

GRADIENT-BASED SOLUTION OF MAXIMUM LIKELIHOOD ANGLE ESTIMATION FOR VIRTUAL ARRAY MEASUREMENTS

Peter Vouras^{*†} Alec Weiss^{*} Maria Becker[†] Ben Jamroz^{*} Jeanne Quimby^{*} Dylan Williams^{*} Kate Remley^{*}

^{*†} National Institute of Standards and Technology, Wireless Networks Division, Gaithersburg, MD

^{*}National Institute of Standards and Technology, Metrology for Wireless Systems Group, Boulder, CO

[†]Hastings College, Hastings, NE

ABSTRACT

Precise measurement and characterization of millimeter wave channels requires antennas capable of high angular resolution to resolve closely spaced multipath sources. To achieve angular resolution on the order of a few degrees these antennas must be electrically large which is impractical for phased array architectures at millimeter wave frequencies. An alternative approach is to synthesize a virtual aperture in space by using an accurate mechanical positioner to move a receive antenna to points along a sampling grid. An advantage of creating virtual apertures is that the received signal is digitized at every spatial sample position which enables the use of sophisticated angle estimation algorithms such as maximum likelihood (ML) techniques. The main contribution of this paper is a new gradient-based implementation of maximum likelihood angle estimation that was demonstrated on virtual array data collected at 28 GHz using a vector network analyzer (VNA).

Index Terms— maximum likelihood angle estimation, synthetic aperture, virtual array, millimeter wave channel sounding, gradient descent algorithm

1. INTRODUCTION

Consider a virtual array that responds to a single polarization created using N spatial samples taken with an omnidirectional antenna on a planar grid. Assume Q narrowband sources in the far field of the receive antenna impinge on the array from the directions $(\theta_0, \phi_0), (\theta_1, \phi_1), \dots, (\theta_{Q-1}, \phi_{Q-1})$ in a spherical coordinate system. The complex envelope of the signal received at the i th virtual array element is

$$x_i(t) = \sum_{k=0}^{Q-1} s_k(t) e^{-j\omega_0 \tau_i(\theta_k, \phi_k)} + n_i(t) \quad (1)$$

where $s_k(t)$ denotes the signal emanating from the k th source as observed at the first array element, ω_0 is the center frequency of the sources, $\tau_i(\theta_k, \phi_k)$ is the incremental delay from the first array element to the i th array element for a plane wave impinging on the array from the direction

(θ_k, ϕ_k) , and $n_i(t)$ is additive zero-mean white measurement noise at the i th element that is assumed to be independent across all array elements. Since the sources are narrowband, the interelement delays of the k th signal, $\tau_i(\theta_k, \phi_k)$ for $i = 0, \dots, N-1$, correspond to incremental phase shifts as the plane wave traverses across the virtual aperture. Thus, (1) can be rewritten as

$$x_i(t) = \sum_{k=0}^{Q-1} s_k(t) e^{-j\frac{2\pi}{\lambda}(md_x u_k + nd_y v_k)} + n_i(t) \quad (2)$$

where for $N = N_x N_y$, $m = 0, \dots, N_x - 1$ is the element index in the x-direction, $n = 0, \dots, N_y - 1$ is the element index in the y-direction, d_x is the spacing between elements in the x-direction, d_y is the spacing between elements in the y-direction, and $u_k = \sin \theta_k \cos \phi_k$, $v_k = \sin \theta_k \sin \phi_k$.

