SUBSPACE-BASED ADAPTIVE DIRECTION ESTIMATION AND TRACKING IN MULTIPATH ENVIRONMENT

Jingmin Xin †, *Naoyuki Hirosaki* ‡, *Hiroyuki Tsuji* §, *Yoji Ohashi* †, *and* *Akira Sano* ‡

† Mobile Communications Development Labs., Fujitsu Laboratories Ltd., Yokosuka 239-0847, Japan
‡ Department of System Design Engineering, Keio University, Yokohama 223-8522, Japan
§ National Institute of Information and Communications Technology, Yokosuka 239-0847, Japan

ABSTRACT

A new computationally efficient subspace-based algorithm is proposed for estimating and tracking the directions of coherent narrowband signals impinging on a uniform linear array (ULA). Specifically the null space is estimated using the least-mean-square (LMS) or normalized LMS (NLMS) algorithm, and the directions are updated using the approximate Newton method. By studying the convergence analyses of the LMS and NLMS algorithms, where the "weight" is in the form of a matrix and there is a correlation between the "additive noise" and "input data" in the updating equation, the step-size stability conditions are derived explicitly. Further the tracking of crossing directions of moving signals is considered. The theoretical analyses and effectiveness of the proposed algorithm are verified.

1. INTRODUCTION

Although subspace-based methods have been extensively studied for direction-of-arrival (DOA) estimation because of their high resolution and computational simplicity, the heavy computational load of eigendecomposition which is usually required for subspace estimation makes subspace-based methods difficult to implement in an on-line manner. Recently some computationally simple subspace-based methods such as the BEWE [1], OPM [2], and SWEDE [3] have been proposed for estimating the directions of narrowband signals efficiently, where the need for computation of eigendecomposition is avoided, and the exact signal/noise subspace is obtained from the array data based on a partition of the array response matrix. Additionally the on-line implementations of the SWEDE and PASTd [8] were considered [3], [4]. However, the performance of these methods degrades severely when the incident signals are coherent (i.e., fully correlated), and/or the signal-to-noise ratio (SNR) is low. By exploiting the array geometry of a uniform linear array (ULA) and its shift-invariance property, we proposed a subspace-based method without eigendecomposition (SUMWE) with good performance in batch mode [5].

This paper investigates a new on-line algorithm for estimating and tracking the coherent signals impinging on a ULA based on the SUMWE, where the least-mean-square (LMS) or normalized LMS (NLMS) algorithm is used for the null space estimation, and the approximate Newton iteration method is used to update the direction finding. The proposed algorithm has the reduced computational load and a remarkable insensitivity to the correlation of signals. By analyzing the transient behaviors of the LMS and NLMS algorithms, where the "weight" is in the form of a matrix and there is a correlation between the "additive noise" and "input data" that involve the instantaneous correlations of the received array data, the step-sizes that guarantee the mean and mean-square stabilities are derived. Further the tracking of crossing directions is considered by introducing a dynamic model of the incident directions. The theoretical analyses and effectiveness of proposed algorithm are substantiated through numerical examples.

2. DATA MODEL AND BASIC ASSUMPTIONS

Consider a ULA of M identical and omnidirectional sensors with adjacent spacing d and assume that p ($p < M/2$) narrowband signals $\{s_i(t)\}$ are in the far-field and impinge from distinct directions $\{\theta_i(t)\}$. The received signal $y_m(t)$ at the mth sensor is given by

$$y_m(t) = \sum_{i=1}^{p} s_i(t)\, e^{j\omega_0(m-1)\tau(\theta_i(t))} + w_m(t) \tag{1}$$

where $w_m(t)$ is the additive noise, $\omega_0 = 2\pi f_0$, $\tau(\theta_i(t)) = (d/c)\sin\theta_i(t)$, and c and f_0 are the propagation speed and center frequency. Then we have a compact data model

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

where $\boldsymbol{y}(t)$, $\boldsymbol{s}(t)$, and $\boldsymbol{w}(t)$ are the vectors of the received signals, incident signals, and additive noise, and $\boldsymbol{A}(\theta(t))$ is the array response matrix given by $\boldsymbol{A}(\theta(t)) = [\boldsymbol{a}(\theta_1(t)), \cdots, \boldsymbol{a}(\theta_p(t))]$, where $\boldsymbol{A}(\theta(t))$ is unambiguous, and $\boldsymbol{a}(\theta_i(t)) = [1, e^{j\omega_0\tau(\theta_i(t))}, \cdots, e^{j\omega_0(M-1)\tau(\theta_i(t))}]^T$.

