}_i(0)$ is not equal to the actual state $\boldsymbol{x}_i(0)$. The design procedure of observer gains $\{\boldsymbol{g}_i\}$ is first to choose the desired observer poles (i.e., the eigenvalues of $\boldsymbol{F} - \boldsymbol{g}_i\boldsymbol{c}^T$) and then to decide the gains $\{\boldsymbol{g}_i\}$ so that they will give the desired poles, where the desired poles should be selected as one pair of conjugate complex poles and one real pole within the unite circle in the z-plane (cf. [29], [30] for the pole placement or assignment technique in control engineering) (see Appendix for the analytical computation).

B. On-Line Implementation of Algorithm

Therefore by utilizing the MENSE detection criterion [34], [39] and the RLS-based subspace updating and the Luenberger observer based state estimation studied above, as shown in Fig. 1, we can summarize the implementation of the AMEND for tracking the directions $\{\theta_i(n)\}$ at the instant n with the N_s snapshots $\{\boldsymbol{y}(k)\}$ measured at $k = nN_s + 1, nN_s + 2, \ldots, (n+1)N_s$ for $n = 0, 1, 2, \ldots$ as follows:

1) **(Initialization-1)** Set the maximum detectable number of signals to $\bar{p} = \lfloor M/2 \rfloor$, and design the gains $\{\boldsymbol{g}_i\}$ of $\bar{p}$ observers through the pole placement, where $\lfloor x \rfloor$ denotes the largest integer not greater than x. ..... $1 + 61\bar{p}$ flops
2) **(Initialization-2)** For the instant index of direction updating $n = 0$, by setting the subarray size to $\bar{p}$, estimate the number of incident signals from the N_s snapshots of

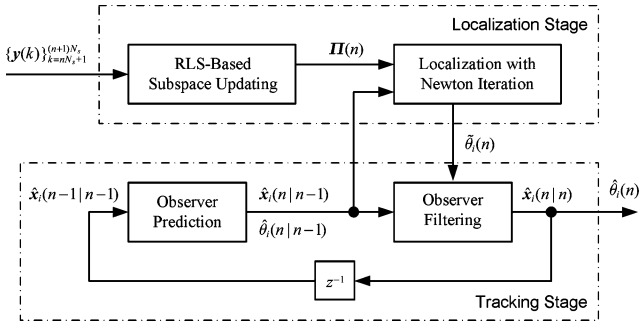

Fig. 1. The simplified flowchart of the proposed AMEND algorithm for direction tracking with Luenberger observer.

$\{\boldsymbol{y}(k)\}_{k=nN_s+1}^{(n+1)N_s}$ with the batch MENSE [34] and denote it by $\hat{p}$.
..... $4(4N_s+3)M+14(M-\bar{p})^3+32(M-\bar{p})^2\bar{p}$ $-3.5(M-\bar{p})^2+2(M-\bar{p})\bar{p}+26.5(M-\bar{p})-4$ flops

3) **(Initialization-3)** By setting the subarray size to $\hat{p}$, estimate the initial values of directions from the N_s snapshots of $\{\boldsymbol{y}(k)\}_{k=nN_s+1}^{(n+1)N_s}$ with the batch SUMWE [2] and denote them by $\hat{\theta}_i(0|0)$ for $i=1,2,\ldots,\hat{p}$.
..... $4(4N_s+3)M+8M(M-2\hat{p})^2+8(M-\hat{p})^2$ $\cdot(M-2\hat{p})+56(M-2\hat{p})\hat{p}^2+32\hat{p}^3+2(M-\hat{p}-1)^2+$ $O(\hat{p}^3)+O(2(M-\hat{p})-1)$ flops

4) **(Initialization-4)** Initialize the Luenberger observer and the RLS algorithm by $\hat{\boldsymbol{x}}(0|0)=[\hat{\theta}_i(0|0),\ \hat{\theta}_i(0|0)/N_s,0]^T$, $\hat{\boldsymbol{P}}(0)=\boldsymbol{O}_{\hat{p}\times(M-2\hat{p})}$ and $\boldsymbol{\Psi}_1(0)=\Psi_0\boldsymbol{I}_{\hat{p}\times\hat{p}}$, and update the instant index n to $n=n+1$. ..... 1 flops

5) **(Observer Prediction)** Calculate the predicted state vector $\hat{\boldsymbol{x}}_i(n|n-1)$ and predicted direction $\hat{\theta}_i(n|n-1)$ from the existing estimate of state vector $\hat{\boldsymbol{x}}_i(n-1|n-1)$ with the Luenberger observer in (3) and (4) as

$$\hat{\boldsymbol{x}}_i(n|n-1)=\boldsymbol{F}\hat{\boldsymbol{x}}_i(n-1|n-1) \tag{26}$$
$$\hat{\theta}_i(n|n-1)=\boldsymbol{c}^T\hat{\boldsymbol{x}}_i(n|n-1). \tag{27}$$

..... 18 flops

6) **(Subspace Updating-1)** During the interval of direction updating $(nT,(n+1)T]$ (i.e., $k=nN_s+1,nN_s+2,\ldots,(n+1)N_s$), calculate the instantaneous correlation vectors $\boldsymbol{\varphi}(k)$ between $\boldsymbol{y}(k)$ and $y_M^*(k)$ and $\bar{\boldsymbol{\varphi}}(k)$ between $\boldsymbol{y}(k)$ and $y_1^*(k)$ at the index of sampling instant k

$$\boldsymbol{\varphi}(k)=\boldsymbol{y}(k)y_M^*(k),\ \bar{\boldsymbol{\varphi}}(k)=\boldsymbol{y}(k)y_1^*(k) \tag{28}$$

and form the instantaneous Hankel matrices $\boldsymbol{\Phi}_f(k)$, $\bar{\boldsymbol{\Phi}}_f(k)$, $\boldsymbol{\Phi}_b(k)$, and $\bar{\boldsymbol{\Phi}}_b(k)$ by using (8) and (28), where p is replaced with $\hat{p}$. ..... $16M+2(M-\hat{p})\hat{p}+2$ flops

7) **(Subspace Updating-2)** Update the weight $\hat{\boldsymbol{P}}(k)$ by using the proposed RLS algorithm shown in Table I.
..... $64(M-2\hat{p})\hat{p}^2+10(M-2\hat{p})\hat{p}$ $+97\hat{p}^3+13\hat{p}^2+24\hat{p}-50$ flops

8) Update the index of sampling instant as $k=k+1$, and go to the next step if $k=(n+1)N_s$, otherwise return to Step 6).

9) **(Subspace Updating-3)** Calculate the instantaneous projector $\bar{\boldsymbol{\Pi}}(k)$ with (21) and denote it by $\bar{\boldsymbol{\Pi}}(n)$. . $8M(M-2\hat{p})^2+8(M-2\hat{p})^3+8(M-2\hat{p})\hat{p}^2+9\hat{p}^3+3\hat{p}^2+2\hat{p}$ flops

10) **(Localization)** Calculate the "measurement" of direction $\tilde{\theta}_i(n)$ by using the projector $\boldsymbol{\Pi}(n)$ and the predicted direction $\hat{\theta}_i(n|n-1)$ with (22) as

$$\tilde{\theta}_i(n)=\hat{\theta}_i(n|n-1)-\left.\frac{\mathrm{Re}\{\bar{\boldsymbol{d}}^H(\theta)\boldsymbol{\Pi}(n)\bar{\boldsymbol{a}}(\theta)\}}{\bar{\boldsymbol{d}}^H(\theta)\boldsymbol{\Pi}(n)\bar{\boldsymbol{d}}(\theta)}\right|_{\theta=\hat{\theta}_i(n|n-1)}. \tag{29}$$

..... $16(M-\hat{p})^2+16(M-\hat{p})+1$ flops

11) **(Observer Filtering)** Calculate the refined state vector $\hat{\boldsymbol{x}}_i(n|n)$ and estimate the directions $\hat{\theta}_i(n)$ by using the predicted state vector $\hat{\boldsymbol{x}}_i(n|n-1)$, the predicted direction $\hat{\theta}_i(n|n-1)$ and the "measurement" of direction $\tilde{\theta}_i(n)$ with the observer in (23) and (24) as

$$\hat{\boldsymbol{x}}_i(n|n)=\hat{\boldsymbol{x}}_i(n|n-1)+\boldsymbol{g}_i(\tilde{\theta}_i(n)-\hat{\theta}_i(n|n-1)) \tag{30}$$
$$\hat{\theta}_i(n)=\boldsymbol{c}^T\hat{\boldsymbol{x}}_i(n|n). \tag{31}$$

..... 7 flops

12) Update the instant index of direction updating as $n=n+1$, and go to Step 5).

Remark 5: By taking account of a compromise between speedy response and sensitivity to the unmodeled disturbances and measurement noises in the dynamic direction model (3) and (4), a careful examination of the overall performance of observer in (23) shows that the desired poles of observer $\boldsymbol{g}_i$ can be selected simply as $z_{i1}=z_{i2}^*=\bar{\kappa}_i e^{j\bar{\phi}_i}$ and $z_{i3}=\bar{\kappa}_i$, where $\bar{\kappa}_i$ is around 0.75, and $\bar{\phi}_i$ is in the region between 9° and 16°, where the observer has an underdamped response. □

Remark 6: In the Luenberger state estimation shown in (23) and (24), the state vector $\hat{\boldsymbol{x}}_i(n)$ (and hence $\hat{\theta}_i(n)$) and the measurement $\theta_i(n)$ are unknown and should be estimated. In the above-mentioned AMEND algorithm, the Luenberger state estimation is implemented as: $\hat{\boldsymbol{x}}_i(n|n)=\boldsymbol{F}\hat{\boldsymbol{x}}_i(n-1|n-1)+\boldsymbol{g}_i(\tilde{\theta}_i(n)-\hat{\theta}_i(n|n-1))$, and correspondingly $\hat{\theta}_i(n)=\hat{\theta}_i(n|n-1)+g_{i1}(\tilde{\theta}_i(n)-\hat{\theta}_i(n|n-1))$. Obviously the estimate $\hat{\boldsymbol{x}}_i(n|n)$ in (30) is based on the predicted estimate $\hat{\boldsymbol{x}}_i(n|n-1)$ in (26), the measured direction $\tilde{\theta}_i(n)$ in (29) and the predicted direction $\hat{\theta}_i(n|n-1)$ in (27), while the predicted estimate $\hat{\boldsymbol{x}}_i(n|n-1)$ is based on the existing estimate $\hat{\boldsymbol{x}}_i(n-1|n-1)$ at the previous $(n-1)$th instant, and the measured direction $\tilde{\theta}_i(n)$ is the estimate of $\theta_i(n)$ at the current nth instant obtained by the approximate Newton iteration with the array data through $\boldsymbol{\Pi}(n)$ and the predicted direction $\hat{\theta}_i(n|n-1)$. As a result, the Luenberger observer plays important roles of refining (or smoothing) the measured direction $\tilde{\theta}_i(n)$ obtained from the array data and maintaining the association between the estimates $\{\hat{\theta}_i(n)\}$ at different instants of direction updating. □

