

## Research Article

# Direct Signal Space Construction Luenberger Tracker with Quadratic Least Square Regression for Computationally Efficient DoA Tracking

**Bukeun Byeon, Ho-Kyoung Lee, and Do-Sik Yoo** 

*School of Electronic and Electrical Engineering, Hongik University, Map-Gu Wausan Ro 94, Seoul 04066, Republic of Korea*

Correspondence should be addressed to Do-Sik Yoo; yoodosik@hongik.ac.kr

Received 28 August 2019; Accepted 8 May 2020; Published 17 June 2020

Academic Editor: Ludovic Chamoin

Copyright © 2020 Bukeun Byeon et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we propose a computationally efficient direction-of-arrival (DoA) tracking scheme called the direct signal space construction Luenberger tracker (DSPCLT) with quadratic least square (QLS) regression. Also, we study analytically how the choice of observer gain affects the algorithm performance and illustrate how we can use the theoretical results in determining optimal observer gain value. The proposed scheme (DSPCLT) has several distinct features compared with existing algorithms. First, it requires only a fraction of computational complexity compared with other schemes. Secondly, it maintains robustness by treating separately the special case of object overlap in which subspace-based algorithms often suffer from lack of resolvability. Thirdly, the proposed scheme achieves enhanced performance by a method of delay compensation, which accounts for observation delay. Through numerical analysis, we show that DSPCLT achieves performance similar or superior to existing algorithms with only a fraction of computational requirement.

## 1. Introduction

In this paper, we propose a computationally efficient direction-of-arrival (DoA) tracking scheme called the direct signal space construction Luenberger tracker (DSPCLT) with quadratic least square (QLS) regression. During the last several decades, various subspace-based schemes have been proposed for DoA estimation such as multiple signal classification algorithms [1], estimation of signal parameter via rotational invariance technique (ESPRIT) [2], and root-MUSIC method [3]. Recently, because of their high resolution, these algorithms have been studied in various applications such as nonlinear arrays [4–6], 2-dimensional arrays [7, 8], and MIMO radar [9]. However, early subspace-based algorithms generally suffer from the issue of large computational complexity, typically due to the need for eigenvalue decomposition (ED) or singular-value decomposition (SVD). For this reason, various schemes have been proposed to reduce the system complexity, such as the propagator method (PM) [10], orthogonal propagator

method (OPM) [11], subspace method without eigendecomposition (SWEDE) [12], direct signal space construction method (DSPCM) [13–15], and projection approximation subspace tracking-based deflation (PASTd) algorithm [16].

However, these algorithms exhibit difficulty in estimating the DoAs when one or more signals suddenly disappear or when DoAs overlap. [17] As means to overcome such issues, a number of DoA tracking schemes have been proposed in [18–20]. Recently, authors in [21, 22] proposed to combine existing tracking techniques with computational efficient subspace techniques to avoid the issues of overlapping or suddenly disappearing targets while keeping the complexity minimal. For example, the authors in [21] proposed an algorithm by combining PASTd and Kalman filtering [23]. We shall refer to this algorithm as PASTd Kalman in this paper. PASTd Kalman reduces the computational complexity of constructing the subspace by avoiding ED and SVD with recursive least square (RLS) technique. However, the algorithm still requires relatively high complexity due to the adaptation of Kalman filtering. More

recently, the authors in [22] proposed an algorithm called the adaptive method of estimating DoA (AMEND) which replaces the Kalman filter with Luenberger state observer [24]. However, the complexity of AMEND is not smaller than PASTd Kalman because it employs, for subspace construction, a relatively computationally expensive algorithm called the subspace-based method without eigendecomposition (SUMWE) [25].

In this paper, we propose DSPCLT that achieves similar or better performance with computational complexity significantly lower than existing algorithms. The proposed scheme has several distinct features compared with existing schemes. First, it achieves very low complexity by employing DSPCM [14, 15] for subspace construction and Luenberger observer in place of the Kalman filter for estimation and filtering. Secondly, it achieves robustness by treating separately the special case of object overlap in which subspace-based algorithms suffer from the lack of resolvability. Thirdly, the proposed scheme achieves enhanced performance by a method of delay compensation, which takes into account the fact that the DoA estimation at present time is performed with observed data collected previously. We show, through numerical analysis, that the proposed scheme achieves performance similar or superior to existing algorithms with only a fraction of computational requirement.

Another important topic studied in this paper is the choice of observer gains used to implement Luenberger observer. In most existing schemes employing Luenberger observer, optimal observer gain values are determined through simulations. For example, a certain observer gain value was suggested in [22] after simulations. However, we note that there is infinite variety of situations depending on the number of antenna sensor elements, the numbers of objects, the speeds of movements, the signal-to-noise ratios of the target, the shape of object trajectory, and so on. Hence, for a selection of a certain observer gain value to be accepted as a valid choice, there must be some degree of guarantee that a particular observer gain value will provide optimal or near-optimal performance in other situations, which is generally very difficult with pure simulations. For this reason, we study analytically how the choice of observer gain affects the algorithm performance and how we can choose optimal observer gain value.

This paper is organized as follows. In Section 2, the system model and basic assumptions are described together with the notational definitions used throughout this paper. In Sections 3 ~ 5, the proposed scheme (DSPCLT) is delineated, which consists of three stages called initialization, prediction, and estimation. The stages of initialization and prediction are introduced in Section 3 and that of estimation is presented in Sections 4 and 5 depending on whether objects are resolvable by the subspace-based algorithm employed. In Section 6, we study theoretically how the selection of observer gain value affects the algorithm performance and describe how we can make optimal observer gain value selection using the analytical results. Next, we show, in Section 7, that the proposed scheme achieves similar or better performance

with only a fraction of computational requirements compared with other algorithms. Finally, we draw conclusion in Section 8.

The following notations are used throughout this paper. Bold-faced letters are used for matrices and vectors. The complex conjugation, the transposition, and the Hermitian transposition of matrix  $\mathbf{A}$  are denoted, respectively, by  $\mathbf{A}^*$ ,  $\mathbf{A}^T$ , and  $\mathbf{A}^H$ . The  $M \times N$  zero matrix and  $N \times N$  identity matrix are denoted, respectively, by  $\mathbf{O}_{M \times N}$  and  $\mathbf{I}_N$ .

## 2. System Model and Basic Assumptions

We consider a uniform linear array (ULA) consisting of  $N$  sensors with intersensor spacing  $d$  and assume that  $K$  narrowband far-field signal waves  $s_1(t), s_2(t), \dots, s_K(t)$  of wavelength  $\lambda$  impinge on the array at distinct directions, respectively, making angles  $\theta_1(t), \theta_2(t), \dots, \theta_K(t)$  with the linear array. We denote, by  $x_1(t), x_2(t), \dots, x_N(t)$ , the received noisy signals collected at the  $N$  sensors. Then, if we define the steering vector  $\mathbf{a}(\theta_k(t))$  by  $\mathbf{a}(\theta_k(t)) = [1, e^{j\gamma_k(t)}, \dots, e^{j(N-1)\gamma_k(t)}]^T$  with  $\gamma_k(t) = 2\pi d \cos \theta_k(t)/\lambda$ , the system can be modeled with the following equation:

$$\mathbf{x}(t) = \mathbf{A}(\theta_1(t), \dots, \theta_K(t))\mathbf{s}(t) + \mathbf{n}(t), \quad (1)$$

where  $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_N(t)]^T$ ,  $\mathbf{A}(\theta_1(t), \dots, \theta_K(t)) = [\mathbf{a}(\theta_1(t)), \dots, \mathbf{a}(\theta_K(t))]$ ,  $\mathbf{s}(t) = [s_1(t), \dots, s_K(t)]^T$ , and  $\mathbf{n}(t)$  is the vector consisting of the noises added at the sensors. In this paper, we make, on the system model, the following basic assumptions:

- (1) The signal wavelength  $\lambda$  is known a priori and that  $d$  is chosen to be  $\lambda/2$ . And the number  $K$  of impinging signals is predetermined by appropriate methods such as the one in [26].
- (2) The vectors  $\mathbf{s}(t)$  and  $\mathbf{n}(t)$  are statistically independent complex random vector processes with zero mean satisfying

$$\begin{aligned} E[\mathbf{s}(t)\mathbf{s}(s)^H] &= \mathbf{P}\delta_{ts}, \\ E[\mathbf{s}(t)\mathbf{s}(s)^T] &= \mathbf{O}_{K \times K}, \\ E[\mathbf{n}(t)\mathbf{n}(s)^H] &= N_0\delta_{ts}\mathbf{I}_N, \\ E[\mathbf{n}(t)\mathbf{n}(s)^T] &= \mathbf{O}_{N \times N}, \end{aligned} \quad (2)$$

where  $\delta_{ts}$  is the Kronecker delta function. We further assume that  $\mathbf{P}$  is a diagonal matrix with positive real numbers  $p_1, \dots, p_K$  on its diagonal and that the value of  $N_0$  is assumed to be estimated accurately by the receiver through long-term observation of the noise.

- (3) The system tracks the DoAs by obtaining angle estimates at time  $t = 0, T, 2T, 3T, \dots$ , where  $T$  is a predefined positive real number. We assume further that  $T$  is chosen to be small enough that the DoAs are essentially constant during the time duration  $[(n-1)T, nT]$  for each  $n = 1, 2, 3, \dots$

### 3. Initialization and Prediction

In this paper, we propose a computationally efficient DoA tracking scheme called the direct signal space construction Luenberger tracker (DSPCLT) with quadratic least square regression. As in AMEND, the DoA is tracked by sequential update of state that is defined as a three-dimensional vector consisting of direction-of-arrival and its first and second derivatives. The process of update is carried out in two stages, which we call prediction and estimation. In the prediction stage, the state is tentatively updated purely based on the previous state. Then, more accurate updated state is obtained in the estimation stage, which is the major part of the proposed scheme.

As in AMEND, we propose to use, in the estimation stage, the Luenberger observer using a subspace-based algorithm. In particular, we use Newton iteration method to obtain the improved estimate of the DoA for use in the Luenberger observer. While such subspace-based tracking schemes provide excellent performance in general, we need to take special care when the DoAs of objects overlap, in which case the Newton iteration may not provide robust estimate of the DoA. Consequently, it is wise to consider such a special case separately.

Since it is rather complicated to describe the estimation stage, we describe only the initialization stage and the prediction stage in this section. Then, in the next two sections, we consider the estimation stage in nonresolvable and resolvable situations, respectively. The proposed scheme employs, for each object, a 3-dimensional state vector whose components represent the direction-of-arrival and its first and second derivatives, respectively. At the initialization stage, the state vector is assumed to be obtained by a certain existing method such as MUSIC [1]. After the initialization stage, the state vector is updated at a regular interval  $T$  in two stages called prediction and estimation described in the following.

Before proceeding, let us clarify the notations used in the following. First, we denote by  $\hat{\mathbf{y}}_k[n|n]$  the state vector at time  $t = nT$  and by  $\hat{\theta}_k[n|n]$  the first component of  $\hat{\mathbf{y}}_k[n|n]$ . We shall regard  $\hat{\theta}_k[n|n]$  as the estimate of the DoA  $\theta_k(nT)$  of object  $k$  at time  $nT$ . Next, we denote by  $\hat{\mathbf{y}}_k[n|n-1]$  the predicted state vector obtained in the prediction stage at time  $t = nT$  and by  $\hat{\theta}_k[n|n-1]$  the first component of  $\hat{\mathbf{y}}_k[n|n-1]$ .

**3.1. Initialization Stage.** The initialization stage begins by obtaining the estimates of  $\theta_k(-2T)$ ,  $\theta_k(-T)$ , and  $\theta_k(0)$  for  $k = 1, 2, \dots, K$ , through a certain DoA measurement scheme such as MUSIC [1], ESPRIT [2], or DSPCM [14, 15]. The estimates  $\theta_k(-2T)$ ,  $\theta_k(-T)$ , and  $\theta_k(0)$  shall be denoted, respectively, by  $\tilde{\theta}_k[-2|-2]$ ,  $\tilde{\theta}_k[-1|-1]$ , and  $\tilde{\theta}_k[0|0]$ . Then, the initial state vector  $\hat{\mathbf{y}}_k[0|0]$  is computed by

$$\hat{\mathbf{y}}_k[0|0] = [\tilde{\theta}_k[0|0], (\tilde{\theta}_k[0|0] - \tilde{\theta}_k[-1|-1])/T, (\tilde{\theta}_k[0|0] - 2\tilde{\theta}_k[-1|-1] + \tilde{\theta}_k[-2|-2])/T^2]^T. \quad (3)$$

We note that the third component of  $\hat{\mathbf{y}}_k[0|0]$  is set to be zero in AMEND [22] and PASTd Kalman [21]. In fact, this

initialization provides better performance in the proposed scheme as in AMEND and PASTd Kalman according to our simulations. Consequently, we shall use this method of initialization previously used in AMEND and PASTd Kalman when we evaluate the performance in a later section. However, we shall use initialization (3) for the proposed scheme despite slightly worse performance. This is because mathematical analysis in later sections is slightly simpler with (3).