The array output vector of signals received at each array element can be written as

$$\mathbf{x}(t) = \sum_{k=0}^{Q-1} \mathbf{a}(u_k, v_k) s_k(t) + \mathbf{n}(t) \quad (3)$$

where $\mathbf{x}(t) = [x_0(t), \dots, x_{N-1}(t)]^T$, $\mathbf{n}(t) = [n_0(t), \dots, n_{N-1}(t)]^T$, and $\mathbf{a}(u_k, v_k)$ is the $N \times 1$ steering vector of the array in the direction (u_k, v_k) given by,

$$\mathbf{a}(u_k, v_k) = \left[e^{-j\frac{2\pi}{\lambda}(md_x u_k + nd_y v_k)} \Big|_{0 \leq m \leq N_x - 1, 0 \leq n \leq N_y - 1} \right]^T. \quad (4)$$

Using matrix notation yields,

$$\mathbf{x}(t) = \mathbf{A}(\mathbf{u}, \mathbf{v})\mathbf{s}(t) + \mathbf{n}(t) \quad (5)$$

where $\mathbf{A}(\mathbf{u}, \mathbf{v})$ is the $N \times Q$ matrix of steering vectors

$$\mathbf{A}(\mathbf{u}, \mathbf{v}) = [\mathbf{a}(u_0, v_0), \dots, \mathbf{a}(u_{Q-1}, v_{Q-1})] \quad (6)$$

and $\mathbf{u} = [u_0, \dots, u_{Q-1}]^T$, $\mathbf{v} = [v_0, \dots, v_{Q-1}]^T$, and $\mathbf{s}(t)$ is the $Q \times 1$ vector of signals $\mathbf{s}(t) = [s_1(t) \ s_2(t), \dots, s_{Q-1}(t)]^T$. Assume the signals received at each array element are digitized

over M time instants t_0, \dots, t_{M-1} . The sampled measurements can be expressed as

$$\mathbf{X} = \mathbf{A}(\mathbf{u}, \mathbf{v})\mathbf{S} + \mathbf{N} \quad (7)$$

with

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}(t_0), \dots, \mathbf{x}(t_{M-1})]_{N \times M} \\ \mathbf{N} &= [\mathbf{n}(t_0), \dots, \mathbf{n}(t_{M-1})]_{N \times M} \\ \mathbf{S} &= [\mathbf{s}(t_0), \dots, \mathbf{s}(t_{M-1})]_{Q \times M} \end{aligned} \quad (8)$$

The objective of the maximum likelihood angle estimator is to determine the incoming directions (u_j, v_j) of the Q multipath sources from the M array snapshots $\mathbf{x}(t_0), \dots, \mathbf{x}(t_{M-1})$. The underlying assumptions necessary for deriving the ML estimator are that the number of sources Q is known or can be estimated and that $Q < N$, the steering vectors $\mathbf{a}(u_k, v_k)$ are linearly independent for $k = 0, \dots, Q-1$, and $E[\mathbf{n}(t)\mathbf{n}(t)^H] = \sigma_n^2 \mathbf{I}$. As described in [1], the computation of the ML angle estimator proceeds by determining the joint probability density function of the sampled data and subsequently the log-likelihood function that must then be maximized with respect to the unknown angles. The final result is that the ML estimates of the directions (u_j, v_j) can be obtained by maximizing the function

$$\max_{\mathbf{u}, \mathbf{v}} J(\mathbf{u}, \mathbf{v}) = \text{tr}[\mathbf{P}_{\mathbf{A}(\mathbf{u}, \mathbf{v})} \widehat{\mathbf{R}}] \quad (9)$$

where the sample covariance matrix is

$$\widehat{\mathbf{R}} = \frac{1}{M} \sum_{k=0}^{M-1} \mathbf{x}(t_k) \mathbf{x}(t_k)^H \quad (10)$$

and $\mathbf{P}_{\mathbf{A}(\mathbf{u}, \mathbf{v})}$ is the projection matrix onto the range space of $\mathbf{A}(\mathbf{u}, \mathbf{v})$,

$$\begin{aligned} \mathbf{P}_{\mathbf{A}(\mathbf{u}, \mathbf{v})} &= \mathbf{A}(\mathbf{u}, \mathbf{v}) [\mathbf{A}(\mathbf{u}, \mathbf{v})^H \mathbf{A}(\mathbf{u}, \mathbf{v})]^{-1} \mathbf{A}(\mathbf{u}, \mathbf{v})^H \\ &= \mathbf{A}(\mathbf{u}, \mathbf{v}) \mathbf{A}(\mathbf{u}, \mathbf{v})^\dagger, \end{aligned} \quad (11)$$

with \dagger used to denote the pseudoinverse.