Remark 7: In the localization stage, although the conventional subspace-based methods such as the MUSIC (or spatial smoothing (SS) based MUSIC) with eigendecomposition [33], [44], [45], the SUMWE [2] and the recursion

subspace-based method (RSM) [46] without eigendecomposition can be used to estimate the reliable measurements of DOAs (i.e., $\{\tilde{\theta}_i(n)\}$) from the time-averaged statistics of the array data $\{\boldsymbol{y}(k)\}_{k=nN_s+1}^{(n+1)N_s}$, since the peak picking of the (SS-)MUSIC/SUMWE/RSM spectrum or the polynomial rooting of the root-(SS-)MUSIC/root-SUMWE/root-RSM does not suggest any ordering of the estimated DOAs at contiguous time instants, hence successive direction estimates at different time instants from these batch methods cannot be used for tracking [6], [7], [47]. For the case of p incident signals, there are $p!$ possible combinations between the estimated DOAs $\{\tilde{\theta}_i(n)\}$ and $\{\tilde{\theta}_i(n-1)\}$ (i.e., the signals $\{s_i(n)\}$ and the measurements $\{\tilde{\theta}_i(n)\}$). This results in a tremendous computational burden as the number of signals increases. Moreover batch methods are often difficult to implement because of the memory and/or time required to process the large amount of data [2], [6], [47]. Thus without a sophisticated mechanism to correlate successive estimates of directions, it is not possible to track the multiple directions simply by repeatedly implementing the batch subspace-based methods such as the MUSIC, the RSM and their variations. Therefore the approximate Newton iteration in (22) is used in the localization stage to estimate the measurement of direction and to maintain the association between the estimated directions at successive instants of DOA updating. □

C. Modified AMEND Algorithm

In many practical situations, some incident signals may appear and disappear suddenly [40]. Now we modify the above algorithm and give a variation with "self-initialization", where the change in the number of signals is monitored during each interval $(nN_s, (n+1)N_s]$.

(1) Perform the above Steps 1)–4) for initialization.
(2) Perform the above Steps 5)–11) for DOA tracking.
(3) Estimate the number of incident signals from the N_s snapshots of $\{\boldsymbol{y}(k)\}_{k=nN_s+1}^{(n+1)N_s}$ by using the MENSE [34] and denote it by $\hat{p}(n)$, where the subarray size is set to $\bar{p} = \lfloor M/2 \rfloor$.
(4) Check whether $\hat{p}(n) = \hat{p}$; if yes, update the instant index of direction updating as $n = n+1$ and go to Step (2); otherwise set $\hat{p} = \hat{p}(n)$ and go to the next step.
(5) Estimate $\hat{p}$ directions from $\{\boldsymbol{y}(k)\}_{k=nN_s+1}^{(n+1)N_s}$ by using the SUMWE [2] and denote them by $\hat{\theta}_i(n-1|n-1)$, where the subarray size is set to $\hat{p}$.
(6) Initialize the Luenberger observer and the RLS algorithm for the next instant $n+1$ as $\hat{\boldsymbol{x}}(n-1|n-1) = [\hat{\theta}_i(n-1|n-1), \hat{\theta}_i(n-1|n-1)/N_s, 0]^T$, $\hat{\boldsymbol{P}}(n-1) = \boldsymbol{O}_{\hat{p}\times(M-2\hat{p})}$, and $\boldsymbol{\Psi}_1(n-1) = \Psi_0 \boldsymbol{I}_{\hat{p}\times\hat{p}}$.
(7) Update the instant index as $n = n+1$, and return to Step (2).

V. Statistical Analysis of RLS Algorithm in Stationary Environment

In this section, we investigate the performance of the proposed RLS algorithm in stationary environment, which is a convention of statistical analysis of adaptive filtering theory.

A. Expectation Computation of Inverse Matrix

Due to the nonlinear and recursive nature and the presence of inverse matrix, the exact and theoretical analysis of the RLS algorithm is rather complicated and difficult [27], where the computation of the expectation $E\{\boldsymbol{\Psi}_1^{-1}(k)\}$ is crucial to the statistical analysis of the proposed RLS algorithm. Firstly we consider the expectation of the matrix $\boldsymbol{\Psi}_1(k)$ in (15). By taking expectation on both side of (15) and after some manipulations, we can get

$$\begin{aligned} \mathrm{E}\{\boldsymbol{\Psi}_1(k)\} &= \sum_{i=1}^{k} E\{\boldsymbol{\Phi}_1(i)\boldsymbol{\Phi}_1^H(i)\} + \boldsymbol{\Psi}_o \\ &= k\bar{\boldsymbol{\Psi}}_1 + \boldsymbol{\Psi}_o, \text{ for } \gamma = 1 \end{aligned} \tag{32}$$

and

$$\begin{aligned} \mathrm{E}\{\boldsymbol{\Psi}_1(k)\} &= \sum_{i=1}^{k} \gamma^{k-i}\bar{\boldsymbol{\Psi}}_1 + \gamma^k \boldsymbol{\Psi}_o \\ &= \frac{1-\gamma^k}{1-\gamma}\bar{\boldsymbol{\Psi}}_1 + \gamma^k \boldsymbol{\Psi}_o, \text{ for } 0 \ll \gamma < 1 \end{aligned} \tag{33}$$

where

$$\begin{aligned} \bar{\boldsymbol{\Psi}}_1 &\triangleq E\left\{\bar{\boldsymbol{\Phi}}_1(k)\boldsymbol{\Phi}_1^H(k)\right\} \\ &= \boldsymbol{\Psi}_1 + \boldsymbol{A}_1\left(r_{MM}\left(\bar{\boldsymbol{R}}_s + \boldsymbol{D}\tilde{\boldsymbol{R}}_s\boldsymbol{D}^{-1}\right)\right. \\ &\quad \left. + r_{11}\left(\boldsymbol{D}\bar{\boldsymbol{R}}_s\boldsymbol{D}^{-1} + \tilde{\boldsymbol{R}}_s\right)\right)\boldsymbol{A}_1^H + \alpha\boldsymbol{I}_p \end{aligned} \tag{34}$$

$$r_{im} \triangleq E\{y_i(k)y_m^*(k)\} = \boldsymbol{b}_i^T(\theta)\boldsymbol{R}_s\boldsymbol{b}_m^*(\theta) + \sigma^2\delta_{i,m} \tag{35}$$

$$\bar{\boldsymbol{R}}_s \triangleq \sum_{l=1}^{p} \boldsymbol{D}^{l-1}\boldsymbol{R}_s\boldsymbol{D}^{-(l-1)} \tag{36}$$

$$\tilde{\boldsymbol{R}}_s \triangleq \sum_{l=1}^{p} \boldsymbol{D}^{-(M-l)}\boldsymbol{R}_s^*\boldsymbol{D}^{M-l} \tag{37}$$

with $\boldsymbol{\Psi}_1 \triangleq \boldsymbol{\Phi}_1\boldsymbol{\Phi}_1^H$, and $\alpha \triangleq 2p\sigma^2(r_{11} + r_{MM})$.

Since the received signals $\{y_m(k)\}$ are temporally complex white Gaussian random processes under the model assumptions and $\bar{\boldsymbol{\Psi}}_1(k)$ in (15) is the exponentially weighted time-average of the product $\gamma^{k-i}\boldsymbol{\Phi}_1(i)\boldsymbol{\Phi}_1^H(i)$ for $1 \le i \le k$, we can see that $\boldsymbol{\Psi}_1^{-1}(k)$ in (14) tends to be "slowly" time-varying w.r.t. $\boldsymbol{\Phi}_1(i)$ and hence it is "almost" independent of $\boldsymbol{\Phi}_1(i)$ (i.e., $\boldsymbol{\Psi}_1(k)$) for a non-vanishingly small forgetting factor γ. Then following the fact $\boldsymbol{\Psi}_1(k)\boldsymbol{\Psi}_1^{-1}(k) = \boldsymbol{I}_p$ and by applying the averaging principle (e.g., [26], [36], and [37]), we can obtain

$$\begin{aligned} \boldsymbol{I}_p &= E\{\boldsymbol{\Psi}_1(k)\boldsymbol{\Psi}_1^{-1}(k)\} \\ &\approx E\{\bar{\boldsymbol{\Psi}}_1(k)\}E\{\boldsymbol{\Psi}_1^{-1}(k)\}. \end{aligned} \tag{38}$$

Therefore by combining (32), (33) with (38), we can approximate the expectation of the inverse matrix $\boldsymbol{\Psi}_1^{-1}(k)$ as

$$\begin{aligned} E\{\boldsymbol{\Psi}_1^{-1}(k)\} &\approx (E\{\boldsymbol{\Psi}_1(k)\})^{-1} \\ &= \begin{cases} \frac{1}{k}\left(\bar{\boldsymbol{\Psi}}_1 + \frac{1}{k}\boldsymbol{\Psi}_o\right)^{-1}, & \text{for } \gamma = 1 \\ \left(\frac{1-\gamma^k}{1-\gamma}\bar{\boldsymbol{\Psi}}_1 + \gamma^k\boldsymbol{\Psi}_o\right)^{-1}, & \text{for } 0 \ll \gamma < 1 \end{cases} \\ &\triangleq \boldsymbol{\Psi}_{\mathrm{inv}}(k) \end{aligned} \tag{39}$$

for a non-vanishingly small forgetting factor γ.