**3.2. Prediction Stage.** After initialization, the state vectors are updated periodically, with period  $T$ , in two stages called prediction and estimation. We assume that the state vector  $\hat{\mathbf{y}}_k[n-1|n-1]$  at previous time  $(n-1)T$  is given, where  $n \in \{1, 2, \dots\}$ . We assume that  $\hat{\mathbf{y}}_k[n-1|n-1]$  has been obtained in the initialization stage if  $n = 1$  and in the previous estimation stage if  $n > 1$ . In this stage, namely, in the prediction stage, rough estimates of the next state, which we denote by  $\hat{\mathbf{y}}_k[n|n-1]$ , are obtained by the following dynamic matrix equation:

$$\hat{\mathbf{y}}_k[n|n-1] = \mathbf{F} \cdot \hat{\mathbf{y}}_k[n-1|n-1], \quad (4)$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0.5T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

Here, the first, second, and third components of the vector  $\hat{\mathbf{y}}_k[n|n-1]$  are interpreted to represent predicted values, based on the previous state vector, for  $\theta_k(nT)$ ,  $\dot{\theta}_k(nT)$ , and  $\ddot{\theta}_k(nT)$ , respectively. After the prediction stage, the distance

$$d_{kl} = |\hat{\theta}_k[n|n-1] - \hat{\theta}_l[n|n-1]|, \quad (6)$$

is computed for  $l = 1, \dots, k-1, k+1, \dots, K$ . The object  $k$  is regarded to be in the nonresolvable situation if  $d_{kl}$  is less than or equal to a certain predetermined threshold value for some  $i, j$  with  $l \neq k$ . Naturally, the object  $k$  is regarded to be resolvable if it is not in the nonresolvable situation.

### 4. Estimation Stage in the Nonresolvable Situation

We note that the nonresolvable situation becomes less likely if the number of antenna elements is increased. In other words, the nonresolvable situation does not happen very often and does not last very long even when it happens if the number of antenna elements is large enough. Although the nonresolvable situation may not happen very often and the algorithm used in this situation is not the major contribution of this paper, we consider this situation first because the proposed algorithm for the resolvable situation is much more complicated to describe.

In the nonresolvable situation, the estimated state vector  $\hat{\mathbf{y}}_k[n|n]$  is obtained by using quadratic least square regression using the past  $p$  estimated DoAs  $\hat{\theta}_k[m|m]$ ,  $m = n-p, n-$

$p + 1, \dots, n - 1$ . To be more specific, we assume that the true DoA  $\theta_{[m]}$  satisfies

$$\theta_k[m] = a(m - n)^2 + b(m - n) + c, \quad (7)$$

and attempt to obtain the constants  $a, b$ , and  $c$  by the least square method with the previously estimated DoAs  $\hat{\theta}_k[m|m]$ ,  $m = n - p, n - p + 1, \dots, n - 1$ . We note that the resultant constants  $a, b$ , and  $c$  are given by [27]

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} p & \sum_{m=-p}^{-1} m & \sum_{m=-p}^{-1} m^2 \\ \sum_{m=-p}^{-1} m & \sum_{m=-p}^{-1} m^2 & \sum_{m=-p}^{-1} m^3 \\ \sum_{m=-p}^{-1} m^2 & \sum_{m=-p}^{-1} m^3 & \sum_{m=-p}^{-1} m^4 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum_{m=n-p}^{n-1} \hat{\theta}_k[m|m] \\ \sum_{m=n-p}^{n-1} (m - n)\theta_k[m|m] \\ \sum_{m=n-p}^{n-1} (m - n)^2 \hat{\theta}_k[m|m] \end{bmatrix}. \quad (8)$$

Once the constants  $a, b$ , and  $c$  are computed by (8), the estimated state vector  $\hat{\mathbf{y}}_k[n|n]$  is obtained by

$$\hat{\mathbf{y}}_k[n|n] = \begin{bmatrix} c \\ b \\ 2a \end{bmatrix}. \quad (9)$$

We note that the second and third components of  $\hat{\mathbf{y}}_k[n|n]$  can be obtained by the first and the second derivative of  $\theta_k[m] = a(m - n)^2 + b(m - n) + c$  at  $m = n$ .

## 5. Estimation Stage in the Resolvable Situation

In this section, we describe how we obtain the estimated state vector in the resolvable situation. First, we recall that the predicted state vector  $\hat{\mathbf{y}}_k[n|n-1]$  is obtained, in the prediction stage, by simply multiplying the previous estimated state vector  $\hat{\mathbf{y}}_k[n-1|n-1]$  by a constant matrix  $\mathbf{F}$  without reflecting the data collected between times  $(n-1)T$  and  $nT$ . While the data collected, captured at the antenna, are still not used to obtain the updated estimated state vector in the nonresolvable situation, the data are used to obtain the updated estimated state vector in the resolvable situation. In the following, we describe how the data are used to obtain  $\hat{\mathbf{y}}_k[n|n]$  in the resolvable situation.

In particular, the updated estimated state vector  $\hat{\mathbf{y}}_k[n|n]$  is obtained, using the Luenberger observer, by

$$\hat{\mathbf{y}}_k[n|n] = \hat{\mathbf{y}}_k[n|n-1] + (\hat{\theta}_k[n|n] - \hat{\theta}_k[n|n-1])\mathbf{g}_k, \quad (10)$$

where  $\mathbf{g}_k$  denotes a 3-dimensional column vector called observer gain [24]. Here, the quantity  $\hat{\theta}_k[n|n]$  is an estimate of  $\theta_k(nT)$  obtained by a certain algorithm using the data collected at the antenna. In the following, we describe how

we obtain  $\hat{\theta}_k[n|n]$  in the proposed scheme. Before proceeding, we note that the performance of the system can depend on the choice of the observer gain  $\mathbf{g}_k$ . In many cases as in AMEND, no serious discussion is provided regarding how the choice of the observer gain affects the performance or how we can achieve optimal or near-optimal performance. We discuss this issue in the next section. Hence, we focus, in this section, solely on how we obtain the quantity  $\hat{\theta}_k[n|n]$ , which we call the delay compensated direct signal space construction method Newton estimate (DCDSPCMNE) of  $\theta_k(nT)$ . The DCDSPCMNE  $\hat{\theta}_k[n|n]$  is obtained in two steps. In the first step, we obtain a quantity  $\check{\theta}_k[n|n]$  called the direct signal space construction method Newton estimate (DSPCMNE). Then, in the second step, we obtain more accurate estimate  $\hat{\theta}_k[n|n]$  by a method of motion compensation.

*5.1. Direct Signal Space Construction Method Newton Estimate.* Let us first discuss how we obtain DSPCMNE  $\check{\theta}_k[n|n]$ . In many subspace-based DoA estimation schemes, the DoAs are estimated by minimizing a certain cost function  $f(\theta)$  defined by

$$f(\theta) = \mathbf{a}^H(\theta)\mathbf{\Pi}\mathbf{a}(\theta), \quad (11)$$

where  $\mathbf{\Pi}$  is the orthogonal projector into the noise subspace. One major source of computational complexity in subspace-based schemes is in the construction of the orthogonal projector  $\mathbf{\Pi}$ . In this paper, we propose to use the direct signal space construction method [14, 15] to reduce the computational complexity significantly. In Appendix A.1, it was described how we can construct the orthogonal projector  $\mathbf{\Pi}$  in the proposed scheme.

Since the minimum search process also takes some nonnegligible computational complexity, various methods have been proposed to reduce the burden. One such method, which is particularly useful when some coarse estimate of the DoA is available, is to employ Newton's iteration method of searching zeroes with  $f'(\theta)$  [11, 12, 22, 25]. We also adopt Newton's iteration method in this paper. Let  $\bar{\theta}$  denote the coarse estimate of a zero of  $f'(\theta)$ . Then, according to Newton's method, a refined zero  $\hat{\theta}$  is given by

$$\hat{\theta} = \bar{\theta} - \frac{f'(\bar{\theta})}{f''(\bar{\theta})} = \bar{\theta} - \frac{\text{Re}[\mathbf{a}'^H(\bar{\theta})\mathbf{\Pi}\mathbf{a}(\bar{\theta})]}{\text{Re}[\mathbf{a}''^H(\bar{\theta})\mathbf{\Pi}\mathbf{a}(\bar{\theta})] + \mathbf{a}'^H(\bar{\theta})\mathbf{\Pi}\mathbf{a}'(\bar{\theta})}, \quad (12)$$

where  $\text{Re}[\cdot]$  denotes the real part of  $\cdot$ . Here, we note that the first term of the denominator is much smaller than the second term of the denominator since  $\mathbf{\Pi}\mathbf{a}(\bar{\theta}) \approx 0$ . For this reason, the first term of the denominator is open dropped in implementations.

Using (12) and neglecting the first term of the denominator, we obtain DSPCMNE  $\check{\theta}_k[n|n]$  by

$$\check{\theta}_k[n|n] = \hat{\theta}_k[n|n-1] - \frac{\text{Re}[\mathbf{d}^H[n]\hat{\mathbf{\Pi}}[n]\mathbf{a}[n]]}{\mathbf{d}^H[n]\hat{\mathbf{\Pi}}[n]\mathbf{d}[n]}, \quad (13)$$

where  $\mathbf{a}_k[n]$  and  $\mathbf{d}_k[n]$  are vectors defined, respectively, by

$$\mathbf{a}_k[n] = \mathbf{a}(\widehat{\theta}_k[n|n-1]), \quad (14)$$

and

$$\mathbf{d}_k[n] = \left. \frac{d\mathbf{a}(\theta)}{d\theta} \right|_{\theta=\widehat{\theta}_k[n|n-1]}. \quad (15)$$

Also, the matrix  $\widehat{\Pi}[n]$  in (13) denotes an estimate of the propagator  $\pi$ . We discuss, in Appendix A.2, how we can construct the estimate  $\widehat{\Pi}[n]$  in the proposed scheme.

**5.2. Delay Compensated Direct Signal Space Construction Method Newton Estimate.** We note that the orthogonal projector  $\widehat{\Pi}[n]$  is obtained by using the data collected during the time interval  $((n-1)T, nT]$ . Hence, while DCDSPCMNE  $\check{\theta}_k[n|n]$  is an estimate of  $\theta_k(t)$ , it is not exactly at time  $t = nT$ . In this section, we regard  $\check{\theta}_k[n|n]$  as an estimate of  $\theta_k(t)$  at a certain time  $t = (n-\alpha)T$  ( $0 < \alpha < 1$ ) and obtain the DCDSPCMNE  $\check{\theta}_k[n|n]$  by compensating the movement during the delay  $\alpha T$ . For this purpose, we first need to assess the value of  $\alpha$  and then estimate the amount of change in  $\theta_k(t)$  during this delay.

Before proceeding, we shall assume that  $T$  and  $\check{\theta}_k(t)$  are small enough so that  $\check{\theta}_k(t)$  does not change appreciably over time period  $[(n-1)T, nT]$ . However, we note, even with this simplifying assumption, that it is very difficult to determine theoretically justifiable value for  $\alpha T$  since the process of obtaining  $\widehat{\Pi}[n]$  does not depend linearly on the observation times. As described in Appendix A.2, the data are collected at times  $t = [n - (N_s - 1)\delta]T, [n - (N_s - 2)\delta]T, \dots, [n - (N_s - N_s)\delta]T$ , and forgetting factor  $\rho$  is used to give more weight to more recent data. In the special case in which  $\delta = 1/N_s$  and  $\rho = 1$ , we note that the data are collected at times  $t = (n-1 + 1/N_s)T, (n-1 + 2/N_s)T, \dots, nT$ , and all the data are equally weighted since  $\rho = 1$ . Consequently, we observe that the data are collected symmetrically around time  $t = (n - (N_s - 1)/2N_s)T$  and are equally used. While it is theoretically difficult to justify the validity, we shall assess the value of  $\alpha$  as  $(N_s - 1)/2N_s$  in this special case.

