Parametric array calibration

Shuang Wan

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
February 2011

Abstract

The subject of this thesis is the development of parametric methods for the calibration of array shape errors. Two physical scenarios are considered, the online calibration (self-calibration) using far-field sources and the offline calibration using near-field sources. The maximum likelihood (ML) estimators are employed to estimate the errors. However, the well-known computational complexity in objective function optimization for the ML estimators demands effective and efficient optimization algorithms.

A novel space-alternating generalized expectation-maximization (SAGE)-based algorithm is developed to optimize the objective function of the conditional maximum likelihood (CML) estimator for the far-field online calibration. Through data augmentation, joint direction of arrival (DOA) estimation and array calibration can be carried out by a computationally simple search procedure. Numerical experiments show that the proposed method outperforms the existing method for closely located signal sources and is robust to large shape errors. In addition, the accuracy of the proposed procedure attains the Cramér-Rao bound (CRB).

A global optimization algorithm, particle swarm optimization (PSO) is employed to optimize the objective function of the unconditional maximum likelihood (UML) estimator for the far-field online calibration and the near-field offline calibration. A new technique, decaying diagonal loading (DDL) is proposed to enhance the performance of PSO at high signal-to-noise ratio (SNR) by dynamically lowering it, based on the counter-intuitive observation that the global optimum of the UML objective function is more prominent at *lower* SNR. Numerical simulations demonstrate that the UML estimator optimized by PSO with DDL is optimally accurate, robust to large shape errors, and free of the initialization problem. In addition, the DDL technique is applicable to a wide range of array processing problems where the UML estimator is employed and can be coupled with different global optimization algorithms.

Declaration of originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the School of Engineering at The University of Edinburgh.

Shuang Wan

Acknowledgements

First of all, I would like to thank my supervisors Dr. Pei-Jung Chung and Prof. Bernard Mulgrew for inspiration and technical guidance. Without their help I could never have gone this far. Second, thanks to my colleagues and friends in Edinburgh who help me in work and in everyday life. Third, thanks to my parents, whose love and support warm my heart. Finally, thanks to the Chinese and British governments for financial support.

Contents

List of figures

List of tables

Acronyms and abbreviations

CA circular array

CML conditional maximum likelihood

CML-SAGE conditional maximum likelihood – space generalized expectation-maximization

CRB Cramér-Rao bound

DDL decaying diagonal loading

DE differential evolution

DE-DDL differential evolution – decaying diagonal loading

DF direction finding

DL diagonal loading

DOA direction of arrival

EM expectation-maximization

ESPRIT estimation of signal parameters via rotational invariance techniques

FIM Fisher's information matrix

GA genetic algorithm

LA linear array

LS least squares

LS-ESPRIT least squares – estimation of signal parameters via rotational invariance techniques

MAP maximum *a posteriori*

MC Monte Carlo

ML maximum likelihood

MSE mean squared error

MUSIC multiple signal classification

MVDR minimum variance distortionless response

PSO particle swarm optimization

PSO-DDL particle swarm optimization – decaying diagonal loading

PSO-DDL-simplex particle swarm optimization – decaying diagonal loading – simplex

SA simulated annealing

SAGE space-alternating generalized expectation-maximization

SE squared error

SNR signal-to-noise ratio

STD standard deviation

TLS total least squares

TLS-ESPRIT total least squares – estimation of signal parameters via rotational invariance techniques

UCA uniform circular array

ULA uniform linear array

UML unconditional maximum likelihood

WF1 Weiss and Friedlander algorithm 1 [1]

WF2 Weiss and Friedlander algorithm 2 [2]

WSF weighted subspace fitting

List of principal symbols

a	inter-sensor spacing
b	error tolerance
$\boldsymbol{B}_{\text{CR}}()$	CRB
$\boldsymbol{C}(\boldsymbol{\vartheta})$	covariance matrix of the estimation error of $\boldsymbol{\vartheta}$
$\boldsymbol{C}_{\boldsymbol{X}}(\boldsymbol{\vartheta})$	covariance matrix of random variable $\boldsymbol{X}$ as a function of $\boldsymbol{\vartheta}$
d	distance
d_{nm}	length associated with the distance between the nth sensor and the mth source
D	dimension
D_a	physical dimension of the sensor array
$e_n(t)$	additive noise on the nth sensor at time t
$f_{(\cdot)}()$	probability density function
F	UML objective function
F_c	CML objective function
$\mathcal{G}()$	Gaussian distribution
$\boldsymbol{g}_i(t)$	the ith group's best location at current time step t
$\boldsymbol{h}()$	steering vector
$\boldsymbol{H}()$	array manifold
$\boldsymbol{J}$	Fisher information matrix
K	number of snapshots
L	number of harmonic frequencies
$L()$	log-likelihood function
L_0	initial loading amount
$\boldsymbol{m}(\boldsymbol{\vartheta})$	mean vector as a function of $\boldsymbol{\vartheta}$
M	number of signal sources
N	number of sensors
$\boldsymbol{n}(k)$	additive noise vector at the kth snapshot in the Fourier domain
$n_n(k)$	additive noise on the nth sensor at the kth snapshot in the Fourier domain
$\boldsymbol{p}_i(t)$	the ith particle's best location of its own experience from time step 1 to t
P	number of population