1.1. Alternating Projections (AP) Algorithm

The alternating projections (AP) algorithm maximizes the MLE cost function $J(\mathbf{u}, \mathbf{v})$ with respect to one pair of parameters (u_k, v_k) while holding the other parameters fixed. Since iterations of the AP algorithm perform a maximization at every step, the value of $J(\mathbf{u}, \mathbf{v})$ can never decrease, so the algorithm is guaranteed to converge to a local maximum. Depending on the initial conditions, the local maximum may or may not coincide with the global maximum. Since $J(\mathbf{u}, \mathbf{v})$ will in general have many local maxima, proper initialization is vital for the AP algorithm to converge to the global solution.

The projection matrix $\mathbf{P}_{\mathbf{A}(\mathbf{u}, \mathbf{v})}$ can be decomposed as

$$\begin{aligned} \mathbf{P}_{\mathbf{A}(\mathbf{u}, \mathbf{v})} &= \mathbf{P}_{[\mathbf{A}(\widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k), \mathbf{a}(u_k, v_k)]} \\ &= \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k)} + \mathbf{P}_{\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k)}} \end{aligned} \quad (12)$$

where the $(Q-1) \times 1$ vectors $\widehat{\mathbf{u}}_k$ and $\widehat{\mathbf{v}}_k$ are

$$\begin{aligned} \widehat{\mathbf{u}}_k &= [u_0, u_1, \dots, u_{k-1}, u_{k+1}, \dots, u_{Q-1}]^T \\ \widehat{\mathbf{v}}_k &= [v_0, v_1, \dots, v_{k-1}, v_{k+1}, \dots, v_{Q-1}]^T \end{aligned} \quad (13)$$

and the $N \times (Q-1)$ matrix $\mathbf{A}(\widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k)$ is

$$\mathbf{A}(\widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k) = [\mathbf{a}(u_0, v_0), \dots, \mathbf{a}(u_{k-1}, v_{k-1}), \mathbf{a}(u_{k+1}, v_{k+1}), \dots, \mathbf{a}(u_{Q-1}, v_{Q-1})]. \quad (14)$$

Rewriting the maximization problem in (9) to search along the k th spatial direction (u_k, v_k) at the $(l+1)$ st AP algorithm iteration while holding all other directions fixed yields

$$u_k^{(l+1)}, v_k^{(l+1)} = \arg \max_{u_k, v_k} \text{tr}[\mathbf{P}_{[\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)}), \mathbf{a}(u_k, v_k)]} \widehat{\mathbf{R}}]. \quad (15)$$

Equation (15) states that to obtain the angle estimates $u_k^{(l+1)}, v_k^{(l+1)}$ for the k th source at the $(l+1)$ st AP algorithm iteration, the parameters $\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)}$ are held fixed while the parameters u_k, v_k are free to vary. The equivalent maximization problem is [2]

$$u_k^{(l+1)}, v_k^{(l+1)} = \arg \max_{u_k, v_k} \text{tr}[\mathbf{P}_{\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}} \widehat{\mathbf{R}}] \quad (16)$$

and the vector $\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}$ can be written as

$$\begin{aligned} \mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})} &= [\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}] \mathbf{a}(u_k, v_k) \\ &= [\mathbf{I} - \mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)}) \mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})^\dagger] \mathbf{a}(u_k, v_k). \end{aligned} \quad (17)$$