The assessment of the value of  $\alpha$  is more difficult in the general case. In the general case, considering the fact that the data are collected at  $t = [n - (N_s - 1)\delta]T, [n - (N_s - 2)\delta]T, \dots, [n - (N_s - N_s)\delta]T$  and that data collected at time  $t = (n - (N_s - l)\delta)T$  are weighted by  $(1-\rho)\rho^{N_s-l}/(1-\rho^{N_s})$  in the computation of  $\check{r}_m$  in (A.3) and (A.4), we shall regard the weighted time average

$$\begin{aligned} & [n - (N_s - 1)\delta]T \cdot \frac{(1-\rho)\rho^{N_s-1}}{1-\rho^{N_s}} \\ & + [n - (N_s - 2)\delta]T \cdot \frac{(1-\rho)\rho^{N_s-2}}{1-\rho^{N_s}} \\ & + \dots + [n - (N_s - N_s)\delta]T \cdot \frac{(1-\rho)\rho^{N_s-N_s}}{1-\rho^{N_s}} \\ & = \left[ n - \frac{(N_s - 1)\rho^{N_s+1} - N_s\rho^{N_s} + \rho}{(1-\rho)(1-\rho^{N_s})} \cdot \delta \right] T, \end{aligned} \quad (16)$$

as the estimated observation time  $(n-\alpha)T$ .

Next, we note that

$$\check{\theta}_k[n|n] - \widehat{\theta}_k[n-1|n-1], \quad (17)$$

accounts for the angle change between times  $(n-1)T$  and  $(n-\alpha)T$ . The change between times  $(n-1)T$  and  $nT$  can be estimated as

$$\frac{1}{1-\alpha} \{ \check{\theta}_k[n|n] - \widehat{\theta}_k[n-1|n-1] \}, \quad (18)$$

assuming that angular velocity does not change over  $[(n-1)T, nT]$ . Using this approximation, we obtain the DCDSPCMNE  $\check{\theta}_k[n|n]$  by the following equation:

$$\begin{aligned} \check{\theta}_k[n|n] &= \widehat{\theta}_k[n-1|n-1] + \frac{1}{1-\alpha} \{ \check{\theta}_k[n|n] - \widehat{\theta}_k[n-1|n-1] \}, \\ &= \widehat{\theta}_k[n-1|n-1] \\ &\quad + \frac{(1-\rho)(1-\rho^{N_s}) \{ \check{\theta}_k[n|n] - \widehat{\theta}_k[n-1|n-1] \}}{(1-\rho)(1-\rho^{N_s}) - \{ (N_s-1)\rho^{N_s+1} - N_s\rho^{N_s} + \rho \} \delta}. \end{aligned} \quad (19)$$

Before proceeding, we note that (20) reduces to

$$\begin{aligned} \check{\theta}_k[n|n] &= \widehat{\theta}_k[n-1|n-1] + \frac{N_s(1-\rho)(1-\rho^{N_s})}{[N_s - (N_s+1)\rho + \rho^{N_s+1}]} \\ &\quad \cdot \{ \check{\theta}_k[n|n] - \widehat{\theta}_k[n-1|n-1] \}, \end{aligned} \quad (21)$$

in the special case of  $\delta = 1/N_s$ .

## 6. Selection of Observer Gain

In Section 5, we discussed how the updated estimated state vector is obtained in the normal situation, namely, in the resolvable situation. In particular, the estimated state vector is updated, in the proposed scheme, using the Luenberger observer as described in (10). One important issue that was not discussed in Section 5 was the selection of the observer gain  $\mathbf{g}_k$ . If certain conditions are met, the estimation errors approach zero as time progresses. For example, the state error vectors converge to zero if and only if all the eigenvalues of the matrix  $\mathbf{F} - \mathbf{g}_k [1 \ 0 \ 0]$  have magnitudes strictly less than one in the case of the algorithm in [22]. However, not so much information is available about the effect of observer gain selection beyond such theoretical basics. Hence, observer gains are generally selected through simulations. However, the question arises whether an observer gain selected to be optimal through simulations in certain situations will lead to reasonably good performance in other situations. In this section, we study the effect of observer gain selection through theoretical and numerical analysis and describe how we can obtain reasonably optimal observer gains.

**6.1. State Estimation Error.** We first define the state estimation error  $\mathbf{e}_k[n]$  by

$$\mathbf{e}_k[n] = \widehat{\mathbf{y}}_k[n|n] - \mathbf{y}_k(nT), \quad (22)$$

where  $\mathbf{y}_k(t)$  denotes the state vector defined as  $[\theta_k(t), \dot{\theta}_k(t), \ddot{\theta}_k(t)]^T$ . Also, we define a vector  $\delta_k[n]$ , which we call state evolution mismatch at time  $t = nT$ , by

$$\delta_k[n] = \mathbf{y}_k(nT) - \mathbf{F} \cdot \mathbf{y}_k((n-1)T). \quad (23)$$

As discussed in [22], this vector  $\delta_k[n]$  equals to a zero vector if the angular acceleration does not change between times  $(n-1)T$  and  $nT$ . However, the angular acceleration may change although the amount may be small, and  $\delta_k[n]$  becomes small but nonzero vector. We define the angle measurement error  $\varepsilon_k[n]$  by

$$\varepsilon_k[n] = \tilde{\theta}_k[n|n] - \theta_k(nT). \quad (24)$$

In the next proposition, we obtain recurrence relation between  $\mathbf{e}_k[n]$  and  $\mathbf{e}_k[n-1]$ .

**Proposition 1.** *The state estimation errors satisfy the following recurrent relationship:*

$$\mathbf{e}_k[n] = (\mathbf{F} - \mathbf{g}_k \mathbf{c}^T \mathbf{F}) \cdot \mathbf{e}_k[n-1] + \mathbf{g}_k \varepsilon_k[n] + \mathbf{c}^T \delta_k[n] \mathbf{g}_k - \delta_k[n]. \quad (25)$$

*Proof.* We first note, from (10), (4), and (24), that

$$\hat{\mathbf{y}}_k[n|n] = \mathbf{F} \hat{\mathbf{y}}[n-1|n-1] + \{\theta_k(nT) + \varepsilon_k[n] - \hat{\theta}_k[n|n-1]\} \mathbf{g}_k. \quad (26)$$

Next, from (22), (23), and (26), it follows that

$$\begin{aligned} \mathbf{e}_k[n] &= \mathbf{F} \hat{\mathbf{y}}[n-1|n-1] + \{\theta_k(nT) + \varepsilon_k[n] - \hat{\theta}_k[n|n-1]\} \mathbf{g}_k \\ &\quad - \{\mathbf{F} \cdot \mathbf{y}_k((n-1)T) + \delta_k[n]\} \\ &= \mathbf{F} \{\hat{\mathbf{y}}[n-1|n-1] - \mathbf{y}_k((n-1)T)\} \\ &\quad + \{\theta_k(nT) + \varepsilon_k[n] - \hat{\theta}_k[n|n-1]\} \mathbf{g}_k - \delta_k[n]. \end{aligned} \quad (27)$$

Now, since  $\theta_k(t) = \mathbf{c}^T \mathbf{y}_k(t)$  and  $\hat{\theta}_k[n|n-1] = \mathbf{c}^T \hat{\mathbf{y}}_k[n|n-1]$ , we next obtain

$$\mathbf{e}_k[n] = \mathbf{F} \cdot \mathbf{e}_k[n-1] + \mathbf{c}^T \{\mathbf{y}_k(nT) - \hat{\mathbf{y}}_k[n|n-1]\} \mathbf{g}_k + \varepsilon_k[n] \mathbf{g}_k - \delta_k[n]. \quad (28)$$

$$\begin{aligned} &= \mathbf{F} \cdot \mathbf{e}_k[n-1] + \mathbf{c}^T \{\mathbf{F} \cdot \mathbf{y}_k((n-1)T) + \delta_k[n] \\ &\quad - \mathbf{F} \cdot \mathbf{y}_k[n-1|n-1]\} \mathbf{g}_k + \varepsilon_k[n] \mathbf{g}_k - \delta_k[n]. \end{aligned} \quad (29)$$

Finally, we obtain (25) by rearranging (29).  $\square$   $\square$

**6.2. Root Mean Squared Error and Observer Gain.** In this section, we attempt to study theoretically how the observer gain affects the performance of angle estimation. In particular, we study how the selection of observer gain is related to the root mean squared error (RMSE) of the DoA of the  $k^{\text{th}}$  object, which is defined by

$$\text{RMSE}_k = \sqrt{\frac{1}{N_T} \sum_{n=1}^{N_T} E \left[ \left| \hat{\theta}_k[n|n] - \theta_k(nT) \right|^2 \right]}, \quad (30)$$

where  $N_T$  is the total number of observation intervals. Before proceeding, we note that  $\text{RMSE}_k$  can be expressed, using  $\mathbf{e}_k[n]$ , as

$$\text{RMSE}_k = \sqrt{\frac{1}{N_T} \sum_{n=1}^{N_T} E \left[ \left| \mathbf{c}^T \mathbf{e}_k[n] \right|^2 \right]}, \quad (31)$$

since

$$\begin{aligned} \hat{\theta}_k[n|n] - \theta_k(nT) &= \mathbf{c}^T \hat{\mathbf{y}}_k[n|n] - \mathbf{c}^T \mathbf{y}(nT) \\ &= \mathbf{c}^T \mathbf{e}_k[n]. \end{aligned} \quad (32)$$

First, we diagonalize the matrix  $\mathbf{F} - \mathbf{g}_k \mathbf{c}^T \mathbf{F}$  into  $\mathbf{V} \Lambda \mathbf{V}^{-1}$ , where  $\Lambda$  is a diagonal matrix whose diagonal elements are eigenvalues of  $\mathbf{F} - \mathbf{g}_k \mathbf{c}^T \mathbf{F}$  and  $\mathbf{V}$  is a matrix whose column vectors are the corresponding eigenvectors. Here, we note that it is possible to represent  $\mathbf{g}_k$  in terms of  $\Lambda$  and  $\mathbf{V}$  as described in the following lemma.

**Lemma 1.**

$$\mathbf{g}_k = \mathbf{V} [\mathbf{I} - \Lambda] \mathbf{V}^{-1} \mathbf{c}. \quad (33)$$

*Proof.* Since  $\mathbf{F} - \mathbf{g}_k \mathbf{c}^T \mathbf{F} = \mathbf{V} \Lambda \mathbf{V}^{-1}$ , it follows that

$$\mathbf{g}_k \mathbf{c}^T \mathbf{F} \mathbf{c} = \mathbf{F} \mathbf{c} - \mathbf{V} \Lambda \mathbf{V}^{-1} \mathbf{c}. \quad (34)$$

Here, we note that  $\mathbf{c}^T \mathbf{F} \mathbf{c}$  is simply the (1, 1) component of the matrix  $\mathbf{F}$ , which is 1. Consequently, we obtain

$$\begin{aligned} \mathbf{g}_k &= \mathbf{F} \mathbf{c} - \mathbf{V} \Lambda \mathbf{V}^{-1} \mathbf{c} \\ &= \mathbf{c} - \mathbf{V} \Lambda \mathbf{V}^{-1} \mathbf{c}, \end{aligned} \quad (35)$$

from which (33) readily follows.  $\square$   $\square$

Moreover, the observer gain vector is determined solely by the eigenvalues, namely, the diagonal elements of  $\Lambda$ , as described in the following lemma.

**Lemma 2.** *Let  $\kappa_1$ ,  $\kappa_2 e^{j\phi}$ , and  $\kappa_2 e^{-j\phi}$  denote the diagonal elements of  $\Lambda$ , where  $\kappa_1$  and  $\kappa_2$  are nonnegative real numbers and  $\phi$  is a real number. (Since the characteristic equation of  $\mathbf{F} - \mathbf{g}_k \mathbf{c}^T \mathbf{F}$  is a third-order equation with real coefficients, at least one of the three eigenvalues must be real-valued and the other two must form a complex conjugate pair). Then,*

$$\begin{aligned} g_{k1} &= 1 - \kappa_1 \kappa_2^2, \\ g_{k2} &= \frac{1}{2T} \left[ 3\kappa_1 \kappa_2^2 - \kappa_2^2 - 2\kappa_1 \kappa_2 \cos \phi - \kappa_1 - 2\kappa_2 \cos \phi + 3 \right], \\ g_{k3} &= \frac{1}{T^2} \left[ -\kappa_1 \kappa_2^2 + \kappa_2^2 + 2\kappa_1 \kappa_2 \cos \phi - \kappa_1 - 2\kappa_2 \cos \phi + 1 \right], \end{aligned} \quad (36)$$

where  $g_{ki}$  denotes the  $i^{\text{th}}$  element of  $\mathbf{g}_k$ .