$P()$	probability
P_c	crossover probability
$\boldsymbol{P}_{\boldsymbol{H}(\boldsymbol{\alpha})}$	projection matrix of $\boldsymbol{H}$ as a function of $\boldsymbol{\alpha}$
$\boldsymbol{q}_n$	the nth eigen vector of $\boldsymbol{R}$
$\boldsymbol{Q}$	correction matrix
$Q()$	log-likelihood of the augmented data
$\boldsymbol{Q}_n$	matrix composed of noise eigenvectors
r	random integer
r_l	ratio of loaded iteration
$r_{(\cdot)}$	random number
$\boldsymbol{r}_{(\cdot)}$	random vector
$r_m(t)$	radiated signal from the mth source at time t
R	random number
$\boldsymbol{R}$	covariance matrix of array output in the Fourier domain
$\mathbb{R}^D$	D-dimensional real space
$\boldsymbol{s}(k)$	radiated signal vector at the kth snapshot in the Fourier domain
$\boldsymbol{S}$	covariance matrix of signal in the Fourier domain
t	time
T	maximal time
$\boldsymbol{T}$	linear transformation matrix
$\boldsymbol{T}()$	many-to-one mapping
$u_n(t)$	output of the nth sensor at time t
w	weight
$\boldsymbol{x}\,(k)$	output vector at the kth snapshot in the Fourier domain
$\boldsymbol{x}^{(p,g)}$	parameter vector of the pth member at the gth generation
x_n	x-coordinate of the nth sensor
$x_n\,(k)$	output of the nth sensor at the kth snapshot in the Fourier domain
$\boldsymbol{y}(k)$	augmented data at snapshot k for expectation-maximization (EM) algorithm
y_n	y-coordinate of the nth sensor
$\boldsymbol{z}(k)$	augmented data for SAGE algorithm
$\boldsymbol{\alpha}$	vector of interested parameters
δ	threshhold of DOA separation
$\boldsymbol{\delta}$	the perturbation vector

$\delta()$	the Kronecker delta function
Δx_n	perturbation to the x-coordinate of the nth sensor
Δy_n	perturbation to the y-coordinate of the nth sensor
$\boldsymbol{\vartheta}$	parameter vector
θ_m	bearing of source m
λ	signal wavelength
λ_n	the nth eigen value of $\boldsymbol{R}$
ν	noise power
ν_l	power of loaded noise
$\boldsymbol{\xi}_i(t)$	position of the ith swarm particle at time t
ρ_m	range of source m
σ_p	standard deviation of sensor position errors
τ_{nm}	propagation time from the mth source to the nth sensor
$\varphi_{(\cdot)}$	acceleration constant
χ	constriction factor
$\boldsymbol{\psi}_i\,(t)$	velocity of the ith swarm particle at time t
$\mathit{\Psi}$	maximal velocity
ω_l	lth angular frequency
$\widehat{(\cdot)}$	estimate of $(\cdot)$
$(\cdot)^*$	complex conjugate of $(\cdot)$
$(\cdot)^H$	Hermitian transpose of $(\cdot)$
$(\cdot)^T$	transpose of $(\cdot)$
$(\cdot)^{[i]}$	$(\cdot)$ at the ith iteration
$(\cdot)^{[i,c]}$	$(\cdot)$ at the cth cycle of the ith iteration
$\mathrm{E}(\cdot)$	expectation operator
$\mathrm{m}(\cdot)$	mean operator
$\mathrm{tr}(\cdot)$	trace operator
$\lvert \cdot \rvert$	absolute value operator
$\odot$	element-wise product operator
$\langle\rangle_D$	modulo operator with modulus D

Chapter 1
Introduction

1.1 Calibration in array processing

Array processing lies in the heart of such diverse applications as radar, sonar, communications, radio astronomy, seismology and medical diagnosis. Modern radar and sonar systems use antenna or hydrophone arrays to achieve accurate measurement and detection. Demand from the wireless communication market drives the development of multiple-antenna systems that enjoy higher capacity, better service quality and larger area coverage. In order to detect celestial objects and estimate their characteristics, radio astronomy incorporates arrays with very long baselines. Seismology has long been utilizing sensor arrays to detect and locate underground natural resources or nuclear explosions. Tomography necessitates arrays for cross-section imaging of objects from transmitted or reflected signals in medical diagnosis. Finally, sensor networks are employed in applications of environmental monitoring, military surveillance and agriculture.

