IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 September 2022, accepted 4 October 2022, date of publication 6 October 2022, date of current version 13 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3212832

RESEARCH ARTICLE

Tracking of Moving Sources in a Reverberant Environment Using Evolutionary Algorithms

MINGSIAN R. BAI[1,2], (Senior Member, IEEE), FAN-JIE KUNG[2], AND CHUN-SHIAN TAO[1,3]
[1]Department of Power Mechanical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan
[2]Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan
[3]MediaTek Inc., Hsinchu City 300096, Taiwan

Corresponding author: Mingsian R. Bai (msbai@pme.nthu.edu.tw)

This work was supported by the Ministry of Science and Technology (MOST), Taiwan, under Project 107-2221-E-007-039-MY3.

ABSTRACT This paper describes a source tracking technique in a reverberant environment using a new combination of an adaptive species-based particle swarm optimization (ASPSO) algorithm and a multiple signal classification (MUSIC) algorithm. To mitigate the effects of reverberation, an insightful dereverberation method based on an online autoregressive (AR) array and a minimum variance distortionless response (MVDR) beamformer is developed to dereverberate the microphone signal prior to direction of arrival (DOA) estimation using MUSIC. On the basis of several evolutionary schemes, ASPSO enables rapid tracking by finding local maxima in the MUSIC pseudospectrum. In the ASPSO algorithm, particles are divided into different species, where each species is associated with a sound source. As the sound source moves, the DOA information is dynamically updated using ASPSO, in which the inertia weight decreases progressively to prevent premature convergence. Two update rules for adapting the filter coefficients are employed for drastically moving sources. Simulations and experiments are conducted using a circular microphone array to validate the proposed ASPSO with AR (ASPSO-AR) algorithm. The results demonstrate that ASPSO-AR requires one-third of the processing time of the grid search (GS) method. In addition, the root-mean-square error (RMSE) of the ASPSO-AR algorithm is 10° less than that of the GS method.

INDEX TERMS Autoregression, direction of arrival, grid search, multiple signal classification, particle swarm optimization.

I. INTRODUCTION

Sound source localization using microphone arrays finds modern applications in voice assistants [1], smart homes [2], mobile phones [3], hands-free systems [4], and meeting transcription systems [5]. Direction of arrival (DOA) estimation of sources is central to many acoustic signal processing problems, such as source separation [6] and speech enhancement [7]. In practical application scenarios, a moving sound source such as a human speaker can pose difficulties for localization, particularly in noisy and reverberant fields. Source tracking refers to the localization of moving sources. A source tracking technique that is robust to acoustically adverse conditions would be most desirable in real-life applications, which motivates a dynamic multisource tracking technique with dereverberation capability in this study.

The associate editor coordinating the review of this manuscript and approving it for publication was Hasan S. Mir.

Source localization methods can be categorized as direct and indirect methods [8]. Direct methods such as delay and sum (DAS) [9], minimum power distortionless response (MPDR) [10], and multiple signal classification (MUSIC) [11] can be used to estimate the DOA directly from the microphone signals, whereas indirect methods can be used to estimate direction-bearing features such as the time difference of arrival (TDOA) [12] between microphone pairs in the first stage, and estimate the DOAs of sources in the second stage [13]. Recently, orthogonal matching pursuit (OMP)-based approaches have been proposed to estimate the M DOAs with high resolution [14], [15], where M is the number of microphone sensors. In [15], a novel scheme was proposed to tackle phase ambiguity and distinguish adjacent signals using a nonuniform array based on the OMP algorithm and DAS analysis. The direct method, the MUSIC algorithm, is employed in this paper because of its superior localization resolution. Similar to all direct localization methods, MUSIC

requires a grid search to find local maxima in its pseudospectrum, which can be very time-consuming for a uniform search in space.

In addition to computational complexity, the tracking accuracy can be severely compromised by reverberation. The MUSIC algorithm is susceptible to reverberation, which is partially correlated with the source signal [16]. Hence, conventional localization methods based on free-field source models are deemed inappropriate for reverberant fields. Techniques have been suggested [17], [18], [19], [20], [21], [22] to alleviate the detrimental effects of reverberation in the context of array beamforming. A covariance matrix based on a diffuse isotropic field model can be assumed for the application of multichannel Wiener filters [23], [24], [25], [26]. Alternatively, the multiple-input/output inverse-filtering theorem (MINT) algorithm [27] can be utilized for dereverberation. However, as a limitation of this technique, the room impulse responses between the sources and sensors must be available *a priori* and accurately identified, which is generally difficult in practice. To counter this, the weighted-prediction-error (WPE) [28], [29], [30], [31] algorithm based on an autoregressive model to estimate late reverberation was proposed. The power spectral density of the anechoic signal is estimated in the WPE such that the late reverberation can be reduced without prior knowledge of the source's direction and room impulse responses. In this paper, a dynamic source tracker inspired by an autoregression (AR) array and minimum variance distortionless response (MVDR) beamforming is formulated for online dereverberation. Thus, the dereverberated microphone signals serve to calculate the MUSIC pseudospectrum for the tracking task.

In source localization, a peak-finding procedure is required to locate the direction of the sound sources from the MUSIC pseudospectrum. Instead of a uniform grid search, which is prohibitively expensive to compute on-the-fly for most applications, a dynamic search procedure based on an evolutionary algorithm (EA) [32] is adopted in this work. Genetic algorithms (GA) [33], evolution strategies (ES) [34], particle swarm optimization (PSO) [35], differential evolution (DE), cuckoo search (CS) [36], [37], and harmony search (HS) [37] are EA-based algorithms that are well suited for nonconvex problems with multiple local optima [38]. EA emulates the process of natural selection, where fitter individuals have a higher chance of reproducing their offspring and surviving in the next generation. For DOA estimation of multiple sources, ''niching'' techniques such as crowding [39], speciation [40], and sequential niche [41] can be employed to address multiobjective optimization problems. Speciation helps accelerate convergence when allocating each particle to a specific species. The number of particles in each species must be restricted to maintain species diversity. Extraneous particles with poor performance for each species are reinitialized at a random location in the search space [42].

Many techniques can be employed to assist the evolution process. Evolutionary state estimation (ESE) [43] divides an evolutionary state into four stages: convergence, exploitation, exploration, and jumping-out. Particles adaptively adjust their evolution trajectories according to the swarm distribution. However, if a sound source abruptly moves away from its stationary state, it is difficult for the particle swarms to escape from the convergence state. To address this problem, the quantum swarm model (QSM) [44] divides all particles into two groups termed ''neutral particles'' and ''quantum particles'' can be considered. Neutral particles follow the update rule of the conventional PSO, whereas quantum particles are positioned as a ''cloud'' centered at the ''species seed'' [40], which is meant for monitoring the peak change. Although the QSM can track the sudden and large movements of the sources, updating quantum particles is computationally expensive. To remedy this, an elitist learning strategy (ELS) [43] can be used. ELS randomly mutates one dimension of the position of the ''global-best'' particle to detect peak changes. Once the fitness value of the mutated particle exceeds the original value, the solution is replaced with the mutated particle. The inertia weights [45], [46], [47], [48] and acceleration coefficients [49], [50], [51], [52] of PSO can be applied to improve the performance of PSO algorithms in this paper. The update rule of particle velocity is modified in each iteration to find new directions for sound sources, where a species-based concept is adopted to divide the particles into multiple species and guide the particles toward the centers of their species. In addition to the attraction force from the species centers, the movement of the particles is also affected by the progressively decreasing inertia weight. The acceleration coefficients are dynamically adapted for sources with significant movements. In this paper, an adaptive species-based particle swarm optimization (ASPSO) algorithm with AR is employed for the dynamic tracking of multiple moving sources under reverberation.

A six-microphone circular array is used for simulations and experiments to assess the tracking performance of the ASPSO-AR algorithm in a reverberant environment. The root-mean-square error (RMSE) is adopted to compare the performance of the proposed system with that of the other baseline methods. The processing times required by the PSO-based algorithms and the grid search are also compared. Numerical simulations are conducted to evaluate the AR algorithm in the context of source localization.

The remainder of this paper is organized as follows. Section II presents the MUSIC-based localization method and autoregression method for dereverberation. In Section III, PSO-based algorithms are introduced. Section IV presents simulations and experiments to validate the proposed techniques, and Section V concludes the paper.