*Proof.* We note that the characteristic equation of  $\mathbf{F} - \mathbf{g}_k \mathbf{c}^T \mathbf{F}$  is given by

$$\begin{aligned} & \lambda^3 + (g_{k1} + g_{k2}T + 0.5g_{k3}T^2 - 3)\lambda^2 \\ & + (-2g_{k1} - g_{k2}T + 0.5g_{k3}T^2 + 3)\lambda + g_{k1} - 1 = 0. \end{aligned} \quad (37)$$

Consequently, for  $\kappa_1$ ,  $\kappa_2 e^{j\phi}$ , and  $\kappa_2 e^{-j\phi}$  to be solutions to (37), the identity equation

$$\begin{aligned} & \lambda^3 + (g_{k1} + g_{k2}T + 0.5g_{k3}T^2 - 3)\lambda^2 \\ & + (-2g_{k1} - g_{k2}T + 0.5g_{k3}T^2 + 3)\lambda + g_{k1} - 1 \\ & = \lambda^3 - (\kappa_1 + 2\kappa_2 \cos \phi)\kappa_2 + \kappa_2(\kappa_2 + 2\kappa_1 \cos \phi)\lambda - \kappa_1\kappa_2^2 = 0, \end{aligned} \quad (38)$$

must hold. Now, comparing the coefficients of the left and right-hand sides of (38) and solving for  $g_{ki}$ , we obtain the results.  $\square$

Although the result in Lemma 1 is simple, it is important in that we can construct the observer gain vector  $\mathbf{g}_k$  by suitably choosing the eigenvectors and eigenvalues that directly affect the system performance. Before proceeding with the performance analysis, let us rewrite (25) as

$$\begin{aligned} \mathbf{e}_k[n] &= \mathbf{V}\Lambda\mathbf{V}^{-1}\mathbf{e}_k[n-1] + \mathbf{V}(\mathbf{I} - \Lambda)\mathbf{V}^{-1}\mathbf{c}\varepsilon_k[n] \\ &+ [\mathbf{V}(\mathbf{I} - \Lambda)\mathbf{V}^{-1}\mathbf{c}\mathbf{c}^T - \mathbf{I}]\delta_k[n], \end{aligned} \quad (39)$$

by using (33). By repeatedly using (39), we can represent the state estimation error  $\mathbf{e}_k[n]$  in terms of its initial value  $\mathbf{e}_k[0]$ , the angle measurement error  $\varepsilon_k[m]$ , and the state evolution mismatch  $\delta_k[m]$  as described in the following:

$$\begin{aligned} \mathbf{e}_k[n] &= \mathbf{V}\Lambda^n\mathbf{V}^{-1}\mathbf{e}_k[0] + \sum_{m=1}^n \mathbf{V}\Lambda^{n-m}(\mathbf{I} - \Lambda)\mathbf{V}^{-1}\mathbf{c}\varepsilon_k[m] \\ &+ \sum_{m=1}^n \mathbf{V}\Lambda^{n-m}[(\mathbf{I} - \Lambda)\mathbf{V}^{-1}\mathbf{c}\mathbf{c}^T - \mathbf{V}^{-1}]\delta_k[m]. \end{aligned} \quad (40)$$

We note that the second and third terms of (38) involve summation from  $m = 1$  to  $m = n$ . Consequently, the second and third terms may grow larger as  $n$  grows. However, if we choose the magnitudes of all the diagonal elements of  $\Lambda$  to be sufficiently less than one, only the error  $\varepsilon_k[m]$  and mismatch  $\delta_k[m]$  with  $m$  close to  $n$  affect the values of  $\mathbf{e}_k[n]$  since  $\Lambda^k$  converges to a zero matrix as  $k$  grows. Moreover, we note that the magnitude of  $\delta_k[m]$  is order of magnitudes smaller than that of  $\varepsilon_k[n]$  unless the observation time interval  $T$  is chosen to be very large, in which the tracking scheme in this paper does not work properly. Consequently, if we choose the magnitudes of all the diagonal elements of  $\Lambda$  to be small enough compared with 1, the third term of (40) is negligibly small in usual situations. Consequently, in the following, we shall assume that  $\mathbf{e}_k[n]$  can be approximated as

$$\mathbf{e}_k[n] = \mathbf{V}\Lambda^n\mathbf{V}^{-1}\mathbf{e}_k[0] + \sum_{m=1}^n \mathbf{V}\Lambda^{n-m}(\mathbf{I} - \Lambda)\mathbf{V}^{-1}\mathbf{c}\varepsilon_k[m], \quad (41)$$

and attempt to compute  $\text{RMSE}_k$  using (31). To obtain reasonably simple and useful result for  $\text{RMSE}_k$ , we shall assume that the angle measurement error  $\varepsilon_k[m]$  has zero mean and that  $\varepsilon_k[m]$  and  $\varepsilon_k[n]$  are uncorrelated if  $n \neq m$ , which are reasonably accurate assumptions in real situations. In the following theorem, we summarize the result.

**Theorem 1.** Assume that

$$E[\varepsilon_k[n]] = 0, \quad (42)$$

and that

$$E[\varepsilon_k[m]\varepsilon_k[n]] = \delta_{mn}\sigma_k^2[n]. \quad (43)$$

Then,

$$\text{RMSE}_k = \sqrt{\frac{1}{N_T} \sum_{n=1}^{N_T} \left[ \mathbf{c}^T \mathbf{V} \Lambda^n \mathbf{V}^{-1} \mathbf{T} (\mathbf{V} \Lambda^n \mathbf{V}^{-1})^T \mathbf{c} + \sum_{m=1}^n (\mathbf{c}^T \mathbf{V} \Lambda^{n-m} (\mathbf{I} - \Lambda) \mathbf{V}^{-1} \mathbf{c})^2 \sigma_k^2[m] \right]}, \quad (44)$$

where

$$\mathbf{T} = \begin{bmatrix} \sigma_k^2[0] & \frac{\sigma_k^2[0]}{T} & \sigma_k^2[0] \\ \frac{\sigma_k^2[0]}{T} & \frac{\sigma_k^2[0] + \sigma_k^2[-1]}{T^2} & \frac{\sigma_k^2[0] + 2\sigma_k^2[-1]}{T^3} \\ \frac{\sigma_k^2[0]}{T^2} & \frac{\sigma_k^2[0] + 2\sigma_k^2[-1]}{T^3} & \frac{\sigma_k^2[0] + 4\sigma_k^2[-1] + \sigma_k^2[-2]}{T^4} \end{bmatrix}. \quad (45)$$

*Proof.* See Appendix B.  $\square$

$\square$

$$\sigma_k[n] = \tilde{\sigma}_k, \quad (46)$$

**Corollary 1.** *If we further assume, in Theorem 1, that there exists some nonnegative real number  $\tilde{\sigma}_k$  such that for all  $n$ , then*

$$\text{RMSE}_k = \tilde{\sigma}_k \sqrt{\frac{1}{N_T} \sum_{n=1}^{N_T} \left[ \mathbf{c}^T \mathbf{V} \Lambda^n \mathbf{V}^{-1} \mathbf{T} (\mathbf{V} \Lambda^n \mathbf{V}^{-1})^T \mathbf{c} + \sum_{m=1}^n (\mathbf{c}^T \mathbf{V} \Lambda^{n-m} (\mathbf{I} - \Lambda) \mathbf{V}^{-1} \mathbf{c})^2 \right]}, \quad (47)$$

where

$$\mathbf{T} = \begin{bmatrix} 1 & \frac{1}{T} & \frac{1}{T^2} \\ \frac{1}{T} & \frac{2}{T^2} & \frac{3}{T^3} \\ \frac{1}{T^2} & \frac{3}{T^3} & \frac{6}{T^4} \end{bmatrix}. \quad (48)$$

According to our test with various simulation environments, assumptions (42) and (43) hold very accurately. However, assumption (46) does not hold, and the value of  $\sigma_k^2[n]$  usually fluctuates between 1/2 and 2 times of the average value. Despite the invalidity of the assumption, the result in Corollary 1 is very useful. First, we note the simplicity of (47) compared with (44). Consequently, it is much easier, with (47), to grasp intuition about the behavior of  $\text{RMSE}_k$  in relation to the choice of  $\Lambda$ . In particular, we note that there is no further information required about the variation of  $\sigma_k^2[n]$  over  $n$ , and the  $\text{RMSE}_k$  is represented in the unit of  $\tilde{\sigma}_k$  in (47). Secondly, we can regard (47) with  $\tilde{\sigma}_k = \sqrt{(1/N_T) \sum_{n=1}^{N_T} \sigma_k^2[n]}$  as a simple and reasonable approximation of (44) since there is no reason to believe that  $\sigma_k^2[n]$  will fluctuate extremely over  $n$ .

Now, we shall show how we can select eigenvalues for optimal or near-optimal performance with (47), which is not environment dependent. Later, in this section, we shall also discuss how much the results with (47) are different from those with (44) using a number of simulation environment setups. We first choose eigenvalues  $\kappa_1, \kappa_2 e^{j\phi}$ , and  $\kappa_2 e^{-j\phi}$  and compute the right-hand side of (47). Apparently, we need to obtain  $\mathbf{V}$  for this purpose. However, we can compute  $\mathbf{F} - \mathbf{g}_k \mathbf{c}^T \mathbf{F}$ , which equals to  $\mathbf{V} \Lambda \mathbf{V}^{-1}$ , using Lemma 2. We further note that the right-hand side of (44) can be evaluated once we obtain  $\mathbf{V} \Lambda \mathbf{V}^{-1}$ .

We recall that  $\mathbf{e}_k[n]$  converges to zero as  $n$  grows if and only if both  $\kappa_1$  and  $\kappa_2$  are smaller than 1. We consider 0.05, 0.10,  $\dots$ , 0.95 as candidate values for  $\kappa_1$  and  $\kappa_2$  and  $0^\circ, 5^\circ, \dots, 175^\circ$  as the values for  $\phi$ . Consequently, we consider a total of  $19 \times 19 \times 36$  eigenvalue set combinations. Out of all considered values, the case of  $\kappa_1 = 0.7, \kappa_2 = 0.5$ , and  $\phi = 35^\circ$  provides the lowest  $\text{RMSE}_k$  value, which is  $0.8646 \tilde{\sigma}_k$ . In the following, we shall express  $\text{RMSE}_k$  in the unit of  $\tilde{\sigma}_k$  for simplicity. In Figure 1, we illustrate the

$\text{RMSE}_k$  values as functions of  $\kappa_1$  and  $\kappa_2$  for the case of  $\phi = 35^\circ$ . In this figure, we note that the  $\text{RMSE}_k$  value rises as  $(\kappa_1, \kappa_2)$  moves away from (0.7, 0.5). In particular, we note that the  $\text{RMSE}_k$  value rises relatively mildly if  $\kappa_1$  or  $\kappa_2$  is decreased. However, the  $\text{RMSE}_k$  value rises significantly as  $\kappa_1$  and  $\kappa_2$  approach 1. We found that the  $\text{RMSE}_k$  values generally rise slowly as the  $\phi$  value moves away from  $35^\circ$ . Although it is not possible to include all results with different  $\phi$  values, we include the results with  $\phi = 15^\circ$  and  $\phi = 55^\circ$  in Figures 2 and 3 to illustrate how the  $\text{RMSE}_k$  value changes with the  $\phi$  value.

So far, we discussed how we can choose the optimal observer gain vector with (45) assuming that  $\sigma_k[n]$  is a constant independent of  $n$ . Hence, the validity of the proposed method of obtaining the observer gain will depend on how the effect of variations in  $\sigma_k[n]$  affects the value of  $\text{RMSE}_k$ . According to our extensive test, the effect of variation of  $\sigma_k[n]$  is very negligible except for extreme situations that do not affect the optimal selection of the observer gain. To illustrate in a space economical way, we consider a situation in which there are five objects with DoA trajectories illustrated in Figure 4. In realistic environments, the DoA trajectories may not change as fast as in Figure 4. However, if we choose a slowly moving environment, the  $\sigma_k[n]$  value would not change very much, and hence the effect would also be negligible. For this reason, we have chosen a very fast changing environment shown in Figure 4. To obtain  $\sigma_k[n]$ , we assume that only object  $k$  moves following the trajectory  $\theta_k(nT)$  and perform simulations with the proposed scheme (DSPCLT). We recall that we divided the situations into resolvable and nonresolvable situations to avoid overlapping situations. For this reason, we assume that only object  $k$  exists in computing  $\sigma_k[n]$ .