A sensor array is a set of sensors located at distinct spatial positions to sense signals travelling in space. At each observation interval, the array records the signal wavefront and produces a data vector. As in any measurement system, using an array as the probe inducts errors into the final measurement, by transforming field strength into recorded data through array response (or steering vector), which in turn is determined by element characteristics and array geometry. Such errors are incurred by imperfect knowledge of the array response, which is unfortunately inevitable in real-world applications. With respect to the array shape error for instance, tactical radar systems require that the array be dismantled and reassembled in the field [1]; towed hydrophone arrays suffer not only as a result of deployment but also of the platform manoeuvring [3]; installation tolerance is expected in cellular communication systems which feature a huge amount of antenna arrays; and antenna arrays in satellite communication systems are prone to deformation resulting from the limited payload weight that degrades the rigidity of the array shape [4]. Besides the array geometry perturbation that affects array response, the characteristics of individual sensor and electronics are different from each other because of manufacture tolerance, whereas only nominal values are known before calibration. Meanwhile, when arranged in an array, there is mutual coupling between sensors induced by neighboring field excitations, which causes sensor response to drift away from the separate measurement made initially. This is a potential agent of performance corruption because there are cases where it is impossible to measure array response in an ideal anechoic chamber due to the array's large physical size. Moreover, these nominal characteristics of elements are subject to long-term degradation and environmental effects such as temperature and moisture, which makes maintaining steering vector knowledge more difficult. These effects all necessitate array calibration or array processing methods that are robust to steering vector errors or both.

Calibration methods can be categorized into online and offline calibrations [5]. Offline calibration refers to the calibration that is carried out before the array is put into use. The array response is measured in ideal environments on proper spacial and frequency grids, and the steering vector measurements are stored in processor storage for subsequent utilization. Online calibration refers to simultaneous signal and array response estimation, in which signal characters such as DOA or power, together with array response parameters such as element position, gain and phase, are calculated in the field. It is known that under certain conditions this auto-calibration is feasible, and Cramér-Rao bounds are used to define the least attainable calibration residual [6].

Generally speaking, sources of estimation error can be classified into finite sample approximation and model uncertainty. The former is caused by approximating the statistics with finite numbers of samples, under which estimators experience performance degradation; the latter is caused by model uncertainties such as noise modeling error and aforementioned steering vector error, on which superresolution methods crucially depend. In this thesis the primary concern is only one type of model uncertainty, the array shape error, which is calibrated by new methods developed herein. The new methods are initially developed under the assumption of the availability of accurate statistics or equivalently infinite sample, i.e., the finite-sample error is ignored. However simulations show that they maintain good performance with small number of samples.

1.2 Thesis structure and contributions

1.2.1 Thesis structure

Chapter 2 gives the background of the research work in this thesis. The data model is introduced first, followed by a description of the parametric array processing methods. Then the literature on sensitivity analysis of the parametric methods is reviewed, concluding that small perturbation can lead to large estimation error, thus the modeling error must be dealt with. Finally, existing robust estimators and calibration methods are reviewed and categorized.

In chapter 3, the far-field online calibration is carried out using the conditional maximum likelihood (CML) estimator with space-alternating generalized expectation-maximization (SAGE)-based optimization. A novel SAGE-based algorithm is developed for joint direction of arrival (DOA) estimation and array shape calibration. Numerical simulations show its superior performance over the existing method.

In chapter 4, the same calibration as in chapter 3 is conducted using the unconditional maximum likelihood (UML) estimator optimized by particle swarm optimization (PSO). A new technique, decaying diagonal loading (DDL) is developed to enhance the performance of PSO at high signal-to-noise ratio (SNR) by dynamically lowering it. In addition to the optimal calibration accuracy demonstrated by the numerical simulations, the DDL development process introduces the “prominence” concept of the global optimum of an objective function which is important to the success of a global optimizer.

In chapter 5, the near-field offline calibration is performed by a UML estimator optimized by PSO with DDL that is developed in chapter 4. The optimal calibration accuracy in this case further confirms the applicability of the DDL technique to a wide range of array processing problems that use the UML estimator.

Chapter 6 concludes the thesis and gives suggestions for further investigation.

1.2.2 Contributions

The first contribution of the thesis is a SAGE-based algorithm for array shape self-calibration. Through data augmentation, the multi-dimensional DOA estimation and array calibration can be carried out by a computationally simple one-dimensional or two-dimensional search procedure. In contrast to existing methods, the SAGE-based algorithm does not introduce additional calibration residuals and is robust to large array shape errors.

The second contribution of the thesis is the development of the UML-PSO-DDL paradigm for array shape calibration. For the first time, the UML estimator is applied to the problem of array shape calibration. Its computation complexity is overcome by the application of the

PSO algorithm. A novel technique, DDL is proposed to enhance PSO's performance at high SNR, based on the counter-intuitive observation that the global optimum of the UML objective function is more prominent at lower SNR. The UML-PSO-DDL procedure is robust to large shape errors and its mean squared error (MSE) approaches the Cramér-Rao bound (CRB). It has been applied to the far-field online shape calibration and near-field offline shape calibration, and is believed to be applicable to a wide range of array processing problems.