Equation (17) shows that the vector $\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}$ is orthogonal to the projection of $\mathbf{a}(u_k, v_k)$ onto the column space of $\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})$. Also by (11)

$$\begin{aligned} \mathbf{P}_{\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}} &= \\ &= \frac{[\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}] [\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}]^H}{[\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}]^H [\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}]}. \end{aligned} \quad (18)$$

Define the unit norm vector

$$\mathbf{b}_k^{(l)} \equiv \mathbf{b}(u_k, v_k; \widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)}) = \frac{\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}}{\|\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\|_2} \quad (19)$$

and substitute (18) into (16). By applying properties of the trace operator including $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$, the optimization problem in (16) to be solved for the k th signal source becomes

$$\begin{aligned} u_k^{(l+1)}, v_k^{(l+1)} &= \arg \max_{u_k, v_k} \mathbf{b}_k^{(l)H} \widehat{\mathbf{R}} \mathbf{b}_k^{(l)} \\ &\equiv \arg \max_{u_k, v_k} J^l(u_k, v_k). \end{aligned} \quad (20)$$

The two primary contributions of this paper described in the next sections are a successful approach to initialize the MLE-AP algorithm and a gradient-based method to maximize the cost function $J^l(u_k, v_k)$ at each AP iteration.

2. GRADIENT-BASED IMPLEMENTATION OF MLE-AP ALGORITHM

2.1. Derivation of Gradient Vector

In this section an analytical expression for the gradient vector of the $(l+1)$ st cost function specified in (20) is derived. To start, rewrite the cost function as

$$J^l(u_k, v_k) = \quad (21)$$

$$\begin{aligned} &\frac{\mathbf{a}(u_k, v_k)^H \left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})} \right]^H \widehat{\mathbf{R}} \left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})} \right] \mathbf{a}(u_k, v_k)}{\mathbf{a}(u_k, v_k)^H \left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})} \right] \mathbf{a}(u_k, v_k)} \\ &\equiv \frac{\mathbf{a}(u_k, v_k)^H \mathbf{B} \mathbf{a}(u_k, v_k)}{\mathbf{a}(u_k, v_k)^H \mathbf{Q} \mathbf{a}(u_k, v_k)} \end{aligned}$$

by substituting (17) into (20) and using the idempotent and self-adjoint properties of orthogonal projection matrices.

Note that the steering vector $\mathbf{a}(u_k, v_k)$ defined in (4) can be rewritten as the Kronecker product of two steering vectors,

$$\mathbf{a}(u_k, v_k) = \left[e^{-j \frac{2\pi}{\lambda} m d_x u_k} \right]^T \otimes \left[e^{-j \frac{2\pi}{\lambda} n d_y v_k} \right]^T \quad (22)$$

with $0 \leq m \leq N_x - 1$ and $0 \leq n \leq N_y - 1$. Hereafter, to simplify notation, the subscript k denoting the k th AOA source will be dropped from the coordinates u, v . Next consider a change in pointing direction corresponding to (δ_u, δ_v) . Then,

$$\begin{aligned} \mathbf{a}(u + \delta_u, v + \delta_v) &= \\ &\left[e^{-j \frac{2\pi}{\lambda} m d_x (u + \delta_u)} \Big|_{0 \leq m \leq N_x - 1} \right]^T \otimes \\ &\left[e^{-j \frac{2\pi}{\lambda} n d_y (v + \delta_v)} \Big|_{0 \leq n \leq N_y - 1} \right]^T. \end{aligned} \quad (23)$$