For numerical evaluations, we assume that the number  $N$  of sensor elements is 12, the total number  $N_T$  of observation intervals is 60, and the signal-to-noise ratio is 5 dB. With  $10^6$  simulation run trials, we computed  $\sigma_k^2[n]$ , which is illustrated in Figure 5. In this figure, we note that the value of  $\sigma_k^2[n]$  is generally related to the angular acceleration. After obtaining the estimate of  $\sigma_k^2[n]$ , we computed  $\text{RMSE}_k$  using (44) in the unit of  $\tilde{\sigma}_k = \sqrt{(1/N_T) \sum_{n=1}^{N_T} \sigma_k^2[n]}$ . To compare the results with the values computed using (47), we evaluated the ratio obtained by dividing the  $\text{RMSE}_k$  value computed using (44) by that using (47) for each set of  $\kappa_1, \kappa_2$ , and  $\phi$  values. Some results of the evaluations are summarized in Table 1. In this table, the number pair  $(a, b)$  means the

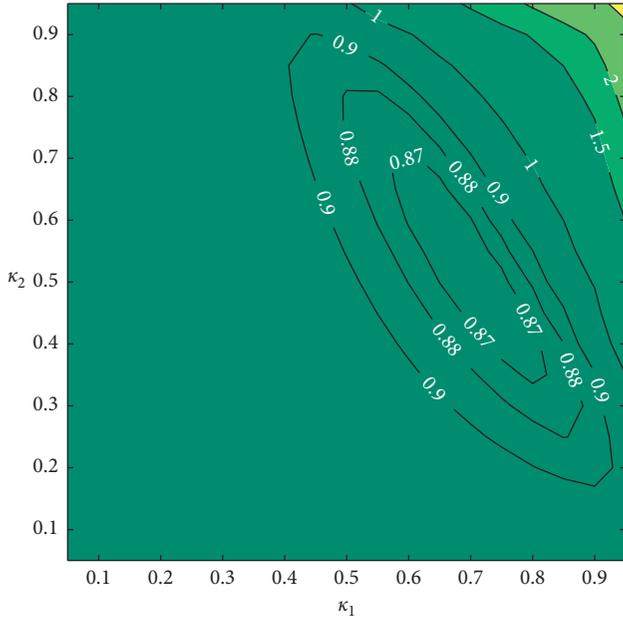


FIGURE 1:  $RMSE_k$  for  $\phi = 35^\circ$  as a function of  $\kappa_1$  and  $\kappa_2$ .

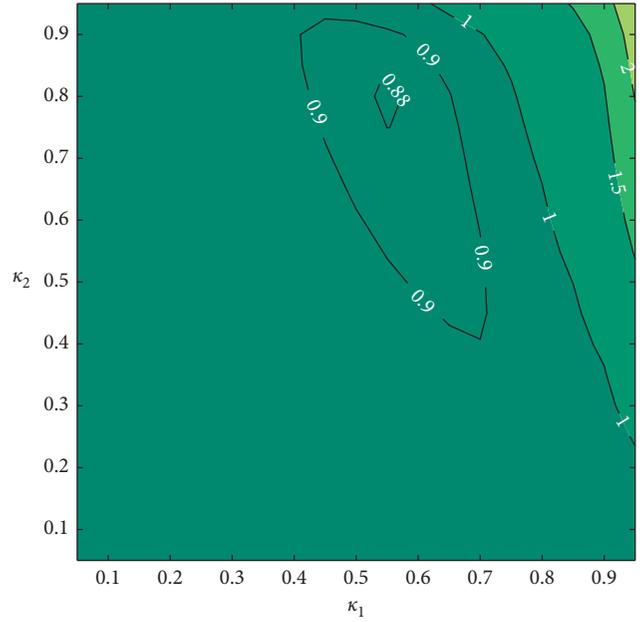


FIGURE 3:  $RMSE_k$  for  $\phi = 55^\circ$  as a function of  $\kappa_1$  and  $\kappa_2$ .

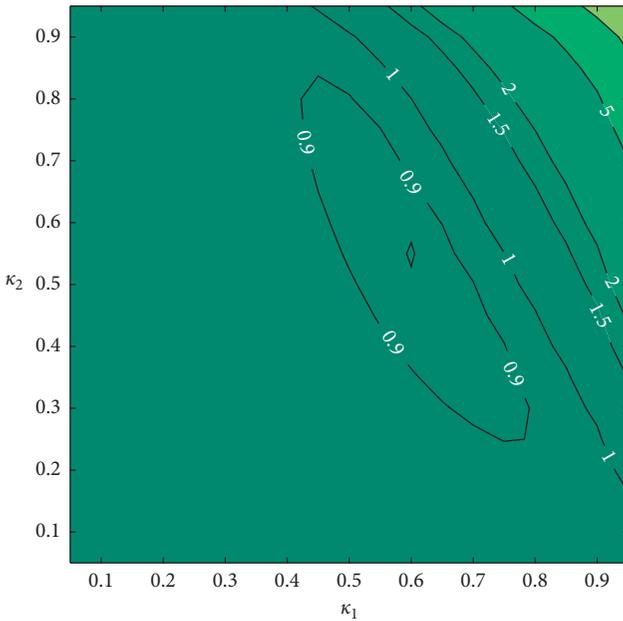


FIGURE 2:  $RMSE_k$  for  $\phi = 15^\circ$  as a function of  $\kappa_1$  and  $\kappa_2$ .

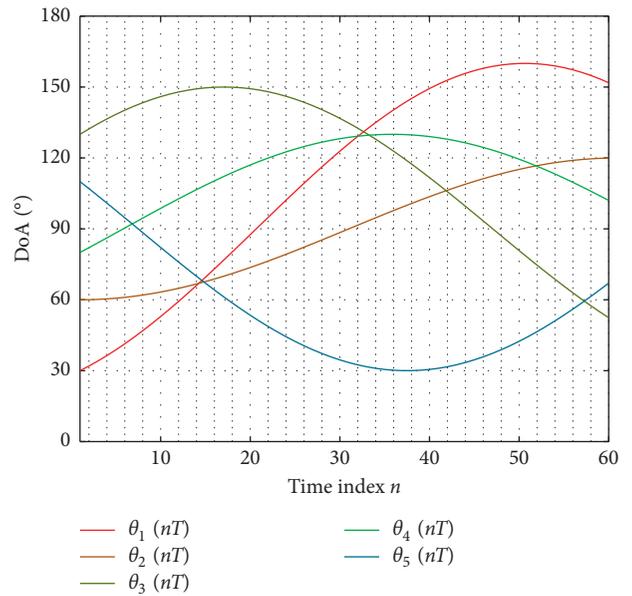
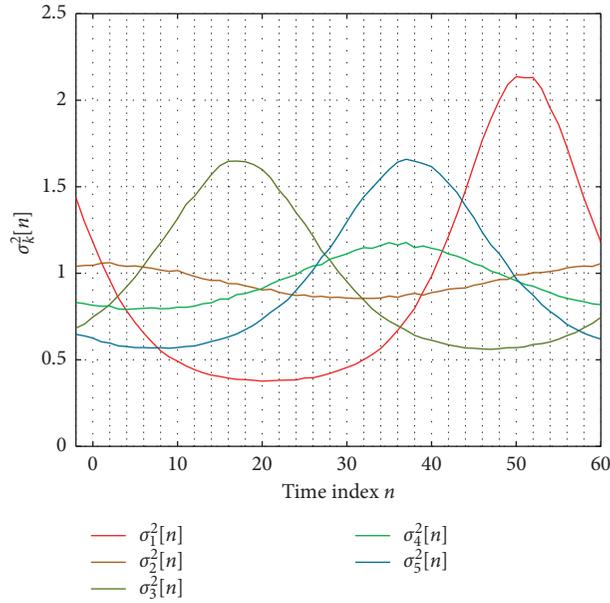


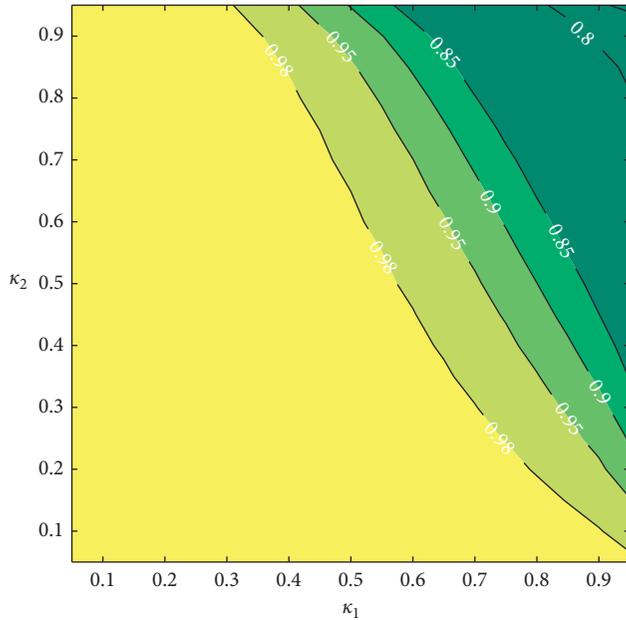
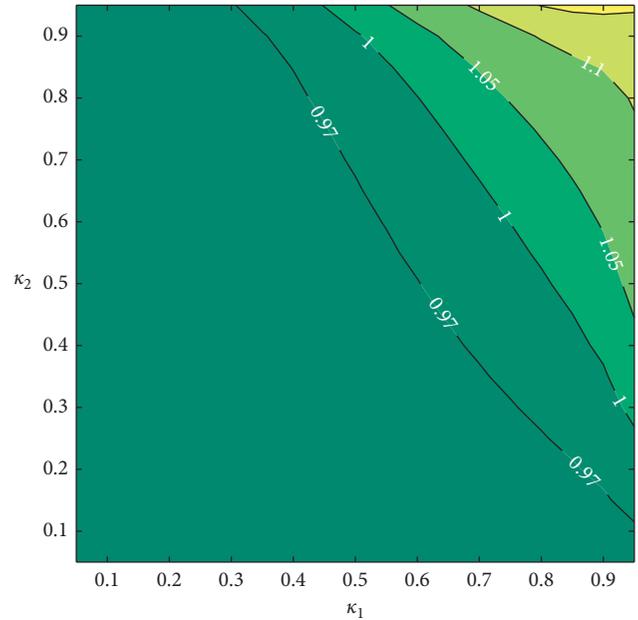
FIGURE 4: DoA trajectories used for simulations.

minimum and maximum values of the ratios over all  $\kappa_1$  and  $\kappa_2$  values for given  $\phi$  value and sequence  $\sigma_k^2[n]$ . For example, for  $\phi = 15^\circ$  and  $\sigma_1^2[n]$ , the ratio falls between 0.965 and 1.141. With this set of numbers, we can conclude how different the two  $RMSE_k$  values computed by (42) and (45) are to the extreme. However, the extreme values do not tell the whole story since the extreme values, namely, the maximum and minimum ratio values, occur at  $\kappa_1$  and  $\kappa_2$  values close to 1 so that the  $RMSE_k$  value is very large. In other words, the extreme values happen with eigenvalue selections for which  $e_k[n]$  do not converge to zero very fast.

To understand what happens, let us first consider the case of  $\phi = 15^\circ$  and  $\sigma_5^2[n]$  in which the minimum value is particularly small. In Figure 6, we plotted the ratios as a function of  $\kappa_1$  and  $\kappa_2$  for this case, namely, for  $\phi = 15^\circ$  and  $\sigma_5^2[n]$ . We observe that the minimum value 0.799 happens near the region  $\kappa_1 \approx 1$  and  $\kappa_2 \approx 1$  for which the  $RMSE_k$  values are very large. In contrast, the ratio is very close to 1.0 except for the region in which  $RMSE_k$  remains to be acceptably small. Moreover, we recall that the results in Figure 6 correspond to some extreme case in which the ratio fluctuates a lot. As a more typical case, we depict, in Figure 7,

FIGURE 5:  $\sigma_k^2[n]$  in the unit of  $\bar{\sigma}_k$ .TABLE 1: (Minimum, maximum) pair of the  $\text{RMSE}_k$  value ratio.