II. SOURCE LOCALIZATION IN REVERBERANT FIELDS

A. ARRAY SIGNAL MODEL AND THE MUSIC ALGORITHM

The MUSIC algorithm is designed on the basis of the orthogonality of the ''signal subspace'' and the ''noise subspace'' derived from the data covariance matrix. Considering D independent source signals impinging on an M – element array,

an array signal model can be established in the short-time Fourier transform (STFT) domain, as follows:

$$\mathbf{y}(l,n) = \mathbf{A}(n)\mathbf{s}(l,n) + \mathbf{r}(l,n) + \mathbf{u}(l,n), \tag{1}$$

where $\mathbf{s}(l,n) \in \mathbb{C}^D$ denotes the STFT of the source signal vector at time frame l and frequency bin n. The reverberation vector $\mathbf{r}(l,n) \in \mathbb{C}^M$ is assumed to be partly correlated with the source signals. The additive noise vector $\mathbf{u}(l,n) \in \mathbb{C}^M$ is assumed to be uncorrelated with the source signals. The steering matrix is denoted as $\mathbf{A}(n) = \left[\mathbf{a}(n,\theta_1)\ \ \mathbf{a}(n,\theta_2)\ \ \cdots\ \mathbf{a}(n,\theta_D)\right] \in \mathbb{C}^{M\times D}$. In the free-field plane-wave model, the steering vector, $\mathbf{a}(n,\theta_d)$, $d = 1, 2, \cdots D$, can be expressed as

$$\mathbf{a}(n,\theta_d) = \left[\exp(-j\mathbf{k}_d\cdot\mathbf{p}_1) \quad \cdots \quad \exp(-j\mathbf{k}_d\cdot\mathbf{p}_M)\right]^T, \tag{2}$$

where $\mathbf{p}_m$, $m = 1, \cdots, M$ is the position vector of the mth microphone, $\mathbf{k}_d = [2\pi n f_s/(N_{FFT}c)]\,\boldsymbol{\kappa}_d$ is the wave vector, f_s denotes the sampling frequency in hertz, N_{FFT} is the fast Fourier transform (FFT) size, and c is the speed of sound. Unit vector $\boldsymbol{\kappa}_d$ signifies the DOA of the dth source. In the MUSIC algorithm, the following pseudospectrum is calculated:

$$S_{MUSIC}(n,\theta) = \frac{1}{\mathbf{a}^H(n,\theta)\mathbf{P}_u(n)\mathbf{a}(n,\theta)}, \tag{3}$$

where $\mathbf{P}_u(n) = \mathbf{U}(n)\mathbf{U}^H(n)$ is the projection matrix of the noise subspace [54]. A peak appears in $S_{MUSIC}(n,\theta)$ if the steering vector points toward the right source direction.

The MUSIC algorithm in (3) is based on the free-field model, and its localization accuracy generally deteriorates in the presence of reverberation. This problem requires dereverberation preprocessing, as detailed below.

B. AUTOREGRESSION ARRAY INSPIRED ONLINE DEREVERBERATION

To counter the degradation of localization performance due to late reverberation, an online dereverberation preprocessor is formulated on the basis of the AR model and MVDR beamformer. The late reverberation signal $\hat{X}_{late}(l,n)$ in the T-F domain can be estimated using linear prediction.

$$\hat{X}_{late}(l,n) = -\tilde{\mathbf{w}}^H\tilde{\mathbf{y}}_{l-\Delta,n}, \tag{4}$$

where "H" denotes the conjugate-transpose operator. $\tilde{\mathbf{w}} = \left[w_1\ \ w_2\ \ \cdots\ w_{K_T}\right]^T \in \mathbb{C}^{K_T\times 1}$ is a complex weight vector. $K_T = MK$ represents the total frame number of the late reverberation, K is the frame number of the late reverberation in each channel, and Δ denotes the number of frames that cover the early reflections. The vector, $\tilde{\mathbf{y}}_{l-\Delta,n} = \left[\bar{\mathbf{y}}_1^T\ \ \bar{\mathbf{y}}_2^T\ \ \cdots\ \bar{\mathbf{y}}_M^T\right]^T \in \mathbb{C}^{MK\times 1}$, accounts for late reflections $\bar{\mathbf{y}}_m = \left[Y_m(l-\Delta,n)\ \ \cdots\ Y_m(l-\Delta-K+1,n)\right]^T$. In this linear prediction model, w_k is the kth complex-valued weighting coefficient to be determined using an MVDR criterion. Using (4), the mth channel linear prediction error (the dereverberated signal) can be expressed as

$$e_m(l,n) = Y_m(l,n) - \hat{X}_{late}(l,n) = \mathbf{w}_m^H\mathbf{y}_m, \tag{5}$$

TABLE 1. The AR-based online dereverberation algorithm.

For $m = 1:M$
  For $n = 1:N_T$
    Initialization: $\mathbf{R}_m^{-1}(0,n) = \mu_0\mathbf{I}$
    For $l = 1:L_T$
      $\hat{\lambda}(l,n) = \beta\hat{\lambda}(l-1,n) + (1-\beta)\sum_{m=1}^{M}\left|Y_m(l,n)\right|^2/M$
      $\mathbf{k}(l,n) = \dfrac{\mathbf{R}_m^{-1}(l-1,n)\mathbf{y}_m}{\alpha\hat{\lambda}(l,n) + \mathbf{y}_m^H\mathbf{R}_m^{-1}(l-1,n)\mathbf{y}_m} \in \mathbb{C}^{(MK+1)\times 1}$
      $\mathbf{R}_m^{-1}(l,n) = \dfrac{\mathbf{R}_m^{-1}(l-1,n) - \mathbf{k}(l,n)\mathbf{y}_m^H\mathbf{R}_m^{-1}(l-1,n)}{\alpha} \in \mathbb{C}^{(MK+1)\times(MK+1)}$
      Using (7) to calculate $\mathbf{w}_m$
      Using (5) to get the dereverberated signal, $e_m(l,n) = \mathbf{w}_m^H\mathbf{y}_m$
    End for
  End for
End for

where $\mathbf{w}_m = \left[w_0\ \ w_1\ \ \cdots\ w_{K_T}\right]^T \in \mathbb{C}^{(K_T+1)\times 1}$ is used to estimate the early and direct signals. $\mathbf{y}_m = \left[Y_m(l,n)\ \ \bar{\mathbf{y}}_1^T\ \ \bar{\mathbf{y}}_2^T\ \ \cdots\ \bar{\mathbf{y}}_M^T\right]^T$. $w_0 = \mathbf{w}_m^H\mathbf{u}_i = 1$ with $[\,\mathbf{u}_i]_j = \delta_{ij}$, $i = 1, j = 1, 2, \ldots, (K_T+1)$ is a one-hot vector. The weighting coefficients can be obtained by minimizing $E\left[e_m^2(l,n)\right]$, which can be posed as the MVDR beamformer design problem:

$$\arg\min_{\mathbf{w}_m}\mathbf{w}_m^H\mathbf{R}_m(l,n)\ \ \mathbf{w}_m \quad \text{st. } \mathbf{w}_m^H\mathbf{u}_i = 1, \tag{6}$$

where $\mathbf{R}_m(l,n) = E\left[\mathbf{y}_m\mathbf{y}_m^H\right] = \mathbf{R}_m^H(l,n) \in \mathbb{C}^{(MK+1)\times(MK+1)}$ denotes the covariance matrix. The solution to the above-constrained optimization problem is [56]

$$\mathbf{w}_m = \lambda\mathbf{R}_m^{-1}(l,n)\mathbf{u}_i = \frac{\mathbf{R}_m^{-1}(l,n)\mathbf{u}_i}{\mathbf{u}_i^H\mathbf{R}_m^{-1}(l,n)\mathbf{u}_i} = \frac{\mathbf{v}_i}{\mathbf{u}_i^H\mathbf{v}_i}, \tag{7}$$

where $\mathbf{R}_m^{-1}(l,n) = \left[\mathbf{v}_1\ \ \mathbf{v}_2\ \ \cdots\ \mathbf{v}_{(MK+1)}\right]$. Using this in (5) yields the dereverberated signal, $e_m(l,n) = \mathbf{w}_m^H\mathbf{y}_m$. The complete procedure of the preceding multichannel AR-based online dereverberation approach is summarized in Table 1.

In Table 1, α and β are forgetting factors for the adaptive recursive least squares (RLS) filter [31]. N_T is the total number of frequency bins. L_T is the total number of time frames. The recursive relation of the inverse matrix $\mathbf{R}_m^{-1}(l,n)$ is derived using the Sherman-Morrison formula [53], which is also known as the Woodbury matrix identity or matrix inverse lemma, as detailed in Appendix A. The dereverberated signals serve as input data for the subsequent MUSIC localization unit.

