Signal Processing 205 (2023) 108879

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Efficient computation of ML DOA estimates under unknown nonuniform sensor noise powers[☆]

J. Selva

University of Alicante, Dept. of Physics, Systems Engineering and Signal Theory (DFISTS), Alicante, E-03080, Spain

ARTICLE INFO

Article history:
Received 31 August 2021
Revised 30 July 2022
Accepted 28 November 2022
Available online 30 November 2022

Keywords:
DOA Estimation
Maximum likelihood
Newton method
Hessian

ABSTRACT

This paper presents an efficient method for computing Maximum Likelihood (ML) direction-of-arrival (DOA) estimates in scenarios in which the sensor noise powers are nonuniform and unknown. The method combines the Alternating Projection (AP) algorithm for coarsely locating additional DOAs and Newton iterations for finally obtaining the ML estimates. Compared with the existing approaches, the method reduces the computational burden significantly due to the small number of Newton iterations required and to the efficient computation of each iteration. Specifically, the iterations are computed in a small number of arithmetic operations thanks to the closed-form formulas for the gradient and Hessian of the ML cost functions presented in this paper. The method's total computational burden is of just a few mega-flops (mega floating-point operations) in typical cases. We present the method for the deterministic and stochastic ML estimators. Then, an analysis of the deterministic ML cost function's gradient reveals an unexpected drawback: its associated estimator is either degenerate or inconsistent. Finally, we assess the method's root-mean-square (RMS) error and computational burden numerically and compare it with other relevant estimators in the literature.

1. Introduction

The DOA estimators for several narrowband sources are usually designed for uniform noise scenarios, i.e, they may be used whenever the noise is uncorrelated, both spatially and temporally, and of the same power at each sensor. This noise uniformity greatly simplifies DOA estimation, because it allows one to describe the noise through a single parameter, its power, and makes it possible to employ estimators such as MUSIC (MUltiple SIgnal Classification) and ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques), [1,2], and simplifies the derivation of ML estimators, [3–5]. However, in a variety of applications, though the noise is spatially and temporally uncorrelated, its power differs from sensor to sensor. This is the case in some seismic and biomedical applications [6], in applications involving sparse arrays [7–12], or whenever there exist imperfections in the sensors' processing chains [13–16]. DOA estimation in these applications becomes more complex, given that the noise powers must be taken into account in the estimation as additional unknown parameters. Actually, the unknown noise powers (UNPs) prevent the direct use of subspace methods such us MUSIC and ESPRIT and significantly increase the complexity of computing ML estimates. There have been attempts to adapt subspace methods to the UNP case, such as [7] and [12], in which the impinging signals are assumed to be uncorrelated. Regarding ML estimation, the UNP ML cost functions depend on a large number of parameters of three different kinds, signal, noise and DOA parameters, and the computation of the corresponding ML estimates involves the location of the cost function's global maximum in all of them. This is a hard optimisation problem that is usually addressed by alternating maximisations in each of these three kinds of parameters in a proper way. In [14,17], the authors proposed an iterative method for the deterministic ML (DML) cost function, that exploited the fact that this function can be concentrated in the signal samples and noise powers separately. They proposed an iteration in which the remaining DOA parameters were obtained through a genetic algorithm. In [18], a more efficient computation method for the UNP DML estimator was presented. This method exploited the relationship between the UNP DML cost function and its equivalent for uniform noise. Fundamentally, the author combined the AP algorithm [19] with a closed-form noise-powers estimation step. This last method outperforms

[☆] This work was supported by the Spanish Ministry of Science and Innovation, the State Agency of Research (AEI) and the European Funds for Regional Development (EFRD) under Project PID2020-117303GB-C22.

E-mail address: jesus.selva@ua.es

https://doi.org/10.1016/j.sigpro.2022.108879

that in [14,17] because it replaces the expensive global-search genetic algorithm with the AP method. In [9], the authors presented a method that resembles that in [18] but for the Stochastic ML (SML) cost function. It is also based on a two-step iteration combining the AP algorithm with an iterative method for the noise powers, termed the Power-Domain (PD) method.

We can see that all these approaches are variants of the standard ascendant direction method in numerical analysis [20, Sec 10.7], in which the maximum of a given cost function is approached iteratively by maximising in sub-sets of the parameters involved. The main advantage of ascendant direction is that it approximately locates the global maximum of the cost function with low complexity. However, it is inefficient for refining an initial coarse estimate, given that the number of iterations required for locating the actual global maximum is usually large. For this refinement, a better alternative is a Newton method given that its convergence rate is quadratic [21, Sec 6.3.1]. The purpose of this paper is to present a method for computing UNP ML estimates based on Newton's method, whose complexity is low due to the efficient implementation of the Newton iterations, which is made possible by the closed-form gradient and Hessian expressions presented in this paper.

The paper has been organised as follows. In the next section, we introduce the signal model and the UNP ML cost functions. Then, we recall the state of the art two-step methods for the computation of the UNP ML estimates in Section 3 and comment on their limitations in Section 4, where we also outline the solution to these limitations. After that, we present the proposed method in Section 5, which is based on Newton iterations, and the gradient and Hessian expressions in Section 6. In Section 7, we check whether the DML and SML estimators are non-degenerate and consistent. Finally, we assess the proposed method in Section 8 numerically.

1.1. Notations and basic concepts

We employ the following notations:

- Column vectors and matrices are written in lower- and upper-case bold font respectively. Thus $\boldsymbol{x}$ denotes a column vector and $\boldsymbol{X}$ a matrix.
- $[\boldsymbol{X}]_{p,q}$ denotes the (p,q) component of matrix $\boldsymbol{X}$, and $[\boldsymbol{X}]_{p,\cdot}$ and $[\boldsymbol{X}]_{\cdot,q}$ its pth row and qth column respectively.
- $\mathrm{diag}(\boldsymbol{x})$ is the diagonal matrix formed by the components of $\boldsymbol{x}$.
- For square $\boldsymbol{X}$, $\mathrm{tr}\{\boldsymbol{X}\}$ denotes the trace of $\boldsymbol{X}$, i.e, the sum of its diagonal components.
- $[\boldsymbol{x};\boldsymbol{y}]$ denotes the column vector formed by concatenating the column vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ vertically.
- $\boldsymbol{X}^H$ stands for the conjugate transpose of $\boldsymbol{X}$.
- $\boldsymbol{X}^\dagger$ stands for the pseudo-inverse of matrix $\boldsymbol{X}$. If $\boldsymbol{X}$ has full column rank then $\boldsymbol{X}^\dagger = (\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H$.
- Given a matrix $\boldsymbol{X}$ of full column rank, its projection matrix is $\boldsymbol{X}\boldsymbol{X}^\dagger$.
- $\boldsymbol{I}_M$ denotes the $M \times M$ identity matrix.
- Given a matrix $\boldsymbol{A}$ of size $M \times K$, its QR decomposition is $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q}$ has size $M \times K$ and follows $\boldsymbol{Q}^H\boldsymbol{Q} = \boldsymbol{I}_K$ and $\boldsymbol{R}$ is upper triangular. For the properties of this decomposition, see [22, Th. 5.2.3].
- The operator '≡' indicates a symbol or function definition.
- Given two variables a and b, the arrow $a \to b$ denotes the replacement of a with b in a given expression.
- $\boldsymbol{\delta}_{K,p}$ denotes a $K \times 1$ Dirac vector

$$[\boldsymbol{\delta}_{K,p}]_k \equiv \begin{cases} 0, & k \neq p \\ 1, & k = p. \end{cases}$$

Throughout the paper, we often omit the dependency on the various parameters in writing for simplicity. Thus, for example, $\boldsymbol{\Phi}$ stands for $\boldsymbol{\Phi}(\boldsymbol{\theta},\boldsymbol{\lambda})$ and $\boldsymbol{\Phi}_o$ for $\boldsymbol{\Phi}_o(\boldsymbol{\theta})$. The actual dependencies are evident from the context.

The starting point for the cost function derivations in the paper is the complex Gaussian probability density function (PDF), [23, Th. 15.1]. Specifically, if the expected value of matrix $\boldsymbol{Z}$ is $\boldsymbol{E}_z$ and the columns of $\boldsymbol{Z}$ are independent and have equal covariance matrix $\boldsymbol{C}_z$, then the PDF of $\boldsymbol{Z}$ is

$$f(\boldsymbol{Z}) = \frac{1}{\pi^{MN}|\boldsymbol{C}_z|^N} \cdot \exp(-\mathrm{tr}\{\boldsymbol{C}_z^{-1}(\boldsymbol{Z}-\boldsymbol{E}_z)(\boldsymbol{Z}-\boldsymbol{E}_z)^H\}). \tag{1}$$

We denote the various ML cost functions in the paper using the subscripts "*o*", "D", and "S",

- $L_{Do}(\boldsymbol{\theta})$ denotes the concentrated deterministic ML cost function for equal but unknown sensor noise power.
- $L_D(\boldsymbol{\theta},\boldsymbol{\lambda})$ and $L_S(\boldsymbol{\theta},\boldsymbol{\lambda})$ respectively denote the concentrated deterministic and stochastic ML cost functions for unknown sensor noise powers.

2. Signal model and UNP maximum likelihood cost functions

We present in the sequel the signal model under unknown noise powers and then the two main ML cost functions, the deterministic and stochastic, in separate subsections. Consider a linear array formed by M sensors and K waves impinging from angles of arrival θ_k, $k = 1,\ldots,K$. If the receiver takes N snapshots, the data model is

$$\boldsymbol{Z} = \boldsymbol{\Phi}_o(\boldsymbol{\theta})\boldsymbol{S} + \boldsymbol{N}, \tag{2}$$

where:

- $[\boldsymbol{Z}]_{m,n}$ is the nth sample from the mth sensor.
- $\boldsymbol{\theta}$ contains the K angles of arrival (AOAs) θ_k, $[\boldsymbol{\theta}]_k \equiv \theta_k$.
- $\boldsymbol{\phi}_o(\theta)$ is the array's response to a wave from angle θ and $\boldsymbol{\Phi}_o(\boldsymbol{\theta})$ is a matrix stacking the responses to the angles in $\boldsymbol{\theta}$, $[\boldsymbol{\Phi}_o(\boldsymbol{\theta})]_{\cdot,k} \equiv \boldsymbol{\phi}_o(\theta_k)$.
- $[\boldsymbol{S}]_{k,n}$ is the nth sample from the kth impinging signal. There are two different models for $\boldsymbol{S}$ called deterministic and stochastic. In the deterministic, the whole matrix $\boldsymbol{S}$ is modeled as a set of unknown parameters while, in the stochastic, the columns of $\boldsymbol{S}$ are modeled as independent samples of a complex Gaussian vector of zero mean and covariance matrix $\boldsymbol{R}_s$.
- All components $[\boldsymbol{N}]_{m,n}$ are independent noise samples that follow a complex normal circularly-symmetric distribution of zero mean and deviation $1/\lambda_m$, where we refer to λ_m as the inverse noise deviation at the mth sensor. Additionally, we define the following vector and diagonal matrix from λ_m,

  $$[\boldsymbol{\lambda}]_m \equiv \lambda_m, \quad \boldsymbol{\Lambda} \equiv \mathrm{diag}(\boldsymbol{\lambda}).$$

  Using $\boldsymbol{\Lambda}$, we have that the columns of $\boldsymbol{N}$ have covariance matrix $\boldsymbol{\Lambda}^{-2}$.