	$\sigma_1^2[n]$	$\sigma_2^2[n]$	$\sigma_3^2[n]$	$\sigma_4^2[n]$	$\sigma_5^2[n]$
$\phi = 15^\circ$	(0.965, 1.141)	(0.973, 1.022)	(0.842, 0.984)	(0.908, 0.980)	(0.799, 0.987)
$\phi = 35^\circ$	(0.965, 1.124)	(0.973, 1.017)	(0.859, 0.984)	(0.916, 0.980)	(0.822, 0.987)
$\phi = 55^\circ$	(0.967, 1.050)	(0.973, 0.995)	(0.922, 0.984)	(0.949, 0.980)	(0.908, 0.987)

FIGURE 6:  $\text{RMSE}_5$  computed with time-varying  $\sigma_5[n]$  for  $\phi = 15^\circ$ .FIGURE 7:  $\text{RMSE}_1$  computed with time-varying  $\sigma_1[n]$  for  $\phi = 35^\circ$ .

the ratio for  $\phi = 35^\circ$  and  $\sigma_1^2[n]$ . We note that the ratio is again very close to 1.0 except for the region in which  $\kappa_1 \approx 1$  and  $\kappa_2 \approx 1$ . Consequently, we observe that the  $\text{RMSE}_k$  value

computed with (47) is very close to that with (44) except at  $\kappa_1, \kappa_2$ , and  $\phi$  values for which  $\mathbf{e}_k[n]$  does not converge to zero fast enough, and  $\text{RMSE}_k$  is very large.

## 7. System Performance and Computational Complexity

In the section, we compare the performance and the computational complexity of the proposed scheme with those of two existing algorithms, namely, AMEND and PASTd Kalman. Most of all, we observe that the proposed scheme (DSPCLT) exhibits excellent performance with significantly low computational complexity in comparison with AMEND and PASTd Kalman.

**7.1. Performance Evaluation of Algorithms.** First, let us compare the performance of DSPCLT with that of AMEND and PASTd Kalman. As measures of performance, we use the following quantities:

$$\text{RMSE}_k[n] = \sqrt{E\left[|\hat{\theta}_k(nT) - \theta_k(nT)|^2\right]}, \quad (49)$$

$$\text{RMSE}_T = \sqrt{\frac{1}{KN_T} \sum_{k=1}^K \sum_{n=1}^{N_T} E\left[|\hat{\theta}_k(nT) - \theta_k(nT)|^2\right]}, \quad (50)$$

where  $\hat{\theta}_k(nT)$  is the estimated DoA of  $\theta_k(nT)$  and  $N_T$  is the total number of observation intervals. We note that  $\text{RMSE}_k[n]$  is the RMSE of the DoA of the  $k^{\text{th}}$  object at time  $t = nT$  and that  $\text{RMSE}_T$  represents the total RMSE over all objects and observation intervals. We recall that we regard  $\hat{\theta}_k[n|n]$  as  $\hat{\theta}_k(nT)$  in the proposed scheme.

However, we need to be very careful in assessing the performance with RMSE. This is because there are some possibilities of tracking wrong objects after passing through a crossover point or of object tracking divergence due to lack of sensitivity to object movements, which can result in huge estimation errors. The value of RMSE can be meaningless when such catastrophic tracking failure happens. To make the value of RMSE more meaningful, we shall regard the situation as tracking failure if and only if the estimated DoA of any of the signals is more than  $5^\circ$  away from the actual DoA and compute the RMSE value disregarding such cases of tracking failure. (In this paper, we shall measure DoAs and RMSE in degrees rather than in radians.) Since the RMSE values do not account for the cases of tracking failure, we shall also consider, in addition to RMSE, the probability  $P_F$  of tracking failure as a performance measure.

For performance evaluations, we first consider a situation in which  $K = 3$  signals impinge upon a ULA with  $N = 12$  sensors separated by half-wavelength from time-varying directions  $\theta_1(nT)$ ,  $\theta_2(nT)$ , and  $\theta_3(nT)$ ,  $n = 1, 2, \dots, N_T = 60$ , illustrated in Figure 8. We assume that the signal-to-noise ratio of each signal at each sensor is given as 5 dB. During each time interval  $[nT, (n+1)T)$ , total  $N_s = 200$  data snapshots are uniformly captured at each sensor with  $\delta = 1/N_s$ .

We note that the choice of forgetting factor  $\rho$  affects the tracking performance. Hence, we compare the performance with various  $\rho$  values from 0.91 to 0.99. In the case of PASTd Kalman, we also consider, instead of  $\rho^{N_s - n}$ , the forgetting

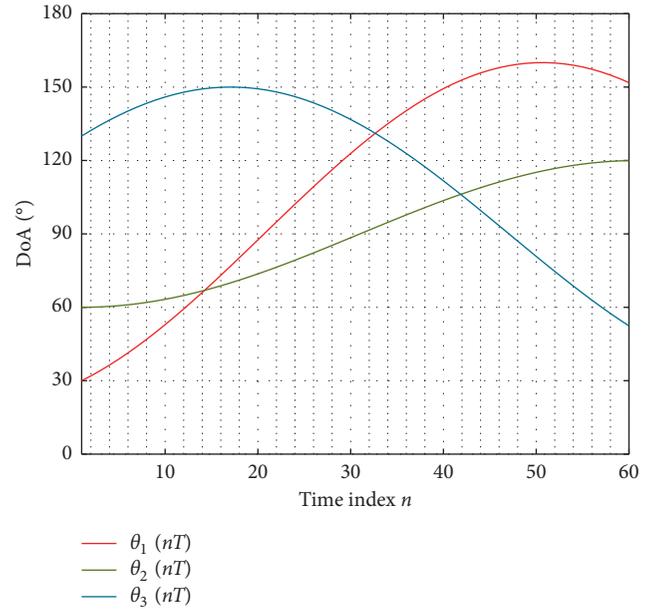


FIGURE 8: DoA trajectories used for simulations.

coefficient  $n/(n+1)$ , which was used in [21]. For AMEND, we use the observer gain vector obtained with the eigenvalues  $0.75$ ,  $0.75e^{j\pi/18}$ , and  $0.75e^{-j\pi/18}$  as suggested in [22]. However, we select the observer gain vector obtained, using Lemma 2, with eigenvalues  $0.7$ ,  $0.5e^{j35\pi/180}$  and  $0.5e^{-j35\pi/180}$  for DSPCLT. We recall that observer gain is not used in PASTd Kalman.

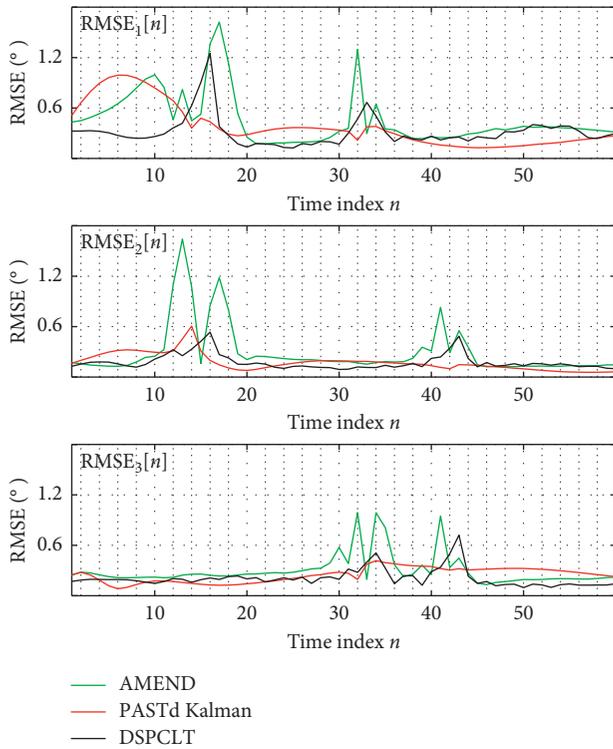
As the value of  $M$  is used to compute  $\bar{r}_m$  in (A.3) and (A.4), we use  $M = K$  since the RMSE performance enhances rapidly as  $M$  increases from 1 to  $K$  and then becomes virtually invariant as it further grows. Also, we separate the resolvable and nonresolvable situations with threshold values  $5^\circ$ ,  $4.5^\circ$ , and  $4^\circ$  for  $N = 12, 16$ , and  $20$ , respectively.

Simulations are performed with  $10^5$  trials. The failure probability  $P_F$  and  $\text{RMSE}_T$  obtained through the simulations are summarized in Table 2. We may say that AMEND exhibits the most reliable performance in that it does not experience failure for any choice of  $\rho$  value. However, AMEND is worst in terms of  $\text{RMSE}_T$  value. In particular, we note that both DSPCLT and PASTd Kalman exhibit not only better  $\text{RMSE}_T$  values than AMEND but also achieve zero failures. The best  $\text{RMSE}_T$  value for each scheme is marked in bold-faced fonts, and we note that DSPCLT achieves the lowest  $\text{RMSE}_T$  value among the three schemes. By the way, the  $\text{RMSE}_T$  value 0.6160 listed on the lowest row for PASTd Kalman corresponds to the forgetting coefficient  $n/(n+1)$  suggested in [21].

To illustrate the performance variation over time,  $\text{RMSE}_k[n]$  is plotted in Figure 9. In this figure, we used  $\rho = 0.95$  for AMEND and  $0.97$  for DSPCLT and PASTd Kalman. In Table 2, we note that AMEND exhibits the worst  $\text{RMSE}_T$  values. Also, we note DSPCLT achieves slightly better  $\text{RMSE}_T$  value compared with PASTd Kalman. In Figure 9, we observe that DSPCLT exhibits best overall  $\text{RMSE}_k[n]$  performance except for the region where the DoAs of objects

TABLE 2:  $P_F$  and  $\text{RMSE}_T$  ( $K = 3, N = 12, M = 3, N_s = 200$ ).

$\rho$	DSPCLT		AMEND		PASTd Kalman	
	$P_F$ [%]	$\text{RMSE}_T$ [°]	$P_F$ [%]	$\text{RMSE}_T$ [°]	$P_F$ [%]	$\text{RMSE}_T$ [°]
0.91	0.40	0.4354	0	0.4719	0.83	0.3450
0.92	0.24	0.4155	0	0.4634	0.57	0.3276
0.93	0.12	0.3742	0	0.4561	0.30	0.3084
0.94	0.09	0.3541	0	0.4510	0.10	0.2928
0.95	0.05	0.3181	0	<b>0.4503</b>	0.03	0.2792
0.96	0.03	0.3014	0	0.4602	0.02	0.2831
0.97	<b>0</b>	<b>0.2715</b>	0	0.4905	<b>0</b>	<b>0.3247</b>
0.98	0	0.2923	0	0.5713	0	0.4308
0.99	0	0.3851	0	0.7496	0	0.6328
—	—	—	—	—	0	0.6160

FIGURE 9: Performance variation over time.  $N = 12, K = 3$ .

overlap. In particular, we note that the performance of DSPCLT is significantly better than others for the first object during  $n \in \{1, 2, \dots, 10\}$ . From Figure 8, we note that the DoA of the first object changes very fast over  $n \in \{1, 2, \dots, 10\}$ . Consequently, we can conclude that the delay compensation algorithm introduced in Section 5.2 indeed provides improved performance in fast changing environments. We also observe that DSPCLT is less susceptible to object crossover compared with AMEND. This is because the nonresolvable situation is separately treated, in the proposed scheme, using the algorithm described in Section 4.

One of the main reasons for the overall performance superiority of the proposed scheme is the comparatively larger effective aperture size with the same number of sensor elements. For illustration of the results of effective aperture size, we compare, in Figure 10, realizations of the function  $f(\theta)$ , defined in (11), at time index  $n = 19$ . In this figure, we observe that  $f(\theta)$  is much sharper for DSPCLT than for AMEND and PASTd Kalman. For this reason, DSPCLT provides better overall RMSE performance than AMEND rivaling PASTd Kalman that employs much more sophisticated Kalman filter algorithm.

We next study the performance variation with increased number of objects. For illustration, we consider the situation in which there are  $K = 5$  objects with trajectories illustrated in Figure 4. Here, we note that AMEND can track only  $K$  objects satisfying  $K < N/2$ , while DSPCLT and PASTd Kalman can track up to  $N - 1$  objects. Consequently, at least  $N = 11$  sensor elements are needed for AMEND to be able to track  $K = 5$  objects. To see the effect of the number of sensor elements, we consider three values 12, 16, and 20 as the number  $N$  of antenna elements of the ULA. Again, the received SNR of each of the 5 objects is set to be 5 dB, the number  $N_s$  of snapshots is chosen to be 200 with  $\delta = 1/N_s$ , and the value of  $M$  is chosen to be the same as  $K$ .