Remark 8: The averaging principle (e.g., [36] and [37]) was applied to facilitate the performance analysis in adaptive filtering literature (e.g., [31] and [38]). Although the approximation of expectation of the inverse matrix by the inverse of expectation of matrix shown in (39) may not be valid for many practical input signals in the ordinary temporal filtering algorithms, it holds in array processing in Gaussian environment for a non-vanishingly small forgetting factor γ. (see the justification of this approximation through simulations in Section VII). □

B. Mean Behavior

By defining the weight-error $\hat{\boldsymbol{P}}(k)$ between the true weight $\boldsymbol{P}$ and the adjusted weight $\hat{\boldsymbol{P}}(k)$ of the adaptive algorithm as $\tilde{\boldsymbol{P}}(k) \triangleq \boldsymbol{P} - \hat{\boldsymbol{P}}(k)$ and after some manipulations, from (18)–(20) and (9), we get

$$\begin{aligned}\tilde{\boldsymbol{P}}(k) =& (\boldsymbol{I}_p - \boldsymbol{\Psi}_1^{-1}(k)\boldsymbol{\Phi}_1(k)\boldsymbol{\Phi}_1^H(k))\tilde{\boldsymbol{P}}(k-1) \\ &+ \boldsymbol{\Psi}_1^{-1}(k)\boldsymbol{\Phi}_1(k)\boldsymbol{G}^H(k)\boldsymbol{Q}. \end{aligned} \tag{40}$$

Under the assumptions that $\boldsymbol{\Psi}_1^{-1}(k)$ and $\hat{\boldsymbol{P}}(k-1)$ are independent of $\boldsymbol{\Phi}_1(k)$, by taking the expectation on both sides of (40) and using the results of expectation evaluation [18], we can get the recursion of the weight-error matrix in the mean sense

$$\begin{aligned} &E\{\tilde{\boldsymbol{P}}(k)\} \\ &\quad= (\boldsymbol{I}_p - \boldsymbol{\Psi}_{\text{inv}}(k)\bar{\boldsymbol{\Psi}}_1)E\{\tilde{\boldsymbol{P}}(k-1)\} + \alpha\boldsymbol{\Psi}_{\text{inv}}(k)\boldsymbol{P} \\ &\quad= \left(\prod_{m=1}^{k}(\boldsymbol{I}_p - \boldsymbol{\Psi}_{\text{inv}}(m)\bar{\boldsymbol{\Psi}}_1)\right)E\{\tilde{\boldsymbol{P}}(0)\} + \alpha\Bigg\{\boldsymbol{\Psi}_{\text{inv}}(k) \\ &\qquad+ \sum_{m=1}^{k-1}\boldsymbol{\Psi}_{\text{inv}}(m)\left(\prod_{i=m+1}^{k}(\boldsymbol{I}_p - \boldsymbol{\Psi}_{\text{inv}}(i)\bar{\boldsymbol{\Psi}}_1)\right)\Bigg\}\boldsymbol{P}. \end{aligned} \tag{41}$$

Obviously the convergence of the weight-error matrix $\tilde{\boldsymbol{P}}(k)$ in the mean sense is governed by the time-varying component $\boldsymbol{F}_{\text{mean}}(k) \triangleq \boldsymbol{I}_p - \boldsymbol{\Psi}_{\text{inv}}(k)\bar{\boldsymbol{\Psi}}_1$, which is more complicated than that of the LMS/NLMS algorithm [18].

C. Mean-Square Behavior

In view of the assumptions that the true weight $\boldsymbol{P}$ and the estimated one $\hat{\boldsymbol{P}}(k-1)$ are independent of $\boldsymbol{\Phi}_1(k)$ and the analyses in Section V-A, we can find that the weight-error $\tilde{\boldsymbol{P}}(k-1)$ and the inverse matrix $\boldsymbol{\Psi}_1^{-1}(k)$ are independent of $\boldsymbol{\Phi}_1(k)$ for a non-vanishingly small γ. By letting $\boldsymbol{K}(k) \triangleq E\{\tilde{\boldsymbol{P}}(k)\tilde{\boldsymbol{P}}^H(k)\}$ and following [18], from (40) and after some calculations, we can obtain

$$\begin{aligned}\boldsymbol{K}(k) =& \left(\boldsymbol{I}_p - \boldsymbol{\Psi}_{\text{inv}}(k)\bar{\boldsymbol{\Psi}}_1\right)\boldsymbol{K}(k-1)(\boldsymbol{I}_p - \bar{\boldsymbol{\Psi}}_1\boldsymbol{\Psi}_{\text{inv}}^H(k)) \\ &+ 2\text{Re}\Big\{\alpha\boldsymbol{\Psi}_{\text{inv}}(k)\boldsymbol{P}E\{\tilde{\boldsymbol{P}}^H(k-1)\} - \boldsymbol{\Psi}_{\text{inv}}(k) \\ &\cdot(\alpha\boldsymbol{P}E\{\tilde{\boldsymbol{P}}^H(k-1)\}\bar{\boldsymbol{\Psi}}_1 + \boldsymbol{K}_5 + \boldsymbol{K}_6)\boldsymbol{\Psi}_{\text{inv}}^H(k)\Big\} \\ &+ \boldsymbol{\Psi}_{\text{inv}}(k)\left(\alpha^2\boldsymbol{P}\boldsymbol{P}^H + \boldsymbol{K}_1 + \boldsymbol{K}_2 + \boldsymbol{K}_3\right. \\ &\left.+\boldsymbol{K}_4\right)\boldsymbol{\Psi}_{\text{inv}}^H(k) \end{aligned} \tag{42}$$

where $\boldsymbol{\Psi}_{\text{inv}}(k)$ denotes the expectation of matrix $\boldsymbol{\Psi}_1^{-1}(k)$ given by (39), while

$$\boldsymbol{K}_1 \triangleq \sum_{l=1}^{p}\sum_{t=1}^{p}\sum_{i=1}^{4}\sum_{k=1}^{4}\bar{\boldsymbol{F}}_{il,mt}\boldsymbol{K}^T(k-1)\bar{\boldsymbol{F}}_{il,mt}^* \tag{43}$$

$$\boldsymbol{K}_2 \triangleq \sum_{l=1}^{p}\sum_{t=1}^{p}\sum_{i=1}^{4}\sum_{k=1}^{4}\boldsymbol{F}_{il,mt}\text{tr}\{\boldsymbol{F}_{il,mt}^H\boldsymbol{K}(k-1)\} \tag{44}$$

$$\boldsymbol{K}_3 \triangleq \sum_{l=1}^{p}\sum_{t=1}^{p}\sum_{i=1}^{4}\sum_{k=1}^{4}\boldsymbol{\Gamma}_{il,mt}\boldsymbol{Q}^*\boldsymbol{Q}^T\boldsymbol{\Gamma}_{mt,il}^H \tag{45}$$

$$\boldsymbol{K}_4 \triangleq \sum_{l=1}^{p}\sum_{t=1}^{p}\sum_{i=1}^{4}\sum_{k=1}^{4}\boldsymbol{F}_{il,mt}\text{tr}\{\tilde{\boldsymbol{\Gamma}}_{il,mt}^H\boldsymbol{Q}\boldsymbol{Q}^H\} \tag{46}$$

$$\boldsymbol{K}_5 \triangleq \sum_{l=1}^{p}\sum_{t=1}^{p}\sum_{i=1}^{4}\sum_{k=1}^{4}\bar{\boldsymbol{F}}_{il,mt}E\{\tilde{\boldsymbol{P}}^*(k-1)\}\boldsymbol{Q}^T\boldsymbol{\Gamma}_{mt,il}^H \tag{47}$$

$$\boldsymbol{K}_6 \triangleq \sum_{l=1}^{p}\sum_{t=1}^{p}\sum_{i=1}^{4}\sum_{k=1}^{4}\boldsymbol{F}_{il,mt}\text{tr}\{\bar{\boldsymbol{\Gamma}}_{mt,il}\boldsymbol{Q}E\{\tilde{\boldsymbol{P}}^H(k-1)\}\} \tag{48}$$

with $\bar{\boldsymbol{F}}_{il,mt} \triangleq E\{\tilde{\boldsymbol{z}}_{il}(k)\tilde{\boldsymbol{z}}_{mt}^T(k)\}$, $\boldsymbol{F}_{il,mt} \triangleq E\{\tilde{\boldsymbol{z}}_{il}(k)\tilde{\boldsymbol{z}}_{mt}^H(k)\}$, $\boldsymbol{\Gamma}_{il,mt} \triangleq E\{\tilde{\boldsymbol{z}}_{il}(k)\tilde{\boldsymbol{g}}_{mt}^T(k)\}$, $\tilde{\boldsymbol{\Gamma}}_{il,mt} \triangleq E\{\tilde{\boldsymbol{g}}_{il}(k)\tilde{\boldsymbol{g}}_{mt}^H(k)\}$, and $\bar{\boldsymbol{\Gamma}}_{mt,il} \triangleq E\{\tilde{\boldsymbol{z}}_{mt}(k)\tilde{\boldsymbol{g}}_{il}^H(k)\}$ for $i,\ m = 1,\ 2,\ 3,\ 4$ and $l,\ t = 1, 2, \ldots, p$, $\tilde{\boldsymbol{z}}_{1l}(k) \triangleq \boldsymbol{y}_{fl}(k)y_M^*(k)$, $\tilde{\boldsymbol{z}}_{2l}(k) \triangleq \boldsymbol{y}_{fl+1}(k)y_1^*(k)$, $\tilde{\boldsymbol{z}}_{3l}(k) \triangleq \boldsymbol{y}_{bl}(k)y_1(k)$, $\tilde{\boldsymbol{z}}_{4l}(k) \triangleq \boldsymbol{y}_{bl+1}(k)y_M(k)$, $\tilde{\boldsymbol{g}}_{1l}(k) \triangleq \tilde{\boldsymbol{w}}_{fl}(k)y_M^*(k)$, $\tilde{\boldsymbol{g}}_{2l}(k) \triangleq \tilde{\boldsymbol{w}}_{fl+1}(k)y_1^*(k)$, $\tilde{\boldsymbol{g}}_{3l}(k) \triangleq \tilde{\boldsymbol{w}}_{bl}(k)y_1(k)$, $\tilde{\boldsymbol{g}}_{4l}(k) \triangleq \tilde{\boldsymbol{w}}_{bl+1}(k)y_M(k)$, $\tilde{\boldsymbol{w}}_{fl}(k) \triangleq [w_l(k), w_{l+1}(k), \ldots, w_{l+L-2}(k)]^T$, and $\tilde{\boldsymbol{w}}_{bl}(k) \triangleq [w_{M-l+1}(k),\ w_{M-l}(k), \ldots, w_{p+2-l}(k)]^H$. Furthermore the analytical expressions of the five matrices $\bar{\boldsymbol{F}}_{il,mt}$, $\boldsymbol{F}_{il,kt}$, $\boldsymbol{\Gamma}_{il,mt}$, $\bar{\boldsymbol{\Gamma}}_{mt,il}$, and $\tilde{\boldsymbol{\Gamma}}_{il,mt}$ can be obtained in Tables II–VI of [18], and they are omitted herein for conciseness.

Clearly this recursion and that in (41) describe the transient behavior of the proposed RLS algorithm in the mean and mean-square senses, respectively.