Here $\{s_i(t)\}$ are coherent signals and expressed as $s_i(t) = \beta_i s_1(t)$ for $i = 1, 2, \cdots, p$, where $\{\beta_i\}$ are the attenuation coefficients with $\beta_i \neq 0$ and $\beta_1 = 1$, and $s_1(t)$ is a temporally complex white Gaussian random process with zero-mean and the variance given by $E\{s_1(n)s_1^*(t)\} = r_s\delta_{n,t}$ and $E\{s_1(n)s_1(t)\} = 0$, while $w_m(t)$ is a temporally and spatially complex white Gaussian random process with zero-mean and the covariance matrix given by $E\{w_m(n)w_k^*(t)\} = \sigma^2\delta_{m,k}\delta_{n,t}$ and $E\{w_m(n)w_k(t)\} = 0$, where $E\{\cdot\}$, $(\cdot)^*$, and $\delta_{n,t}$ denote the expectation, complex conjugate, and Kronecker delta. The additive noise

$\{w_m(t)\}$ are uncorrelated with the signals $\{s_i(t)\}$, and the number of signals p is known or has been estimated.

For tracking the time-varying directions, we assume that $\theta_i(t)$ is slowly time-varying (relative to the sampling rate $1/T_s$ [3]) so that $\theta_i(t) \approx \theta_i(nT)$ for $t \in [nT, (n+1)T)$ and $n = 0, 1, \cdots$ and that N snapshots of array data are available over an interval T of parameter updating, i.e., $T = NT_s$. Hence the direction tracking is formulated as the estimating $\theta_i(n)$ for $n = 0, 1, \cdots$ from N snapshots of $\{\boldsymbol{y}(k)\}$ measured at $k = nN, nN+1, \cdots, (n+1)N-1$ while maintaining the correct association between $\theta_i(n)$ and $\theta_i(n-1)$.

3. ADAPTIVE DOA ESTIMATION AND TRACKING

3.1. Direction Estimation without Eigendecomposition

By defining $\bar{\boldsymbol{A}}$ and $\boldsymbol{A}_1$ be the submatrices of $\boldsymbol{A}(\theta(t))$ in (2) consisting of the first $M-p$ and p rows, since $M > 2p$, we can partition $\bar{\boldsymbol{A}}$ into two parts as [2], [5]

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \begin{matrix} \}p \\ \}M-2p \end{matrix} . \tag{3}$$

As these matrices have full column ranks, the rows of $\boldsymbol{A}_2$ can be expressed as a linear combination of that of $\boldsymbol{A}_1$ [5]

$$\boldsymbol{P}^H \boldsymbol{A}_1 = \boldsymbol{A}_2 \qquad \text{i.e.,} \qquad \boldsymbol{Q}^H \bar{\boldsymbol{A}} = \boldsymbol{O}_{(M-2p)\times p} \tag{4}$$

where $\boldsymbol{P}$ is a linear operator (i.e., "weight" hereafter), and $\boldsymbol{Q} = [\boldsymbol{P}^T, -\boldsymbol{I}_{M-2p}]^T$, while $\boldsymbol{O}_{m\times q}$, $\boldsymbol{I}_m$, and $(\cdot)^H$ denote the $m \times q$ null matrix, $m \times m$ identity matrix, and the Hermitian transpose. Clearly the columns of $\boldsymbol{Q}$ form a basis for the null space of $\bar{\boldsymbol{A}}^H$, and $\{\theta_i(n)\}$ can be estimated by minimizing the cost function [5]

$$f(\theta) = \bar{\boldsymbol{a}}^H(\theta) \boldsymbol{\Pi}_Q \, \bar{\boldsymbol{a}}(\theta) \tag{5}$$

where $\bar{\boldsymbol{a}}(\theta) = [1, e^{j\omega_0\tau(\theta)}, \cdots, e^{j\omega_0(L-2)\tau(\theta)}]^T$, and $\boldsymbol{\Pi}_Q = \boldsymbol{Q}(\boldsymbol{Q}^H\boldsymbol{Q})^{-1}\boldsymbol{Q}^H$.