2.1. Deterministic maximum likelihood (DML) cost function

If $\boldsymbol{S}$ is viewed as a deterministic matrix, then $\boldsymbol{Z}$ in (2) has mean $\boldsymbol{E}_z \to \boldsymbol{\Phi}_o\boldsymbol{S}$ and covariance $\boldsymbol{C}_z \to \boldsymbol{\Lambda}^{-2}$ and, from (1), the PDF of $\boldsymbol{Z}$ is

$$f_D(\boldsymbol{Z}|\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{S}) \equiv \frac{|\boldsymbol{\Lambda}|^{2N}}{\pi^{MN}} \exp\left(-\mathrm{tr}\left\{\boldsymbol{\Lambda}^2(\boldsymbol{Z}-\boldsymbol{\Phi}_o\boldsymbol{S})(\boldsymbol{Z}-\boldsymbol{\Phi}_o\boldsymbol{S})^H\right\}\right), \tag{3}$$

where we have written $\boldsymbol{\Phi}_o$ rather than $\boldsymbol{\Phi}_o(\boldsymbol{\theta})$ for simplicity. If we take the logarithm of this expression and introduce the following "whitened" signature matrix

$$\boldsymbol{\Phi}(\boldsymbol{\theta},\boldsymbol{\lambda}) \equiv \boldsymbol{\Lambda}\boldsymbol{\Phi}_o(\boldsymbol{\theta}), \tag{4}$$

then, after straightforward manipulations, we obtain from (3) the cost function

$$L_D(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{S}) \equiv \begin{array}{l} -NM\log\pi + 2N\log|\boldsymbol{\Lambda}| \\ -\mathrm{tr}\{(\boldsymbol{\Lambda Z} - \boldsymbol{\Phi S})(\boldsymbol{\Lambda Z} - \boldsymbol{\Phi S})^H\}). \end{array} \tag{5}$$

Next, as is well known, this expression is maximized in $\boldsymbol{S}$ for fixed $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$, if the product $\boldsymbol{\Phi S}$ is replaced with $\boldsymbol{P\Lambda Z}$, where $\boldsymbol{P}$ is the projection matrix of $\boldsymbol{\Phi}$. For later use, we express this last matrix as

$$\boldsymbol{P} \equiv \boldsymbol{\Phi M \Phi}^H,$$

where $\boldsymbol{M}$ is the inverse correlation matrix of $\boldsymbol{\Phi}$,

$$\boldsymbol{M} \equiv (\boldsymbol{\Phi}^H\boldsymbol{\Phi})^{-1}.$$

So if we replace $\boldsymbol{\Phi S}$ with $\boldsymbol{P\Lambda Z}$ in (5) and perform straight-forward manipulations, we obtain the new cost function

$$\boxed{L_D(\boldsymbol{\theta}, \boldsymbol{\lambda}) \equiv N\big(2\log|\boldsymbol{\Lambda}| - \mathrm{tr}\{(\boldsymbol{I}_M - \boldsymbol{P})\boldsymbol{R}_{z\lambda}\}\big),} \tag{6}$$

where we have neglected the constant $-NM\log\pi$ in (5) and $\boldsymbol{R}_{z\lambda}$ denotes the "whitened" data correlation matrix

$$\boldsymbol{R}_{z\lambda} \equiv \frac{1}{N}\boldsymbol{\Lambda Z Z}^H\boldsymbol{\Lambda}. \tag{7}$$

(6) is the concentrated DML cost function that is used in the rest of the paper.

For uniform noise, the cost function equivalent to L_D is obtained form (6), simply by setting $\boldsymbol{\Lambda} = \boldsymbol{I}_M/\sigma$, where σ is the noise deviation. Thus, setting $\boldsymbol{\Lambda} = \boldsymbol{I}_M/\sigma$ and noting that $\boldsymbol{P}$ turns out to be independent of σ, we obtain

$$N\Big(-2M\log\sigma - \frac{1}{\sigma^2}\mathrm{tr}\{(\boldsymbol{I}_M - \boldsymbol{P})\boldsymbol{R}_z\}\Big), \tag{8}$$

where

$$\boldsymbol{R}_z \equiv \frac{1}{N}\boldsymbol{ZZ}^H.$$

As can be easily checked, (8) can be maximized in σ and $\boldsymbol{\theta}$ independently. This implies that we may obtain a $\boldsymbol{\theta}$-only cost function simply by setting σ equal to a fixed value, say $\sigma = 1$. Thus, from (8), we obtain the cost function

$$\boxed{L_{Do}(\boldsymbol{\theta}) \equiv -N\mathrm{tr}\{(\boldsymbol{I}_M - \boldsymbol{P}_o)\boldsymbol{R}_z\}\big),} \tag{9}$$

where the sub-script "o" indicates that the matrices are computed with $\boldsymbol{\Lambda} = \boldsymbol{I}_M$.

2.2. Stochastic maximum likelihood (SML) cost function

In the stochastic modeling, the columns of $\boldsymbol{S}$ are viewed as independent trials of a complex Gaussian distribution of zero mean and covariance $\boldsymbol{R}_s$ and, from (2), this leads to the PDF in (1) with $\boldsymbol{E}_z \to \boldsymbol{0}$ and $\boldsymbol{C}_z \to \boldsymbol{\Phi}_o\boldsymbol{R}_s\boldsymbol{\Phi}_o^H + \boldsymbol{\Lambda}^{-2}$. For simplicity, let us write this last covariance matrix as

$$\boldsymbol{\Phi}_o\boldsymbol{R}_s\boldsymbol{\Phi}_o^H + \boldsymbol{\Lambda}^{-2} = \boldsymbol{\Lambda}^{-1}(\boldsymbol{\Phi R}_s\boldsymbol{\Phi}^H + \boldsymbol{I}_M)\boldsymbol{\Lambda}^{-1}.$$

Substituting these values of $\boldsymbol{E}_z$ and $\boldsymbol{C}_z$ into (1), we obtain the PDF in the stochastic case

$$f_S(\boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{R}_s) \equiv \frac{|\boldsymbol{\Lambda}|^{2N}}{\pi^{MN}|\boldsymbol{I}_M + \boldsymbol{\Phi R}_s\boldsymbol{\Phi}^H|^N} \cdot \tag{10}$$

$$\exp\left(-N\mathrm{tr}\left\{(\boldsymbol{I}_M + \boldsymbol{\Phi R}_s\boldsymbol{\Phi}^H)^{-1}\boldsymbol{R}_{z\lambda}\right\}\right), \tag{10}$$

where we have inserted the signature matrix in (4) and the whitened correlation matrix in (7).

Taking the logarithm of (10), we obtain the cost function

$$\begin{array}{l} L_S\big(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{R}_S\big) \equiv -MN\log(\pi) + 2N\log|\boldsymbol{\Lambda}| \\ -N\log|\boldsymbol{I}_M + \boldsymbol{\Phi R}_s\boldsymbol{\Phi}^H| - N\mathrm{tr}\left\{\big(\boldsymbol{I}_M + \boldsymbol{\Phi R}_s\boldsymbol{\Phi}^H\big)^{-1}\boldsymbol{R}_{z\lambda}\right\}. \end{array} \tag{11}$$

This expression can be maximised in $\boldsymbol{R}_s$ for fixed $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ and the maximum is attained at

$$\hat{\boldsymbol{R}}_s \equiv \boldsymbol{\Phi}^\dagger\boldsymbol{R}_{z\lambda}(\boldsymbol{\Phi}^\dagger)^H - \boldsymbol{M}. \tag{12}$$

(See [24] for a proof.)

In order to replace $\boldsymbol{R}_s$ with $\hat{\boldsymbol{R}}_s$ in (11), it is convenient to start by performing this same replacement on the covariance matrix appearing twice in (11), namely the matrix

$$\boldsymbol{C} \equiv \boldsymbol{I}_M + \boldsymbol{\Phi}\hat{\boldsymbol{R}}_s\boldsymbol{\Phi}^H. \tag{13}$$

More precisely, we proceed to derive compact expressions of $\boldsymbol{C}$ and $\boldsymbol{C}^{-1}$ in terms of $\boldsymbol{\Phi}$ and $\boldsymbol{P}$.

First, noting that $\boldsymbol{\Phi M \Phi}^H = \boldsymbol{P}$ and $\boldsymbol{\Phi\Phi}^\dagger = \boldsymbol{P}$, the substitution of (12) into (13) yields the desired expression for $\boldsymbol{C}$,

$$\begin{aligned} \boldsymbol{C} \quad &= \boldsymbol{I}_M + \boldsymbol{\Phi}(\boldsymbol{\Phi}^\dagger\boldsymbol{R}_{z\lambda}(\boldsymbol{\Phi}^\dagger)^H - \boldsymbol{M})\boldsymbol{\Phi}^H \\ &= \boldsymbol{I}_M - \boldsymbol{P} + \boldsymbol{P R}_{z\lambda}\boldsymbol{P}. \end{aligned} \tag{14}$$

Second, regarding $\boldsymbol{C}^{-1}$, consider the QR decomposition $\boldsymbol{\Phi} = \boldsymbol{QR}$, with $\boldsymbol{Q}^H\boldsymbol{Q} = \boldsymbol{I}_K$ and invertible $\boldsymbol{R}$, and an $M \times (M-K)$ matrix $\boldsymbol{Q}_\perp$ spanning the orthogonal complement to $\boldsymbol{Q}$, ($\boldsymbol{Q}_\perp^H\boldsymbol{Q}_\perp = \boldsymbol{I}_{M-K}$, $\boldsymbol{Q}^H\boldsymbol{Q}_\perp = \boldsymbol{0}$). Noting that $\boldsymbol{P} = \boldsymbol{QQ}^H$ and $\boldsymbol{I}_M - \boldsymbol{P} = \boldsymbol{Q}_\perp\boldsymbol{Q}_\perp^H$, we may write $\boldsymbol{C}$ in (14) as

$$\boldsymbol{C} = \big[\boldsymbol{Q}_\perp, \, \boldsymbol{Q}\big]\begin{bmatrix} \boldsymbol{I}_{M-K} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^H\boldsymbol{R}_{z\lambda}\boldsymbol{Q} \end{bmatrix}\begin{bmatrix} \boldsymbol{Q}_\perp^H \\ \boldsymbol{Q}^H \end{bmatrix}.$$