D. Learning Curves and Steady-State Performance

Since the derivations of the MSE and MSD learning curves of the proposed RLS algorithm are almost same to that of the LMS algorithm [18] and the results can be directly applied, hence from (9), (20) and (42), the MSE and MSD learning curves $J_{\text{RLS}}^{\text{MSE}}(k)$ and $J_{\text{RLS}}^{\text{MSD}}(k)$ of the RLS algorithm are obtained

$$\begin{aligned} J_{\text{RLS}}^{\text{MSE}}(k) &\triangleq \text{tr}\{E\{\boldsymbol{E}^H(k)\boldsymbol{E}(k)\}\} \\ &= \text{tr}\{\bar{\boldsymbol{\Psi}}_1\boldsymbol{K}(k-1)\} + \alpha\text{tr}\{\boldsymbol{P}^H\boldsymbol{P} + \boldsymbol{I}_{M-2p}\} \\ &\quad - 2\alpha\text{Re}\{\text{tr}\{\boldsymbol{P}^HE\{\tilde{\boldsymbol{P}}(k-1)\}\}\} \end{aligned} \tag{49}$$

$$J_{\text{RLS}}^{\text{MSD}}(k) \triangleq E\{\text{tr}\{\tilde{\boldsymbol{P}}^H(k)\tilde{\boldsymbol{P}}(k)\}\} = \text{tr}\{\boldsymbol{K}(k)\}. \tag{50}$$

Furthermore from (41), the mean of the weight-error $\tilde{\boldsymbol{P}}(k)$ in the steady-state is given by

$$\lim_{k\to\infty}E\{\tilde{\boldsymbol{P}}(k)\} = \alpha\bar{\boldsymbol{\Psi}}_1^{-1}\boldsymbol{P}. \tag{51}$$

And the steady-state MSE (SSMSE) and steady-state MSD (SSMSD) of the RLS algorithm are obtained as

$$\begin{aligned} J_{\mathrm{RLS}}^{\mathrm{SSMSE}} &\triangleq \lim_{k\to\infty} J_{\mathrm{RLS}}^{\mathrm{MSE}}(k) \\ &= \mathrm{tr}\{\bar{\boldsymbol{\Psi}}_1 \boldsymbol{K}(\infty)\} - 2\alpha^2 \mathrm{Re}\{\mathrm{tr}\{\boldsymbol{P}^H \bar{\boldsymbol{\Psi}}_1^{-1} \boldsymbol{P}\}\} \\ &\quad + \alpha \mathrm{tr}\{\boldsymbol{P}^H \boldsymbol{P} + \boldsymbol{I}_{M-2p}\} \qquad (52) \\ J_{\mathrm{RLS}}^{\mathrm{SSMSD}} &\triangleq \lim_{k\to\infty} J_{\mathrm{RLS}}^{\mathrm{MSD}}(k) = \mathrm{tr}\{\boldsymbol{K}(\infty)\} \qquad (53) \end{aligned}$$

where $\boldsymbol{K}(\infty) \triangleq \lim_{k\to\infty} \boldsymbol{K}(k)$, which is the mean-square weight-error given by (42) in the steady-state.

Remark 9: Due to the fact that the expectation of the inverse matrix is derived for a nonvanishingly small forgetting factor, the statistical analyses of the RLS algorithm mentioned above are valid for large forgetting factor. □

VI. Analytical Study of RLS Performance

Here we consider the statistical properties of the RLS algorithm in the case of one single signal with constant direction and the quantitative performance comparison of the RLS and LMS algorithms in the steady-state to gain some insights into the impacts of the forgetting factor, the initialization and the SNR on the performance of the proposed algorithm.

In this case (i.e., $p = 1$ and $L = M$), we readily get [18]

$$\boldsymbol{P} = \boldsymbol{p} = [e^{-j\omega_0\tau(\theta_1)}, \ldots, e^{-j\omega_0(M-2)\tau(\theta_1)}] \qquad (54)$$

$$\bar{\boldsymbol{\Psi}}_1 = \bar{\Psi}_1 = 4r_s^2 + 4(r_s + \sigma^2)^2 \qquad (55)$$

where $r_s = r_{s_1}$. By performing some manipulations, the mean and mean-squared weight-errors in (41) and (42) are given by

$$\begin{aligned} E\{\tilde{\boldsymbol{P}}(k)\} &= E\{\tilde{\boldsymbol{p}}(k)\} \\ &= (1 - \Psi_{\mathrm{inv}}(k)\bar{\Psi}_1) E\{\tilde{\boldsymbol{p}}(k-1)\} + \alpha \Psi_{\mathrm{inv}}(k) \boldsymbol{p} \qquad (56) \\ \boldsymbol{K}(k) &= K(k) \\ &= (1 - \Psi_{\mathrm{inv}}(k)\bar{\Psi}_1)^2 K(k-1) + 2\mathrm{Re}\{\alpha \Psi_{\mathrm{inv}}(k) \boldsymbol{p} \\ &\quad \cdot E\{\tilde{\boldsymbol{p}}^H(k-1)\} - \Psi_{\mathrm{inv}}^2(k)(\alpha \boldsymbol{p} E\{\tilde{\boldsymbol{p}}^H(k-1)\} \\ &\quad \cdot \bar{\Psi}_1 + K_5 + K_6)\} + \Psi_{\mathrm{inv}}^2(k)(\alpha^2 \boldsymbol{p}\boldsymbol{p}^H + K_1 \\ &\quad + K_2 + K_3 + K_4) \\ &= \left((1 - \Psi_{\mathrm{inv}}(k)\bar{\Psi}_1)^2 + \Psi_{\mathrm{inv}}(k)(\bar{K}_1 + \bar{K}_2)\right) K(k-1) \\ &\quad + 2\mathrm{Re}\{\alpha \Psi_{\mathrm{inv}}(k) \boldsymbol{p} E\{\tilde{\boldsymbol{p}}^H(k-1)\} \\ &\quad - \Psi_{\mathrm{inv}}^2(k)(\alpha \boldsymbol{p} E\{\tilde{\boldsymbol{p}}^H(k-1)\}\bar{\Psi}_1 + K_5 + K_6)\} \\ &\quad + \Psi_{\mathrm{inv}}^2(k)(\alpha^2 \boldsymbol{p}\boldsymbol{p}^H + K_3 + K_4) \qquad (57) \end{aligned}$$

where $\alpha = 4\sigma^2(r_s + \sigma^2)$, and

$$\begin{aligned} K_1 &= 8r_s^2(4r_s^2 + (2r_s + \sigma^2)^2)K(k-1) \\ &\triangleq \bar{K}_1 K(k-1) \qquad (58) \\ K_2 &= \left(40r_s^4 + 6(r_s^2 + (r_s + \sigma^2)^2)^2\right) K(k-1) \\ &\triangleq \bar{K}_2 K(k-1) \qquad (59) \\ K_3 &= 2\sigma^4(r_s^2 + 2\sigma^2(r_s + \sigma^2)) \qquad (60) \\ K_4 &= 2(M-2)\sigma^2(r_s^2 + (r_s + \sigma^2)^2)(4(r_s + \sigma^2) + \sigma^2) \\ &\quad + 4\sigma^2 r_s^2(\sigma^2 - 2r_s) \qquad (61) \\ K_5 &= c_{51} \boldsymbol{p} E\{\tilde{\boldsymbol{p}}^H(k-1)\} \\ &\quad - c_{52} e^{j\omega_0(2-M)\tau(\theta_1)} (E\{\tilde{\boldsymbol{p}}^*(k-1)\})_{M-2} \\ &\quad - c_{53} e^{j\omega_0(3-M)\tau(\theta_1)} (E\{\tilde{\boldsymbol{p}}^*(k-1)\})_{M-3} \qquad (62) \\ K_6 &= c_{61} \boldsymbol{p} E\{\tilde{\boldsymbol{p}}^H(k-1)\} \\ &\quad - c_{53} e^{-j\omega_0\tau(\theta_1)} (E\{\tilde{\boldsymbol{p}}^*(k-1)\})_1 \\ &\quad - c_{53} e^{j\omega_0(2-M)\tau(\theta_1)} (E\{\tilde{\boldsymbol{p}}^*(k-1)\})_{M-2} \qquad (63) \\ c_{51} &= 2\sigma^2 r_s^2 (2r_s + \sigma^2) \qquad (64) \\ c_{52} &= 6\sigma^2 r_s (r_s + \sigma^2)(2r_s + \sigma^2) \qquad (65) \\ c_{53} &= 4\sigma^2 r_s^3 \qquad (66) \\ c_{61} &= 6\sigma^2 (r_s + \sigma^2)(r_s^2 + (r_s + \sigma^2)^2) \qquad (67) \end{aligned}$$

while $(\cdot)_i$ denotes the ith element of the bracket vector, and the approximated expectation of the inverse matrix $\Psi_1^{-1}(k)$ is given by

$$\Psi_{\mathrm{inv}}(k) = \begin{cases} (k\bar{\Psi}_1 + \Psi_o)^{-1}, & \text{for } \gamma = 1 \\ \left(\frac{1-\gamma^k}{1-\gamma}\bar{\Psi}_1 + \gamma^k \Psi_o\right)^{-1}, & \text{for } 0 \ll \gamma < 1. \end{cases} \qquad (68)$$

where this expectation in the steady-state is given by

$$\lim_{k\to\infty} \Psi_{\mathrm{inv}}(k) = \begin{cases} 0, & \text{for } \gamma = 1 \\ (1-\gamma)\bar{\Psi}_1^{-1}, & \text{for } 0 \ll \gamma < 1. \end{cases} \qquad (69)$$

Then from (56) and (57), we can see that the convergences in the mean and mean-square senses are controlled by the time-varying factors $f_{\mathrm{mean}}(k) \triangleq 1 - \Psi_{\mathrm{inv}}(k)\bar{\Psi}_1$ and $f_{\mathrm{m.s.}}(k) \triangleq (1 - \Psi_{\mathrm{inv}}(k)\bar{\Psi}_1)^2 + \Psi_{\mathrm{inv}}^2(k)(\bar{K}_1 + \bar{K}_2)$, respectively. Under the assumptions that $0 \ll \gamma \leq 1$ and $\Psi_0 > 0$, from (68), we easily get $0 < f_{\mathrm{mean}}(k) \leq 1$, which means that the convergence is always guaranteed in the mean sense. On the other hand, the convergence factor $f_{\mathrm{m.s.}}(k)$ is affected by γ and Ψ_0 in a rather complicated manner through the time-varying $\Psi_{\mathrm{inv}}(k)$ during the transient phase. By analyzing $f_{\mathrm{m.s.}}(k)$ with (68) and (69), we can find that following observations on the performance of the proposed RLS algorithm (some slightly tedious analyses are omitted herein):

1) The small initialization Ψ_0 should to be chosen to gain fast convergence;
2) When the forgetting factor γ increases, the rate of convergence will decreases;
3) The rate of convergence decreases with the increasing SNR.

Furthermore, by comparing the updating equation of the RLS algorithm in (19) and that of the LMS algorithm in [21, Eq. (25)], we see that the inverse matrix $\bar{\boldsymbol{\Psi}}_1^{-1}(k)$ (i.e., $\tilde{\bar{\boldsymbol{\Psi}}}_1(k)$) in (19) has the effect similar to the step-size μ of the LMS algorithm. Now we discuss the relative performance of the RLS algorithm and the previously proposed LMS algorithm [18] in the steady state.