Chapter 2
Model uncertainty in parametric array processing

The resolution of conventional array processing is bounded by the Fourier limit that is associated with the array's physical size. Parametric array processing, by further exploiting the signal model, can achieve super-resolution that is finer than the Fourier limit [7]. Array processing by parameter estimation can be categorized into maximum likelihood (ML) and subspace methods. The former uses conditional maximum likelihood (CML) and unconditional maximum likelihood (UML) estimators [8] [9] and the latter multiple signal classification (MUSIC) [10] and estimation of signal parameters via rotational invariance techniques (ESPRIT) [11], etc. The ML estimators are derived from the ML principle and the subspace estimators are based on the orthogonality of the signal and noise subspaces. In some special cases the subspace estimators are asymptotically equivalent to their ML counterparts and thus can be regarded as fast implementations of the ML estimators [12].

Theoretically, the resolution of parametric methods is limited only by signal-to-noise ratio (SNR) and observation time, regardless of the array aperture. However, this relies on the condition that the model is free of uncertainty. In practice, model uncertainty is unavoidable and even a small error can severely degrade the performance of array processing methods [13]. Thus the model uncertainty must be addressed. In this chapter, the background of array processing under model uncertainty is provided. The signal model for parametric array processing is introduced in section 2.1, followed by a description of the estimation methods in section 2.2, then the impact of model uncertainty is discussed in section 2.3, afterwards the calibration methods are reviewed in section 2.4. Section 2.6 concludes this chapter.

2.1 Signal model

A two-dimensional far-field case is used as an example to interpret the signal model. Consider an array of N sensors receiving signals emitted by M far-field sources, as shown in Fig. 2.1.

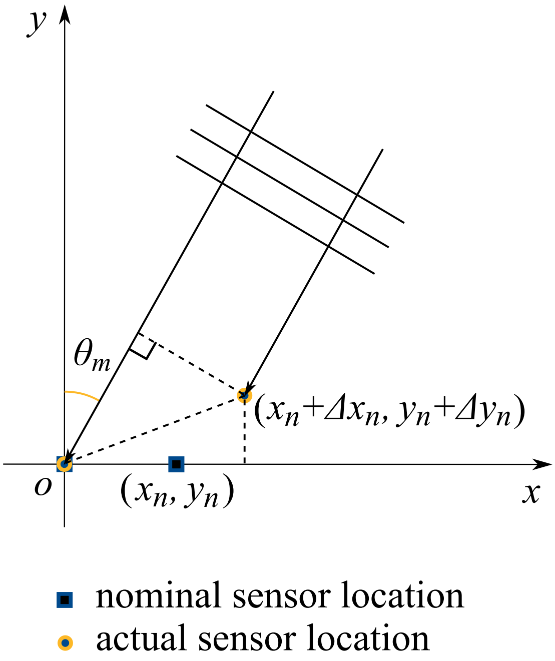

Figure 2.1: *Far-field array-source geometry.*

The known nominal location of the nth sensor is given by $[x_n, y_n]^T$, which is perturbed by unknown error $[\Delta x_n, \Delta y_n]^T$ to the actual location $[x_n + \Delta x_n, y_n + \Delta y_n]^T$. The perturbations are summarized in vector $\boldsymbol{\delta} = [\Delta x_1, \Delta y_1, \dots, \Delta x_N, \Delta y_N]^T$, which is considered as constant in a calibration process.

In the time domain, the output of the nth sensor, $u_n(t)$, can be expressed as

$$u_n(t) = \sum_{m=1}^{M} r_m(t - \tau_{nm}) + e_n(t), \tag{2.1}$$

where $r_m(t - \tau_{nm})$ is the delayed version of radiated signal from the mth source, $e_n(t)$ is the additive noise on the nth sensor. t denotes time, and τ_{nm} is associated with the propagation time from the mth source to the nth sensor.

For the array shape calibration problem, all the information of interest is carried by τ_{nm}, therefore a separation of τ_{nm} from $r_m(t - \tau_{nm})$ provides algebraic and conceptual convenience. A Fourier transform implements the separation:

$$x_n(\omega_l) = \frac{1}{\sqrt{T}} \int_{-T/2}^{T/2} u_n(t)\, e^{-j\omega_l t} \mathrm{d}t, \tag{2.2}$$

where $\omega_l = (l' + l)(2\pi/T)$ is the lth angular frequency, $l = 0, 1, \cdots, L-1$, L being the number of harmonic frequencies, and l' a constant.

In principle, infinite L is required to capture all the signal information, however, if the signal energy concentrates in a finite frequency band, finite L suffices. Moreover, if a narrowband signal is considered, that is, the signal energy concentrates in a frequency band whose width is small compared to $2\pi/T$, then a single Fourier coefficient of a frequency at the center of the band, ω, suffices to represent the time domain signal. In the following narrow band signal is considered and the dependence on the single frequency ω is suppressed for notational simplicity.