III. PARTICLE SWARM OPTIMIZATION-BASED MULTI-SOURCE TRACKING

A. PARTICLE SWARM OPTIMIZATION

MUSIC localization is computationally expensive to compute in real time for moving-source tracking if a uniform grid

search is performed in the space. To address this problem, we exploit particle swarm optimization (PSO) [39] to accelerate the tracking process. The general update rule of PSO can be written as

$$x_\delta(l,t) = x_\delta(l,t-1) + v_\delta(l,t-1), \tag{8}$$

where $x_\delta(l,t)$ contains the information of the angle θ in (3). δ is the particle index. t is the iteration index. The term $v_\delta(l,t)$ is referred to as "velocity" in PSO terminology and is updated according to

$$\begin{aligned} v_\delta(l,t) = {} & w v_\delta(l,t-1) + b_1 c_1 \left[p_\delta(l,t) - x_\delta(l,t)\right] \\ & + b_2 c_2 \left[g(l,t) - x_\delta(l,t)\right], \end{aligned} \tag{9}$$

where $p_\delta(l,t)$ represents the "personal-best" and $g(l,t)$ denotes the "global-best" for all particles in the search space. In addition, w denotes the inertia weight, c_1 and c_2 denote the acceleration coefficients, and $b_1, b_2 \in [0, 1]$ denote random numbers drawn from the uniform distribution. The particle swarm continues to move at each iteration until the following stopping criterion is satisfied:

$$\left|f(g(l,t)) - f(g(l,t-1))\right| / f(g(l,t)) < \varepsilon_0, \tag{10}$$

where ε_0 presets a threshold. The following frequency-averaged MUSIC spectrum is adopted as the fitness function:

$$f(\theta) = \sum_{n=n_{start}}^{n_{stop}} S_{MUSIC}(n,\theta)/N_{freq}, \tag{11}$$

where $N_{freq} = n_{stop} - n_{start} + 1$ is the frequency range from the frequency bin n_{start} to n_{stop}.$p_\delta(l,t)$ and $g(l,t)$ in (9) can be determined from the maximal fitness function in (11). Using the PSO peak-finding update rule, the direction of a single source can be determined from the converged particles. The PSO algorithm is summarized in Table 2.

In Table 2, N_{iter} denotes the total number of iterations, and N_{all} denotes the total number of particles. To track multiple sources, the PSO algorithm must be modified using the species seed technique [40]. The modified approach is referred to as PSO-S in this paper, as detailed below.

B. PARTICLE SWARM OPTIMIZATION WITH SPECIES SEED

Instead of using the global-best position, PSO-S utilizes species seed particles in its neighborhood. The species seed approach facilitates PSO for tracking multiple sound sources. In addition, the species seed technique can accelerate processing when searching for sound sources. A random number is generated as the inertia weight to prevent particles from being trapped in the local-suboptimal position. The velocity update equation of the δth particle in the PSO-S algorithm is modified as follows:

$$\begin{aligned} v_\delta(l,t) = {} & b_3 w v_\delta(l,t-1) + b_1 c_1 \left[p_\delta(l,t) - x_\delta(l,t)\right] \\ & + b_2 c_2 \left[s_\delta(l,t) - x_\delta(l,t)\right], \end{aligned} \tag{12}$$

where b_3 is a random number uniformly distributed between 0 and 1. $s_\delta(l,t)$ denotes the species seed inside a circle with radius σ_s and is centered at the δth particle in the search space. The complete procedure for determining seed species is summarized in Table 3.

TABLE 2. The PSO algorithm.

Initialize current position $x_\delta(l,0)$ and personal-best position $p_\delta(l,0)$
$x_\delta(l,0)$ and $p_\delta(l,0)$ are uniformly distributed in [0°, 360°]
$v_\delta(l,0) = 0$, and $w = 1$, $c_1 = 2$, $c_2 = 2$.
For $l = 1 : L_T$
  For $t = 1 : N_{iter}$
    For $\delta = 1 : N_{all}$
      Update the position of particle
      $x_\delta(l,t) = x_\delta(l,t-1) + v_\delta(l,t-1)$
      Update the personal-best position
      If $f(x_\delta(l,t)) > f(p_\delta(l,t-1))$
        $p_\delta(l,t) = x_\delta(l,t)$
      End if
    End for
    Find the global-best position
    $g(l,t) = \arg\max_{p_\delta(l,t)} \left[f(p_\delta(l,t))\right]$
    Update the velocity $v_\delta(l,t)$ of each particle using (9)
    Check the stopping criterion using (10)
    Exit the iteration if the stop criterion is met.
  End for
End for

In Table 3, s refers to the subset of S_{seed}. The species seed technique exploits the fitness value of the local-best particles. The PSO-S algorithm first searches for the local-best particles and then uses them as the species seed to find the source locations. The procedure for finding the local-best particles is known as "crowding" in PSO literature [55]. The directions of the sources can be estimated using the centroids of the clusters. The "crowding" procedure is summarized in Table 4.

By counting the number of each particle being selected as the species seed in a cluster, the particle positions with the top D highest counts are regarded as the best-estimated source directions in the current iteration. The stopping criterion of the PSO-S algorithm is

$$\sum_{\delta=1}^{D} \left|f(s_\delta(l,t)) - f(s_\delta(l,t-1))\right| / f(s_\delta(l,t)) < D\varepsilon_0. \tag{13}$$

C. MODIFIED PARTICLE SWARM OPTIMIZATION

The modified particle swarm optimization (MPSO) [57] algorithm utilizes a set of adaptive parameters as its inertia weight and acceleration coefficients in (12). That is,

$$w(t) = w(t-1)\eta, \quad c_1(t) = c_1(t-1)\eta, \; c_2(t) = c_2(t-1)\eta^{-1}, \tag{14}$$

where

$$\eta = \frac{N_{all}}{N_{all} + N_{cluster}} \in [0, 1], \tag{15}$$

TABLE 3. The species assignment algorithm [40].

L_{sorted} a list of particle position sorted by the fitness value of local-best in descending order
S_{seed} a list of particle position of species seeds, initial $S_{seed} = \varnothing$
s_δ the species seed position of particle δ
σ_s the radius of species seed
For $\delta = 1 : N_{all}$
  Get $x_\delta = \delta^{th}$ particle position in L_{sorted}
  $found\ flag$ = FALSE
  For $s \in S_{seed}$ do
    If Euclidean distance $(s, x_\delta) < \sigma_s$
      $s_\delta = s$
      $found\ flag$ = TRUE
      break
    End if
  End for
  If $found\ flag$ = FALSE
    $S_{seed} = S_{seed} \cup x_\delta$
    $s_\delta = x_\delta$
  End if
End for

where $N_{cluster}$ denotes the number of particles in each cluster. The MPSO algorithm helps particles efficiently search the neighborhood for a suboptimal solution. The stopping criterion of the MPSO algorithm is given by (13).

D. ADAPTIVE SPECIES-BASED PARTICLE SWARM OPTIMIZATION

The adaptive species-based particle swarm optimization (ASPSO) algorithm utilizes a progressive inertia weight $[w_p(t)]$ and two adaptive acceleration coefficients $[c_{1,a}(t), c_{2,a}(t)]$ [43] in (12).

$$w_p(t) = a^{t-1}, \tag{16}$$

where a is a numerical value between zero and one. $c_{1,a}(t)$ and $c_{2,a}(t)$ are updated on the basis of exploration, exploitation, convergence, and jumping-out states [43].

$$c_{1,a}(t), c_{2,a}(t) = \begin{cases} c_{1,a}(t-1) + \mu, c_{2,a}(t-1) - \mu \\ \quad \text{for exploration state} \\ c_{1,a}(t-1) + 0.5\mu, c_{2,a}(t-1) - 0.5\mu \\ \quad \text{for exploitation state} \\ c_{1,a}(t-1) + 0.5\mu, c_{2,a}(t-1) + 0.5\mu \\ \quad \text{for convergence state} \\ c_{1,a}(t-1) - \mu, c_{2,a}(t-1) + \mu \\ \quad \text{for jumping-out state,} \end{cases} \tag{17}$$

where $u \in [0.05, 0.1]$ is a random value with a uniform distribution. The four evolutionary states can be determined by an evolutionary factor (f_e), which is an ESE technique.

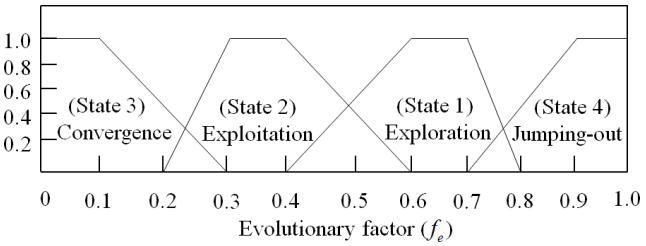


FIGURE 1. The relation between the evolutionary factor and four evolutionary states (convergence, exploitation, exploration, and jumping-out) [43].

TABLE 4. The pseudo-code of the crowding algorithm.

Initialization: $l_\delta(l,t) = p_\delta(l,t)$
For $\delta = 1 : N_{all}$ and $\delta' = 1 : N_{all}$, $\delta' \neq \delta$
  If $\left|p_\delta(l,t) - p_{\delta'}(l,t)\right| < \sigma_{local}$ and $f(l_\delta(l,t)) < f(p_{\delta'}(l,t))$
    $l_\delta(l,t) = p_{\delta'}(l,t)$
  End if
End for

Figure 1 indicates that the evolutionary factor (f_e) can determine the current evolutionary state by which the acceleration coefficients can be obtained. The evolutionary factor (f_e) is calculated as follows:

$$f_e = \frac{d_s - d_{\min}}{d_{\max} - d_{\min}} \in [0, 1], \tag{18}$$

where $d_i = \frac{1}{N_s - 1} \sum_{j=1, j \neq i}^{N_s} \left|x_i(l,t) - x_j(l,t)\right|$ is the average distance between the $i$thparticle and the other $N_s - 1$ particles. N_s represents the number of particles in each species. d_s denotes the average distance when the $i$thparticle is a species seed. $d_{\max}$ denotes the maximum average distance, and $d_{\min}$ represents the minimum average distance. In addition, the ELS [43] technique is applied to the convergence state to search for potential local maxima.

$$s'_\delta(l,t) = s_\delta(l,t) + \sigma_{ELS} e, \tag{19}$$

where $e \sim N(0, 1)$ is a Gaussian random variable and σ_{ELS} controls the search range of the ELS. The stopping criterion of the ASPSO algorithm is given in (13). The ASPSO algorithm is summarized in Table 5.