When the instant index k is sufficiently large so that $E\{\tilde{\boldsymbol{p}}(k)\} = E\{\tilde{\boldsymbol{p}}(k-1)\}$ and $K(k) = K(k-1)$, from (56) and (57), we can obtain by virtue of (54), (62), and (63)

$$E\{\tilde{\boldsymbol{p}}(k)\} = \alpha \bar{\Psi}_1^{-1} \boldsymbol{p} \qquad (70)$$

$$K(k) = \frac{N_{\mathrm{RLS}}(k)}{D_{\mathrm{RLS}}(k)} \qquad (71)$$

for sufficiently large iteration index k, where

$$\begin{aligned} N_{\mathrm{RLS}}(k) &= 2(M-2)\alpha^2 \bar{\Psi}^{-1} + \Psi_{\mathrm{inv}}(k)(K_3 + K_4 \\ &\quad - (M-2)\alpha^2 - 2(K_5 + K_6)) \qquad (72) \\ D_{\mathrm{RLS}}(k) &= 2\bar{\Psi}_1 - \Psi_{\mathrm{inv}}(k)(\bar{\Psi}_1^2 + \bar{K}_1 + \bar{K}_2) \qquad (73) \end{aligned}$$

$$K_5 = \alpha\bar{\Psi}_1^{-1}((M-2)c_{51} - c_{52} - c_{53}) \quad (74)$$

$$K_6 = \alpha\bar{\Psi}_1^{-1}((M-2)c_{61} - 2c_{53}). \quad (75)$$

Then by substituting (69) into (71), from (53), the SSMSD of the proposed RLS algorithm is obtained for different forgetting factor γ as

$$J_{\mathrm{RLS}}^{\mathrm{SSMSD}}\big|_{\gamma=1} = (M-2)\alpha^2\bar{\Psi}_1^{-2} \quad (76)$$

$$\begin{aligned} J_{\mathrm{RLS}}^{\mathrm{SSMSD}}&\big|_{0\ll\gamma<1} \\ &= (2\bar{\Psi}_1^2 - (1-\gamma)(\bar{\Psi}_1^2 + \bar{K}_1 + \bar{K}_2))^{-1}\Big\{2(M-2)\alpha^2 \\ &\quad + (1-\gamma)(K_3 + K_4 - (M-2)\alpha^2 - 2\alpha\Psi_1^{-1} \\ &\quad \cdot ((M-2)(c_{51}+c_{61}) - (c_{52}+3c_{53})))\Big\}. \end{aligned} \quad (77)$$

Similarly we can get the analytical expression of the SSMSD $J_{\mathrm{LMS}}^{\mathrm{SSMSD}}$ of the LMS algorithm proposed in [18] as

$$\begin{aligned} J_{\mathrm{LMS}}^{\mathrm{SSMSD}} &= (2\bar{\Psi}_1^2 - \mu\bar{\Psi}_1(\bar{\Psi}_1^2 + \bar{K}_1 + \bar{K}_2))^{-1}\{2(M-2)\alpha^2 \\ &\quad + \mu\bar{\Psi}_1(K_3 + K_4 - (M-2)\alpha^2 - 2\alpha\Psi_1^{-1} \\ &\quad \cdot ((M-2)(c_{51}+c_{61}) - (c_{52}+3c_{53})))\} \end{aligned} \quad (78)$$

where μ is the positive step-size, and its stability region is given by $0 < \mu < \bar{x}_{\mathrm{up}}$. [18].

Hence by comparing (76)–(78), we can easily obtain the following results on the steady-state performance of the proposed RLS algorithm and the LMS algorithm in [18] for the case of one incident signal

$$J_{\mathrm{RLS}}^{\mathrm{SSMSD}}\big|_{\gamma=1} < J_{\mathrm{RLS}}^{\mathrm{SSMSD}}\big|_{0\ll\gamma<1} \quad (79)$$

$$J_{\mathrm{RLS}}^{\mathrm{SSMSD}}\big|_{\gamma_2} < J_{\mathrm{RLS}}^{\mathrm{SSMSD}}\big|_{\gamma_1} \quad (80)$$

for $0 \ll \gamma_1 < \gamma_2 < 1$, while

$$J_{\mathrm{RLS}}^{\mathrm{SSMSD}}\big|_{\gamma=1} < J_{\mathrm{LMS}}^{\mathrm{SSMSD}} \quad (81)$$

and

$$J_{\mathrm{RLS}}^{\mathrm{SSMSD}} = J_{\mathrm{LMS}}^{\mathrm{SSMSD}} \quad (82)$$

for $\mu = (1-\gamma)\bar{\Psi}_1^{-1}$, $0 \ll \gamma < 1$, and $0 < \mu < \bar{x}_{\mathrm{up}}$.. Obviously the RLS algorithm with $\gamma = 1$ has superior performance in the steady-state than the RLS algorithm with $0 \ll \gamma < 1$ and the LMS algorithm with any step-size, and the RLS and LMS algorithms have the similar steady-state performance when the RLS forgetting factor and the LMS step-size are chosen appropriately.

VII. Numerical Examples

Now we verify the effectiveness of the proposed AMEND algorithm and the statistical analyses of the proposed RLS-based subspace updating algorithm through some numerical examples. The ULA is separated by a half-wavelength, and the SNR is defined as the ratio of the power of the incident signal to that of the additive noise at each sensor. The simulation results are obtained by the ensemble-averaging over 1000 independent trials.

1) Example 1: Verification of Statistical Analyses: In this example, we inspect the theoretical analyses of the statistical performance of the proposed RLS-based algorithm for subspace updating studied in Sections V and VI. Here we define the relative measure factor $\kappa(k)$ of closeness between the ensemble-averaged value of $E\{\mathbf{\Psi}_1^{-1}(k)\}$ and the approximated expectation $\bar{\mathbf{\Psi}}_{\mathrm{inv}}(k)$ in (39) as

$$\kappa(k) \triangleq \frac{\left\| \frac{1}{\bar{N}} \sum_{\bar{i}=1}^{\bar{N}} (\mathbf{\Psi}_1^{-1}(k))^{(\bar{i})} - \mathbf{\Psi}_{\mathrm{inv}}(k) \right\|_F^2}{\|\mathbf{\Psi}_{\mathrm{inv}}(k)\|_F^2} \quad (83)$$

where $(\mathbf{\Psi}_1^{-1}(k))^{(\bar{i})}$ denotes the inverse of $\mathbf{\Psi}_1(k)$ obtained in the $\bar{i}$th trial at the instant k, $\bar{N}$ is the number of trials, and the ensemble-averaging of $\{(\mathbf{\Psi}_1^{-1}(k))^{(\bar{i})}\}$ is used as the experimental value for the expectation $E\{\mathbf{\Psi}_1^{-1}(k)\}$. Obviously the smaller value of $\kappa(k)$ indicates the better approximation.

Firstly we examine the theoretical derivations of expectation computation of the inverse matrix and the MSD learning curve of the null space updating for several forgetting factors, where one signal impinges on the array along $\theta_1 = 5^\circ$ with the signal power $r_{s_1} = 1$ and $\mathrm{SNR} = 10$ dB (i.e., $\sigma^2 = 0.1$), and the initial value Ψ_o is set to one, while the RLS forgetting factor is set to $\gamma = 1$, 0.975, 0.9, 0.8, and 0.6. When $\gamma = 1$, the RLS algorithm has growing memory with infinite window length L_{eff}, and $\mathbf{\Psi}_1(k)$ in (15) is a function of all data $\mathbf{\Phi}_1(i)\mathbf{\Phi}_1^H(i)$ for $i = 1,\ 2, \ldots, k$ with same weight $\gamma = 1$, hence the assumption of independence between $\mathbf{\Psi}_1^{-1}(k)$ and $\mathbf{\Psi}_1(k)$ (i.e., $\mathbf{\Phi}_1(i)\mathbf{\Phi}_1^H(i)$ for $i = 1,\ 2, \ldots, k$) mentioned in Section V is valid. When $\gamma < 1$, the algorithm has exponentially growing window with a small window length L_{eff}, and it gives less weight $\gamma^{k-i}(< 1)$ to the past data $\mathbf{\Phi}_1(i)\mathbf{\Phi}_1^H(i)$ for $i < k$ and more weight $\gamma^0(= 1)$ to the current one $\mathbf{\Phi}_1(k)\mathbf{\Phi}_1^H(k)$. Thus the independence assumption between $\mathbf{\Psi}_1^{-1}(k)$ and $\mathbf{\Psi}_1(k)$ becomes weak and correspondingly it results in the degeneration of the approximation in (39) for small γ. As shown in Fig. 2, we can see that there is a good matching between $E\{\mathbf{\Psi}_1^{-1}(k)\}$ and $\mathbf{\Psi}_{\mathrm{inv}}(k)$ in (39) (i.e., (68)) for a non-vanishingly small γ and especially for large γ that is close to one, as the instant k increases. As a result, the ensemble-averaged MSD learning curves of the RLS algorithm agree well with the theoretical ones given by (42) (i.e., (57)) for large γ as shown in Fig. 3, when the instant k becomes large. Moreover, as clarified in (79) and (80), the steady-state MSD $J_{\mathrm{RLS}}^{\mathrm{SSMSD}}$ decreases with the increasing forgetting factor γ, and the minimum one is achieved at $\gamma = 1$, while the convergence speed becomes much lower for larger γ.

Next we test the convergence behavior of the proposed RLS algorithm in terms of the initialization Ψ_0, where the SNR and the forgetting factor are set to 10 dB and $\gamma = 1$, and Ψ_0 is chosen as $\Psi_0 = 1000$, 100, 10, 1, and 0.1, while the other simulation parameters are same to that above. From Fig. 4, we easily find that there is a perfect agreement between the theoretical MSD learning curves (42) [i.e., (57)] with the simulation results for the forgetting factor $\gamma = 1$, and smaller initial value Ψ_o is desirable for fast convergence speed. Note the rate of convergence is relatively insensitive to the variation of SNR when Ψ_0 is relatively small.

Then we consider the influence of the SNR on the convergence of the RLS algorithm, where the simulation parameters are similar to the above, except that $\gamma = 1$, $\Psi_0 = 1$, and the SNR is varied from 0 dB to 20B. Fig. 5 shows that the theoretical results agree well with the simulation ones, and the rate of

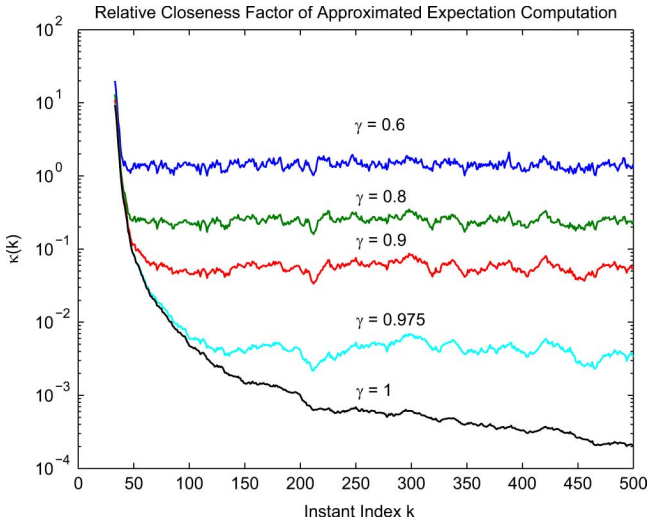

Fig. 2. Relative measure of closeness between $E\{\mathbf{\Psi}_1^{-1}(k)\}$ and $\mathbf{\Psi}_{\rm inv}(k)$ for several forgetting factors in Example 1 ($M = 16$, $p = 1$, SNR $= 10$ dB, $\Psi_0 = 1$, and $\gamma = 1$, 0.975, 0.9, 0.8, and 0.6).