3.2. Null Space Estimation with LMS/NLMS Algorithm

By dividing the array into L overlapping subarrays with p sensors forwards and backwards and by defining the signal vectors of the lth forward and backward subarrays as $\boldsymbol{y}_{fl}(k) = [y_l(k), y_{l+1}(k), \cdots, y_{l+p-1}(k)]^T$ and $\boldsymbol{y}_{bl}(k) = [y_{M-l+1}(k), y_{M-l}(k), \cdots, y_{L-l+1}(k)]^H$, we obtain four instantaneous Hankel correlation matrices at the instant k [6]

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

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

where $\boldsymbol{Y}_f(k) = [\boldsymbol{y}_{f1}(k), \cdots, \boldsymbol{y}_{fL-1}(k)]^T$, $\bar{\boldsymbol{Y}}_f(k) = [\boldsymbol{y}_{f2}(k), \cdots, \boldsymbol{y}_{fL}(k)]^T$, $\boldsymbol{Y}_b(k) = [\boldsymbol{y}_{b1}(k), \cdots, \boldsymbol{y}_{bL-1}(k)]^T$, $\bar{\boldsymbol{Y}}_b(k) = [\boldsymbol{y}_{b2}(k), \cdots, \boldsymbol{y}_{bL}(k)]^T$, and $L = M - p + 1$. From (3), the $(M-p) \times p$ matrices in (6) and (7) can be parted into the $p \times p$ and $(M-2p) \times p$ submatrices in the downward direction (e.g., $\boldsymbol{\Phi}_f(k) = [\boldsymbol{\Phi}_{f1}^T(k), \boldsymbol{\Phi}_{f2}^T(k)]^T$), and we get [6]

$$\boldsymbol{\Phi}_2(k) = \boldsymbol{P}^H \boldsymbol{\Phi}_1(k) + \boldsymbol{E}_o^H(k) \tag{8}$$

where $\boldsymbol{\Phi}_1(k) = [\boldsymbol{\Phi}_{f1}(k), \bar{\boldsymbol{\Phi}}_{f1}(k), \boldsymbol{\Phi}_{b1}(k), \bar{\boldsymbol{\Phi}}_{b1}(k)]$, $\boldsymbol{\Phi}_2(k) = [\boldsymbol{\Phi}_{f2}(k), \bar{\boldsymbol{\Phi}}_{f2}(k), \boldsymbol{\Phi}_{b2}(k), \bar{\boldsymbol{\Phi}}_{b2}(k)]$, $\boldsymbol{E}_o(k) = -\boldsymbol{G}^H\boldsymbol{Q}$, $\boldsymbol{G} = [y_M^*(k)\boldsymbol{W}_f, y_1^*(k)\bar{\boldsymbol{W}}_f, y_1(k)\boldsymbol{W}_b, y_M(k)\bar{\boldsymbol{W}}_b]$, $\boldsymbol{W}_f = [\boldsymbol{w}_{f1}(k), \cdots, \boldsymbol{w}_{fL-1}(k)]^T$, $\bar{\boldsymbol{W}}_f = [\boldsymbol{w}_{f2}(k), \cdots, \boldsymbol{w}_{fL}(k)]^T$, $\boldsymbol{W}_b = [\boldsymbol{w}_{b1}(k), \cdots, \boldsymbol{w}_{bL-1}(k)]^T$, $\bar{\boldsymbol{W}}_b = [\boldsymbol{w}_{b2}(k), \cdots, \boldsymbol{w}_{bL}(k)]^T$, $\boldsymbol{w}_{fl}(k) = [w_l(k), \cdots, w_{l+p-1}(k)]^T$, and $\boldsymbol{w}_{bl}(k) = [w_{M-l+1}(k), \cdots, w_{L-l+1}(k)]^H$.