Define the diagonal matrices,

$$\begin{aligned} \Delta u &= \delta_u \text{diag} \left[0, \frac{2\pi}{\lambda} d_x, \dots, \frac{2\pi}{\lambda} (N_x - 1) d_x \right] \\ &= \delta_u \mathbf{T}_x \end{aligned} \quad (24)$$

and similarly $\Delta v = \delta_v \mathbf{T}_y$ using d_y instead of d_x . Then using the identity

$$\Delta u \oplus \Delta v = \Delta u \otimes \mathbf{I} + \mathbf{I} \otimes \Delta v \quad (25)$$

and the properties of the matrix exponential, the perturbed steering vector $\mathbf{a}(u + \delta_u, v + \delta_v)$ can be written as

$$\begin{aligned} \mathbf{a}(u + \delta_u, v + \delta_v) &= e^{-j \Delta u} \otimes e^{-j \Delta v} \mathbf{a}(u, v) \\ &= e^{-j(\Delta u \oplus \Delta v)} \mathbf{a}(u, v). \end{aligned} \quad (26)$$

At this point it is useful to clarify the overarching strategy for computing the gradient vector of the cost function $J(u, v)$ in (21), where the superscript iteration index l has been dropped for simplicity. A related approach is also described in [3]. The desired gradient vector of $J(u, v)$ to be computed is defined as $\nabla J = [\partial J / \partial u \quad \partial J / \partial v]^T$. In terms of numerator and denominator functions, $J(u, v) = N(u, v) / D(u, v)$, so using the quotient rule for differentiation yields

$$\frac{\partial J}{\partial u} = \frac{\frac{\partial N}{\partial u} D(u, v) - \frac{\partial D}{\partial u} N(u, v)}{D(u, v)^2} \quad (27)$$

$$\frac{\partial J}{\partial v} = \frac{\frac{\partial N}{\partial v} D(u, v) - \frac{\partial D}{\partial v} N(u, v)}{D(u, v)^2}.$$

It is clear that to apply the quotient rule for computing ∇J it is also necessary to compute $\nabla N = [\partial N / \partial u \quad \partial N / \partial v]^T$ and $\nabla D = [\partial D / \partial u \quad \partial D / \partial v]^T$. A useful fact is that since the numerator function N is continuously differentiable with respect to u and v , the directional derivative $N'(\mathbf{p}; \mathbf{d})$ of N at the point $\mathbf{p} = [u \quad v]^T$ in the direction $\mathbf{d} = [\delta_u \quad \delta_v]^T$ is equal to [4]

$$N'(\mathbf{p}; \mathbf{d}) = \nabla N(\mathbf{p})^T \mathbf{d}. \quad (28)$$

In the case at hand, ∇N is unknown and the quantity to be determined, but the directional derivative $N'(\mathbf{p}; \mathbf{d})$ can also be calculated as the derivative with respect to t of the function $G_N(t) = N(\mathbf{p} + t\mathbf{d})$ evaluated at $t = 0$,

$$N'(\mathbf{p}; \mathbf{d}) = \left. \frac{d}{dt} G_N(t) \right|_{t=0} = \left. \frac{d}{dt} N(\mathbf{p} + t\mathbf{d}) \right|_{t=0}. \quad (29)$$

Thus ∇N can be recovered by using (29) to compute the directional derivative $N'(\mathbf{p}; \mathbf{d})$ and then writing the result in a form compatible with (28) to recover the gradient vector. The same procedure also applies to the denominator function $D(u, v)$ using the derivative with respect to t of the function $G_D(t) = D(\mathbf{p} + t\mathbf{d})$ evaluated at $t = 0$.

Continuing along this tack and starting with $D(u, v)$ yields,

$$\begin{aligned} G_D(t) &= D(u + t\delta_u, v + t\delta_v) = \\ &= \mathbf{a}(u + t\delta_u, v + t\delta_v)^H \mathbf{Q} \mathbf{a}(u + t\delta_u, v + t\delta_v) \\ &= \mathbf{a}(u, v)^H e^{j(\Delta u \oplus \Delta v)t} \mathbf{Q} e^{-j(\Delta u \oplus \Delta v)t} \mathbf{a}(u, v) \end{aligned} \quad (30)$$