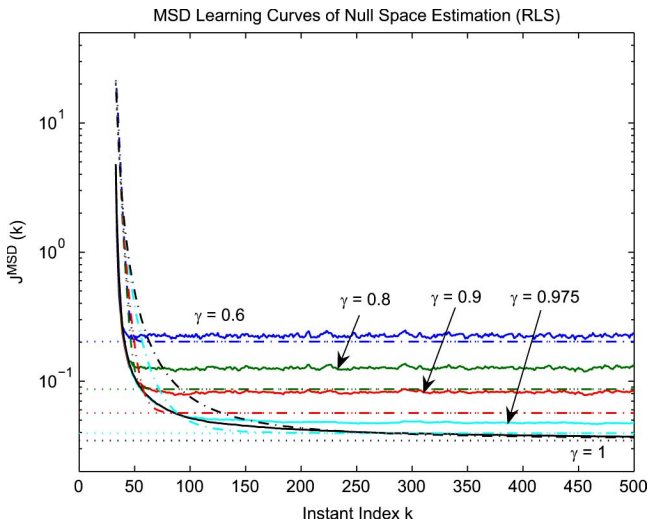

Fig. 3. MSD learning curves of null space estimation for several forgetting factors in Example 1 (solid line: ensemble-averaged MSD; dash-dot line: theoretical MSD; and dotted line: steady-state MSD; $M = 16, p = 1,$ SNR $= 10$ dB, $\Psi_0 = 1$, and $\gamma = 1$, 0.975, 0.9, 0.8, and 0.6).

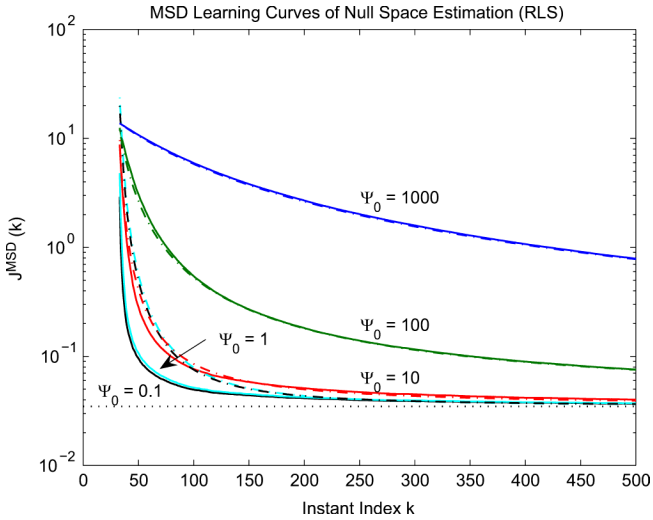

Fig. 4. MSD learning curves of null space estimation for several initial values of Ψ_0 in Example 1 (solid line: ensemble-averaged MSD; dash-dot line: theoretical MSD; and dotted line: steady-state MSD; $M = 16, p = 1,$ SNR $= 10$ dB, $\gamma = 1$, and $\Psi_0 = 1000$, 100, 10, 1 and 0.1).

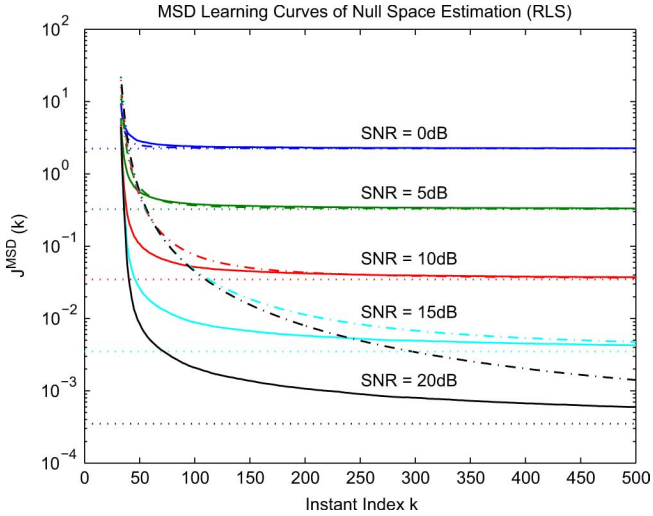

Fig. 5. MSD learning curves of null space estimation for several SNRs in Example 1 (solid line: ensemble-averaged MSD; dash-dot line: theoretical MSD; and dotted line: steady-state MSD; $M = 16$, $p = 1$, $\gamma = 1$, $\Psi_0 = 1$, and SNR $= 0$ dB, 5 dB, 10 dB, 15 dB, and 20 dB).

convergence in the mean-square sense becomes relatively fast at low SNR as described in Section VI.

Finally we compare the relative performance of the RLS algorithm and the LMS algorithm [18] for the null space estimation, where $\theta_1 = 5^\circ$ with $r_{s_1} = 1$, the SNR is set to 10 dB (i.e., $\bar{\Psi}_1 = 8.84$), and $\Psi_0 = 1$. When the forgetting factor of the RLS algorithm is chosen as $\gamma = 1$, while the step-size of the LMS algorithm is chosen as $\mu = 0.005$, 0.01, and 0.02, which satisfy the stability condition $0 < \mu < \bar{x}_{\rm up.}$, where $\bar{x}_{\rm up.} = 8.2336 \times 10^{-2}$[18]. The theoretical and ensemble-averaged MSD learning curves of the LMS algorithm are plotted and compared with that of the RLS algorithm in Fig. 6. Obviously the RLS algorithm with $\gamma = 1$ provides smaller steady-state MSD for the null space estimation than the LMS algorithm with any step-size μ as derived in (81). On the other hand, when the step-size of the LMS algorithm is chosen as $\mu = (1-\gamma)/\bar{\Psi} = 2.8281 \times 10^{-3}$, where the forgetting factor of the RLS algorithm is set to $\gamma = 0.975$, the theoretical and ensemble-averaged MSD learning curves of the LMS and RLS algorithms are depicted in Fig. 7. We can see that the LMS algorithm has the similar behavior in the steady-state as that of the RLS algorithm with $\gamma < 1$ by choosing the step-size μ properly as shown in (82), however it performs worse than the RLS algorithm during the transient phase.

2) Example 2: Tracking Performance of Proposed Amend Algorithm: Now we evaluate the performance of the proposed AMEND algorithm shown in Section IV-B for tracking the directions of coherent signals with crossings. The number of sensors is $M = 12$, and there are four coherent signals come from nonlinearly or linearly time-varying directions $\theta_1(n) \sim \theta_4(n)$ with the SNR's of 10 dB, where the initial values are -60°, -20°, 8.28°, and 60°, respectively. The directions are tracked

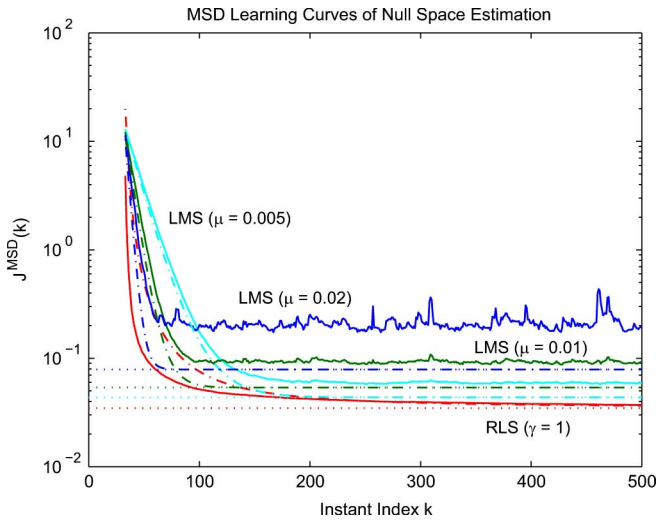

Fig. 6. MSD learning curves of the RLS and LMS algorithms in Example 1 (solid line: ensemble-averaged MSD; dash-dot line: theoretical MSD; and dotted line: steady-state MSD; $M = 16, p = 1, \mathrm{SNR} = 10$ dB, $\gamma = 1$, $\Psi_0 = 1$, and $\mu = 0.005$, 0.01, and 0.02).

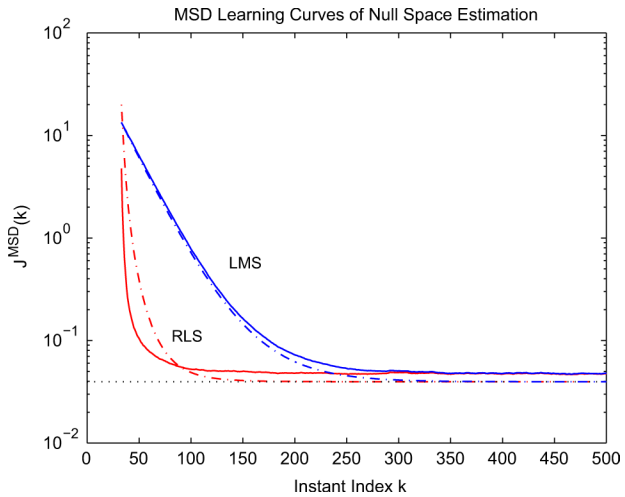

Fig. 7. MSD learning curves of the RLS and LMS algorithms in Example 1 (solid line: ensemble-averaged MSD; dash-dot line: theoretical MSD; and dotted line: steady-state MSD; $\mathrm{SNR} = 10$ dB, $\gamma = 0.975$, $\Psi_0 = 1$, and $\mu = (1-\gamma)/\bar{\Psi} = 2.8281 \times 10^{-3}$).

over 60 s with $T = 1$ second, and during each interval T, $N_s = 100$ snapshots of array data $\{\boldsymbol{y}(k)\}_{k=nN_s+1}^{(n+1)N_s}$ are measured for $n = 0, 1 \ldots, 59$, which are used to estimate the weight $\hat{\boldsymbol{P}}(k)$ and the orthogonal projector $\boldsymbol{\Pi}(k)$ at $k = (n+1)N_s$ (i.e., $\boldsymbol{\Pi}(n)$). The AMEND algorithm with the RLS-based subspace updating shown in Section IV-B (referred as RLS-AMEND) is carried out, where the forgetting factor and the initialization are set to $\gamma = 0.98$ and $\Psi_0 = 1$, and the parameters of poles of Luenberger observers as described in Remark 5 are chosen as $\bar{\kappa}_1 = 0.79$, $\bar{\psi}_1 = 10°$, $\bar{\kappa}_2 = 0.79$, $\bar{\psi}_2 = 12°$, $\bar{\kappa}_3 = 0.77$, $\bar{\psi}_5 = 14°$, $\bar{\kappa}_4 = 0.75$, $\bar{\psi}_4 = 15°$, $\bar{\kappa}_5 = 0.76$, $\bar{\psi}_5 = 13°$, $\bar{\kappa}_6 = 0.78$, and $\bar{\psi}_6 = 11°$, respectively. The NLMS-based subspace updating [18] is also performed and introduced into the AMEND algorithm (referred to as NLMS-AMEND) for comparison, where the step-size is set to $\bar{\mu} = 0.8$.