From this factorisation, its clear that $\boldsymbol{C}^{-1}$ is given by

$$\boldsymbol{C}^{-1} = \big[\boldsymbol{Q}_\perp, \, \boldsymbol{Q}\big]\begin{bmatrix} \boldsymbol{I}_{M-K} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{Q}^H\boldsymbol{R}_{z\lambda}\boldsymbol{Q})^{-1} \end{bmatrix}\begin{bmatrix} \boldsymbol{Q}_\perp^H \\ \boldsymbol{Q}^H \end{bmatrix}.$$

And, finally, replacing $\boldsymbol{Q} \to \boldsymbol{\Phi R}^{-1}$, we obtain an expression for $\boldsymbol{C}^{-1}$ in terms of $\boldsymbol{\Phi}$ and $\boldsymbol{P}$ only:

$$\begin{aligned} \boldsymbol{C}^{-1} &= \boldsymbol{Q}_\perp\boldsymbol{Q}_\perp^H + \boldsymbol{Q}\big(\boldsymbol{Q}^H\boldsymbol{R}_{z\lambda}\boldsymbol{Q}\big)^{-1}\boldsymbol{Q}^H \\ &= \boldsymbol{I}_M - \boldsymbol{P} \\ &\quad + \boldsymbol{\Phi R}^{-1}\Big(\big(\boldsymbol{R}^{-1}\big)^H\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}\boldsymbol{\Phi R}^{-1}\Big)^{-1}\big(\boldsymbol{R}^{-1}\big)^H\boldsymbol{\Phi}^H \\ &= \boldsymbol{I}_M - \boldsymbol{P} + \boldsymbol{\Phi}\big(\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}\boldsymbol{\Phi}\big)^{-1}\boldsymbol{\Phi}^H. \end{aligned}$$

We may write this formula concisely as

$$\boldsymbol{C}^{-1} = \boldsymbol{I}_M - \boldsymbol{P} + \boldsymbol{P}_z, \tag{15}$$

where

$$\boldsymbol{P}_z \equiv \boldsymbol{\Phi M}_{z\lambda}\boldsymbol{\Phi}^H \text{ and } \boldsymbol{M}_{z\lambda} \equiv (\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}\boldsymbol{\Phi})^{-1}. \tag{16}$$

Coming back to (11), the replacement of $\boldsymbol{R}_s$ with $\hat{\boldsymbol{R}}_s$ can be effected by substituting into that equation the identities (14), (15), and

$$\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{R}_{z\lambda}\} = K.$$

This last identity can be easily deduced from (16). The result of these substitutions, neglecting constant summands, is the concentrated cost functions

$$\boxed{L_S(\boldsymbol{\theta}, \boldsymbol{\lambda}) \equiv N\big(2\log|\boldsymbol{\Lambda}| - \mathrm{tr}\{(\boldsymbol{I}_M - \boldsymbol{P})\boldsymbol{R}_{z\lambda}\} - \log|\boldsymbol{C}|\big).} \tag{17}$$

This is the concentrated stochastic ML cost function that is analysed in the rest of the paper. Note that L_S in (17) is formed by adding a single term to L_D in (6). Actually, we have

$$\boxed{L_S = L_D + L_C, \text{ where } L_C \equiv -N\log|\boldsymbol{C}|.} \tag{18}$$

3. State of the art two-step approach for the computation of UNP ML estimates

Let us now recall the usual two-step approach for computing UNP ML estimators [9,18]. Fundamentally, this approach consists of maximising a given cost function in $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ alternately until convergence is achieved. Since it has the same form for either the DML or SML cost function, let us only recall the version in [18] for the DML estimator. The DML estimate of the pair $(\boldsymbol{\theta}, \boldsymbol{\lambda})$ is the abscissa of the global maximum of $L_D(\boldsymbol{\theta}, \boldsymbol{\lambda})$ in (6), and the method in [18] exploits two basic properties of this last cost function:

1. For a fixed value of $\boldsymbol{\lambda}$, say $\boldsymbol{\lambda}_o$, $L_D(\boldsymbol{\theta}, \boldsymbol{\lambda}_o)$ can be easily cast as a uniform-noise DML cost function. This allows one to maximise $L_D(\boldsymbol{\theta}, \boldsymbol{\lambda}_o)$ in $\boldsymbol{\theta}$ using the AP algorithm [19]. This last algorithm is based on line searches and consists of two steps:
   - **AP initial step.** A $K \times 1$ vector $\hat{\boldsymbol{\theta}}_0$ of coarse estimates is computed in K sub-steps as follows. At the kth sub-step, a $(k-1)$-length vector $\hat{\boldsymbol{\theta}}_{0,k-1}$ is extended to a k-length vector $\hat{\boldsymbol{\theta}}_{0,k}$ by appending the result of a line search, i.e,

     $$\hat{\boldsymbol{\theta}}_{0,k} = \Big[\hat{\boldsymbol{\theta}}_{0,k-1}; \arg\max_{\theta} L_D([\hat{\boldsymbol{\theta}}_{0,k-1}; \theta], \boldsymbol{\lambda}_o)\Big]. \tag{19}$$

     The initial vector $\hat{\boldsymbol{\theta}}_{0,0}$ is empty and the output coarse estimate is $\hat{\boldsymbol{\theta}}_0 = \hat{\boldsymbol{\theta}}_{0,K}$.
   - **AP refinement step.** This sub-step is iterative and, in each iteration, performs one line search for each component of the previous iteration's output $\hat{\boldsymbol{\theta}}_{k-1}$ independently. In other words, if $\hat{\boldsymbol{\theta}}_{r,k,\theta}$ denotes $\hat{\boldsymbol{\theta}}_r$ but with its kth component replaced with the variable θ, then the next iteration output is given by ($r \geq 0$, $k = 1, 2, \ldots, K$)

     $$[\hat{\boldsymbol{\theta}}_{r+1}]_k = \arg\max_{\theta} L_D(\hat{\boldsymbol{\theta}}_{r,k,\theta}, \boldsymbol{\lambda}_o). \tag{20}$$
2. If $\boldsymbol{\theta}_0$ is the output of step 1) and is close to the true value of $\boldsymbol{\theta}$, then we may expect

   $$\boldsymbol{R}_z \approx \boldsymbol{\Phi}_o(\boldsymbol{\theta}_0)\boldsymbol{S}\boldsymbol{S}^H\boldsymbol{\Phi}_o(\boldsymbol{\theta}_0)^H + \boldsymbol{\Lambda}^{-2}$$

   and, therefore, the columns of $\boldsymbol{R}_z - \boldsymbol{\Lambda}^{-2}$ approximately lie in the span of $\boldsymbol{\Phi}_o(\boldsymbol{\theta}_0)$. This implies that their projection onto the orthogonal complement to this last span is approximately zero, i.e,

   $$(\boldsymbol{I}_M - \boldsymbol{P}_o(\boldsymbol{\theta}_0))(\boldsymbol{R}_z - \boldsymbol{\Lambda}^{-2}) \approx \boldsymbol{0}.$$

   Thus, we may estimate $\boldsymbol{\lambda}$ as the vector minimising the Frobenius norm of this last matrix, given by

   $$\mathrm{tr}\Big\{(\boldsymbol{I}_M - \boldsymbol{P}_o(\boldsymbol{\theta}_0))(\boldsymbol{R}_z - \boldsymbol{\Lambda}^{-2})(\boldsymbol{R}_z - \boldsymbol{\Lambda}^{-2})^H\Big\}.$$

   As can be readily checked [18], the resulting estimate of $\boldsymbol{\lambda}$ is, $(m = 1, 2, \ldots, M)$,

   $$[\boldsymbol{\lambda}_0]_m = \sqrt{\frac{1 - [\boldsymbol{P}_o(\boldsymbol{\theta}_0)]_{m,m}}{\mathrm{Re}\{[\boldsymbol{R}_z(\boldsymbol{I}_M - \boldsymbol{P}_o(\boldsymbol{\theta}_0))]_{m,m}\}}}. \tag{21}$$

These two procedures can be used to approach the maximum of $L_D(\boldsymbol{\theta}, \boldsymbol{\lambda})$ by means of an iterative procedure, in which 1) and 2) are performed alternatively. In [18], this procedure is simplified further by substituting (21) into the alternative cost function

$$L_{Dmd}(\boldsymbol{\theta}, \boldsymbol{\lambda}) \equiv -N\mathrm{tr}\{(\boldsymbol{I}_M - \boldsymbol{P})\boldsymbol{R}_{z\lambda}\}, \tag{22}$$

which is identical to L_D but neglecting the term $\log|\boldsymbol{\Lambda}|$. The result of this substitution is a, so called, PD cost function, given by

$$L_{PD}(\boldsymbol{\theta}) \equiv -\sum_{m=1}^{M}\left(\boldsymbol{r}_m^H(\boldsymbol{I}_M - \boldsymbol{P}_o)\boldsymbol{r}_m - \frac{\mathrm{Re}\{[\boldsymbol{I}_M - \boldsymbol{P}_o]_{m,\cdot}\boldsymbol{r}_m\}}{[\boldsymbol{I}_M - \boldsymbol{P}_o]_{m,m}}\right), \tag{23}$$

where $\boldsymbol{r}_m \equiv [\boldsymbol{R}_z]_{\cdot,m}$.