Denoting the Fourier coefficients of $r_m(t)$ and $e_n(t)$ by $s_m(k)$ and $n_n(k)$, respectively, the observation equation (2.1) is transformed into the frequency domain as

$$x_n\left(k\right) = \sum_{m=1}^{M} e^{-j\omega\tau_{nm}} s_m\left(k\right) + n_n\left(k\right), \; k = 1, \ldots, K, \tag{2.3}$$

where K is the number of snapshots. In practice, a total observation period is divided into K intervals, each with duration T and the data set in each interval subject to a Fourier transform, Eq. (2.2). Each interval is called a snapshot and the K snapshots of transformed data are used to form the statistics.

Eq. (2.3) can be organized into matrix form

$$\boldsymbol{x}\left(k\right) = \boldsymbol{H}\boldsymbol{s}\left(k\right) + \boldsymbol{n}\left(k\right), \; k = 1, \ldots, K, \tag{2.4}$$

where

$$\boldsymbol{x}\left(k\right) = \left[x_1\left(k\right), x_2\left(k\right), \ldots, x_N\left(k\right)\right]^T$$

$$\boldsymbol{s}\left(k\right) = \left[s_1\left(k\right), s_2\left(k\right), \ldots, s_M\left(k\right)\right]^T$$

$$\boldsymbol{n}\left(k\right) = \left[n_1\left(k\right), n_2\left(k\right), \ldots, n_N\left(k\right)\right]^T$$

and the nmth element of the array manifold $\boldsymbol{H}$ is

$$H_{nm} = \exp\left\{-j\omega\tau_{nm}\right\}, \; n = 1, 2, \ldots, N, \; m = 1, 2, \ldots, M. \tag{2.5}$$

For the near-field case, however, the range effect must be taken into account and consequently the nmth element of the array manifold $\boldsymbol{H}$ is

$$H_{nm} = \frac{1}{d_{nm}} \exp\left\{-j\omega\tau_{nm}\right\},\ n = 1, 2, \ldots, N,\ m = 1, 2, \ldots, M. \tag{2.6}$$

The relation between far field and near field is elucidated in the following sub-section.

2.1.1 Near field and far field

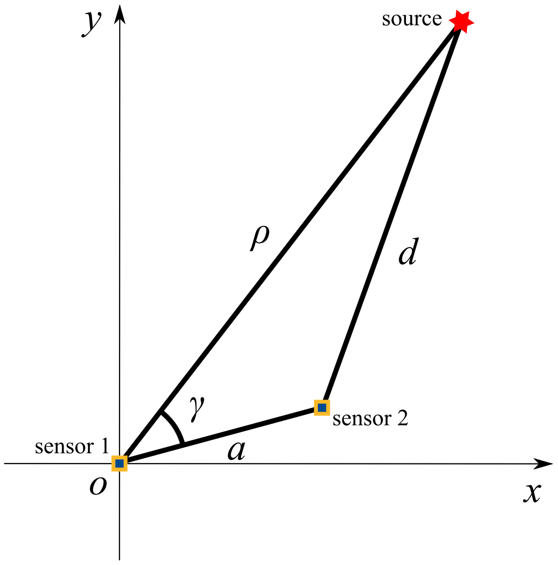

Figure 2.2: *The geometry of wave propagation.*

Fig. 2.2 illustrates the geometry of a source and an array of two sensors, where d is the distance between the source and sensor 2:

$$d = \sqrt{\rho^2 + a^2 - 2\rho a \cos\gamma} = \rho\sqrt{1 + \left[\left(\frac{a}{\rho}\right)^2 - 2\frac{a}{\rho}\cos\gamma\right]}.$$

Here ρ is the distance between the source and sensor 1, which lies on the origin of the coordinates without loss of generality. a is the inter-sensor spacing, and γ is the angle between ρ and a.

For the far-field case, $\rho \gg a$. Using a Taylor expansion regarding $\left(\frac{a}{\rho}\right)^2 - 2\frac{a}{\rho}\cos\gamma$, and after simple manipulation one has

$$d = \rho - a\cos\gamma + \frac{1}{2}\frac{a^2}{\rho}\sin^2\gamma + aO\left(\left(\frac{a}{\rho}\right)^2\right). \tag{2.7}$$

As the intersensor spacing a is comparable to the wavelength λ, the first two terms of the Taylor series have to be included for the approximation of the phase factor $\omega\tau$ as in Eq. (2.6).

For the magnitude factor $1/d$ in Eq. (2.6),

$$\frac{1}{d} = \frac{1}{\rho}\frac{1}{\sqrt{1+\left[\left(\frac{a}{\rho}\right)^2 - 2\frac{a}{\rho}\cos\gamma\right]}} = \frac{1}{\rho} + \frac{a}{\rho^2}\cos\gamma + \frac{1}{\rho}O\left(\left(\frac{a}{\rho}\right)^2\right). \tag{2.8}$$

As $a \ll \rho$, the first term of the Taylor series is a good approximation of the amplitude factor.