The simulations are performed with  $10^5$  runs again, and the results are summarized in Table 3 and Figure 11. As in the case of  $K = 3$ , we obtained simulation results with various forgetting factor values and included the best results in the table and the figure. In particular, the value  $\rho$  indicates the forgetting factor value that leads to the best  $\text{RMSE}_T$  value. In Table 3 and Figure 11, we observe that AMEND exhibits the worst overall performance in almost every aspect. We observe that both AMEND and DSPCLT experienced nonzero failure probabilities, which decrease as the number of sensor elements increases. Consequently, we can conclude that PASTd Kalman achieves the most reliable performance among the three considered schemes. However, we note that DSPCLT exhibits performance better than PASTd Kalman with a sufficiently large number of sensor elements. We also note that DSPCLT, compared with AMEND, not only achieves lower failure probability but also exhibits lower  $\text{RMSE}_T$ . In particular, we observe that the  $\text{RMSE}_T$  value of DSPCLT is significantly lower than that of AMEND despite the fact that  $\text{RMSE}_T$  is computed excluding the cases in which tracking fails. Next, in Figure 11, the sequence of  $\text{RMSE}_k[n]$  values for each of the five objects is depicted for the case of  $N = 20$ . As in the previous case of  $K = 3$  objects, AMEND exhibits the worst performance, and DSPCLT achieves similar or better performance, compared with PASTd Kalman, except around crossover points.

**7.2. Computational Complexity.** In this section, we compare the computational complexities of DSPCLT, AMEND, and PASTd Kalman. Before proceeding, we note that the complexity of DSPCLT varies depending on whether the object is in the resolvable or nonresolvable situation. Since the complexity of DSPCLT in the nonresolvable situation is

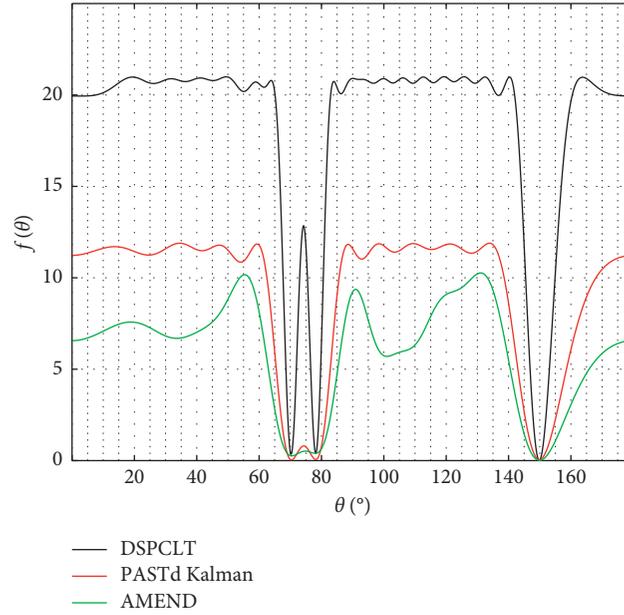

 FIGURE 10: Comparison of  $f(\theta)$  functions at time index  $n = 19$ .

 TABLE 3:  $P_F$  and  $RMSE_T$  ( $K = 5, M = 5, N_s = 200, N = 12, 16, 20$ ).

$N$	DSPCLT			AMEND			PASTd Kalman		
	$\rho$	$P_F$ [%]	$RMSE_T$ [°]	$\rho$	$P_F$ [%]	$RMSE_T$ [°]	$\rho$	$P_F$ [%]	$RMSE_T$ [°]
12	0.97	4.41	0.5080	0.97	22.16	1.1589	0.97	0	0.3787
16	0.97	0.04	0.2791	0.95	0.31	0.4360	0.97	0	0.3135
20	0.95	0	0.1884	0.93	0	0.2937	0.96	0	0.2433

significantly smaller than that in the nonresolvable situation and nonresolvable situation happens only very rarely when the number of sensor elements is large enough, we shall assume that only resolvable situation happens when computing the complexity of DSPCLT. Before proceeding, we note that all the three tracking schemes DSPCLT, AMEND, and PASTd Kalman can be divided into three processes, namely, “subspace construction,” “projector construction,” and “filtering.” Here, the subspace construction is in essence the construction of the vectors that generate the signal subspace.

In PASTd Kalman [21] and AMEND [22], subspace construction is carried out using the recursive least square (RLS) method, while the vectors in DSPCLT are constructed directly from the estimated correlation matrix using (A.1). In the projector construction stage, an orthogonal projector matrix such as  $\Pi[n]$  is constructed. And state vectors are updated using Kalman filter or Luenberger observer in the final stage of filtering. As described in Section 1, Kalman filter is used in PASTd Kalman, and Luenberger observer is used in DSPCLT and AMEND. In Table 4, the numbers of real floating point operations needed to carry out each of the three processes for each of the three schemes as well as the total numbers are presented.

For more concrete picture of computational complexity, we evaluated the numbers in Table 4 with the choice of parameters used in obtaining Figures 9 and 11, the results of which are presented, respectively, in Tables 5 and 6. We note that a significant portion of computational complexity is needed in subspace construction. We note that AMEND requires particularly large numbers of operations in the subspace construction compared with DSPCLT. This is because AMEND uses matrix inversion and QR decomposition even though it does not use ED nor SVD. We note that DSPCLM does not require matrix inversion nor QR decomposition. We note that PASTd Kalman, compared with DSPCLT and AMEND, requires larger amount of operations in the filtering stage since it uses computationally expensive Kalman filtering. Finally, we note that DSPCLT requires 2 ~ 6 times more operations to carry out projector construction and filtering compared with AMEND. This is because DSPCLT has larger effective aperture size despite that the same number of sensors is used. However, the overall computational complexity of DSPCLT is significantly smaller than that of AMEND and PASTd Kalman because the computational complexity of the second stage is much smaller than that of either the first or the third stage. Overall, DSPCLT requires only about 11.99% (18.78%) and 7.70%

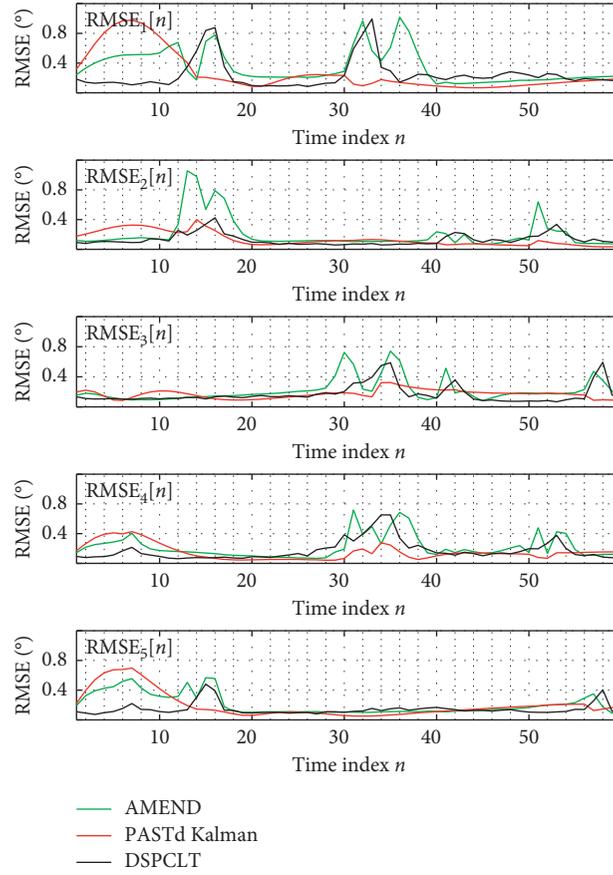
FIGURE 11: Performance variation over time.  $N = 20, K = 5$ .

TABLE 4: Required floating point operations.

	DSPCLT	AMEND	PASTd Kalman
Subspace construction	$5M(2N - K + 1)N_s$	$64K^2(N - 2K)N_s$	$K(28N + 7)N_s$
Projector construction	$8K(2N - K)^2$	$16(N - K)(N - 2K)^2$	$8KN^2$
Filtering	$16(2N - K)^2$	$16(N - K)^2$	$8N^2N_s + 8(K + 2)N^2$
Total	$5M(2N - K + 1)N_s + (8K + 16)(2N - K)^2$	$64K^2(N - 2K)N_s + 16(N - K)[(N - 2K)^2 + N - K]$	$(28KN + 8N^2 + 7K)N_s + 16(1 + K)N^2$

TABLE 5: Required floating point operations:  $N = 12, M = K = 3, N_s = 200$ .

	DSPCLT	AMEND	PASTd Kalman
Subspace construction	66,000	691,200	205,800
Projector construction	10,584	5,184	3,456
Filtering	7,056	1,296	236,160
Total	83,640	697,680	445,416

TABLE 6: Required floating point operations:  $N = 20, M = K = 5, N_s = 200$ .

	DSPCLT	AMEND	PASTd Kalman
Subspace construction	180,000	3,200,000	567,000
Projector construction	49,000	24,000	16,000
Filtering	19,600	3,600	662,400
Total	248,600	3,227,600	1,245,400

(19.96%) real floating point operations compared with AMEND and PASTd Kalman, respectively, for the case of  $N = 12, M = K = 3, N_s = 200$  and  $N = 20, M = K = 5, N_s = 200$ .

## 8. Conclusion

In this paper, we proposed the direct signal space construction method Luenberger tracker with quadratic least square regression for computationally efficient DoA tracking. Also, we studied analytically how the selection of observer gain value affects the system performance and described how we can use the theoretical results to optimize the observer gain value. We note that the proposed algorithm combines the direct signal space construction method and Luenberger observer to achieve computational complexity significantly lower than other schemes. Also, the algorithm achieves enhanced performance by employing a method of delay compensation which accounts for the difference in the times of estimation and data collection. The proposed scheme avoids the possible issue of non-resolvability by treating separately the situation of object overlap. As a result, the proposed scheme achieves performance similar or superior to existing algorithms with significantly lower computational complexity, which was verified with a variety of simulation results.

## Appendix

### A. Orthogonal Projector Construction with DSPCM

In this paper, we propose to use a method of constructing an orthogonal projector based on DSPCM in [14, 15]. We first describe how the subspace and the orthogonal projector are defined. Then, we describe how we obtain an estimate of the orthogonal projector is computed based on the data collected at antenna sensor elements.

*A.1. Orthogonal Projector Based on DSPCM.* Before discussing the concept of the orthogonal projector, we need to define the subspace we are going to work with. For this purpose, we first define the signal-only correlation matrix  $\mathbf{R}_s$  by

$$\mathbf{R}_s(t) \equiv E[\mathbf{x}(t)\mathbf{x}(t)^H] - N_0\mathbf{I}_N, \quad (\text{A.1})$$

where the value of noise power spectral density  $N_0$  shall be assumed to be estimated with long-term observation of signal-free data collected at the sensors. It is not difficult to show, by inserting (1) into (49), that  $\mathbf{R}_s(t) = \mathbf{A}\mathbf{P}\mathbf{A}^H$  and hence that  $\mathbf{R}_s(t)$  is a Hermitian Toeplitz matrix with its  $(m, n)$  element given by  $R_{mn}(t) = \sum_{k=1}^K p_k e^{j(m-n)\nu_k(t)}$ . Due to the Toeplitz structure, there are at most  $2N - 1$  distinct elements in  $\mathbf{R}_s(t)$ , namely,  $R_{mN}, R_{m-1N}, \dots, R_{NN}, R_{NN-1}, \dots$ , and  $R_{N1}$ . For simplicity of notation, we shall denote these  $2N - 1$  elements of  $\mathbf{R}_s$  by  $r_1(t), r_2(t), \dots, r_{2N-1}(t)$ . We note that  $r_m(t)$  satisfies

$$r_m(t) = \sum_{k=1}^K p_k e^{j(-N+m)\nu_k(t)}, \quad m = 1, 2, \dots, 2N - 1, \quad (\text{A.2})$$

and contains information about the DoAs, namely,  $\theta_1(t), \dots, \theta_K(t)$ .