Standard PSO can locate only a single source, whereas PSO-S can locate multiple sources. A set of modified inertia weights $[w(t)]$ and two acceleration coefficients $[c_1(t), c_2(t)]$ are employed in the MPSO algorithm, which makes it well-suited for tracking fast-moving sources. ASPSO relies on progressively decreasing weights, $[w_p(t)]$, to guide particles toward the directions of sound sources and two adaptive coefficients $[c_{1,a}(t), c_{2,a}(t)]$ to better track the trajectory of moving sources with large movements. All three PSO-based algorithms exploit the species tracking procedure for multiple-source tracking. Despite its complexity, the

TABLE 5. The ASPSO algorithm.

Initialize current position $x_\delta(l,0)$ and personal-best position $p_\delta(l,0)$
$x_\delta(l,0)$ and $p_\delta(l,0)$ are uniformly distributed in [0°, 360°]
$v_\delta(l,0)=0,$ and $c_{1,a}(l,0)=2,\ c_{2,a}(l,0)=2.$
For $l=1:L_T$
  For $t=1:N_{iter}$
    For $\delta=1:N_{all}$
      Update the position of particle
      $x_\delta(l,t)=x_\delta(l,t-1)+v_\delta(l,t-1)$
      Update the personal-best position
      If $f(x_\delta(l,t))>f(p_\delta(l,t-1))$
        $p_\delta(l,t)=x_\delta(l,t)$
      End if
    End for
    Find the local-best position using crowding procedure
    in TABLE IV
    Find the species seed position using species procedure
    in TABLE III
    Calculate the inertia weight and acceleration coefficients
    using (16)-(19)
    Update the velocity $v_\delta(l,t)$ of each particle by using (12)
    in terms of $w_p(t)$, $c_{1,a}(t),$ and $c_{2,a}(t)$
    Check the stopping criterion.
    Exit the iteration if the stop criterion is met.
  End for
End for

proposed ASPSO algorithm exhibits robustness in dynamic source tracking, as shown by the simulation and experimental results.

IV. SIMULATIONS AND EXPERIMENTS

A. PARAMETER SETTINGS

Male speech signals [58] and female speech signals [59] are used as source signals in the simulation. The reverberant signals for different angles are produced by convolving the clean signal with the room impulse response generated using the image source method [60], [61]. The parameter settings for (10), (11), (16), (19), (20) and Table 1 are listed in Table 6. d_{source} refers to the distance between the sound sources and the array center. θ_{GS} denotes the angular resolution of the grid search (GS) method. ε_λ represents the threshold for estimating the number of sound sources, as described in detail below:

B. SOURCE COUNTING ESTIMATE

In this paper, the number of sound sources for each time frame is estimated using eigenvalue decomposition as follows:

$$\hat{p}(l)=\left\{i-1 \mid i\in 2,3,\ldots M, \bar{\lambda}_i(l)<\varepsilon_\lambda\bar{\lambda}_1(l)\right\}, \quad (20)$$

where $\hat{p}(l)$ is the estimated number of sound sources for each time frame. $\bar{\lambda}_m(l)=\frac{1}{N_{freq}}\sum_{n=n_{start}}^{n_{stop}}|\lambda_m(l,n)|$ is the average eigenvalue in descending order for each time frame, where $m=1,2,\ldots M$. The parameter ε_λ is used to estimate the number of sources. Using (20), $\hat{p}(l)$ can be determined by $i-1$when $\bar{\lambda}_i(l)<\varepsilon_\lambda\bar{\lambda}_1(l)$. This paper uses a UCA to estimate the number of sources. Hence, the maximum number of sources detected is $M-1$, and if the coarray MUSIC [62], [63] is used, this work can detect more than M sources.

TABLE 6. The parameter settings.

f_s (kHz)	N_{FFT}	Overlap (%)	Window
16	512	75	Hanning
n_{start} (Hz)	n_{stop} (Hz)	σ_{local} (deg)	σ_s (deg)
585	3500	30	30
σ_{ELS} (deg)	ε_0	a	N_{all}
7	0.01	0.9	20
N_{iter}	α	β	μ_0
40	0.9999	0.5	0.2256
d_{source} (m)	θ_{GS} (deg)	ε_λ	
1	1	7.57×10^{-3}	

C. SIMULATIONS

A rectangular room of dimensions, L4.6 m × W5.0 m × H2.6 m, is assumed for the simulation. The signal-to-noise ratio (SNR) is set to 40 dB. For simplicity, this study only considers azimuthal angles (the signal sources and microphone sensors are in the same plane). Three simulation cases are employed to validate the proposed dynamic source-tracking algorithm with an AR dereverberation preprocessor. Case A aims to evaluate the proposed localization algorithm for two sources with various sets of reverberation times (T_{60}). Case B evaluates the different angular spacings that the proposed localization algorithm can distinguish. Case C is intended to assess the tracking ability of the proposed PSO-based source tracking algorithms in a reverberant field.

1) CASE A

The simulation settings for Case A are shown in Fig. 2.

Figure 2 depicts the simulation setting of Case A for a 6-microphone uniform circular array (UCA) (diameter,$d_a=7$ cm). The two sources are 1 m from the center of the UCA. Nine combinations of the two source positions are selected for Case A. The angular spacing $\Delta\theta$ is 40°, with reverberation times of $T_{60}=0.25$ s, 0.40 s, and 0.60 s. The root-mean-square error for Case A is calculated as

$$\varepsilon_{rms,A}=\sqrt{\frac{1}{DQ}\sum_{d=1}^{D}\sum_{q=1}^{Q}\left[\hat{\theta}_d^q-\theta_d^q\right]^2}, \quad (21)$$

where the number of evaluated cases is $Q=27$.

Three different reverberation times are selected. For each reverberation time, nine combinations of the source positions

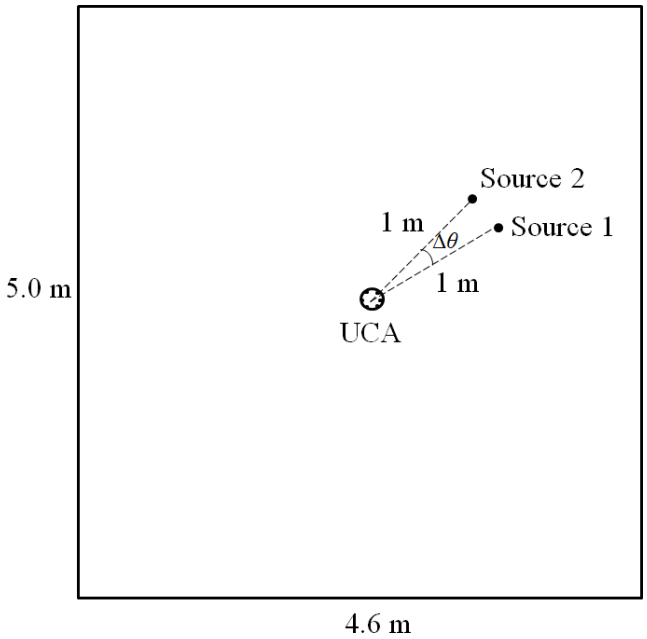


FIGURE 2. The simulation settings for two sources in Case A. The diameter of the circular array is 7 cm. The distance between the sound sources and the center of the UCA is 1 m. The simulated room is L4.6 m × W5.0 m × H2.6 m. T_{60} = 0.25 s, 0.40 s, and 0.60 s. SNR = 40 dB.

are examined. The localization results of Case A obtained using the grid search (GS) with and without AR dereverberation are shown in Fig. 3. Figure 3 shows that room reverberation has a detrimental effect on the two-source localization. Without AR preprocessing, the RMSE obtained using the GS method is 14.54°, whereas the RMSE drops sharply to 4.74° with GS-AR preprocessing. Figure 3 shows that AR dereverberation can reduce the localization error of multiple sources under various reverberation conditions. Next, the GS and ASPSO-AR methods for different angular spacings are presented in Case B.

2) CASE B

In Case B, $\Delta\theta$ for two sources is set to 30°, 20°, and 10° to evaluate the performance of the speech source localization under T_{60} = 0.25 s and SNR = 40 dB. The GS and ASPSO-AR methods are used. In this case, σ_{local} and σ_s are set to 7°. N_{all} is set to 70. The results are depicted in Fig. 4. The GS and ASPSO-AR methods can localize speech sound sources when $\Delta\theta$ = 30°. However, the performance of the GS method decays when $\Delta\theta$ = 20° due to reverberation. The ASPSO-AR method can detect two speech sources when $\Delta\theta$ = 10°. Hence, the proposed ASPSO-AR algorithm can resist reverberation and distinguish adjacent speech sound sources compared to the GS method. Table 7 displays the source localization results. Next, the tracking performance by using the PSO-based algorithms will be presented in Case C.