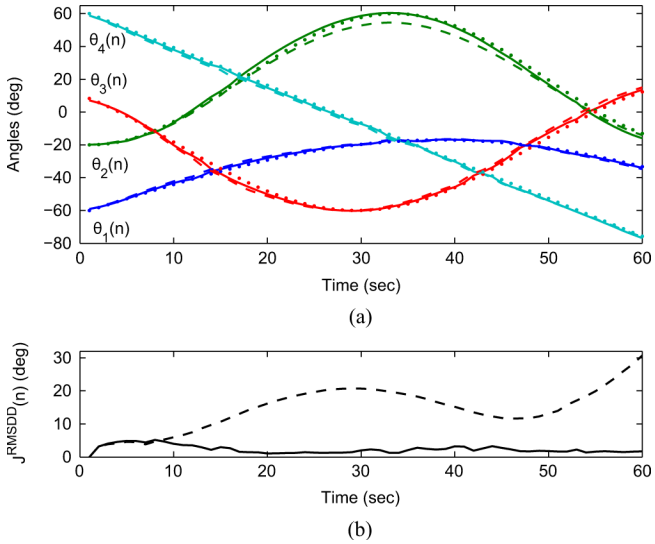

Fig. 8. (a) Tracking results for the directions of coherent signals with crossings. (b) Root-MSD learning curves of estimated directions in Example 2 (dotted line: actual values; solid line: RLS-AMEND algorithm; dashed line: NLMS-AMEND algorithm; $M = 12, p = 4, \mathrm{SNR} = 10$ dB, $N_s = 100, \gamma = 0.98$, $\Psi_0 = 1$, and $\bar{\mu} = 0.8$).

In order to measure the overall performance of estimating the directions, we define a root-MSD learning curve of estimated directions (RMSDD) as

$$J^{\mathrm{RMSDD}}(n) \triangleq \sqrt{\frac{1}{\bar{N}} \sum_{\bar{i}=1}^{\bar{N}} \sum_{i=1}^{p} \left(\hat{\theta}_i^{(\bar{i})}(n) - \theta_i(n)\right)^2} \qquad (84)$$

where $\hat{\theta}_i^{(\bar{i})}(n)$ is the estimate obtained in the $\bar{i}$th trial at the instant n. In the initial phase, by using the first N_s snapshots $\{\boldsymbol{y}(k)\}_{k=1}^{N_s}$, the number of incident signals is estimated correctly as $\hat{p} = 4$ by using the batch MENSE [34], while the initial values of directions are estimated by the batch SUMWE [2]. Then for $n = 1, 2, \ldots$, the directions are estimated and tracked by using the proposed RLS-based subspace updating and the Luenberger observer, where the eigendecomposition for subspace updating, the estimation of noise variances required by the ordinary Kalman filtering and the estimate association of directions are avoided. The trajectories of the actual directions and the estimates obtained by the proposed RLS- and NLMS-based AMEND algorithms are plotted in Fig. 8(a), while the corresponding $J^{\mathrm{RMSDD}}(n)$ learning curves are shown in Fig. 8(b). As aforementioned, the inverse matrix $\boldsymbol{\Psi}_1^{-1}(k)$ in (19) plays a similar role to the step-size of the LMS/NLMS algorithm, and it can be seen as a time-varying parameter in terms of γ, $\boldsymbol{\Psi}_o$ and "input data" $\boldsymbol{\Phi}_1(i)$ for $i = 1$ to k, while the step-size of the LMS/NLMS algorithm is constant herein. Clearly, the proposed RLS-AMEND algorithm provides remarkable tracking ability than the NLMS-based one in the multipath environment for the time-varying directions with crossings, where the estimated DOAs are always very close to the actual values.

3) Example 3: Tracking Performance of Modified Amend Algorithm: In this example, we assess the tracking performance of the modified AMEND algorithm with "self-initialization" described in Section IV-C (referred to as RLS-MAMEND) in

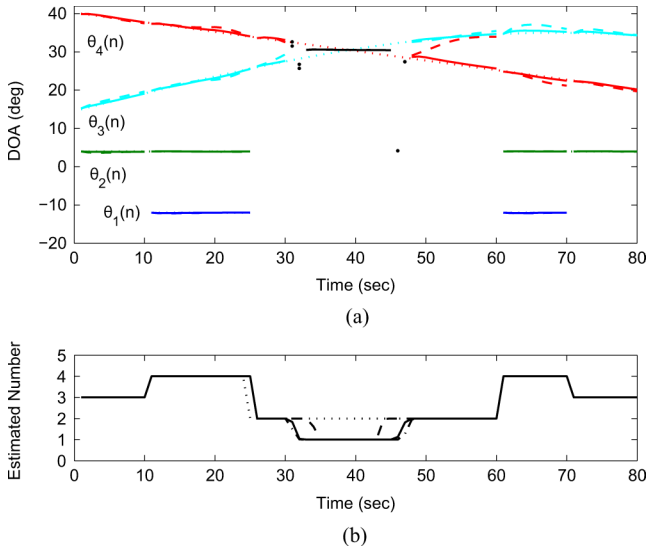

Fig. 9. (a) Tracking results for the directions of coherent signals with crossing and appearing/disappearing (dotted line: actual values; solid line: RLS-AMEND algorithm; dashed line: NLMS-AMEND algorithm). (b) Averaged estimates of the number of signals (dotted line: actual value; solid line: AMEND algorithm; dashed line: FBSS-MDL; dash-dotted line: SS-MDL) in Example 2 ($M = 12$, SNR $= 15$ dB, $N_s = 50$, $\gamma = 0.98$, $\Psi_0 = 1$, and $\bar{\mu} = 0.9$).

a difficult scenario, where four coherent signals with SNR's of 15 dB imping on the array from $\theta_1(n) \sim \theta_4(n)$, where $s_1(n)$ appears at $T = 11$ second and $T = 61$ second and disappears at $T = 25$ second and $T = 70$ second, $s_2(n)$ disappears during the interval between $T = 25$ second and $T = 60$ second, while $\theta_3(n)$ and $\theta_4(n)$ are nonlinearly or linearly time-varying with crossing as shown in Fig. 9(a), where the initial values of directions are $-12°$, $4°$, $15°$ and $40°$, respectively. The other simulation parameters are similar to that of the previous example, except that $N_s = 50$ and $\bar{\mu} = 0.9$, where the NLMS-based subspace updating [18] is also introduced into the modified AMEND algorithm (referred to as NLMS-MAMEND) for comparison.

The performance for tracking the directions and that for estimating the number of incident signals are depicted in Figs. 9(a) and 9(b), respectively, where the QR decomposition with *a priori* column pivoting (QRPP) is used in the MENSE to improve the detection performance (cf. [34] for details). The most popular eigenvalue-based minimum description length (MDL) criterion [41] with the spatial smoothing (SS) and forward-backward SS (FBSS) preprocessing [42] is carried out, where the instantaneous array covariance matrix at the instant n is calculated from the N snapshots $\{\boldsymbol{y}(k)\}_{k=nN_s+1}^{(n+1)N_s}$ and the subarray size is chosen as $m = 6$, and the averaged estimates of the signal number are also plotted in Fig. 9(b) for reference. When two incident signals (i.e., $\theta_3(n)$ and $\theta_4(n)$) become close, the ranks of $\boldsymbol{A}(\theta(n))$ will begin to collapse and result in the fact that only one signal will be detected. As shown in Fig. 9(b), the proposed MAMEND algorithm has similar detection performance to the SS-based MDL method with eigendecomposition, and it can estimate the decreasing and increasing number of incident signals accurately and immediately. Hence by monitoring this change of the number of incident signal during each interval, the proposed RLS-MAMEND algorithm can initialize the RLS-based subspace updating and the Luenberger observer automatically, and we easily find that the directions can be tracked well by using the proposed RLS-MAMEND algorithm. Note that there are some stray estimates of directions in the cross region shown in Fig. 9(a) due to the spurious estimate of the number of incident signals as shown in Fig. 9(b).

VIII. Conclusion

In many applications of array processing, the directions of incident signals may be time-varying with crossings, and some incident signals may appear and disappear sometimes. In this paper, a new computationally simple and efficient subspace-based adaptive method was proposed for estimating DOAs of multiple coherent narrowband signals impinging on a ULA. Especially in the proposed AMEND algorithm, the null space is estimated by using the RLS algorithm, and the direction tracking is accomplished by employing the Luenberger observer, where the eigendecomposition for subspace updating, the estimation of noise variances required by the ordinary Kalman filtering and the estimate association of directions are avoided. The statistical performance of the RLS algorithm in stationary environment was analyzed in the mean and mean-squares senses, and further the MSE and MSD learning curves were derived explicitly. Moreover an analytical study of the RLS algorithm was carried out for one incident signal, where the effects of the forgetting factor, the initialization and the SNR on the convergence were examined, and the relative performance of the RLS and LMS algorithms in the steady-state was discussed. Finally the effectiveness of the proposed algorithm and the theoretical derivations were verified and substantiated through numerical examples.

Appendix
Computation of Observer Gain

By letting the desired pole locations of the $\boldsymbol{F} - \boldsymbol{g}_i\boldsymbol{c}^T$ in the z-plane be z_{i1}, z_{i2}, and z_{i3} and defining $\boldsymbol{g}_i \triangleq [g_{i1}, g_{i2}, g_{i3}]^T$ for $i = 1, 2, \ldots, p$, from (5), the characteristic equation for the error dynamics in (25) is obtained

$$\begin{aligned}&\left|z\boldsymbol{I}_3 - (\boldsymbol{F} - \boldsymbol{g}_i\boldsymbol{c}^T)\right| \\ &\quad = (z-1)^2(z + g_{i1} - 1) \\ &\qquad + (z-1)(0.5g_{i3}T^2 + g_{i2}T) + g_{i3}T = 0 \end{aligned} \tag{A1}$$

while the desired observer characteristic equation is given by

$$\begin{aligned}&(z - z_{i1})(z - z_{i2})(z - z_{i3}) \\ &\quad = z^3 + \alpha_{i1}z^2 + \alpha_{i2}z + \alpha_{i3} = 0 \end{aligned} \tag{A2}$$

where $\alpha_{i1} = -(z_{i1} + z_{i2} + z_{i3})$, $\alpha_{i2} = z_{i1}z_{i2} + z_{i1}z_{i3} + z_{i2}z_{i3}$, and $\alpha_{i3} = z_{i1}z_{i2}z_{i3}$.