Thus we easily obtain the LMS and NLMS algorithms for updating $\boldsymbol{P}(k)$ respectively [6]

$$\boldsymbol{P}(k) = \boldsymbol{P}(k-1) + \mu \boldsymbol{\Phi}_1(k) \boldsymbol{E}(k) \tag{9}$$

$$\boldsymbol{P}(k) = \boldsymbol{P}(k-1) + \bar{\mu}\, \mathrm{inv}\{\tilde{\boldsymbol{R}}\} \tilde{\boldsymbol{Q}}^H \boldsymbol{\Phi}_1(k) \boldsymbol{E}(k) \tag{10}$$

where $\boldsymbol{E}(k) = \boldsymbol{\Phi}_2^H(k) - \boldsymbol{\Phi}_1^H(k)\boldsymbol{P}(k-1)$, $\tilde{\boldsymbol{Q}}$ and $\tilde{\boldsymbol{R}}$ are the unitary and upper-triangular matrices of the QR decomposition of $\boldsymbol{\Phi}_1(k)\boldsymbol{\Phi}_1^H(k)$, μ and $\bar{\mu}$ are the LMS and NLMS positive step-sizes, and $\mathrm{inv}\{\cdot\}$ denotes the inversion operation of the bracketed matrix with a simple back-substitution. From (9) or (10), then the instantaneous orthogonal projector $\boldsymbol{\Pi}(k)$ of $\boldsymbol{\Pi}_Q$ in (5) is obtained [5], [6]

$$\boldsymbol{\Pi}(k) = \boldsymbol{Q}(k)(\boldsymbol{I}_{M-2p} - \boldsymbol{P}^H(k)\mathrm{inv}\{\bar{\boldsymbol{R}}\}\bar{\boldsymbol{Q}}^H\boldsymbol{P}(k))\boldsymbol{Q}^H(k) \tag{11}$$

where $\bar{\boldsymbol{Q}}$ and $\bar{\boldsymbol{R}}$ are the unitary and upper-triangular matrices of the QR decomposition of $\boldsymbol{P}(k)\boldsymbol{P}^H(k) + \boldsymbol{I}_p$.

3.3. Direction Finding with Newton Iteration Method

Now by considering the Taylor series expansion of $f(\theta)$ in (5) and by using (11), we get the approximate Newton iteration formula for direction updating [5], [6]

$$\tilde{\theta}_i(n) = \hat{\theta}_i(n-1) - \left.\frac{\mathrm{Re}\{\bar{\boldsymbol{d}}^H(\theta)\boldsymbol{\Pi}(n)\bar{\boldsymbol{a}}(\theta)\}}{\bar{\boldsymbol{d}}^H(\theta)\boldsymbol{\Pi}(n)\bar{\boldsymbol{d}}(\theta)}\right|_{\theta=\hat{\theta}_i(n-1)} \tag{12}$$

where $\bar{\boldsymbol{d}}(\theta) = j\omega_0(d/c)\cos\theta[0, e^{j\omega_0\tau(\theta)}, 2e^{j2\omega_0\tau(\theta)}, \cdots, (L-2)e^{j\omega_0(L-2)\tau(\theta})]^T$, and $\boldsymbol{\Pi}(n) = \boldsymbol{\Pi}(k)|_{k=(n+1)N-1}$.

3.4. Tracking of Crossing Directions with Luenberger Observer

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 $\boldsymbol{x}_i(n) = [\theta_i(n), \dot{\theta}_i(n), \ddot{\theta}_i(n)]^T$, in the absence of process and measurement noises, we have the dynamics and measurement equations of $\theta_i(n)$

$$\boldsymbol{x}_i(n+1) = \boldsymbol{F}\boldsymbol{x}_i(n), \qquad \theta_i(n) = \boldsymbol{c}^T\boldsymbol{x}_i(n) \tag{13}$$

where $\boldsymbol{F}$ and $\boldsymbol{c}$ are the transition matrix and measurement vector given by $\boldsymbol{F} = [1, T, 0.5T^2; 0, 1, T; 0, 0, 1]$ and $\boldsymbol{c} = [1, 0, 0]^T$. Here the estimate $\tilde{\theta}_i(n)$ obtained by (12) is treated as the "measurement", then the estimate $\hat{\boldsymbol{x}}_i(n|n)$ of the state of this dynamical system is obtained

$$\hat{\boldsymbol{x}}_i(n|n) = \boldsymbol{F}\hat{\boldsymbol{x}}_i(n-1|n-1) + \boldsymbol{g}_i(\tilde{\theta}_i(n) - \boldsymbol{c}^T\hat{\boldsymbol{x}}_i(n-1|n-1)) \tag{14}$$

where $\boldsymbol{g}_i$ is called the observer gain.