4. Limitations of the state of the art two-step approach and outline of the proposed method

The computation of UNP ML estimates consists of locating the global maximum of either L_D and L_S and this process has two fundamental stages. In the first, the global maximum is coarsely located by avoiding the convergence to any local maximum and, in the second stage, the coarse location just obtained is refined until the global maximum is attained. The state of the art approach in the previous section performs these two stages in the same way, i.e, by ascending to the global maximum through searches performed in either a single parameter (AOA θ) or a parameter subset (vector $\boldsymbol{\lambda}$). This is probably the best approach for the first stage, given that it avoids the convergence to any local minimum due to the line search in (19). However, once a coarse location $[\boldsymbol{\theta}_0; \boldsymbol{\lambda}_0]$ is known, there is additional information available, namely, the fact that the cost function's shape is approximately known around $[\boldsymbol{\theta}_0; \boldsymbol{\lambda}_0]$, and this additional information can be exploited by a suitable method, such as a Newton method [21, Ch. 5], [25,26]. To be more precise, if $[\boldsymbol{\theta}_0; \boldsymbol{\lambda}_0]$ is close to the global maximum position, say position $[\boldsymbol{\theta}; \boldsymbol{\lambda}]$, then a second-order Taylor expansion around $[\boldsymbol{\theta}_0; \boldsymbol{\lambda}_0]$ may have some accuracy at $[\boldsymbol{\theta}; \boldsymbol{\lambda}]$; that is, if L denotes either L_D or L_S, we have the approximation

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\lambda}) \approx {} & L(\boldsymbol{\theta}_0, \boldsymbol{\lambda}_0) + \boldsymbol{g}(\boldsymbol{\theta}_0, \boldsymbol{\lambda}_0)^T\begin{bmatrix}\boldsymbol{\theta} - \boldsymbol{\theta}_0 \\ \boldsymbol{\lambda} - \boldsymbol{\lambda}_0\end{bmatrix} \\ & + \tfrac{1}{2}[\boldsymbol{\theta} - \boldsymbol{\theta}_0; \boldsymbol{\lambda} - \boldsymbol{\lambda}_0]^T\boldsymbol{H}(\boldsymbol{\theta}_0, \boldsymbol{\lambda}_0)\begin{bmatrix}\boldsymbol{\theta} - \boldsymbol{\theta}_0 \\ \boldsymbol{\lambda} - \boldsymbol{\lambda}_0\end{bmatrix}, \end{aligned} \tag{24}$$

where $\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{\lambda})$ and $\boldsymbol{H}(\boldsymbol{\theta}, \boldsymbol{\lambda})$ denote the gradient and Hessian of $L(\boldsymbol{\theta}, \boldsymbol{\lambda})$ respectively. This implies that we may approach the position of the global maximum by iteratively maximising (24) or, in other words, by performing Newton iterations of the form ($r = 0, 1\ldots$)

$$\begin{bmatrix}\boldsymbol{\theta}_{r+1} \\ \boldsymbol{\lambda}_{r+1}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\theta}_r \\ \boldsymbol{\lambda}_r\end{bmatrix} - \mu_r\boldsymbol{H}^{-1}(\boldsymbol{\theta}_r, \boldsymbol{\lambda}_r)\,\boldsymbol{g}(\boldsymbol{\theta}_r, \boldsymbol{\lambda}_r), \tag{25}$$

where $[\boldsymbol{\theta}_r; \boldsymbol{\lambda}_r]$ is the current iterate and the parameter μ_r is usually set to 1 but may be between 0 and 1 in case there is no increase in the cost function value; (see [21, Ch. 6] for further details on Newton's method.).

This Newton method improves on the state of the art approach for the following two reasons. First, (25) involves no repeated one-dimensional maximisations such as (20). And second, (25) converges quadratically to the global maximum [21, Sec. 6.3.1] and this implies, in practice, that a small number of iterations is sufficient. However, a drawback is that (25) requires the value of the gradient and Hessian of either L_D or L_S and this may, in principle, involve a high computational burden. In this paper, we overcome this drawback by presenting closed-form formulas of the gradient and Hessian of both L_D and L_S in Section 6, that make it possible to obtain the values of these differentials in a small number of flops. These formulas have been derived following the approach in [27], [28, Ch. 5].

One unexpected spin-off of these closed-form formulas is the analysis in Section 7, where it is shown that the DML estimator is either inconsistent or degenerate, i.e, either its performance fails to improve with the number of snapshots or the DML cost function has no global maximum. This is a drawback that is readily observed in the numerical examples (Section 8.1): the Newton iterations for the DML cost function produce diverging values of some component of $\boldsymbol{\lambda}$.

In the next section, we present the proposed method in detail and then the expressions of the gradients and Hessians in Section 6.

5. Proposed method

The proposed method consists of the following steps:

1.1 Computation of coarse estimate of $\boldsymbol{\theta}$, denoted $\boldsymbol{\theta}'_0$.
1.2 Computation of coarse estimate of $\boldsymbol{\lambda}$, denoted $\boldsymbol{\lambda}'_0$.
2 Refinement of $[\boldsymbol{\theta}'_0; \boldsymbol{\lambda}'_0]$ using a Newton method.

We explain these steps in corresponding sub-sections.

5.1. Step 1.1: Initial DOA estimates θ_0

This sub-step computes the argument $\boldsymbol{\theta}_0$ that globally maximises L_{Do} in (9). $\boldsymbol{\theta}_0$ will be the input argument of Step 2. The method employed for obtaining this last vector is the AP initial step in Section 3, but complemented with a Newton refinement after each angle is added. Specifically, starting with an empty vector $\boldsymbol{\theta}'_0$, $\boldsymbol{\theta}'_{k+1}$ is constructed from $\boldsymbol{\theta}'_k$ in the following two sub-steps,

1. **Add angle.** The method finds out the approximate location of the global maximum of $L_{Do}([\boldsymbol{\theta}'_k; \theta])$ through a line search in θ, and appends the corresponding abscissa to $\boldsymbol{\theta}'_k$, in order to form a new vector $\boldsymbol{\theta}'_{k+1,o}$, $(\boldsymbol{\theta}'_{k+1,o} = [\boldsymbol{\theta}'_{k,o}; \theta])$.
2. **Refinement.** A Newton refinement of the form in (25) is applied to $\boldsymbol{\theta}'_{k+1,o}$. Specifically, with initial vector $\boldsymbol{\theta}'_{k+1,o,0} = \boldsymbol{\theta}'_{k+1,o}$, the corresponding Newton iteration is $(r = 0, 1, \ldots)$,

$$\boldsymbol{\theta}'_{k,o,r+1} = \boldsymbol{\theta}'_{k,o,r} - \mu_{k,r} \boldsymbol{H}^{-1}_{Do}(\boldsymbol{\theta}'_{k,o,r}) \boldsymbol{g}_{Do}(\boldsymbol{\theta}'_{k,o,r}), \tag{26}$$

where $\mu_{k,r}$ is the iteration's scale factor, and $\boldsymbol{g}_{Do}(\boldsymbol{\theta}_{k,o,r})$ and $\boldsymbol{H}_{Do}(\boldsymbol{\theta}_{k,o,r})$ are the gradient and Hessian (or Hessian approximation) of L_{Do} respectively. (26) is repeated until $\|\boldsymbol{\theta}'_{k,o,r+1} - \boldsymbol{\theta}'_{k,o,r}\|$ is sufficiently small. Then, the last iteration's vector is the output $\boldsymbol{\theta}'_k$.

1) and 2) are repeated K times and the final output $\boldsymbol{\theta}'_K$ is the input to Step 2, $\boldsymbol{\theta}_0 = \boldsymbol{\theta}'_K$.

5.2. Step 1.2: Initial noise parameter estimates λ_0

In this sub-step, we employ the estimate of $\boldsymbol{\lambda}$ in (21) presented in [18].

5.3. Step 2: Newton refinement

In this final step, we refine the initial estimate $[\boldsymbol{\theta}_0; \boldsymbol{\lambda}_0]$ through the Newton iterations in (25). They are computed until $\|[\boldsymbol{\theta}_{r+1}; \boldsymbol{\lambda}_{r+1}] - [\boldsymbol{\theta}_r; \boldsymbol{\lambda}_r]\|$ is sufficiently small. The final vector $[\boldsymbol{\theta}_{r+1}; \boldsymbol{\lambda}_{r+1}]$ is the corresponding DML or SML estimate.

6. Gradients and Hessians of the DML and SML cost functions

For simplicity, we introduce the gradients and Hessians of L_D and L_C, given that the corresponding differentials of L_S can be readily obtained through the equation $L_S = L_D + L_C$. We let $\boldsymbol{g}_D$ and $\boldsymbol{g}_C$ denote the gradients and $\boldsymbol{H}_D$ and $\boldsymbol{H}_C$ the Hessians of L_D and L_C respectively. Besides, we divide all these differentials into blocks corresponding to the $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ vectors. Thus, for the gradient $\boldsymbol{g}_D$ we have the block structure

$$\boldsymbol{g}_D \equiv \begin{bmatrix} \boldsymbol{g}_{D\theta} \\ \boldsymbol{g}_{D\lambda} \end{bmatrix}, \qquad \boldsymbol{g}_C \equiv \begin{bmatrix} \boldsymbol{g}_{C\theta} \\ \boldsymbol{g}_{C\lambda} \end{bmatrix}.$$

and the corresponding expressions are

$$\begin{aligned} \boldsymbol{g}_{D\theta} &= 2N\mathrm{Re}\Big\{\mathrm{diag}\Big\{\boldsymbol{\Phi}^{\dagger}\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M - \ \boldsymbol{P})\boldsymbol{D}\Big\}\Big\} \\ g_{D\lambda} &= 2N\boldsymbol{\Lambda}^{-1}\mathrm{diag}\{\boldsymbol{I}_M - (\boldsymbol{I}_M - \boldsymbol{P})\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M - \boldsymbol{P})\} \\ g_{C\theta} &= -\ 2N\mathrm{Re}\Big\{\mathrm{diag}\Big\{\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M - \boldsymbol{P})\boldsymbol{D}\Big\}\Big\} \\ g_{C\lambda} &= 2N\boldsymbol{\Lambda}^{-1}\mathrm{Re}\{\mathrm{diag}\{\boldsymbol{P} - 2\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\}\}, \end{aligned} \tag{27}$$

where

$$[\boldsymbol{D}]_{\cdot,k} \equiv \boldsymbol{\Lambda}\frac{\partial}{\partial\theta_k}\boldsymbol{\phi}_o(\theta_k), \quad k = 1, \ldots, K.$$

The derivation of $\boldsymbol{g}_{D\lambda}$ and $\boldsymbol{g}_{C\lambda}$ can be found in Appendix B and that of $\boldsymbol{g}_{D\theta}$ and $\boldsymbol{g}_{C\theta}$ in the complementary material.

Regarding the Hessians, we have the structure

$$\boldsymbol{H}_D \equiv \begin{bmatrix} \boldsymbol{H}_{D\theta\theta} & \boldsymbol{H}_{D\theta\lambda} \\ \boldsymbol{H}^T_{D\theta\lambda} & \boldsymbol{H}_{D\lambda\lambda} \end{bmatrix}, \quad \boldsymbol{H}_C \equiv \begin{bmatrix} \boldsymbol{H}_{C\theta\theta} & \boldsymbol{H}_{C\theta\lambda} \\ \boldsymbol{H}^T_{C\theta\lambda} & \boldsymbol{H}_{C\lambda\lambda} \end{bmatrix}.$$

The expressions of these blocks are in Appendix A and the corresponding derivations are in the complementary material.