3) CASE C

The simulation settings for Case C are shown in Fig. 5.

TABLE 7. The source localization results of case B.

	$\Delta\theta = 30^\circ$	$\Delta\theta = 20^\circ$	$\Delta\theta = 10^\circ$
GS	(44.00°, 69.00°)	(54.00°, 57.00°)	(45.00°, 49.00°)
ASPSO-AR	(41.53°, 69.32°)	(41.76°, 57.39°)	(41.29°, 48.81°)
Ground truth	(40.00°, 70.00°)	(40.00°, 60.00°)	(40.00°, 50.00°)

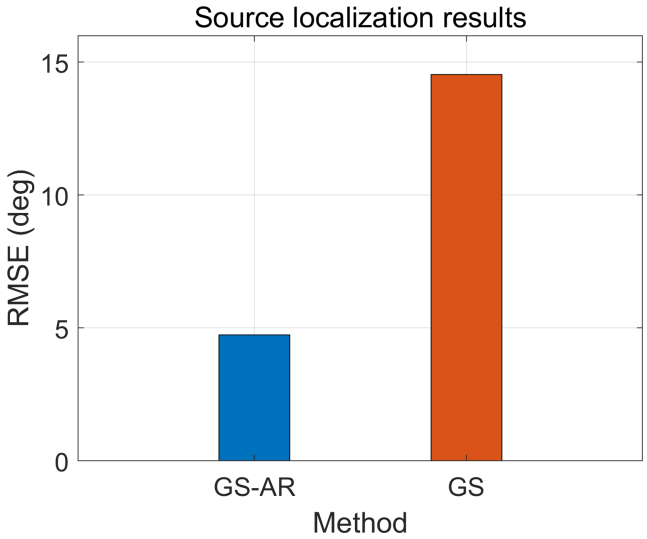


FIGURE 3. The localization results of RMSE with the GS-AR and GS methods in Case A under SNR = 40 dB and various T_{60}.

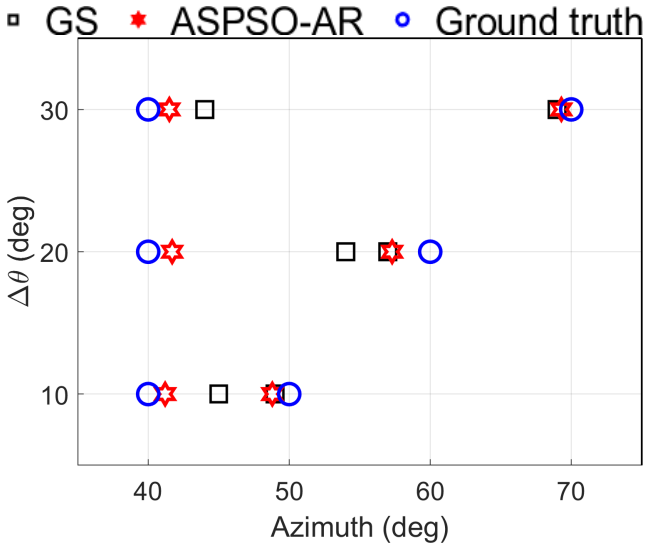


FIGURE 4. The source localization results for $\Delta\theta$ = 30°, 20°, and 10° using the GS and ASPSO-AR methods under T_{60} = 0.25 s and SNR = 40 dB.

In Case C, the ground-truth source angle is scheduled in five stages. In 0-5 s, the first and second sources are stationary at 40° and 130°, respectively. At Time = 5 s, the first source starts to rotate counterclockwise with a constant angular speed (0.1 rad/s) from 40° to 130° and stops at Time = 20 s. Similarly, at Time = 5 s, the second sound

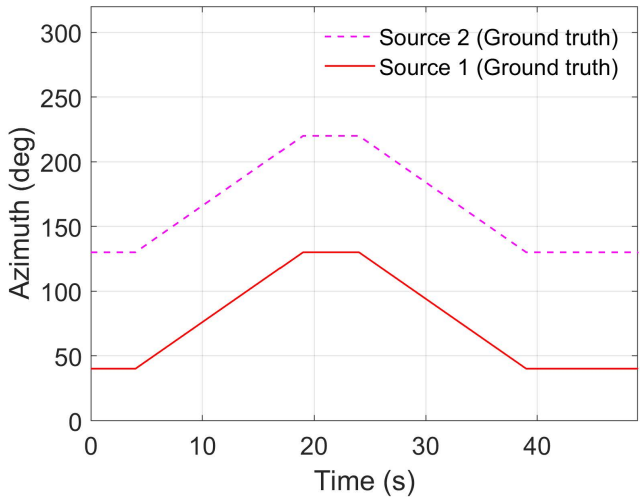


FIGURE 5. The azimuth trajectory of two moving sources in the simulation for Case C.

source starts to rotate counterclockwise at a constant angular speed (0.1 rad/s) from 130° to 220° and stops at Time = 20 s. Between Time = 20-25 s, both sources remain at rest. A 6-microphone UCA with a diameter $d_a = 7$ cm, positioned at the center of a room with $T_{60} = 0.25$ s and SNR = 40 dB, is assumed in this simulation case. To quantify the tracking performance of the proposed PSO algorithms, the root-mean-square error (RMSE) for Case C is computed as follows:

$$\varepsilon_{rms,C} = \sqrt{\frac{1}{DL}\sum_{d=1}^{D}\sum_{l=1}^{L_T}\left[\hat{\theta}_d(l) - \theta_d(l)\right]^2}, \qquad (22)$$

where $\hat{\theta}_d(l)$ and $\theta_d(l)$ denote the estimated and ground-truth azimuthal angles, respectively. The source-tracking results for Case C are shown in Figs. 6(a)-(c).

In this simulation, four tracking algorithms with AR dereverberation (GS-AR, PSO-S-AR, MPSO-AR, and ASPSO-AR) and two algorithms without AR dereverberation (ASPSO and GS) are compared. The trajectories of the GS-based and PSO-based algorithms are shown in Fig. 6(a). Only the tracking results of the ASPSO and ASPSO-AR algorithms are presented because PSO-S-AR, MPSO-AR, ASPSO, and ASPSO-AR perform similarly. Note that the RMSE in Fig. 6(b) for GS-AR is reduced by 15° in comparison with that of the GS method, which suggests that AR dereverberation is effective in improving the tracking performance. The performances of these six methods can be ranked as ASPSO-AR ($\varepsilon_{rms,C} = 2.81°$), MPSO-AR ($\varepsilon_{rms,C} = 3.07°$), PSO-S-AR ($\varepsilon_{rms,C} = 4.05°$), ASPSO ($\varepsilon_{rms,C} = 5.78°$), GS-AR ($\varepsilon_{rms,C} = 9.04°$), and GS ($\varepsilon_{rms,C} = 24.22°$). The results show that PSO-based algorithms with AR perform better than the GS-AR algorithm. The MPSO-AR and ASPSO-AR methods outperform the PSO-S-AR method because of the dynamic parameters (inertia weight and acceleration coefficients). In addition, the ASPSO-AR method benefits from the ESE procedure in adjusting the acceleration coefficients,