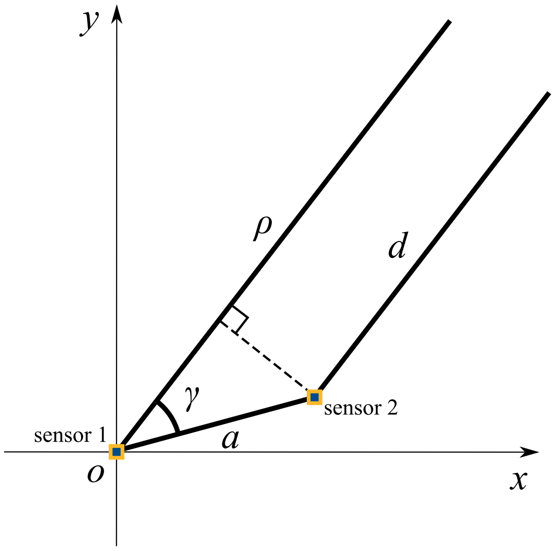

Figure 2.3: *Far-field approximation of the geometry of wave propagation.*

The far-field approximation effectively reduces the wavefront to planar, or in other words the ray of propagation to sensor 1 is parallel to that of sensor 2, as shown in Fig. 2.3. The boundary between far-field and near-field is where the errors due to the parallel ray approximation starts to become insignificant. The insignificance is accepted as the path length deviation due to neglecting the third term of Eq. (2.7) is $\frac{1}{16}$ of a wavelength. This corresponds to a phase error of $\frac{2\pi}{\lambda} \cdot \frac{\lambda}{16} = \frac{\pi}{8}$ rad [14].

For an array consisting of multiple sensors, let D_a denote the dimension of the array, then the far-field distance ρ_f is found by equating the maximal value of the third term of Eq. (2.7), which occurs for $a = D_a/2$ and $\gamma = 90°$, to $\frac{1}{16}$ of a wavelength:

$$\frac{(D_a/2)^2}{2\rho_f} = \frac{\lambda}{16},$$

which gives

$$\rho_f = \frac{2D_a^2}{\lambda} \tag{2.9}$$

In addition to the condition Eq. (2.9) induced by the approximation of the electric field, a similar approximation of the magnetic field induces the condition [14]

$$\rho \gg \lambda. \tag{2.10}$$

This condition is significant for arrays operating in lower frequencies, where the array can be small compared to the wavelength due to physical constraints.

Summarizing the constraints induced by the approximations of the phase Eq. (2.7) and magnitude Eq. (2.8) of the electric field, and the approximation of the magnetic field Eq. (2.10), the far-field conditions can be expressed as follows:

$$\begin{cases} \rho > \frac{2D_a^2}{\lambda} \\ \rho \gg D_a \\ \rho \gg \lambda. \end{cases}$$

Usually, the far-field is taken to begin at a distance given by Eq. (2.9). This is usually a sufficient condition for arrays operating in higher frequencies where the array dimension is comparable to the wavelength. In this thesis it is assumed that the arrays operate in this mode and thus the far-field distance ρ_f is defined by Eq. (2.9).

The region interior to ρ_f is called the near-field and can be divided into two subregions. The subregion closer to the array is called the reactive near-field region, which extends to a distance $0.62\sqrt{D_a^3/\lambda}$ from the array, as long as $D_a \gg \lambda$. Between the reactive near-field and far-field region is the radiating near-field region. For an array focused at infinity, the region is sometimes referred to as the Fresnel region, where a Fresnel approximation (a second-order Taylor expansion regarding the dimension of the array) is applicable. A detailed discussion of the Fresnel approximation and its accuracy can be found in [15].

In conclusion, the field region distances can be summarized as follows:

Region	Distance from array (ρ)
Reactive near field	0 to $0.62\sqrt{D_a^3/\lambda}$
Radiating near field (Fresnel)	$0.62\sqrt{D_a^3/\lambda}$ to $2D_a^2/\lambda$
Far field (Fraunhofer)	$2D_a^2/\lambda$ to ∞

Table 2.1: *Field regions.*

2.2 Parameter estimation methods

Parameter estimation methods for array processing can be classified as ML methods and subspace methods. Sections 2.2.1 and 2.2.2 introduce them, respectively.

2.2.1 ML methods

The ML methods can be categorized into CML and UML, depending on whether the source signals are assumed to be deterministic or random. For CML, the source signals are assumed deterministic and subsequently their waveforms have to be estimated, along with other parameters such as those geometric parameters that are of major interest. For UML, the source signals are assumed random and their covariances have to be estimated instead, along with other parameters.