Then by comparing the coefficients of equal powers of z in (A1) and (A2) and after some simple manipulations, we can obtain the elements of gain $\boldsymbol{g}_i$ as

$$g_{i1} = 3 + \alpha_{i1} \tag{A3}$$

$$g_{i2} = \frac{1}{2T}(5 + 3\alpha_{i1} + \alpha_{i2} - \alpha_{i3}) \tag{A4}$$

$$g_{i3} = \frac{1}{T^2}(1 + \alpha_{i1} + \alpha_{i2} + \alpha_{i3}). \tag{A5}$$

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the Associate Editor Prof. M. Greco for their careful review, helpful comments, and valuable suggestions.

REFERENCES

[1] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, 1996.

[2] J. Xin and A. Sano, "Computationally efficient subspace-based method for direction-of-arrival estimation without eigendecomposition," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 876–893, 2004.

[3] Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Trans. Autom. Control*, vol. 23, pp. 618–626, 1978.

[4] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.

[5] C.-B. Chang and J. A. Tabaczynski, "Application of state estimation to target tracking," *IEEE Trans. Autom. Control*, vol. 29, pp. 98–109, 1984.

[6] C. K. Sword, M. Simaan, and E. W. Kamen, "Multiple target angle tracking using sensor array outputs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, pp. 367–373, 1990.

[7] C. R. Sastry, E. W. Kamen, and M. Simaan, "An efficient algorithm for tracking the angles of arrival of moving targets," *IEEE Trans. Signal Process.*, vol. 39, no. 1, pp. 242–246, 1991.

[8] K. W. Lo and C. K. Li, "An improved multiple target angle tracking algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, pp. 797–805, 1992.

[9] C. R. Rao, L. Zhang, and L. C. Zhao, "Multiple target angle tracking using sensor array outputs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, pp. 268–271, 1993.

[10] S. B. Park, C. S. Ryu, and K. K. Lee, "Multiple target angle tracking algorithm using predicted angles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, pp. 643–648, 1994.

[11] C. R. Rao, C. R. Sastry, and B. Zhou, "Tracking the direction of arrival of multiple moving targets," *IEEE Trans. Signal Process.*, vol. 42, no. 5, pp. 1133–1144, 1994.

[12] J. M. Goldberg, "Joint direction-of-arrival and array shape tracking for multiple moving targets," *IEEE J. Ocean. Eng.*, vol. 23, pp. 118–126, 1998.

[13] L. Frenkel and M. Feder, "Recursive expectation-maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking," *IEEE Trans. Signal Process.*, vol. 47, no. 2, pp. 306–320, 1999.

[14] Y. Zhou, P. C. Yip, and H. Leung, "Tracking the direction-of-arrival of multiple moving targets by passive arrays: Algorithm," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2655–2666, 1999.

[15] R. E. Zarnick, K. L. Bell, and H. L. Van Trees, "A unified method for measurement and tracking of contacts from an array of sensors," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 2950–2961, 2001.

[16] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, 1995.

[17] J. Sanchez-Araujo and S. Marcos, "An efficient PASTD-algorithm implementation for multiple direction of arrival tracking," *IEEE Trans. Signal Process.*, vol. 47, no. 8, pp. 2321–2324, 1999.

[18] J. Xin and A. Sano, "Efficient subspace-based algorithm for adaptive bearing estimation and tracking," *IEEE Trans. Signal Process.*, vol. 53, no. 12, pp. 4485–4505, 2005.

[19] S. Marcos, A. Marsal, and M. Benidir, "The propagator method for source bearing estimation," *Signal Process.*, vol. 42, pp. 121–138, 1995.

[20] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Trans. Autom. Control*, vol. 15, pp. 175–184, 1970.

[21] P. Gutman and M. Velger, "Tracking targets using adaptive Kalman filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, pp. 691–699, 1990.

[22] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 10, pp. 1783–1795, 1990.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME-J. Basic Eng.*, vol. 82, pp. 35–45, 1960.

[24] D. G. Luenberger, "Observing the state of a linear system," *IEEE Trans. Mil. Electron.*, vol. 8, pp. 74–80, 1964.

[25] D. G. Luenberger, "An introduction to observers," *IEEE Trans. Autom. Control*, vol. 16, pp. 596–602, 1971.

[26] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[27] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[28] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and Adaptive Signal Processing—Spectral Estimation, Signal Modelling, Adaptive Filtering and Array Processing*. New York: McGraw-Hill, 2000.

[29] K. Ogata, *Discrete-Time Control Systems*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[30] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Melon Park: Addison-Wesley Longman, 1998.

[31] S. H. Ardalan and S. T. Alexander, "Fixed-point roundoff error analysis of the exponentially windowed RLS algorithm for time-varying systems," *IEEE Trans. Acoust., Specch, Signal Process.*, vol. 35, no. 6, pp. 770–783, 1987.

[32] N. E. Hubing and S. T. Alexander, "Statistical analysis of initialization methods for RLS adaptive filters," *IEEE Trans. Signal Process.*, vol. 38, no. 8, pp. 1793–1804, 1991.

[33] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," in *Proc. RADC Spectrum Estimation Workshop*, Rome, NY, Oct. 1979, pp. 243–258.

[34] J. Xin, N. Zheng, and A. Sano, "Simple and efficient nonparametric method for estimating the number of signals without eigendecomposition," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1405–1420, 2007.

[35] H. Yan and H. H. Fan, "Signal-selective DOA tracking for wideband cyclostationary sources," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2007–2015, 2007.

[36] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. Autom. Control*, vol. 22, pp. 551–575, 1977.

[37] C. G. Samson and V. U. Reddy, "Fixed point error analysis of the normalized ladder algorithm," *IEEE Trans. Acoust., Specch, Signal Process.*, vol. 31, no. 5, pp. 1177–1191, 1983.

[38] P. S. R. Diniz, M. L. R. de Campos, and A. Antoniou, "Analysis of LMS-Newton adaptive filtering algorithms with variable convergence factor," *IEEE Trans. Signal Process.*, vol. 43, no. 3, pp. 617–627, 1995.

[39] J. Xin, N. Zheng, and A. Sano, "On-line detection of the number of narrowband signals with a uniform linear array," in *Proc. 2008 Eur. Signal Process. Conf.*, Lausanne, Switzerland, Aug. 25–29, 2008, Paper #1569102006.

[40] G. Xu, H. Zha, G. Golud, and T. Kailath, "Fast algorithms for updating signal subspaces," *IEEE Trans. Circuits Syst.—II*, vol. 41, no. 8, pp. 537–549, 1994.

[41] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 387–392, 1985.

[42] G. Xu, R. H. Roy, III, and T. Kailath, "Detection of number of sources via exploitation of centro-symmetry property," *IEEE Trans. Signal Process.*, vol. 42, no. 1, pp. 102–112, 1994.

[43] A. Eriksson, P. Stoica, and T. Söderström, "On-line subspace algorithms for tracking moving sources," *IEEE Trans. Signal Process.*, vol. 42, no. 9, pp. 2319–2330, 1994.

[44] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, pp. 806–811, 1985.

[45] S. U. Pillai and B. H. Kwon, "Forward/backward spatial smoothing techniques for coherent signals identification," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 1, pp. 8–15, 1989.

[46] L. Huang, X. Li, L. Cai, and S. Wu, "Recursion subspace-based method for bearing estimation," in *Proc. 3rd Int. Worshop on Signal Design and Its Appl. Commun.*, Chengdu, China, Sep. 23–27, 2007, pp. 190–193.

[47] A. Satish and R. L. Kashyap, "Multiple target tracking using maximum likelihood principle," *IEEE Trans. Signal Process.*, vol. 43, no. 7, pp. 1677–1695, 1995.

Jingmin Xin (S'92–M'96–SM'06) received the B.E. degree in information and control engineering from Xi'an Jiaotong University, Xi'an, China, in 1988, and the M.E. and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1993 and 1996, respectively.

From 1988 to 1990, he was with the Tenth Institute of Ministry of Posts and Telecommunications (MPT) of China, Xi'an. He was with the Communications Research Laboratory, Japan, as an Invited Research Fellow of the Telecommunications Advancement Organization of Japan (TAO) from 1996 to 1997 and as a Postdoctoral Fellow of the Japan Science and Technology Corporation (JST) from 1997 to 1999. He was also a Guest (Senior) Researcher with YRP Mobile Telecommunications Key Technology Research Laboratories Company, Limited, Yokosuka, Japan, from 1999 to 2001. From 2002 to 2007, he was with Fujitsu Laboratories Limited, Yokosuka, Japan. Since 2007, he has been a Professor at Xi'an Jiaotong University. His research interests are in the areas of adaptive filtering, statistical and array signal processing, system identification, and pattern recognition.

Nanning Zheng (SM'93–F'06) graduated from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China, in 1975, and received the M.S. degree in information and control engineering from Xi'an Jiaotong University in 1981 and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985.

He joined Xi'an Jiaotong University in 1975, and is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, pattern recognition and image processing, and hardware implementation of intelligent systems.

Dr. Zheng became a member of the Chinese Academy of Engineering in 1999. He is the Chinese Representative on the Governing Board of the International Association for Pattern Recognition. He also serves as an executive deputy editor of the *Chinese Science Bulletin*.

Akira Sano (M'89) received the B.E., M.E., and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Tokyo, Japan, in 1966, 1968, and 1971, respectively.

In 1971, he joined the Department of Electrical Engineering, Keio University, Yokohama, Japan, where he is currently a Professor with the Department of System Design Engineering. He was a Visiting Research Fellow with the University of Salford, Salford, U.K., from 1977 to 1978. His current research interests are in adaptive modeling and design theory in control, signal processing and communication, and applications to control of sounds and vibrations, mechanical systems, and mobile communication systems. He is a coauthor of the textbook *State Variable Methods in Automatic Control* (New York: Wiley, 1988).

Dr. Sano received the Kelvin Premium from the Institute of Electrical Engineering in 1986. He is a Fellow of the Society of Instrument and Control Engineers and is a Member of the Institute of Electrical Engineering of Japan and the Institute of Electronics, Information and Communications Engineers of Japan. He was General Co-chair of 1999 IEEE Conference of Control Applications and an IPC Chair of 2004 IFAC Workshop on Adaptation and Learning in Control and Signal Processing. He served as Chair of IFAC Technical Committee on Modeling and Control of Environmental Systems from 1996 to 2001. He has also been Vice Chair of IFAC Technical Committee on Adaptive Control and Learning since 1999 and has been Chair of IFAC Technical Committee on Adaptive and Learning Systems since 2002. He was also on the Editorial Board of *Signal Processing*.