Further from (13) and (14), we get the recursion of the predicted state-error vector of $\boldsymbol{x}_i(n)$ at the instant n

$$\boldsymbol{\xi}_i(n) = (\boldsymbol{F} - \boldsymbol{g}_i \boldsymbol{c}^T)\boldsymbol{\xi}_i(n-1) \tag{15}$$

where $\boldsymbol{\xi}_i(n) = \boldsymbol{x}_i(n) - \hat{\boldsymbol{x}}_i(n|n)$. Obviously 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, then $\boldsymbol{\xi}_i(n) \to \mathbf{0}$ as $n \to \infty$. Here this convergence condition can be satisfied by designing the observer gains $\{\boldsymbol{g}_i\}$ with the pole assignment.

3.5. On-Line Implementation of Algorithm

1) Predict the directions by using (13) as $\hat{\boldsymbol{x}}_i(n|n-1) = \boldsymbol{F}\hat{\boldsymbol{x}}_i(n-1|n-1)$ and $\hat{\theta}_i(n|n-1) = \boldsymbol{c}^T\hat{\boldsymbol{x}}_i(n|n-1)$. .... 18 *flops*

2) With the N snapshots of $\{\boldsymbol{y}(k)\}_{k=nN}^{(n+1)N-1}$, calculate the correlation vectors $\boldsymbol{\varphi}(k)$ between $\boldsymbol{y}(k)$ and $y_M^*(k)$ and $\bar{\boldsymbol{\varphi}}(k)$ between $\boldsymbol{y}(k)$ and $y_1^*(k)$ as

$$\boldsymbol{\varphi}(k) = \boldsymbol{y}(k)y_M^*(k), \quad \bar{\boldsymbol{\varphi}}(k) = \boldsymbol{y}(k)y_1^*(k) \tag{16}$$

where $\boldsymbol{\varphi}(k) = [\hat{r}_{1M}(k), \cdots, \hat{r}_{MM}(k)]^T$, and $\bar{\boldsymbol{\varphi}}(k) = [\hat{r}_{11}(k), \cdots, \hat{r}_{M1}(k)]^T$, and form the Hankel matrices $\boldsymbol{\Phi}_f(k)$, $\bar{\boldsymbol{\Phi}}_f(k)$, $\boldsymbol{\Phi}_b(k)$, and $\bar{\boldsymbol{\Phi}}_b(k)$. .... $12M$ *flops*

3) Then update $\boldsymbol{P}(k)$ by using (9) (LMS) or (10) (NLMS), and calculate the projector $\boldsymbol{\Pi}(k)$ by using (11), where $\boldsymbol{\Pi}((n+1)N-1)$ is denoted as $\boldsymbol{\Pi}(n)$. ... $2(M-2p)p(32p+5) + 8(M-2p)(M(M-2p) + (M-p)^2 + 2p^2) + 33p^3 + 31p^2 + 25p + \kappa$ *flops*

4) Estimate $\tilde{\theta}_i(n)$ by using (12), where $\hat{\theta}_i(n-1)$ is replaced with $\hat{\theta}_i(n|n-1)$. ..... $16(M-p)(M-p+1)$ *flops*

5) Refine the state vector with the observer as $\hat{\boldsymbol{x}}_i(n|n) = \hat{\boldsymbol{x}}_i(n|n-1) + \boldsymbol{g}_i(\tilde{\theta}_i(n) - \hat{\theta}_i(n|n-1))$, and estimate $\tilde{\theta}_i(n)$ from $\hat{\theta}_i(n|n) = \boldsymbol{c}^T\hat{\boldsymbol{x}}_i(n|n-1)$. .... 6 *flops*

The computational complexity of each step is roughly indicated in terms of the number of MATLAB flops, where $\kappa = 0$ or $\kappa = 97p^3 - 11p^2 + 22p$ for the LMS or NLMS. Additionally the first $K_0 = 2M$ snapshots of the received data are accumulated for the off-line SUMWE [5] to provide the initial values of directions $\{\hat{\theta}_i(n-1|n-1)\}$, the LMS and NLMS algorithms are initialized by $\boldsymbol{P}(K_0) = \boldsymbol{O}_{p\times(M-2p)}$, and the initial values $\dot{\theta}_i(n-1)$ and $\ddot{\theta}_i(n-1)$ are set to zero.