Since in the deterministic signal model the likelihood function is conditional on the signal waveforms, the ML method based on this model is named CML, whereas the ML method based on the stochastic signal model is named UML. These conceptual assumptions do not change any physical fact of the array processing problem, but determine what the unknown parameters are and consequently determine the formats of the estimators. Owing to the parametric difference the CML and UML estimators feature different performances and the Cramér-Rao bounds (CRB's) based on the conditional and unconditional models are also different.

Despite the differences in modeling assumptions, the derivations of the CML and UML estimators both involve a separation of the signal parameters, i.e. the signal waveforms for CML and signal covariances for UML. The likelihood function is first maximized with respect to the signal parameters to yield an explicit expression of the signal parameters as a function of the other parameters, and this expression is then inserted back to the likelihood function to yield a likelihood function of the other parameters only, which are of major interest. In the following the derivations of the CML and UML estimators are briefly introduced.

2.2.1.1 CML estimation

Under the conditional model, the signals are deterministic. They are frozen while the noises are different from realization to realization. The signal waveforms have to be estimated in addition to other parameters. Hence the unknown parameters are

$$\boldsymbol{\alpha}, \{\boldsymbol{s}(k)\}_{k=1}^{K}, \nu, \tag{2.11}$$

where $\boldsymbol{\alpha}$ is the vector of parameters of interest, the content of which will be specified according to the problems considered in the following chapters. The notations $\{\boldsymbol{s}(k)\}_{k=1}^{K}$ and ν are the signal waveforms and the noise power, respectively. They are the nuisance parameters of no interest but have to be estimated alongside the parameters of interest.

Because the noise is subject to a Gaussian distribution,

$$\boldsymbol{n}(k) \sim \mathcal{G}\left(\boldsymbol{0}, \nu \boldsymbol{I}\right).$$

Under the conditional model assumption the array output is subject to Gaussian distribution, with mean $\boldsymbol{H}\boldsymbol{s}(k)$:

$$\boldsymbol{x}(k) \sim \mathcal{G}\left(\boldsymbol{H}\boldsymbol{s}(k), \nu \boldsymbol{I}\right).$$

Then the joint probability density of K snapshots is

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{k=1}^{K} \frac{1}{\det(\pi \nu \boldsymbol{I})} \exp\left\{-\frac{1}{\nu}|\boldsymbol{x}(k) - \boldsymbol{H}(\boldsymbol{\alpha})\boldsymbol{s}(k)|^2\right\},$$

and the log-likelihood function is

$$L(\boldsymbol{\alpha}, \{\boldsymbol{s}(k)\}_{k=1}^{K}, \nu) = -KN \ln \pi - KN \ln \nu - \frac{1}{\nu}\sum_{k=1}^{K} |\boldsymbol{x}(k) - \boldsymbol{H}(\boldsymbol{\alpha})\boldsymbol{s}(k)|^2. \tag{2.12}$$

Let us first assume ν is known, and solve for it later. In this case the minimization of

$$\sum_{k=1}^{K} |\boldsymbol{x}(k) - \boldsymbol{H}(\boldsymbol{\alpha})\boldsymbol{s}(k)|^2$$

in Eq. (2.12) is a separable problem [16] [17]: one can first solve for $\{\boldsymbol{s}(k)\}_{k=1}^{K}$ as a function of $\boldsymbol{\alpha}$, and then substitute the result back to form a function of only $\boldsymbol{\alpha}$. The separated solution of $\{\boldsymbol{s}(k)\}_{k=1}^{K}$ is the standard solution of the least squares (LS) problem:

$$\hat{\boldsymbol{s}}(k) = \left[\boldsymbol{H}^H(\boldsymbol{\alpha})\boldsymbol{H}(\boldsymbol{\alpha})\right]^{-1}\boldsymbol{H}^H(\boldsymbol{\alpha})\boldsymbol{x}(k), \tag{2.13}$$

or in array processing context it is the output of the minimum variance distortionless response (MVDR) beamformer.

Inserting Eq. (2.13) back into (2.12) and dropping the constant term $-KN\ln\pi$, we have

$$L_1(\boldsymbol{\alpha},\nu) = -KN\ln\nu - \frac{1}{\nu}\sum_{k=1}^{K}|\boldsymbol{x}(k) - \boldsymbol{P}_{\boldsymbol{H}(\boldsymbol{\alpha})}\boldsymbol{x}(k)|^2, \tag{2.14}$$

where

$$\boldsymbol{P}_{\boldsymbol{H}(\boldsymbol{\alpha})} = \boldsymbol{H}(\boldsymbol{\alpha})\left[\boldsymbol{H}^H(\boldsymbol{\alpha})\boldsymbol{H}(\boldsymbol{\alpha})\right]^{-1}\boldsymbol{H}^H(\boldsymbol{\alpha})$$

is the projection matrix onto the column space of $\boldsymbol{H}(\boldsymbol{\alpha})$.