7. Consistency and non-degeneracy of the DML and SML estimators

Let us consider first the value of $\boldsymbol{R}_z$ for high N and with high probability. This value is approximately

$$\boldsymbol{R}_z \approx \boldsymbol{\Phi}_o\boldsymbol{R}_s\boldsymbol{\Phi}_o + \boldsymbol{\Lambda}^{-2}, \tag{28}$$

where the right hand side is evaluated at the true values of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$; and where $\boldsymbol{R}_s$ is the signal covariance matrix for the SML estimator, or the asymptotic covariance

$$\boldsymbol{R}_s = \lim_{N\to\infty}\frac{1}{N}\boldsymbol{S}\boldsymbol{S}^H.$$

for the DML estimator. (We assume the existence of this limit. Note that the number of columns of $\boldsymbol{S}$ is N.)

We have the following two desirable properties of both the DML and SML estimators for high N and with high probability,

1. **Consistency.** The DML or SML estimate is close to the true values of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$. Therefore, (28) also holds if its right-hand side is evaluated at the estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ rather than at the true values of these vectors.
2. **Non-degeneracy.** The DML or SML estimate corresponds to a critical point of either L_D or L_S, i.e, we either have

$$\boldsymbol{g}_{D\theta}(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \boldsymbol{0}, \quad \boldsymbol{g}_{D\lambda}(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \boldsymbol{0} \tag{29}$$

or

$$\boldsymbol{g}_{S\theta}(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \boldsymbol{0}, \quad \boldsymbol{g}_{S\lambda}(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \boldsymbol{0}. \tag{30}$$

Now we can show that if 1) is true then 2) is false for the DML estimator, while if 1) is true then 2) is also true for the SML estimator. For this, let us assume 1) and then prove that all equations in condition 2) hold except for the second one in (29), i.e, $\boldsymbol{g}_{D\lambda} \neq \boldsymbol{0}$.

First, note that the expressions of $\boldsymbol{g}_{D\theta}$ and $\boldsymbol{g}_{C\theta}$ in (27) are formed by products containing the factor

$$\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M - \boldsymbol{P}). \tag{31}$$

But (28) implies the approximation

$$\boldsymbol{R}_{z\lambda} = \boldsymbol{\Lambda}\boldsymbol{R}_z\boldsymbol{\Lambda} \approx \boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{\Phi}^H + \boldsymbol{I}_M, \tag{32}$$

and we can easily combine the property $(\boldsymbol{I}_M - \boldsymbol{P})\boldsymbol{\Phi} = \boldsymbol{0}$ with this last approximation to show that (31) is approximately zero. So, we deduce $\boldsymbol{g}_{D\theta} \approx 0$ and $\boldsymbol{g}_{C\theta} \approx 0$ and in turn

$$\boldsymbol{g}_{S\theta} = \boldsymbol{g}_{D\theta} + \boldsymbol{g}_{C\theta} \approx \boldsymbol{0}.$$

Thus, the first equations of (29) and (30) hold.

Second, let us check whether $\boldsymbol{g}_{D\lambda} \approx \boldsymbol{0}$. For this, substitute (32) into the expression for $\boldsymbol{g}_{D\lambda}$ in (27) and use $(\boldsymbol{I}_M - \boldsymbol{P})\boldsymbol{\Phi} = \boldsymbol{\Phi}$ and $(\boldsymbol{I}_M - \boldsymbol{P})^2 = \boldsymbol{I}_M - \boldsymbol{P}$. We have

$$\begin{aligned} \boldsymbol{g}_{D\lambda} &\approx 2N\mathrm{diag}\big\{\boldsymbol{\Lambda}^{-1} \\ &\quad -\boldsymbol{\Lambda}^{-1}(\boldsymbol{I}_M - \boldsymbol{P})(\boldsymbol{I}_M + \boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{\Phi}^H)(\boldsymbol{I}_M - \boldsymbol{P})\big\} \\ &= 2N\mathrm{diag}\{\boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-1}(\boldsymbol{I}_M - \boldsymbol{P})\} \\ &= 2N\mathrm{diag}\{\boldsymbol{\Lambda}^{-1}\boldsymbol{P}\}. \end{aligned} \tag{33}$$

If $\boldsymbol{Q}$ denotes a matrix whose columns are an ortho-normal basis for the span of $\boldsymbol{\Phi}$, then $\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{Q}^H$ and the mth component of (33) can be expressed as

$$[\boldsymbol{g}_{D\lambda}]_m \approx 2N\frac{\|[\boldsymbol{Q}]_{\cdot,m}\|^2}{\lambda_m} \geq 0.$$

Since $\|[\boldsymbol{Q}]_{\cdot,m}\|^2$ is positive for at least one index m, we have $\boldsymbol{g}_{D\lambda} \neq \boldsymbol{0}$. This proves that 1) and 2) are incompatible for the DML estimator.

And finally, let us check whether $\boldsymbol{g}_{S\lambda} \approx \boldsymbol{0}$. For this, we need to consider the expression of $\boldsymbol{g}_{C\lambda}$ in (27), but let first us prove that $\boldsymbol{P}_z\boldsymbol{R}_{z\lambda} \approx \boldsymbol{P}$. We have

$$\begin{aligned} \boldsymbol{P}_z\boldsymbol{R}_{z\lambda} &= \boldsymbol{\Phi}(\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda} \\ &\approx \boldsymbol{\Phi}(\boldsymbol{\Phi}^H(\boldsymbol{I}_M + \boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{\Phi}^H)\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^H(\boldsymbol{I}_M + \boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{\Phi}^H) \\ &= \boldsymbol{\Phi}(\boldsymbol{\Phi}^H\boldsymbol{\Phi} + \boldsymbol{\Phi}^H\boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{\Phi}^H\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^H(\boldsymbol{I}_M + \boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{\Phi}^H) \\ &= \boldsymbol{\Phi}((\boldsymbol{I}_K + \boldsymbol{\Phi}^H\boldsymbol{\Phi}\boldsymbol{R}_s)\boldsymbol{\Phi}^H\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^H(\boldsymbol{I}_M + \boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{\Phi}^H) \\ &= \boldsymbol{\Phi}(\boldsymbol{\Phi}^H\boldsymbol{\Phi})^{-1}(\boldsymbol{I}_K + \boldsymbol{\Phi}^H\boldsymbol{\Phi}\boldsymbol{R}_s)^{-1}\boldsymbol{\Phi}^H(\boldsymbol{I}_M + \boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{\Phi}^H) \\ &= \boldsymbol{\Phi}(\boldsymbol{\Phi}^H\boldsymbol{\Phi})^{-1}(\boldsymbol{I}_K + \boldsymbol{\Phi}^H\boldsymbol{\Phi}\boldsymbol{R}_s)^{-1}(\boldsymbol{\Phi}^H\boldsymbol{\Phi}\boldsymbol{R}_s\boldsymbol{\Phi}^H + \boldsymbol{\Phi}^H) \\ &= \boldsymbol{\Phi}(\boldsymbol{\Phi}^H\boldsymbol{\Phi})^{-1}(\boldsymbol{I}_K + \boldsymbol{\Phi}^H\boldsymbol{\Phi}\boldsymbol{R}_s)^{-1}(\boldsymbol{I}_K + \boldsymbol{\Phi}^H\boldsymbol{\Phi}\boldsymbol{R}_s)\boldsymbol{\Phi}^H \\ &= \boldsymbol{\Phi}(\boldsymbol{\Phi}^H\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^H = \boldsymbol{P}. \end{aligned} \tag{34}$$

Now, let us operate on the expression of $\boldsymbol{g}_{C\lambda}$ in (27) assuming this last approximation, recalling that $\boldsymbol{R}_{z\lambda} = \boldsymbol{\Lambda}\boldsymbol{R}_z\boldsymbol{\Lambda}$, and using the property $\mathrm{diag}\{\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\} = \mathrm{diag}\{\boldsymbol{\Lambda}^{-1}\boldsymbol{P}\}$. We have

$$\begin{aligned} \boldsymbol{g}_{C\lambda} &= 2N\Big(\mathrm{Re}\Big\{\mathrm{diag}\Big\{\boldsymbol{P}\boldsymbol{\Lambda}^{-1} - 2\boldsymbol{R}_z\boldsymbol{\Lambda}\boldsymbol{P}_z\Big\}\Big\}\Big) \\ &= 2N\Big(\mathrm{Re}\Big\{\mathrm{diag}\Big\{\boldsymbol{P}\boldsymbol{\Lambda}^{-1} - 2\boldsymbol{\Lambda}^{-1}\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\Big\}\Big\}\Big) \\ &\approx 2N\Big(\mathrm{Re}\Big\{\mathrm{diag}\Big\{\boldsymbol{P}\boldsymbol{\Lambda}^{-1} - 2\boldsymbol{\Lambda}^{-1}\boldsymbol{P}\Big\}\Big\}\Big) \\ &= 2N\Big(\mathrm{Re}\Big\{\mathrm{diag}\Big\{\boldsymbol{\Lambda}^{-1}\boldsymbol{P} - 2\boldsymbol{\Lambda}^{-1}\boldsymbol{P}\Big\}\Big\}\Big) \\ &= -2N\mathrm{diag}\Big\{\boldsymbol{\Lambda}^{-1}\boldsymbol{P}\Big\}. \end{aligned}$$

Therefore, from (33), we have

$$\boldsymbol{g}_{S\lambda} = \boldsymbol{g}_{D\lambda} + \boldsymbol{g}_{C\lambda} \approx \boldsymbol{0}.$$

So, we conclude that 1) implies 2) for the SML estimator.

8. Numerical examples

We have assessed the proposed method for the L_D and L_S cost functions numerically in the following scenario:

- **Received signals.** The received signals were uncorrelated complex Gaussian process of equal variance and there were $K = 2$ of them.
- **Sensor array.** Uniform linear array formed by $M = 10$ sensors with half-wavelength spacing.
- **Angles of arrival.** The angles of arrival were $\theta_1 = 7^\circ$ and $\theta_2 = 13^\circ$ relative to the broadside.
- **Sensor noise inverse deviations.** They were the following:

$$\boldsymbol{\Lambda} = \mathrm{diag}([10, 2, 1.5, 0.5, 8, 0.7, 1.1, 3, 6, 3])^{-1/2}. \tag{35}$$

- **Initial estimate.** In all simulations, the initial estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ were computed using the AP algorithm followed by the estimator in ().
- **Refinement methods.** We tested the following estimators,
    - **MUSIC.** Multiple Signal Classification estimator.
    - **DMLo.** DML estimator for uniform noise. This is the output of Step 1.1 of the proposed method.
    - **Chen.** SML estimator computed using the method in [9].
    - **MDS.** Approximate DML estimator in [18]
    - **DML-alt.** Proposed method for the DML cost function but alternating the iterations in $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$.
    - **DML.** Proposed method for the DML cost function.
    - **SML.** Proposed method for the SML cost function using the full Hessian.
- **Number of Monte Carlo trials.** We performed 500 Monte Carlo trials.