4. STABILITY ANALYSIS OF LMS AND NLMS ALGORITHMS

By analyzing the mean and mean-square behaviors of the weight-error $\tilde{\boldsymbol{P}}(k) = \boldsymbol{P} - \boldsymbol{P}(k)$, the step-size convergence conditions that guarantee both the mean and mean-square stabilities of the LMS and NLMS algorithms are given as follows, when the incident signals are constant.

$$0 < \mu < \min\left\{\frac{2}{\lambda_{\max}(\bar{\boldsymbol{\Psi}}_1)}, \frac{1}{\lambda_{\max}(\boldsymbol{C}^{-1}\tilde{\boldsymbol{C}})}, \frac{1}{\lambda_{\max}(\boldsymbol{C}^{-1}\breve{\boldsymbol{C}})}, \frac{1}{\max\{\lambda(\tilde{\boldsymbol{L}}) \in \mathcal{R}^+\}}, \frac{1}{\max\{\lambda(\breve{\boldsymbol{L}}) \in \mathcal{R}^+\}}\right\} \tag{17}$$

$$0 < \bar{\mu} < 2 \tag{18}$$

where $\bar{\boldsymbol{\Psi}}_1 = \boldsymbol{\Psi}_1 + \sum_{l=1}^{p}\{r_{MM}(\boldsymbol{M}_{fll} + \boldsymbol{M}_{bl+1,l+1}) + r_{11} \cdot (\boldsymbol{M}_{fl+1,l+1} + \boldsymbol{M}_{bll})\}$, $\boldsymbol{\Psi}_1 = \boldsymbol{\Phi}_1\boldsymbol{\Phi}_1^H$, $\boldsymbol{\Phi}_1 = E\{\boldsymbol{\Phi}_1(k)\}$, $r_{im} = E\{y_i(k)y_m^*(k)\}$, $\boldsymbol{M}_{fim} = E\{\boldsymbol{y}_{fi}(k)\boldsymbol{y}_{fm}^H(k)\}$, $\boldsymbol{M}_{bim} = E\{\boldsymbol{y}_{bi}(k)\boldsymbol{y}_{bm}^H(k)\}$, $\boldsymbol{C} = (\boldsymbol{I}_p \otimes \bar{\boldsymbol{\Psi}}_1) + (\bar{\boldsymbol{\Psi}}_1 \otimes \boldsymbol{I}_p)$, $\tilde{\boldsymbol{C}} = \bar{\boldsymbol{C}} + \bar{\boldsymbol{K}}_1$, $\breve{\boldsymbol{C}} = \bar{\boldsymbol{C}} - \bar{\boldsymbol{K}}_1$, and $\bar{\boldsymbol{C}} = (\bar{\boldsymbol{\Psi}}_1^T \otimes \bar{\boldsymbol{\Psi}}_1) + \bar{\boldsymbol{K}}_2$, while

$$\bar{\boldsymbol{K}}_1 = \sum_{l=1}^{p}\sum_{t=1}^{p}\sum_{i=1}^{4}\sum_{m=1}^{4} \bar{\boldsymbol{F}}_{il,mt}^H \otimes \bar{\boldsymbol{F}}_{il,mt} \tag{19}$$

$$\bar{\boldsymbol{K}}_2 = \sum_{l=1}^{p}\sum_{t=1}^{p}\sum_{i=1}^{4}\sum_{m=1}^{4} \mathrm{vec}(\boldsymbol{F}_{il,mt})\mathrm{vec}^H(\boldsymbol{F}_{il,mt}) \tag{20}$$

$$\tilde{\boldsymbol{L}} = \begin{bmatrix} \boldsymbol{C}/2, & -\tilde{\boldsymbol{C}}/2 \\ \boldsymbol{I}_{p^2}, & \boldsymbol{O}_{p^2\times p^2} \end{bmatrix}, \ \breve{\boldsymbol{L}} = \begin{bmatrix} \boldsymbol{C}/2, & -\breve{\boldsymbol{C}}/2 \\ \boldsymbol{I}_{p^2}, & \boldsymbol{O}_{p^2\times p^2} \end{bmatrix} \tag{21}$$