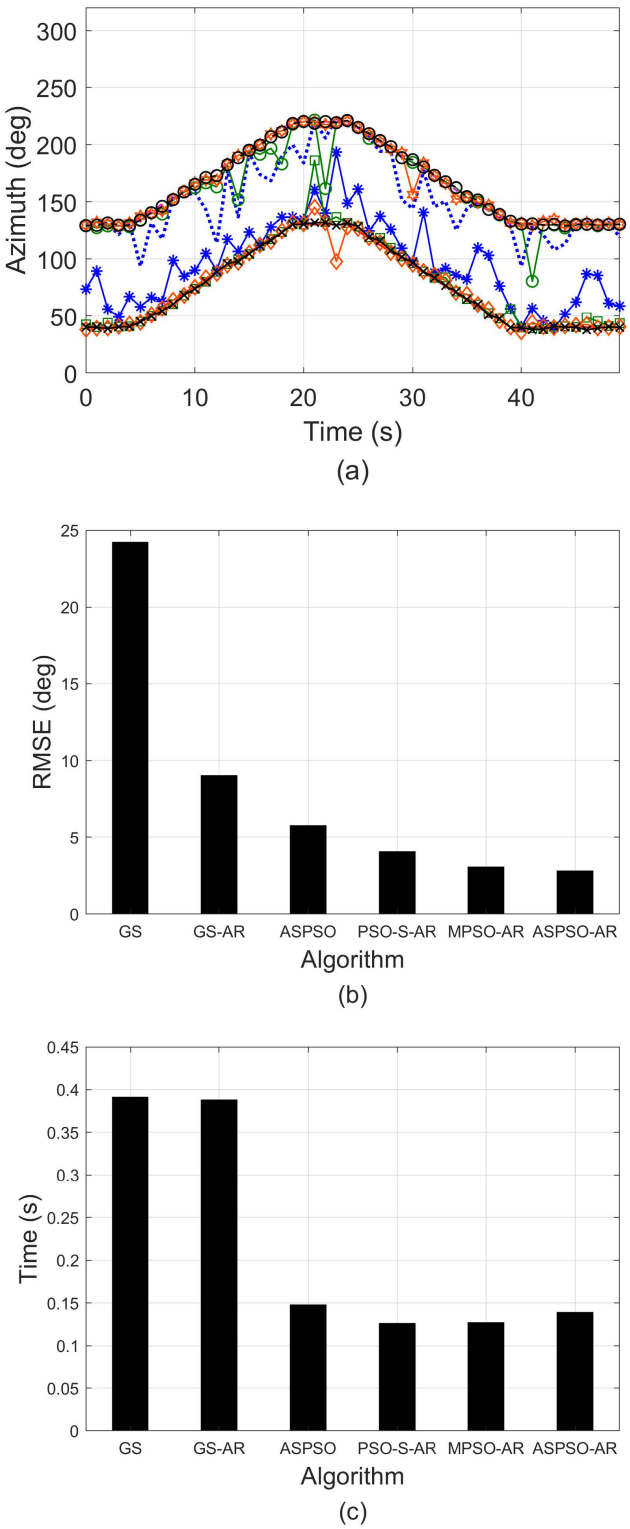


FIGURE 6. The source tracking results of Case C. (a) Source direction estimated, (b) RMSE, and (c) processing time. $T_{60} = 0.25$ s and SNR = 40 dB.

as well as the ELS technique. This explains the slightly better performance of the ASPSO-AR method than that of the MPSO-AR method. Speech pauses may cause errors in

source localization for each time frame. The ASPSO algorithm is robust to speech pauses because the species seed information can be utilized in the next time frame. The results of the GS and ASPSO methods are confirmed. The ASPSO-AR method performs better than the ASPSO method owing to the AR dereverberation. The processing time required for the MUSIC localization algorithm with and without AR dereverberation, based on 6247 time frames, is compared in Fig. 6(c). A laptop computer, Swift SFX14-41G, with a Core AMD Ryzen 7 5800U, was used for comparison. The package R2018b MATLAB 9.5®was used. It can be seen from the results that the PSO-based algorithms spend only one-third of the processing time of the GS-based algorithms, which lends themselves very well to real-time implementation. The processing times required by the six algorithms can be ranked as PSO-S-AR (0.126 s), MPSO-AR (0.127 s), ASPSO-AR (0.139 s), ASPSO (0.148 s), GS-AR (0.388 s), and GS (0.391 s). Owing to its simplicity, the PSO-S method performs a faster search than MPSO and ASPSO. Therefore, ASPSO requires more processing time than MPSO owing to the greater complexity of the former algorithm.

D. EXPERIMENTS

To further validate the proposed tracking algorithms, an experiment was conducted using the same settings as in Case C in the simulation. The MUSIC pseudospectrum is frequency-averaged in the 2500-3000 Hz band to obtain the localization result. However, instead of rotating the sources, an equivalent rotating-array scenario was created by placing the UCA on a turntable, as shown in Fig. 7.

Two fixed loudspeaker sources are placed at $\theta = 40°, 130°$, and both are 1 m away from the origin. A UCA with a diameter of 7 cm is mounted on the turntable and rotated in opposite directions to simulate moving source scenarios. For example, a 90-degree rotation clockwise of the microphone array on the turntable equals a 90-degree counterclockwise of the sound sources. The motion of the turntable follows the same five-stage angular displacement schedule for Case C in the simulation (Fig. 5). The experimental results are shown in Fig. 8. From Fig. 8, we see the degradation of the tracking performance if AR dereverberation is not used to preprocess the microphone signals. The RMSE of tracking is reduced by approximately 11° with the aid of AR dereverberation. The performances of the tracking approaches can be ranked as ASPSO-AR ($\varepsilon_{rms} = 9.55°$), MPSO-AR ($\varepsilon_{rms} = 9.69°$), PSO-S-AR ($\varepsilon_{rms} = 9.80°$), ASPSO ($\varepsilon_{rms} = 9.90°$), GS-AR ($\varepsilon_{rms} = 13.04°$), and GS ($\varepsilon_{rms} = 24.86°$). The PSO-based algorithms, combined with AR dereverberation, perform better in terms of RMSE than the GS-AR algorithm. The MPSO-AR method exploits the adaptive inertia weight and acceleration coefficients to update the velocity of the particles, which leads to a better performance of the MPSO-AR method than the PSO-S-AR method. The ASPSO-AR method outperforms PSO-S-AR and MPSO-AR, owing to the ESE and ELS. The ASPSO method performs better than the GS-based algorithms due to the species seed tracking procedure. The processing times required by the tracking algorithms can be ranked as PSO-S-AR (0.085 s), MPSO-AR (0.091 s), ASPSO (0.102 s), ASPSO-AR (0.103 s), GS-AR (0.150 s), and GS (0.153 s).

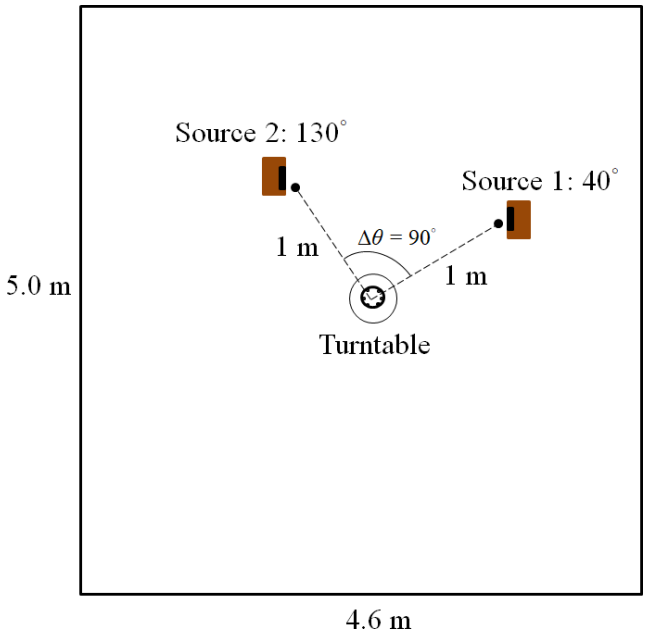


FIGURE 7. The experimental arrangement for the two-source tracking scenario in a L4.6 m × W5.0 m × H2.6 m listening room. $T_{60} = 0.25$ s and SNR = 38.03 dB.

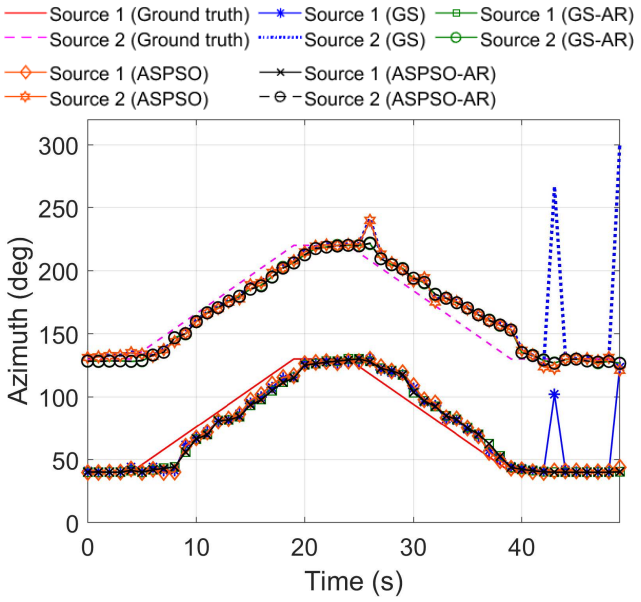


FIGURE 8. The experimental results of the source tracking trajectory using GS-based and PSO-based algorithms. $T_{60} = 0.25$ s and SNR = 38.03 dB.

V. CONCLUSION

PSO-based algorithms combined with the AR method for online dereverberation have been proposed to track dynamically moving sources under reverberant conditions. Several

mechanisms are incorporated into the PSO search process to improve tracking ability. Three PSO-based approaches equipped with AR dereverberation are compared to two benchmarking methods (GS and GS-AR). The simulation results show that the PSO-based algorithms can reduce the processing time to approximately one-third that of the GS algorithm. In addition, the AR method aimed at dereverberation proves effective in increasing the tracking performance. In comparison with the PSO-S and MPSO algorithms, the ASPSO algorithm exhibits superior tracking performance because of the ESE and ELS procedures. The MPSO method performs slightly better at tracking sources than the PSO-S method. While the ASPSO algorithm shows excellent tracking performance, it requires more processing time than the PSO-S and MPSO methods. The experimental results demonstrate the superior tracking performance of PSO-based algorithms with AR dereverberation in the presence of reverberation.