While traditional subspace-based algorithms such as MUSIC or ESPRIT attempt to define signal or noise subspace by making orthogonal decomposition of the given vector space, DSPCM attempts to define not by a subspace of a given vector space but by rearranging the elements of  $\mathbf{R}_s$ . To be more specific, we define  $K$  signal vectors  $\mathbf{r}_1(t), \dots, \mathbf{r}_K(t)$  by

$$\mathbf{r}_k(t) = [r_k(t), r_{k+1}(t), \dots, r_{k+2N-K-1}(t)]^T, \quad k = 1, 2, \dots, K, \quad (\text{A.3})$$

and regard as the signal space at time  $t$  the vector space spanned by these  $K$  signal vectors. If we let  $\mathcal{V}$  as the complex vector space consisting of all  $L = 2N - K$ -dimensional column vectors, then the signal space is a subspace of  $\mathcal{V}$ . We shall call the orthogonal complement of the signal space within  $\mathcal{V}$  the noise space. In the first and the third stage of the Luenberger observer, we need the orthogonal projector onto the noise subspace, which we shall denote by  $\Pi(t)$ . From the knowledge of elementary linear algebra [28], it is not difficult to see that  $\Pi(t)$  can be written as

$$\Pi(t) = \mathbf{I}_L - (\hat{\mathbf{r}}_1(t)\hat{\mathbf{r}}_1^H(t) + \dots + \hat{\mathbf{r}}_K(t)\hat{\mathbf{r}}_K^H(t)), \quad (\text{A.4})$$

where the vectors  $\hat{\mathbf{r}}_1(t), \dots, \hat{\mathbf{r}}_K(t)$  are the orthonormal vectors obtainable by applying Gram-Schmidt process on  $\mathbf{r}_1(t), \dots, \mathbf{r}_K(t)$ . From the results in [14, 15], we note that  $\theta$  equals to one of  $\theta_1(t), \dots, \theta_K(t)$  if and only if  $\mathbf{a}^H(\theta)\Pi(t)\mathbf{a}(\theta) = 0$ , which can be used to make initial estimates of the DoAs.

*A.2. Estimation of the Orthogonal Projector.* Next, to build an estimate  $\hat{\Pi}[n]$  of the projector  $\Pi(nT)$ , data samples at times between  $(n-1)T$  and  $nT$  are used. In particular, we assume that there are total  $N_s$  available samples of  $x_k(t)$ , for each  $k = 1, 2, \dots, N$ , snapshotted at times  $t = [n - (N_s - 1)\delta]T, [n - (N_s - 2)\delta]T, \dots, [n - (N_s - N_s)]T$ . Here,  $\delta$  is a certain positive real number such that  $N_s\delta \leq 1$ . For notational simplicity, we shall write  $x_{k,l}$  for  $x_k(nT - (N_s - l)\delta T)$ . From these data samples, the estimates  $\tilde{r}_m$  of  $r_m$  in (50) are computed by

$$\tilde{r}_m = \frac{1}{m} \sum_{k=1}^m \sum_{l=1}^{N_s} \frac{(1-\rho)\rho^{N_s-l}}{1-\rho^{N_s}} x_{k,l} x_{N-m+k,l}^*, \quad (\text{A.5})$$

for  $m = 1, 2, \dots, M$ , and

$$\tilde{r}_m = \frac{1}{M} \sum_{k=1}^M \sum_{l=1}^{N_s} \frac{(1-\rho)\rho^{N_s-l}}{1-\rho^{N_s}} x_{k,l} x_{N-m+k,l}^* - N_0\delta_{Nm}, \quad (\text{A.6})$$

for  $m = M, M+1, \dots, N$ . Here,  $M$  is a positive integer not exceeding  $N$ , and  $\rho$  denotes a forgetting factor chosen to satisfy  $0 < \rho < 1$ . We, then, obtain  $\tilde{r}_{N+1}, \dots, \tilde{r}_{2N-1}$  from

$\tilde{r}_1, \dots, \tilde{r}_{N-1}$  using the relation  $\tilde{r}_{2N-m} = \tilde{r}_m^*$ ,  $m = 1, 2, \dots, N-1$ . Next, we obtain  $K$  orthonormal vectors  $\hat{r}_1, \dots, \hat{r}_K$  by applying Gram-Schmidt procedure on  $\tilde{r}_1, \dots, \tilde{r}_K$ , where  $\tilde{r}_k$ 's are  $N$ -dimensional column vectors defined by  $\tilde{r}_k = [\tilde{r}_k, \tilde{r}_{k+1}, \dots, \tilde{r}_{k+L-1}]^T$ . Next, we obtain  $\hat{\Pi}[n]$  by

$$\hat{\Pi}[n] = \mathbf{I}_L - (\hat{r}_1 \hat{r}_1^T + \dots + \hat{r}_K \hat{r}_K^T). \quad (\text{A.7})$$

## B. Proof of Theorem 1

First, using assumptions (42) and (43), we obtain

$$\begin{aligned} e_k[0] &= \hat{y}_k[0|0] - y_k(0) \\ &= \begin{bmatrix} \tilde{\theta}_k[0|0] - \theta_k(0) \\ \frac{\tilde{\theta}_k[0|0] - \tilde{\theta}_k[-1|-1] - \theta_k(0) - \theta_k(-T)}{T} \\ \frac{\tilde{\theta}_k[0|0] - 2\tilde{\theta}_k[-1|-1] + \tilde{\theta}_k[-2|-2] - \theta_k(0) - 2\theta_k(-T) + \theta_k(-2T)}{T^2} \end{bmatrix}, \end{aligned} \quad (\text{B.2})$$

$$= \begin{bmatrix} \varepsilon_k[0] \\ \frac{\varepsilon_k[0] - \varepsilon_k[-1]/T}{\varepsilon_k[0] - 2\varepsilon_k[-1] + \varepsilon_k[-2]/T^2} \end{bmatrix}. \quad (\text{B.3})$$

Now, by directly plugging (B.3) into  $E[\mathbf{e}_k[0]\mathbf{e}_k^T[0]]$  and using (43), we obtain

$$E[\mathbf{e}_k[0]\mathbf{e}_k^T[0]] = \begin{bmatrix} \sigma_k^2[0] & \frac{\sigma_k^2[0]}{T} & \sigma_k^2[0] \\ \frac{\sigma_k^2[0]}{T} & \frac{\sigma_k^2[0] + \sigma_k^2[-1]}{T^2} & \frac{\sigma_k^2[0] + 2\sigma_k^2[-1]}{T^3} \\ \frac{\sigma_k^2[0]}{T^2} & \frac{\sigma_k^2[0] + 2\sigma_k^2[-1]}{T^3} & \frac{\sigma_k^2[0] + 4\sigma_k^2[-1] + \sigma_k^2[-2]}{T^4} \end{bmatrix}, \quad (\text{B.4})$$

which completes the proof.

## Data Availability

The data used to support the findings of this study are included in the two supplementary information files.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

$$\begin{aligned} E\left[\left|\mathbf{c}^T \mathbf{e}_k[n]\right|^2\right] &= E\left[\mathbf{c}^T \mathbf{V} \Lambda^n \mathbf{V}^{-1} \mathbf{e}_k[0] + \sum_{m=1}^n \mathbf{c}^T \mathbf{V} \Lambda^{n-m} (\mathbf{I} - \Lambda) \mathbf{V}^{-1} \mathbf{c} \varepsilon_k[m]\right]^2 \\ &= \mathbf{c}^T \mathbf{V} \Lambda^n \mathbf{V}^{-1} E[\mathbf{e}_k[0]\mathbf{e}_k^T[0]] (\mathbf{V} \Lambda^n \mathbf{V}^{-1})^T \mathbf{c} \\ &\quad + \sum_{m=1}^n (\mathbf{c}^T \mathbf{V} \Lambda^{n-m} (\mathbf{I} - \Lambda) \mathbf{V}^{-1} \mathbf{c})^2 E[\varepsilon_k[m]\varepsilon_k[m]]. \end{aligned} \quad (\text{B.1})$$

Next, to compute  $E[\mathbf{e}_k[0]\mathbf{e}_k^T[0]]$ , we note that

## Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1B07045719) and Korea Electric Power Corporation (Grant number: R18XA02).

## Supplementary Materials

The data used to support the findings of this study are included within the supplementary information files named 'DoA\_k.xlsx' and 'sigma2\_k.xlsx': (1) 'DoA\_k.xlsx' includes

the numerical values of  $\theta_k(nT)$  in Figures 4 and 8; (2) 'sigma2\_k.xlsx' includes the numerical values of  $\sigma_k^2[n]$  in Figure 5. (*Supplementary Materials*)

## References

- [1] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [2] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.
- [3] A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *Proceedings of the ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8, pp. 336–339, Boston, MA, USA, April 1983.
- [4] P. Pal and P. P. Vaidyanathan, "Nested arrays: a novel approach to array processing with enhanced degrees of freedom," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4167–4181, 2010.
- [5] P. Pal and P. P. Vaidyanathan, "Coprime sampling and the music algorithm," in *Proceedings of the 2011 Digital Signal Processing and Signal Processing Education meeting (DSP/SPE)*, IEEE, Sedona, AZ, USA, pp. 289–294, January 2011.
- [6] C. El Kassisi, J. Picheral, G. Fleury, and C. Mokbel, "Direction of arrival estimation using em-esprit with nonuniform arrays," *Circuits, Systems, and Signal Processing*, vol. 31, no. 5, pp. 1787–1807, 2012.
- [7] K. Huang, J. Sha, W. Shi, and Z. Wang, "An efficient fpga implementation for 2-d music algorithm," *Circuits, Systems, and Signal Processing*, vol. 35, no. 5, pp. 1795–1805, 2016.
- [8] J. Chen, S. Guan, Y. Tong, and L. Yan, "Two-dimensional direction of arrival estimation for improved archimedean spiral array with music algorithm," *IEEE Access*, vol. 6, pp. 49740–49745, 2018.
- [9] X. Wang, W. Wang, and D. Xu, "Low-complexity ESPRIT-root-MUSIC algorithm for non-circular source in bistatic MIMO radar," *Circuits, Systems, and Signal Processing*, vol. 34, no. 4, pp. 1265–1278, 2015.
- [10] J. Munier and G. Y. Delisle, "Spatial analysis using new properties of the cross-spectral matrix," *IEEE Transactions on Signal Processing*, vol. 39, no. 3, pp. 746–749, 1991.
- [11] S. Marcos, A. Marsal, and M. Benidir, "The propagator method for source bearing estimation," *Signal Processing*, vol. 42, no. 2, pp. 121–138, 1995.
- [12] A. Eriksson, P. Stoica, and T. Soderstrom, "On-line subspace algorithms for tracking moving sources," *IEEE Transactions on Signal Processing*, vol. 42, no. 9, pp. 2319–2330, 1994.
- [13] N. Xi and L. Liping, "A computationally efficient subspace algorithm for 2-d doa estimation with l-shaped array," *IEEE Signal Processing Letters*, vol. 21, no. 8, pp. 971–974, 2014.
- [14] D.-S. Yoo, "Subspace-based doa estimation with sliding signal-vector construction for ula," *Electronics Letters*, vol. 51, no. 17, pp. 1361–1363, 2015.
- [15] D.-S. Yoo, "A low complexity subspace-based doa estimation algorithm with uniform linear array correlation matrix subsampling," *International Journal of Antennas and Propagation*, vol. 2015, Article ID 323545, 10 pages, 2015.
- [16] B. Yang, "Projection approximation subspace tracking," *IEEE Transactions on Signal Processing*, vol. 43, no. 1, pp. 95–107, 1995.
- [17] G. Xu, H. Zha, G. Golub, and T. Kailath, "Fast algorithms for updating signal subspaces," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 41, no. 8, pp. 537–549, 1994.
- [18] C. K. Sword, M. Simaan, and E. W. Kamen, "Multiple target angle tracking using sensor array outputs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 2, pp. 367–373, 1990.
- [19] K. W. Lo and C. K. Li, "An improved multiple target angle tracking algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 797–805, 1992.
- [20] J. M. Goldberg, "Joint direction-of-arrival and array-shape tracking for multiple moving targets," *IEEE Journal of Oceanic Engineering*, vol. 23, no. 2, pp. 118–126, 1998.
- [21] J. Sanchez-Araujo and S. Marcos, "An efficient pastd-algorithm implementation for multiple direction of arrival tracking," *IEEE Transactions on Signal Processing*, vol. 47, no. 8, pp. 2321–2324, 1999.
- [22] J. Xin, N. Zheng, and A. Sano, "Subspace-based adaptive method for estimating direction-of-arrival with luenberger observer," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 145–159, 2010.
- [23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [24] D. Luenberger, "An introduction to observers," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 596–602, 1971.
- [25] J. Xin and A. Sano, "Computationally efficient subspace-based method for direction-of-arrival estimation without eigendecomposition," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 876–893, 2004.
- [26] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 387–392, 1985.
- [27] J. Makhoul, "Linear prediction: a tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.
- [28] S. Lang, *Linear Algebra*, Springer, Berlin, Germany, 3rd edition, 2004.