$\bar{\boldsymbol{F}}_{il,mt} = E\{\tilde{\boldsymbol{z}}_{il}(k)\tilde{\boldsymbol{z}}_{mt}^T(k)\}$, $\boldsymbol{F}_{il,mt} = E\{\tilde{\boldsymbol{z}}_{il}(k)\tilde{\boldsymbol{z}}_{mt}^H(k)\}$, $\tilde{\boldsymbol{z}}_{1l}(k) = \boldsymbol{y}_{fl}(k)y_M^*(k)$, $\tilde{\boldsymbol{z}}_{2l}(k) = \boldsymbol{y}_{fl+1}(k)y_1^*(k)$, $\tilde{\boldsymbol{z}}_{3l}(k) = \boldsymbol{y}_{bl}(k)y_1(k)$, $\tilde{\boldsymbol{z}}_{4l}(k) = \boldsymbol{y}_{bl+1}(k)y_M(k)$, and $\otimes$ denotes the Kronecker product. Here we assume that the real and positive eigenvalues $\lambda(\tilde{\boldsymbol{L}})$ and $\lambda(\breve{\boldsymbol{L}})$ of the $2p^2 \times 2p^2$ matrices in (21) exist; if they do not, the corresponding condition should be removed from (17).

Proof: Omitted (see [6] for details).

5. NUMERICAL EXAMPLES

The ULA with M sensors is separated by a half-wavelength, and the simulation results shown below are obtained by the ensemble-averaging over 1000 independent trials.

Example 1—Verification of Stability Analysis: The number of sensors is $M = 16$, and one signal impinges the array along $\theta_1 = 10^\circ$ with the signal power $r_s = 1$. The additive noise is assumed to be absent. The step-sizes of the LMS and NLMS algorithms are set to $\mu = 0.25, 1/12, 1/24$, and 1, and $\bar{\mu} = 2, 1.5, 1$, and 0.1, respectively.

From the analyses in Section 4, the stability bounds of the LMS step-size μ in the mean and mean-square senses are $\mu_{\text{mean}} = 0.25$ and $\mu_{\text{m.s.}} = 1/12$, then the convergence condition is given by $0 < \mu < \min\{\mu_{\text{mean}}, \mu_{\text{m.s.}}\} = \mu_{\text{m.s.}}$, and the optimum step-size is $\mu_{\text{o}} = \mu_{\text{m.s.}}/2$ [6]. Fig. 1 shows that there is an almost perfect agreement between the LMS theoretical mean-square error (MSE) learning curve (see [6]) and the ensemble-averaged ones for μ which are smaller than the stability supremum $\mu_{\text{m.s.}}$ and that the fastest convergence is achieved with $\mu = \mu_o$. Though there are appreciable differences between the behaviors of the ensemble-averaged curves and those of the theoretical ones for μ that are out of the stability region, these phenomena essentially

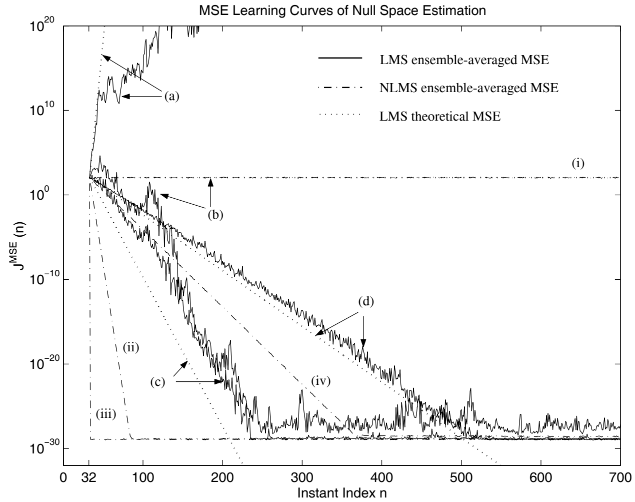

Fig. 1. MSE learning curves of null space estimation in the case of one signal without additive noise ((a) $\mu = 0.25$, (b) $\mu = 1/12$, (c) $\mu = 1/24$, and (d) $\mu = 0.01$; (i) $\bar{\mu} = 2$, (ii) $\bar{\mu} = 1.5$, (iii) $\bar{\mu} = 1$, and (iv) $\bar{\mu} = 0.1$) for Example 1.