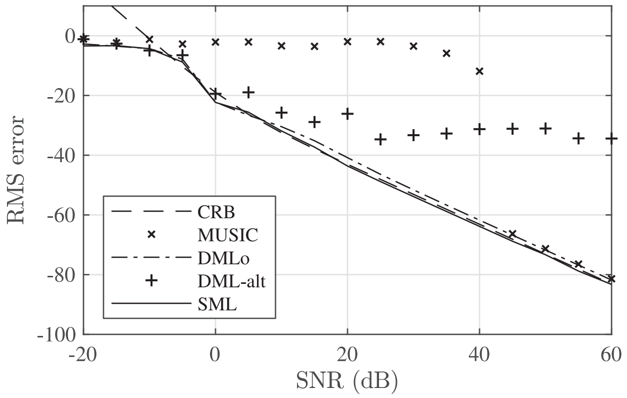


Fig. 1. RMS error performance of MUSIC, DMLo, DML-alt, and SML for uncorrelated signals.

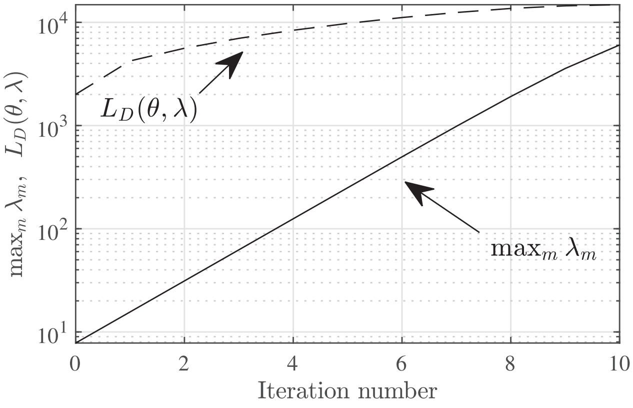
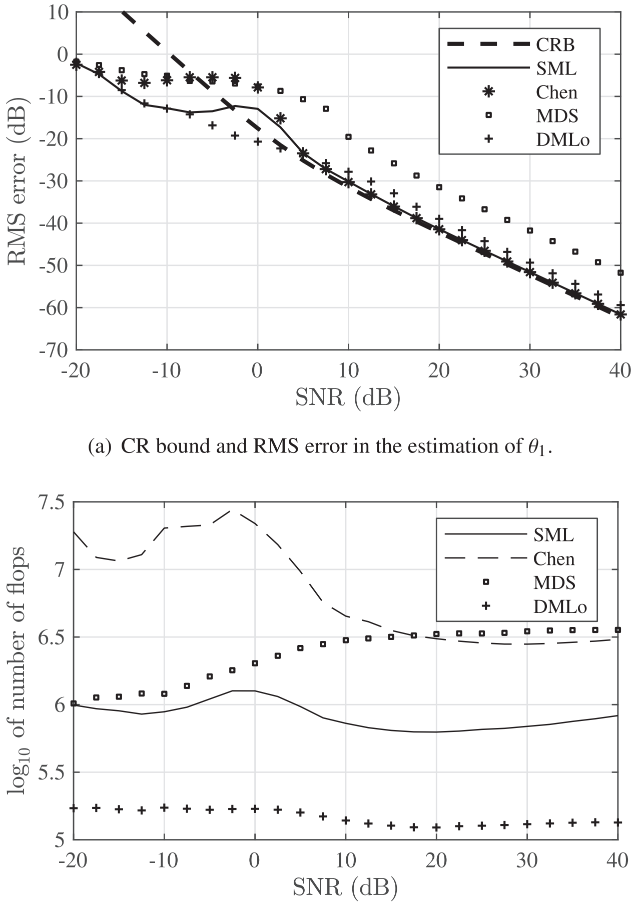


Fig. 2. Value of $L_D(\boldsymbol{\theta}, \boldsymbol{\lambda})$ and λ_m. The x-axis variable is the iteration number in Newton's method but applied to the $\boldsymbol{\lambda}$ vector of $L_D(\boldsymbol{\theta}, \boldsymbol{\lambda})$ only. $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ are initialised with the true values of these vectors and SNR = 40 dB.

8.1. Degeneracy of the DML estimator

Fig. 1 shows the RMS error performance of MUSIC, DMLo, DML-alt, and SML. We can see that DML-alt reaches an RMS error floor above the Cramer-Rao bound (CRB), while MUSIC becomes close to the Cramer-Rao (CR) bound only at high SNRs. Finally, SML reaches the CR bound at intermediate and high SNRs. In this figure, DML is missing because Newton's method produces a divergent $\boldsymbol{\lambda}$ estimate and, therefore, DML is unavailable. This can be explained by the problem related with the DML cost function already commented in Section 7.

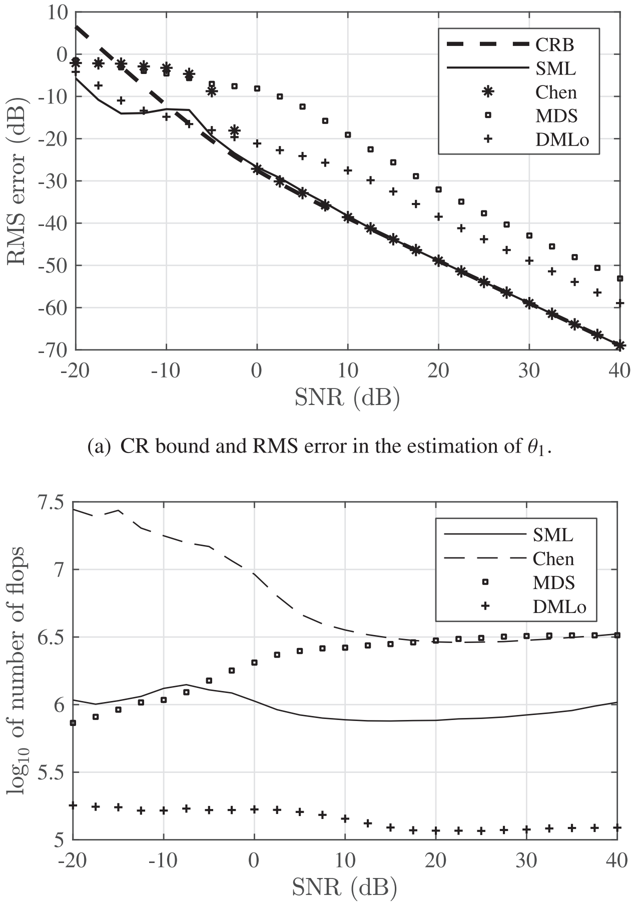
Fig. 2 shows this phenomenon for a specific realisation in which Newton's method is initialised with the true values of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ and SNR = 40 dB. We can see in this figure that the New-



(a) CR bound and RMS error in the estimation of θ_1.

(b) Average number of flops required for the computation of the estimators in Fig. 3(a).

Fig. 3. RMS error and average number of flops for several estimators.



(a) CR bound and RMS error in the estimation of θ_1.

(b) Average number of flops required for the computation of the estimators in Fig. 4(a).

Fig. 4. RMS error and average number of flops for several estimators and noise parameters in (35).

ton iteration achieves an increase in the cost function value every time, but $\max_m \lambda_m$ diverges, i.e, at least one sensor noise power is taken as zero approximately. Obviously, this is a degenerate result.

8.2. Comparison with other estimators

Fig. 3 (a) shows the RMS in the estimation of θ_1 for several estimators. Note that SML and Chen have the best performance, while the RMS error of DMLo is slightly above the error of SML and Chen. MDS's error is above the CR bound by roughly 10 dB at mid to high SNRs. Fig. (b) shows the average computational burden of the estimators in 3(a) measured in flops. Note that the computational burden of Chen is between factor 3.7 and 22.9 larger than that of SML; (difference between 0.56 and 1.36 in log-10 scale). Finally, though the RMS error of DMLo is slightly larger than that of SML and Chen [Fig. 3(a)], its computational burden is far smaller than that of SML and Chen.

The difference in performance between DMLo and SML becomes more significant if the disparity in the noise powers is increased. Figs. 4(a) and 35(b) are the equivalents of Figs. 3(a) and 3(b) for $\boldsymbol{\lambda}$ given by

$$\boldsymbol{\Lambda} = \mathrm{diag}([80, 2, 1.5, 0.5, 64, 0.7, 1.1, 3, 48, 3])^{-1/2}, \qquad (36)$$

where the noise variances are the same as in (34), except components 1, 5 and 9, which have been multiplied by 8. Note that SML and Chen outperform DMLo by roughly 10 dB.

9. Conclusions

We have presented an efficient method for computing maximum likelihood (ML) estimates of the directions of arrival (DOA) to an array of sensors under unknown nonuniform sensor-noise powers. The method is based on the Newton method combined with an initial Alternate Projection (AP) search. Its efficiency is based on three factors. First, the Newton iterations converge in a small number of iterations. Second, they involve no one-dimensional searches. And third, the gradient and Hessian required in each iteration can be obtained with low complexity thanks to the closed-form formulas for these differentials presented in this paper. The analysis of the deterministic ML (DML) cost function's gradient reveals an unexpected drawback of the corresponding estimator: it is either degenerate or inconsistent. The method is assessed in the paper numerically and compared with other methods in the literature.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Declaration of Competing Interest

None.

Data availability

Data will be made available on request.