and the desired directional derivative

$$\begin{aligned} D'(\mathbf{p}; \mathbf{d}) &= j \left[\delta_u \text{tr} \left([\mathbf{T}_x \otimes \mathbf{I}, \mathbf{Q}] \mathbf{a}(u, v) \mathbf{a}(u, v)^H \right) \right. \\ &\quad \left. + \delta_v \text{tr} \left([\mathbf{I} \otimes \mathbf{T}_y, \mathbf{Q}] \mathbf{a}(u, v) \mathbf{a}(u, v)^H \right) \right] \end{aligned} \quad (31)$$

where the notation $[\mathbf{A}, \mathbf{B}]$ denotes the Lie bracket, $[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA}$. Rewriting (31) in matrix form and comparing to (28) yields

$$D'(\mathbf{p}; \mathbf{d}) = \begin{bmatrix} -\text{imag}(\text{tr}([\mathbf{T}_x \otimes \mathbf{I}, \mathbf{Q}] \mathbf{a}(u, v) \mathbf{a}(u, v)^H)) \\ -\text{imag}(\text{tr}([\mathbf{I} \otimes \mathbf{T}_y, \mathbf{Q}] \mathbf{a}(u, v) \mathbf{a}(u, v)^H)) \end{bmatrix}^T \begin{bmatrix} \delta_u \\ \delta_v \end{bmatrix} \equiv \nabla D(\mathbf{p})^T \mathbf{d}. \quad (32)$$

Repeating the same argument for the numerator function $N(u, v)$ results in

$$N'(\mathbf{p}; \mathbf{d}) = \begin{bmatrix} -\text{imag}(\text{tr}([\mathbf{T}_x \otimes \mathbf{I}, \mathbf{B}] \mathbf{a}(u, v) \mathbf{a}(u, v)^H)) \\ -\text{imag}(\text{tr}([\mathbf{I} \otimes \mathbf{T}_y, \mathbf{B}] \mathbf{a}(u, v) \mathbf{a}(u, v)^H)) \end{bmatrix}^T \begin{bmatrix} \delta_u \\ \delta_v \end{bmatrix} \equiv \nabla N(\mathbf{p})^T \mathbf{d}. \quad (33)$$

Now the components of ∇N and ∇D are clearly available to substitute into (27) to compute ∇J .

2.2. Conjugate Gradient Algorithm

The conjugate gradient (CG) algorithm for maximizing the cost function in (21) for the k th AOA source at the l th iteration of the MLE-AP algorithm is [4],

Require: Initial or a priori angle estimates u_0 and v_0 . One approach for obtaining u_0, v_0 is to compute (35) on a discrete angular grid and then choose the output peaks.

- 1: Set the initial search direction $\mathbf{d}_0 = \nabla J(u_0, v_0)$
 - 2: Until $\|\nabla J(u_j, v_j)\|_2 \leq \epsilon$, where j denotes the conjugate gradient iteration index, do the following:
 - 3: Determine the step-size μ_j
 - 4: Set $\mathbf{p}_{j+1} = \mathbf{p}_j + \mu_j \mathbf{d}_j$ where $\mathbf{p}_j = [u_j \quad v_j]^T$
 - 5: Set $\mathbf{g}_{j+1} = \nabla J(u_{j+1}, v_{j+1})$
 - 6: Set $\mathbf{d}_{j+1} = \mathbf{g}_{j+1} + \alpha_j \mathbf{d}_j$
 - 7: Set $\alpha_j = \frac{\mathbf{g}_{j+1}^T (\mathbf{g}_{j+1} - \mathbf{g}_j)}{\mathbf{g}_j^T \mathbf{g}_j}$
 - 8: Set $j = j + 1$
-

The step-size μ_j for the j th conjugate gradient iteration can be set equal to a constant value small enough to ensure algorithm convergence or it can be chosen via a one-dimensional line search. The preferred approach is to use Armijo's rule. Given an initial stepsize, $0 < \rho < 1$, Armijo's rule chooses the final stepsize μ_j to be the first value in the sequence $1, \rho, \rho^2, \rho^3, \dots$ that satisfies the condition

$$J(\mathbf{p}_j + \mu_j \mathbf{d}_j) \geq J(\mathbf{p}_j) + \mu_j \alpha \nabla J(\mathbf{p}_j)^T \mathbf{d}_j \quad (34)$$

for a fixed scalar $0 < \alpha < 0.5$. In other words, $\mu_j = \rho^m$ for some integer m .