APPENDIX

The inverse matrix $\mathbf{R}_m^{-1}(l,n)$ in Table 1 can be derived from the Sherman-Morrison formula [53].

$$\left(\mathbf{A}+\mathbf{u}\mathbf{v}^H\right)^{-1}=\mathbf{A}^{-1}-\frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^H\mathbf{A}^{-1}}{1+\mathbf{v}^H\mathbf{A}^{-1}\ \mathbf{u}}, \tag{A1}$$

where $\mathbf{A}$ is assumed to be an invertible square matrix. $\mathbf{u}$ and $\mathbf{v}$ are the column vectors. If we set $\mathbf{A}$ to be $\alpha\mathbf{R}_m(l-1,n)$ and set $\mathbf{u}$ and $\mathbf{v}$ to be $\frac{\mathbf{y}_m}{\sqrt{\hat{\lambda}(l,n)}}$, (A1) can be expressed as

$$\begin{aligned}
&\left(\alpha\mathbf{R}_m(l-1,n)+\frac{\mathbf{y}_m}{\sqrt{\lambda(l,n)}}\frac{\mathbf{y}_m^H}{\sqrt{\lambda(l,n)}}\right)^{-1}\\
&=(\alpha\mathbf{R}_m(l-1,n))^{-1}\\
&\quad-\frac{(\alpha\mathbf{R}_m(l-1,n))^{-1}\mathbf{y}_m\mathbf{y}_m^H/\lambda(l,n)\,(\alpha\mathbf{R}_m(l-1,n))^{-1}}{1+\mathbf{y}_m^H\,(\alpha\mathbf{R}_m(l-1,n))^{-1}\mathbf{y}_m/\lambda(l,n)}\\
&=\frac{1}{\alpha}\left(\mathbf{R}_m^{-1}(l-1,n)-\frac{\mathbf{R}_m^{-1}(l-1,n)\mathbf{y}_m\mathbf{y}_m^H\mathbf{R}_m^{-1}(l-1,n)}{\alpha\lambda(l,n)+\mathbf{y}_m^H\mathbf{R}_m^{-1}(l-1,n)\mathbf{y}_m}\right)\\
&=\frac{\mathbf{R}_m^{-1}(l-1,n)-\mathbf{k}(l,n)\mathbf{y}_m^H\mathbf{R}_m^{-1}(l-1,n)}{\alpha},
\end{aligned} \tag{A2}$$

where $\mathbf{k}(l,n)$ is a $(MK+1)\times 1$ vector defined in Table 1.

ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers for their valuable comments and hard work that improved this paper. This research was made possible by Dr. Mingsian R. Bai's three-month visit to the LMS, FAU Erlangen-Nürnberg.

REFERENCES

[1] C. Rascon and I. Meza, "Localization of sound sources in robotics: A review," *Robot. Auton. Syst.*, vol. 96, pp. 184–210, Oct. 2017, doi: 10.1016/j.robot.2017.07.011.

[2] X. Bian, G. D. Abowd, and J. M. Rehg, "Using sound source localization in a home environment," in *Proc. Int. Conf. Pervasive Comput.*, Berlin, German, 2005, pp. 19–36, doi: 10.1007/11428572_2.

[3] P. K. B. Mugagga and S. Winberg, "Sound source localisation on Android smartphones: A first step to using smartphones as auditory sensors for training A.I systems with big data," in *Proc. AFRICON*, Addis Ababa, Ethiopia, Sep. 2015, pp. 1–5, doi: 10.1109/AFRCON.2015.7331970.

[4] N. Grbic and S. Nordholm, "Soft constrained subband beamforming for hands-free speech enhancement," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Orlando, FL, USA, May 2002, pp. 885–888, doi: 10.1109/ICASSP.2002.5743881.

[5] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 7, pp. 2011–2022, Sep. 2007, doi: 10.1109/TASL.2007.902460.

[6] L. Sun and Q. Cheng, "Real-time microphone array processing for sound source separation and localization," in *Proc. 47th Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2013, pp. 20–22, doi: 10.1109/CISS.2013.6552257.

[7] K. Farrell, R. J. Mammone, and J. L. Flanagan, "Beamforming microphone arrays for speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, San Francisco, CA, USA, Mar. 1992, pp. 285–288, doi: 10.1109/ICASSP.1992.225915.

[8] K. W. Cheung, H. C. So, W.-K. Ma, and Y. T. Chan, "A constrained least squares approach to mobile positioning: Algorithms and optimality," *EURASIP J. Appl. Signal Process.*, vol. 2006, no. 1, pp. 1–23, Jan. 2006, doi: 10.1155/ASP/2006/20858.

[9] A. L. L. Ramos, S. Holm, S. Gudvangen, and R. Otterlei, "Delay-and-sum beamforming for direction of arrival estimation applied to gunshot acoustics," *Proc. SPIE*, vol. 8019, Jun. 2011, Art. no. 80190U, doi: 10.1117/12.886833.

[10] S. A. Vorobyov, "Principles of minimum variance robust adaptive beamforming design," *Signal Process.*, vol. 93, no. 12, pp. 3264–3277, 2013, doi: 10.1016/j.sigpro.2012.10.021.

[11] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986, doi: 10.1109/TAP.1986.1143830.

[12] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Hong Kong, Apr. 2003, pp. 553–556, doi: 10.1109/ICASSP.2003.1201741.

[13] B. Kwon, Y. Park, and Y.-S. Park, "Analysis of the GCC-PHAT technique for multiple sources," in *Proc. ICCAS*, Gyeonggi-do, South Korea, Oct. 2010, pp. 2070–2073, doi: 10.1109/ICCAS.2010.5670137.

[14] M. Dehghani and K. Aghababaiyan, "FOMP algorithm for direction of arrival estimation," *Phys. Commun.*, vol. 26, pp. 170–174, Feb. 2018, doi: 10.1016/j.phycom.2017.12.012.

[15] K. Aghababaiyan, V. Shah-Mansouri, and B. Maham, "High-precision OMP-based direction of arrival estimation scheme for hybrid non-uniform array," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 354–357, Feb. 2020, doi: 10.1109/LCOMM.2019.2952595.

[16] A. Paulraj, V. U. Reddy, T. J. Shan, and T. Kailath, "Performance analysis of the music algorithm with spatial smoothing in the presence of coherent sources," in *Proc. IEEE Mil. Commun. Conf., Commun.-Comput., Teamed (MILCOM)*, Monterey, CA, USA, Oct. 1986, p. 41, doi: 10.1109/MILCOM.1986.4805849.

[17] C. Marro, Y. Mahieux, and K. U. Simmer, "Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 3, pp. 240–259, May 1998, doi: 10.1109/89.668818.

[18] S. M. Griebel, "A microphone array system for speech source localization, denoising, and dereverberation," Ph.D. dissertation, Dept. Eng. Appl. Sci., Harvard Univ., Cambridge, MA, USA, 2002.

[19] E. P. Habets, S. Gannot, and I. Cohen, "Dual-microphone speech dereverberation in a noisy environment," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, Vancouver, BC, Canada, Aug. 2006, pp. 651–655, doi: 10.1109/ISSPIT.2006.270881.

[20] E. A. R. Habets and S. Gannot, "Dual-microphone speech dereverberation using a reference signal," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Honolulu, HI, USA, Apr. 2007, pp. 901–904, doi: 10.1109/ICASSP.2007.367216.

[21] E. A. P. Habets and J. Benesty, "A two-stage beamforming approach for noise reduction and dereverberation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 5, pp. 945–958, May 2013, doi: 10.1109/TASL.2013.2239292.

[22] O. Schwartz, S. Gannot, and E. A. P. Habets, "Nested generalized sidelobe canceller for joint dereverberation and noise reduction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, South Brisbane, QLD, Australia, Apr. 2015, pp. 106–110, doi: 10.1109/ICASSP.2015.7177941.

[23] S. Braun, A. Kuklasinski, O. Schwartz, O. Thiergart, E. A. P. Habets, S. Gannot, S. Doclo, and J. Jensen, "Evaluation and comparison of late reverberation power spectral density estimators," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 6, pp. 1056–1071, Jun. 2018, doi: 10.1109/TASLP.2018.2804172.

[24] O. Thiergart, M. Taseska, and E. A. P. Habets, "An informed MMSE filter based on multiple instantaneous direction-of-arrival estimates," in *Proc. 21st Eur. Signal Process. Conf. (EUSIPCO)*, Marrakech, Morocco, Sep. 2013, pp. 1–5.

[25] S. Braun and E. A. P. Habets, "Dereverberation in noisy environments using reference signals and a maximum likelihood estimator," in *Proc. 21st Eur. Signal Process. Conf. (EUSIPCO)*, Marrakech, Morocco, Sep. 2013, pp. 1–5.

[26] I. Kodrasi and S. Doclo, "Joint late reverberation and noise power spectral density estimation in a spatially homogeneous noise field," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 441–445, doi: 10.1109/ICASSP.2018.8462142.

[27] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 36, no. 2, pp. 145–152, Feb. 1988, doi: 10.1109/29.1509.