Appendix A. Hessian expressions

For introducing the Hessian blocks, define first the matrix

$$[\boldsymbol{D}_2]_{\cdot,k} \equiv \boldsymbol{\Lambda}\frac{\partial^2}{\partial\theta_k^2}\boldsymbol{\phi}_o(\theta_k),\ k=1,\ldots,K.$$

The blocks are the following:

$$\begin{aligned}\boldsymbol{H}_{D\theta\theta} = 2N\mathrm{Re}\Big\{ & \\ & \boldsymbol{M}\odot(\boldsymbol{D}^H(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D})^T \\ & -(\boldsymbol{\Phi}^\dagger\boldsymbol{D})\odot(\boldsymbol{\Phi}^\dagger\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D})^T \\ & -(\boldsymbol{\Phi}^\dagger\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D})\odot(\boldsymbol{\Phi}^\dagger\boldsymbol{D})^T \\ & -(\boldsymbol{\Phi}^\dagger\boldsymbol{R}_{z\lambda}(\boldsymbol{\Phi}^\dagger)^H)\odot(\boldsymbol{D}^H(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D})^T \\ & +\boldsymbol{I}_K\odot(\boldsymbol{\Phi}^\dagger\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D}_2)^T\Big\}.\end{aligned} \tag{A.1}$$

$$\begin{aligned}\boldsymbol{H}_{D\theta\lambda} = 4N\mathrm{Re}\Big\{&(\boldsymbol{\Phi}^\dagger\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P}))\odot((\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D})^T \\ &+(\boldsymbol{D}^H(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P}))\odot(\boldsymbol{\Phi}^\dagger)^*\Big\}\boldsymbol{\Lambda}^{-1},\end{aligned} \tag{A.2}$$

$$\begin{aligned}&\boldsymbol{H}_{D\lambda\lambda} = 2N\boldsymbol{\Lambda}^{-1}\Big(\\ &\mathrm{Re}\Big\{(4\boldsymbol{P}-\boldsymbol{I}_M)\odot((\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P}))^T\Big\} \\ &\qquad\qquad -\boldsymbol{I}_M\Big)\boldsymbol{\Lambda}^{-1},\end{aligned} \tag{A.3}$$

$$\begin{aligned}\boldsymbol{H}_{C\theta\theta} = 2N\mathrm{Re}\Big\{&(\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D})\odot(\boldsymbol{\Phi}^\dagger\boldsymbol{D})^T \\ &+\boldsymbol{M}\odot(\boldsymbol{D}^H(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D})^T \\ &-\boldsymbol{I}_K\odot(\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D}_2)^T \\ &-\boldsymbol{M}_{z\lambda}\odot(\boldsymbol{D}^H(\boldsymbol{I}_M-\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z)\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D})^T \\ &+(\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}\boldsymbol{D})\odot(\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}^H\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{D})^T\Big\},\end{aligned} \tag{A.4}$$

$$\begin{aligned}&\boldsymbol{H}_{C\theta\lambda} = 4N\mathrm{Re}\Big\{(\boldsymbol{D}^H(\boldsymbol{I}_M-\boldsymbol{P}))\odot(\boldsymbol{\Phi}^\dagger)^* \\ &(\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}^H)\odot(\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P}_z\boldsymbol{R}_{z\lambda})\boldsymbol{D})^T \\ &(\boldsymbol{D}^H(\boldsymbol{I}_M-\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z))\odot(\boldsymbol{R}_{z\lambda}\boldsymbol{\Phi}\boldsymbol{M}_{z\lambda})^T\Big\}\boldsymbol{\Lambda}^{-1},\end{aligned} \tag{A.5}$$

$$\begin{aligned}\boldsymbol{H}_{C\lambda\lambda} = 2N\boldsymbol{\Lambda}^{-1}\mathrm{Re}\Big\{&(\boldsymbol{I}_M-2\boldsymbol{P})\odot\boldsymbol{P}^T \\ &-4(\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P}_z\boldsymbol{R}_{z\lambda}))\odot\boldsymbol{P}_z^T \\ &-2(\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z)\odot(\boldsymbol{I}_M-2\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z)^T\Big\}\boldsymbol{\Lambda}^{-1}.\end{aligned} \tag{A.6}$$

The derivations of these blocks can be found in the complementary material.

Appendix B. Derivation of gradient blocks $\boldsymbol{g}_{D\lambda}$ and $\boldsymbol{g}_{C\lambda}$

In the sequel, we let x denote any of the components of $\boldsymbol{\lambda}$ and the sub-script $()_x$ denote differentiation in that variable. Thus, for instance, if x is λ_m then $\boldsymbol{P}_x$ denotes

$$\frac{\partial}{\partial\lambda_m}\boldsymbol{P}.$$

We require the following formulas,

- We repeatedly use the fact that the product of two diagonal matrices can be commuted, i.e, $\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x = \boldsymbol{\Lambda}_x\boldsymbol{\Lambda}^{-1}$.
- For a square invertible matrix $\boldsymbol{A}$, Jacobi's formula states that the derivative in a variable x of $\log|\boldsymbol{A}|$ is
$$(\log|\boldsymbol{A}|)_x = \mathrm{tr}\{\boldsymbol{A}^{-1}\boldsymbol{A}_x\}. \tag{B.1}$$
- The derivatives in x of $\boldsymbol{R}_{z\lambda}$ and $\boldsymbol{P}$ can be computed by means of the product derivative rule, and be concisely expressed as
$$\boldsymbol{R}_{z\lambda,x} = \boldsymbol{\Lambda}_x\boldsymbol{\Lambda}^{-1}\boldsymbol{R}_{z\lambda}+\boldsymbol{R}_{z\lambda}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x, \tag{B.2}$$
$$\boldsymbol{P}_x = \boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x+\boldsymbol{\Lambda}_x\boldsymbol{\Lambda}^{-1}\boldsymbol{P}-2\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\boldsymbol{P}. \tag{B.3}$$
- The product $\boldsymbol{P}_z\boldsymbol{P}_x$ can be concisely written in terms of $\boldsymbol{\Lambda}_x$ through orthogonality properties. Specifically, since $\boldsymbol{P}_z\boldsymbol{P} = \boldsymbol{P}_z$, we have
$$\begin{aligned}\boldsymbol{P}_z\boldsymbol{P}_x &= (\boldsymbol{P}_z\boldsymbol{P})_x-\boldsymbol{P}_{z,x}\boldsymbol{P} = \boldsymbol{P}_{z,x}(\boldsymbol{I}-\boldsymbol{P}) \\ &= (\boldsymbol{\Phi}_x\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}^H+\boldsymbol{\Phi}\boldsymbol{M}_{z\lambda,x}\boldsymbol{\Phi}^H+\boldsymbol{\Phi}\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}_x^H)(\boldsymbol{I}-\boldsymbol{P}) \\ &= \boldsymbol{\Phi}\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}_x^H(\boldsymbol{I}-\boldsymbol{P}) = \boldsymbol{\Phi}\boldsymbol{M}_{z\lambda}\boldsymbol{\Phi}^H\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x(\boldsymbol{I}-\boldsymbol{P}) \\ &= \boldsymbol{P}_z\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x(\boldsymbol{I}-\boldsymbol{P}).\end{aligned} \tag{B.4}$$

B1. Gradient of L_D in λ, $g_{D\lambda}$

Let us derive the expression of $\boldsymbol{g}_{D\lambda}$. First, we differentiate (6) in x using (B.1),

$$L_{D,x} = N\Big(2\,\mathrm{tr}\{\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}+\mathrm{tr}\{\boldsymbol{P}_x\boldsymbol{R}_{z\lambda}\}-\mathrm{tr}\{(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{R}_{z\lambda,x}\}\Big). \tag{B.5}$$

Second, we substitute (B.3) and (B.2) into this last expression and expand the product with $(\boldsymbol{I}_M-\boldsymbol{P})$. The result of these operations is

$$\begin{aligned}L_{D,x} = N\Big(&2\mathrm{tr}\{\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}+\mathrm{tr}\{\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\boldsymbol{R}_{z\lambda}\} \\ &+\mathrm{tr}\{\boldsymbol{\Lambda}_x\boldsymbol{\Lambda}^{-1}\boldsymbol{P}\boldsymbol{R}_{z\lambda}\}-2\mathrm{tr}\{\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\boldsymbol{P}\boldsymbol{R}_{z\lambda}\} \\ &-\mathrm{tr}\{\boldsymbol{\Lambda}_x\boldsymbol{\Lambda}^{-1}\boldsymbol{R}_{z\lambda}\}-\mathrm{tr}\{\boldsymbol{R}_{z\lambda}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\} \\ &+\mathrm{tr}\{\boldsymbol{P}\boldsymbol{\Lambda}_x\boldsymbol{\Lambda}^{-1}\boldsymbol{R}_{z\lambda}\}+\mathrm{tr}\{\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}\Big).\end{aligned} \tag{B.6}$$

Third, we rotate the products inside the trace operators in order to get $\boldsymbol{\Lambda}_x$ on the right hand side. Besides, we use the property $\boldsymbol{\Lambda}_x\boldsymbol{\Lambda}^{-1} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x$. We obtain

$$\begin{aligned}L_{D,x} &= 2N\Big(\mathrm{tr}\{\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}+\mathrm{tr}\{\boldsymbol{R}_{z\lambda}\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\} \\ &\quad+\mathrm{tr}\{\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}-\mathrm{tr}\{\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}-\mathrm{tr}\{\boldsymbol{R}_{z\lambda}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}\Big) \\ &= 2N\Big(\mathrm{tr}\{\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}-\mathrm{tr}\{(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{R}_{z\lambda}(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}\Big).\end{aligned} \tag{B.7}$$

If x is one of the components of $\boldsymbol{\lambda}$, say λ_m, then $\boldsymbol{\Lambda}_x = \boldsymbol{\delta}_{M,m}\boldsymbol{\delta}_{M,m}^T$, and we have $\mathrm{tr}\{\boldsymbol{A}\boldsymbol{\Lambda}_x\} = [\boldsymbol{A}]_{m,m}$ for any matrix $\boldsymbol{A}$ of proper size. So, to obtain the gradient, we just need to replace $\mathrm{tr}\{\boldsymbol{A}\boldsymbol{\Lambda}_x\}$ with $\mathrm{diag}\{\boldsymbol{A}\}$ in (B.5) for every possible $\boldsymbol{A}$. The result of this operation is

$$\begin{aligned}\boldsymbol{g}_{D\lambda} &= 2N\mathrm{diag}\{-\boldsymbol{\Lambda}^{-1}(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})+\boldsymbol{\Lambda}^{-1}\} \\ &= 2N\boldsymbol{\Lambda}^{-1}\mathrm{diag}\{\boldsymbol{I}_M-(\boldsymbol{I}_M-\boldsymbol{P})\boldsymbol{R}_{z\lambda}(\boldsymbol{I}_M-\boldsymbol{P})\},\end{aligned}$$

which is the second formula in (27).