3. MEASURED RESULTS AND CONCLUSIONS

Since the cost function $J(\mathbf{p})$ has multiple peaks it is necessary to properly initialize the conjugate gradient algorithm to ensure that it converges to the correct solution. A simple approach is to compute the output of a spatial matched filter s for all u, v directions in space as given by

$$s(u, v) = \mathbf{a}(u, v)^H \hat{\mathbf{R}} \mathbf{a}(u, v) \quad (35)$$

and then choose the largest peaks for initial angle estimates. Figure 1 illustrates the dB output of the spatial matched filter in sine space coordinates for array data collected between 26-40 GHz in a lab room. The white crosses correspond to multipath peaks detected using a thresholding algorithm. These peaks are used to initialize the CG algorithm. Figure 2 illustrates the algorithm's path to convergence within 40 iterations from initial condition (IC) to global maximum.

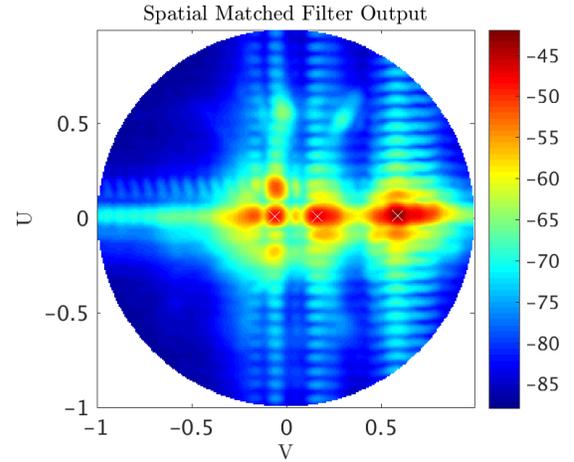


Fig. 1. Output of Spatial Matched Filter

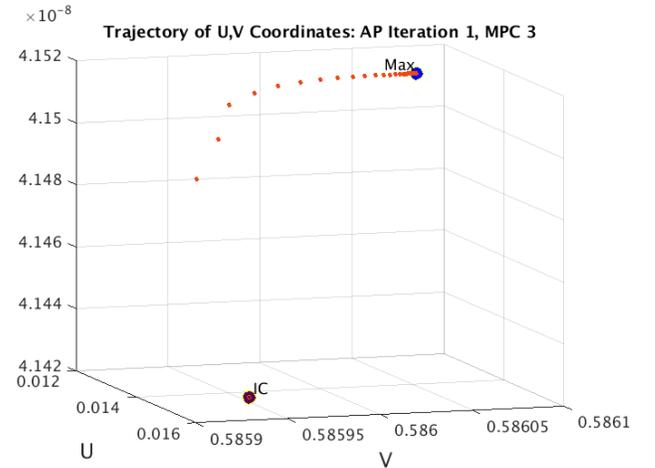


Fig. 2. Algorithm Convergence Trajectory: Multipath Cluster 3

4. REFERENCES

- [1] I. Ziskind and M. Wax, "Maximum Likelihood Localization of Multiple Sources by Alternating Projection," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, No. 10, October 1988, pp. 1553-1560.
- [2] I. Ziskind and M. Wax, "Maximum Likelihood Estimation via the Alternating Projection Maximization Algorithm," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 6-9, 1987.
- [3] S. T. Smith, "Optimum Phase-Only Adaptive Nulling," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 47, No. 7, July 1999, pp. 1835-1843.
- [4] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1995.