[28] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 7, pp. 1717–1731, Sep. 2010, doi: 10.1109/TASL.2010.2052251.

[29] A. Jukic and S. Doclo, "Speech dereverberation using weighted prediction error with Laplacian model of the desired signal," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 5172–5176, doi: 10.1109/ICASSP.2014.6854589.

[30] M. Parchami, W.-P. Zhu, and B. Champagne, "Speech dereverberation using weighted prediction error with correlated inter-frame speech components," *Speech Commun.*, vol. 87, pp. 49–57, Mar. 2017, doi: 10.1016/j.specom.2017.01.004.

[31] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A Python package for weighted prediction error dereverberation in numpy and tensorflow for online and offline processing," in *Proc. Speech Commun.; 13th ITG-Symp.*, Oldenburg, Germany, Oct. 2018, pp. 216–220.

[32] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evol. Comput.*, vol. 1, pp. 1–23, Dec. 1993, doi: 10.1162/evco.1993.1.1.1.

[33] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989, pp. 1–25.

[34] T. Bäck and F. Kursawe, "Evolutionary algorithms for fuzzy logic: A brief overview," in *Fuzzy Logic and Soft Computing*. Singapore: World Scientific, 1995, pp. 3–10.

[35] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, Nov. 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[36] M. N. A. Wahab, S. N. Meziani, and A. Atyabi, "A comprehensive review of swarm optimization algorithms," *PLoS ONE*, vol. 10, no. 5, pp. 1–36, 2015, doi: 10.1371/journal.pone.0122827.

[37] A. Gotmare, S. S. Bhattacharjee, R. Patidar, and N. V. George, "Swarm and evolutionary computing algorithms for system identification and filter design: A comprehensive review," *Swarm Evol. Comput.*, vol. 32, pp. 68–84, Feb. 2017, doi: 10.1016/j.swevo.2016.06.007.

[38] S. Kiranyaz, J. Pulkkinen, and M. Gabbouj, "Multi-dimensional particle swarm optimization in dynamic environments," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2212–2223, Mar. 2011, doi: 10.1016/j.eswa.2010.08.009.

[39] K. A. D. Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Dept. Comput. Commun. Sci., Univ. Michigan, Ann Arbor, MI, USA, 1975.

[40] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 207–234, 2002, doi: 10.1162/106365602760234081.

[41] D. Beasley, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimization," *Evol. Comput.*, vol. 1, no. 2, pp. 101–125, Jun. 1993, doi: 10.1162/evco.1993.1.2.101.

[42] D. Parrott and X. Li, "A particle swarm model for tracking multiple peaks in a dynamic environment using speciation," in *Proc. Congr. Evol. Comput.*, Portland, OR, USA, Jun. 2004, pp. 98–103, doi: 10.1109/CEC.2004.1330843.

[43] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 39, no. 6, pp. 1362–1381, Dec. 2009, doi: 10.1109/TSMCB.2009.2015956.

[44] X. Li, J. Branke, and T. Blackwell, "Particle swarm with speciation and adaptation in a dynamic environment," in *Proc. 8th Annu. Conf. Genetic Evol. Comput. (GECCO)*, Seattle, WA, USA, 2006, pp. 51–58, doi: 10.1145/1143997.1144005.

[45] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput. IEEE World Congr. Comput. Intell.*, Anchorage, AK, USA, May 1998, pp. 69–73, doi: 10.1109/ICEC.1998.699146.

[46] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. Congr. Evol. Comput. (CEC)*, Washington, DC, USA, Jul. 1999, pp. 1945–1950, doi: 10.1109/CEC.1999.785511.

[47] A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization," *Comput. Oper. Res.*, vol. 33, no. 3, pp. 859–871, Mar. 2006, doi: 10.1016/j.cor.2004.08.012.

[48] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proc. Congr. Evol. Comput.*, Seoul, South Korea, May 2001, pp. 101–106, doi: 10.1109/CEC.2001.934377.

[49] A. Ratnaweera, S. Halgamuge, and H. Watson, "Particle swarm optimization based on self-adaptive acceleration factors," in *Proc. Int. Conf. Fuzzy Syst. Knowl. Discovery*, Guilin, China, 2003, pp. 264–268, doi: 10.1109/WGEC.2009.55.

[50] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, Jun. 2004, doi: 10.1109/TEVC.2004.826071.

[51] T. Yamaguchi and K. Yasuda, "Adaptive particle swarm optimization; self-coordinating mechanism with updating information," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Taiwan, Oct. 2006, pp. 2303–2308, doi: 10.1109/ICSMC.2006.385206.

[52] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Adaptive multi-objective particle swarm optimization based on competitive learning," in *Proc. 11th Int. Conf. Prognostics Syst. Health Manag. (PHM-Jinan)*, Singapore, Oct. 2020, pp. 2281–2288, doi: 10.1109/PHM-Jinan48558.2020.00047.

[53] N. Egidi and P. Maponi, "A Sherman–Morrison approach to the solution of linear systems," *J. Comput. Appl. Math.*, vol. 189, pp. 703–718, May 2006, doi: 10.1016/j.cam.2005.02.013.

[54] W.-J. Zen and X.-L. Li, "High-resolution multiple wideband and non-stationary source localization with unknown number of sources," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3125–3136, Jun. 2010, doi: 10.1109/TSP.2010.2046041.

[55] R. Brits, "Niching strategies for particle swarm optimization," M.S. thesis, Dept. Fac. Natural Agricult. Sci., Pretoria Univ., Pretoria, South Africa, 2002.

[56] M. R. Bai, J.-G. Ih, and J. Benesty, *Acoustic Array Systems: Theory, Implementation, and Application*. Singapore: Wiley, 2013, p. 298.

[57] M. R. Bai, S.-S. Lan, J.-Y. Huang, Y.-C. Hsu, and H.-C. So, "Audio enhancement and intelligent classification of household sound events using a sparsely deployed array," *J. Acoust. Soc. Amer.*, vol. 147, no. 1, pp. 11–24, 2020, doi: 10.1121/10.0000492.

[58] S. Raval. *Artificial Intelligence Education*. San Bruno, CA, USA: YouTube, May 2018.

[59] Deeplizard. *Deep Learning Fundamentals-Classic Edition*, 2017.

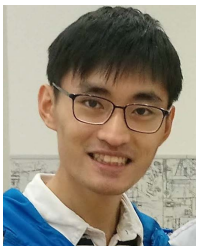
[60] E. A. Lehmann and A. M. Johansson, "Diffuse reverberation model for efficient image-source simulation of room impulse responses," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 6, pp. 1429–1439, Aug. 2010, doi: 10.1109/TASL.2009.2035038.

[61] E. A. Lehmann and A. M. Johansson, "Prediction of energy decay in room impulse responses simulated with an image-source model," *J. Acoust. Soc. Amer.*, vol. 124, no. 1, pp. 269–277, Jul. 2008, doi: 10.1121/1.2936367.

[62] P. Pal and P. P. Vaidyanathan, "Nested arrays: A novel approach to array processing with enhanced degrees of freedom," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4167–4181, Aug. 2010, doi: 10.1109/TSP.2010.2049264.

[63] S. K. Yadav and N. V. George, "Coarray MUSIC-group delay: High-resolution source localization using non-uniform arrays," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9597–9601, Sep. 2021, doi: 10.1109/TVT.2021.3101254.

MINGSIAN R. BAI (Senior Member, IEEE) was born in Taipei, Taiwan, in 1959. He received the B.S. degree in power mechanical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1981, the M.S. degree in business management from the National Chen-Chi University, Taipei, in 1984, and the M.S. degree in mechanical engineering and the Ph.D. degree in engineering mechanics and aerospace engineering from the Graduate School, Iowa State University, in 1985 and 1989, respectively. He currently serves as a Distinguished Professor with the Department of Power Mechanical Engineering and the Department of Electrical Engineering and the Director of the Telecom-Electroacoustics-Audio (TEA) Laboratory. He was also a Visiting Scholar at the Center of Vibration and Acoustics, Penn State University, USA, University of Adelaide, Australia, the Institute of Sound and Vibration Research (ISVR), U.K., and LMS, FAU, Erlangen-Nürnberg, in 1997, 2000, 2002, and 2019, respectively. His research interests include acoustics, spanning acoustic array systems, audio signal processing, active noise, and vibration control.



FAN-JIE KUNG was born in Taichung, Taiwan, in 1988. He received the B.S. degree in electrical engineering from the National Taipei University of Technology, Taipei, Taiwan, in 2011, and the M.S. degree in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 2013, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering. He worked at the National Chung-Shan Institute of Science and Technology, from 2013 to 2019. His research interests include statistical signal processing, speech dereverberation, noise reduction, and source localization.

CHUN-SHIAN TAO was born in Hualien, Taiwan, in 1996. He received the B.S. degree in electrical and mechanical engineering from the National Taipei University of Technology, Taipei, Taiwan, in 2018, and the M.S. degree in power mechanical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 2020. He is currently an Audio Software Engineer at MediaTek Inc., Hsinchu. His research interests include source localization, source separation, and speech enhancement.