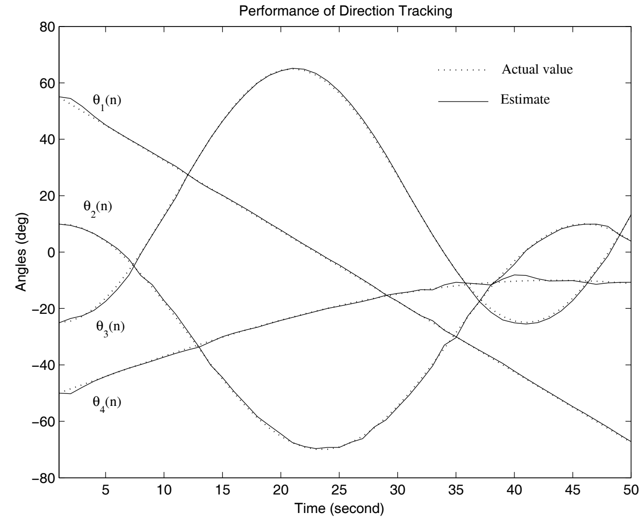

Fig. 2. Averaged estimates for tracking time-varying directions of coherent signals (dotted line: actual value; and solid line: estimate) for Example 2.

conform with the learning mechanism clarified and studied in [7]. The convergence of the NLMS ensemble-averaged MSE learning curves is guaranteed for the step-size $\bar{\mu}$ satisfying $0 < \bar{\mu} < 2$, and the learning curve converges fastest with $\bar{\mu} = 1$. Additionally the convergence condition for the NLMS algorithm is independent of the statistics of the incident signal and the NLMS algorithm converges faster than the LMS one.

Example 2—Tracking of Crossing Directions: The number of sensors is $M = 9$, and there are four coherent signals come from time-varying directions $\theta_1(n) \sim \theta_4(n)$ with SNR's of 15dB, 10dB, 13dB and 13dB. The directions are tracked over 50s with $T = 1$s, and during each interval T, $N = 100$ snapshots of array data are measured and used to estimate the orthogonal projector $\mathbf{\Pi}(n)$. The poles of system $\boldsymbol{F} - \boldsymbol{g}_i \boldsymbol{c}^T$ are set as $\rho_{11} = 0.7021 + j0.6945$, $\rho_{12} = \rho_{11}^*$, $\rho_{13} = 0.9192$, $\rho_{21} = 0.5621 + j0.6145$, $\rho_{22} = \rho_{21}^*$, $\rho_{23} = 0.5343$, $\rho_{31} = 0.7021 + j0.6656$, $\rho_{32} = \rho_{31}^*$, $\rho_{33} = 0.8905$, $\rho_{41} = 0.6912 + j0.6956$, $\rho_{42} = \rho_{41}^*$, $\rho_{43} = 0.8243$, respectively. The trajectories of the actual directions and the estimates obtained by the proposed NLMS-based algorithm are plotted in Fig. 2, where $\bar{\mu} = 0.88$. Clearly the proposed algorithm has superior tracking ability in the multipath environment even the time-varying directions cross at some instants.

6. CONCLUSION

This paper proposed a computationally efficient subspace-based algorithm for adaptive direction estimation and tracking of uncorrelated and correlated narrowband signals impinging on a ULA. In this algorithm, the null space is estimated using the LMS or NLMS algorithm, and the directions are updated using the approximate Newton method. The convergence conditions for the step-sizes of the LMS and NLMS algorithms that guarantee the mean and mean-square stabilities were explicitly derived. The effectiveness of the proposed algorithm was verified through numerical examples, and it was shown that the proposed algorithm has good adaptation and tracking abilities.

7. REFERENCES

[1] C.-C. Yeh, "Simple computation of projection matrix for bearing estimations," *IEE Proc., Pt. F*, vol. 134, no. 2, pp. 146-150, 1987.

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

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

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

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

[6] J. Xin and A. Sano, "Efficient subspace-based algorithm for adaptive bearing estimation and tracking," submitted to *IEEE Trans. Signal Processing*, 2004.

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

[8] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 95-107, 1995.