B2. Gradient of L_C in λ, $g_{C\lambda}$

First, differentiate (18) in x using (B.1), (14), and (15):

$$\begin{aligned}L_{C,x} &= (-N\log|\boldsymbol{C}|)_x = -N\mathrm{tr}\{\boldsymbol{C}^{-1}\boldsymbol{C}_x\} \\ &= -N\mathrm{tr}\{(\boldsymbol{I}_M-\boldsymbol{P}+\boldsymbol{P}_z)\{\boldsymbol{I}_M-\boldsymbol{P}+\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}\}_x\} \\ &= -N\mathrm{tr}\{(\boldsymbol{I}_M-\boldsymbol{P}+\boldsymbol{P}_z)\{-\boldsymbol{P}+\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}\}_x\}.\end{aligned} \tag{B.8}$$

From (B.3), it can be easily checked that $\{-\boldsymbol{P}+\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}\}_x$ is equal to a sum of terms whose row or column span lies in the span of $\boldsymbol{\Phi}$. This implies $\mathrm{tr}\{(\boldsymbol{I}_M-\boldsymbol{P})\{-\boldsymbol{P}+\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}\}_x\}=\boldsymbol{0}$ and, therefore, (B.6) simplifies to

$$L_{C,x} = -N\mathrm{tr}\{\boldsymbol{P}_z\{-\boldsymbol{P}+\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}\}_x\}.$$

Applying the product derivative rule, we have

$$\begin{aligned} L_{C,x} = {} & -N(-\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{P}_x\}+\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{P}_x\boldsymbol{R}_{z\lambda}\boldsymbol{P}\} \\ & +\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{P}\boldsymbol{R}_{z\lambda,x}\boldsymbol{P}\}+\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}_x\}). \end{aligned} \tag{B.9}$$

Next, we rotate the trace arguments, leaving the derivatives on the right, and apply the property

$$\boldsymbol{P}_z\boldsymbol{P} = \boldsymbol{P}\boldsymbol{P}_z = \boldsymbol{P}_z. \tag{B.10}$$

We obtain

$$\begin{aligned} L_{C,x} = {} & -N(-\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{P}_x\}+\mathrm{tr}\{\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{P}_x\} \\ & +\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{R}_{z\lambda,x}\})+\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{R}_{z\lambda}\boldsymbol{P}_x\}. \end{aligned} \tag{B.11}$$

At this point, the fact that $\boldsymbol{P}$, $\boldsymbol{P}_z$, and $\boldsymbol{R}_{z\lambda}$ are Hermitian implies that $-\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{P}_x\}$ and $\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{R}_{z\lambda,x}\}$ are real and $\mathrm{tr}\{\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{P}_x\}^*=\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{R}_{z\lambda}\boldsymbol{P}_x\}$. Using these two properties, we may write (B.8) more concisely as

$$L_{C,x} = -N\mathrm{Re}\Big\{-\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{P}_x\}\}+2\mathrm{tr}\{\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{P}_x\}\}+\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{R}_{z\lambda,x}\}\Big\}. \tag{B.12}$$

Now, the orthogonality properties and (B.4) imply $\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{P}_x\}=0$. So, we have

$$L_{C,x} = -N\mathrm{Re}\Big\{2\mathrm{tr}\{\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{P}_x\}\}+\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{R}_{z\lambda,x}\}\Big\}.$$

Next, we insert the formulas for $\boldsymbol{P}_x$ and $\boldsymbol{R}_{z\lambda,x}$ in (B.2) and (B.3),

$$\begin{aligned} L_{C,x} = {} & -N\mathrm{Re}\Big\{2\mathrm{tr}\Big\{\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z(\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x+\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\boldsymbol{P} \\ & -2\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\boldsymbol{P})\Big\}+\mathrm{tr}\Big\{\boldsymbol{P}_z(\boldsymbol{R}_{z\lambda}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x+\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\boldsymbol{R}_{z\lambda})\Big\}\Big\}. \end{aligned} \tag{B.13}$$

This expression can be readily expanded into a sum of trace terms. Then, rotating the trace arguments so that $\boldsymbol{\Lambda}_x$ appears on the right-hand side, applying (B.7) and noting that $\mathrm{tr}\{\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}^*=\mathrm{tr}\{\boldsymbol{P}_z\boldsymbol{R}_{z\lambda}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\}$, we obtain

$$\begin{aligned} L_{C,x} &= 2N\mathrm{Re}\Big\{\mathrm{tr}\Big\{\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\Big\}-2\mathrm{tr}\Big\{\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\Big\}\Big\} \\ &= 2N\mathrm{Re}\Big\{\mathrm{tr}\Big\{\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\Big\}-2\mathrm{tr}\Big\{\boldsymbol{\Lambda}^{-1}\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{\Lambda}_x\Big\}\Big\} \\ &= 2N\mathrm{Re}\Big\{\mathrm{tr}\Big\{\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\Big\}-2\mathrm{tr}\Big\{\boldsymbol{R}_z\boldsymbol{\Lambda}\boldsymbol{P}_z\boldsymbol{\Lambda}_x\Big\}\Big\}. \end{aligned}$$

Finally, noting that $\boldsymbol{P}\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z=\boldsymbol{P}$, we obtain

$$L_{C,x} = 2N\mathrm{Re}\Big\{\mathrm{tr}\Big\{\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x\Big\}-2\mathrm{tr}\Big\{\boldsymbol{R}_z\boldsymbol{\Lambda}\boldsymbol{P}_z\boldsymbol{\Lambda}_x\Big\}\Big\}. \tag{B.14}$$

If we let x run through the variables in $\boldsymbol{\lambda}$ in the same way as we did for (B.5), the result is

$$\begin{aligned} \boldsymbol{g}_{C\lambda} &= 2N\mathrm{Re}\{\mathrm{diag}\{\boldsymbol{P}\boldsymbol{\Lambda}^{-1}-2\boldsymbol{R}_z\boldsymbol{\Lambda}\boldsymbol{P}_z\}\} \\ &= 2N\boldsymbol{\Lambda}^{-1}\mathrm{Re}\{\mathrm{diag}\{\boldsymbol{P}-2\boldsymbol{R}_{z\lambda}\boldsymbol{P}_z\}\}, \end{aligned}$$

where we have used $\boldsymbol{R}_z\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^{-1}\boldsymbol{R}_{z\lambda}$. This is the formula for $\boldsymbol{g}_{C\lambda}$ in (27).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.sigpro.2022.108879

References

[1] R.O. Schmidt, Multiple emitter location and signal parameter estimation, IEEE Trans. Antennas Propagat. 34 (3) (1986) 276–280.
[2] R. Roy, A. Paulraj, et al., ESPRIT–A subspace rotation approach to estimation of parameters of cisoids in noise, IEEE Trans. Acoustics Speech Signal Process. (1986) 1340–1342.
[3] P. Stoica, K.C. Sharman, Maximum Likelihood methods for direction of arrival estimation, IEEE Trans. Acoustic. Speech Signal Process. 38 (7) (1990) 1132–1142.
[4] P. Stoica, B. Ottersten, M. Viberg, R.L. Moses, Maximum Likelihood array processing for stochastic coherent sources, IEEE Trans. Signal Process. 44 (1) (1996) 96–105.
[5] M. Viberg, B. Ottersten, Sensor array processing based on subspace fitting, IEEE Trans. Signal Process. 39 (5) (1991) 1110–1121.
[6] E. Fishler, H.V. Poor, Estimation of the number of sources in unbalanced arrays via information theoretic criteria, IEEE Trans. Signal Process. 53 (9) (2005) 3543–3553.
[7] Y. Wu, C. Hou, G. Liao, Q. Guo, Direction-of-arrival estimation in the presence of unknown nonuniform noise fields, IEEE J. Oceanic Eng. 31 (2) (2006) 504–510.
[8] C.-E. Chen, F. Lorenzelli, R.E. Hudson, K. Yao, Maximum Likelihood DOA estimation of multiple wideband sources in the presence of nonuniform sensor noise, EURASIP J. Adv. Signal Process. 2008 (1) (2007) 1–12.
[9] C.E. Chen, F. Lorenzelli, R.E. Hudson, K. Yao, Stochastic Maximum-Likelihood DOA estimation in the presence of unknown nonuniform noise, IEEE Trans. Signal Process. 56 (7) (2008) 3038–3044.
[10] L. Lu, H. Wu, K. Yan, S.S. Iyengar, Robust Expectation-Maximization algorithm for multiple wideband acoustic source localization in the presence of nonuniform noise variances, IEEE Sens. J. 11 (3) (2011) 536–544.
[11] B. Liao, S.-C. Chan, L. Huang, C. Guo, Iterative methods for subspace and DOA estimation in nonuniform noise, IEEE Trans. Signal Process. 64 (12) (2016) 3008–3020.
[12] B. Liao, L. Huang, C. Guo, H.C. So, New approaches to direction-of-arrival estimation with sensor arrays in unknown nonuniform noise, IEEE Sens. J. 16 (24) (2016) 8982–8989.
[13] A.L. Matveyev, A.B. Gershman, J.F. Bohme, On the direction estimation Cramer-Rao bounds in the presence of uncorrelated unknown noise, Circuit. Syst. Signal Process. (1999).
[14] M. Pesavento, Gershman, Maximum-Likelihood direction-of-arrival estimation in the presence of unknown nonuniform noise, IEEE Trans. Signal Process. 49 (2001) 1310–1324.
[15] Z.-Q. He, Z.-P. Shi, L. Huang, Covariance sparsity-aware DOA estimation for nonuniform noise, Digit Signal Process. 28 (2014) 75–81.
[16] B. Liao, C. Guo, H.C. So, Direction-of-arrival estimation in nonuniform noise via low-rank matrix decomposition, in: Digital Signal Processing (DSP), 2017 22nd International Conference on, IEEE, 2017, pp. 1–4.
[17] S.A. Vorobyov, A.B. Gershman, K.M. Wong, Maximum Likelihood direction-of-arrival estimation in unknown noise fields using sparse sensor arrays, IEEE Trans. Signal Process. 53 (1) (2005) 34–43.
[18] D. Madurasinghe, A new DOA estimator in nonuniform noise, IEEE Signal Process. Lett. 12 (4) (2005) 337–339.
[19] I. Ziskind, M. Wax, Maximum Likelihood localization of multiple sources by alternating projection, IEEE Trans. Acoustic. Speech Signal Process. 36 (1988) 1553–1560.
[20] W.H. Press, et al., Numerical recipes in C, Cambridge University Press, 1997.
[21] J.E. Dennis Jr., R.B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, number 16 in Classics in Applied Mathematics, SIAM, 1996.
[22] G.H. Golub, C.F.V. Loan, Matrix computations, The Johns Hopkins University Press, 2013. Fourth edition
[23] S.M. Kay, Fundamentals of Statistical Signal Processing. Estimation Theory, vol. 1, of Signal Processing, Prentice Hall, first edition, 1993.
[24] A.G. Jaffer, Maximum Likelihood Direction Finding of Stochastic Sources, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, 1988, pp. 2893–2896.
[25] J. Selva, An efficient Newton-type method for the computation of ML estimators in a uniform linear array, IEEE Trans. Signal Process. 53 (6) (2005) 2036–2045.
[26] J. Selva, ML estimation and detection of multiple frequencies through periodogram estimate refinement, IEEE Signal Process. Lett. 24 (3) (2017) 249–253.
[27] J. Selva, Newton-type methods in Array Processing, IEEE Signal Process. Lett. 11 (2) (2004) 104–107.
[28] J. Selva, Efficient multipath mitigation in navigation systems, PhD thesis, Universitat Politècnica de Catalunya, 2004.