Notice that

$$\begin{aligned}
\sum_{k=1}^{K}|\boldsymbol{x}(k) - \boldsymbol{P}_{\boldsymbol{H}(\boldsymbol{\alpha})}\boldsymbol{x}(k)|^2 &= \sum_{k=1}^{K}|\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\boldsymbol{x}(k)|^2 = \sum_{k=1}^{K}\boldsymbol{x}^H(k)\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\boldsymbol{x}(k) \\
&= \mathrm{tr}\sum_{k=1}^{K}\boldsymbol{x}^H(k)\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\boldsymbol{x}(k) = \mathrm{tr}\sum_{k=1}^{K}\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\boldsymbol{x}(k)\boldsymbol{x}^H(k) \\
&= K\mathrm{tr}\left[\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{x}(k)\boldsymbol{x}^H(k)\right] \\
&= K\mathrm{tr}\left[\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\widehat{\boldsymbol{R}}\right],
\end{aligned}$$

where

$$\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})} = \boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{H}(\boldsymbol{\alpha})}$$

is the orthogonal complement of $\boldsymbol{P}_{\boldsymbol{H}(\boldsymbol{\alpha})}$, and

$$\widehat{\boldsymbol{R}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{x}(k)\,\boldsymbol{x}^H(k)$$

is the sample covariance matrix, then Eq. (2.14) can be rewritten as

$$L_1(\boldsymbol{\alpha},\nu) = -K\left\{N\ln\nu + \frac{1}{\nu}\mathrm{tr}\left[\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\widehat{\boldsymbol{R}}\right]\right\}. \tag{2.15}$$

Now let us solve for ν. Setting

$$\frac{\partial}{\partial\nu}L_1(\boldsymbol{\alpha},\nu) = 0$$

we obtain

$$\hat{\nu} = \frac{1}{N}\mathrm{tr}\left[\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\widehat{\boldsymbol{R}}\right]. \tag{2.16}$$

Substituting Eq. (2.16) into Eq. (2.15) gives

$$L_1(\boldsymbol{\alpha}) = -KN\left\{\ln\mathrm{tr}\left[\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\widehat{\boldsymbol{R}}\right] - \ln N + 1\right\}. \tag{2.17}$$

Maximizing Eq. (2.17) is equivalent to minimizing the argument of the logarithm, so the CML estimator can be expressed as

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}}\left\{\mathrm{tr}\left[\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})}\widehat{\boldsymbol{R}}\right]\right\}. \tag{2.18}$$

It is remarked that the minimizer $\hat{\boldsymbol{\alpha}}$ of Eq. (2.18) also maximizes Eq. (2.15) when ν is known, thus knowing the noise power ν does not affect the CML estimate.

2.2.1.2 UML estimation

Under the unconditional model, the signals are stochastic. It is the signal covariances that have to be estimated in addition to other parameters. Hence the unknown parameters are

$$\boldsymbol{\alpha}, \boldsymbol{S}, \nu,$$

where $\boldsymbol{\alpha}$ and ν are the same as those in CML estimation, (2.11), and $\boldsymbol{S} = \mathrm{E}\left[\boldsymbol{s}(k)\boldsymbol{s}^H(k)\right]$ is the signal covariance matrix. $\boldsymbol{S}$ and ν are the nuisance parameters of no interest but have to be estimated alongside the parameters of interest, $\boldsymbol{\alpha}$.

It is further assumed in Eq. (2.4) that $\boldsymbol{s}(k)$ and $\boldsymbol{n}(k)$ are zero mean and mutually independent Gaussian random variables, then the array output is subject to a Gaussian distribution

$$\boldsymbol{x}\left(k\right) \sim \mathcal{G}\left(\boldsymbol{0}, \boldsymbol{R}\right),$$

where $\boldsymbol{R} = \mathrm{E}\left[\,\boldsymbol{x}\left(k\right)\boldsymbol{x}^H\left(k\right)\right] = \boldsymbol{H}\boldsymbol{S}\boldsymbol{H}^H + \nu\boldsymbol{I}$ is the covariance matrix of $\boldsymbol{x}\left(k\right)$.

Under mild regularity conditions $\{\boldsymbol{x}(k)\}_{k=1}^K$ are mutually independent [18], then the joint probability density of K snapshots is

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{k=1}^{K} \frac{1}{\det(\pi\boldsymbol{R})} \exp\left\{-\boldsymbol{x}^H(k)\boldsymbol{R}^{-1}\boldsymbol{x}(k)\right\},$$

and the log-likelihood function is

$$L(\boldsymbol{\alpha}, \{\boldsymbol{s}(k)\}_{k=1}^K, \nu) = -KN\ln\pi - K\ln\det\boldsymbol{R} - \sum_{k=1}^{K}\boldsymbol{x}^H(k)\boldsymbol{R}^{-1}\boldsymbol{x}(k). \tag{2.19}$$

When the noise power ν is known, a necessary condition for maximizing the log-likelihood function is that

$$\frac{\partial L(\boldsymbol{\alpha}, \boldsymbol{R})}{\partial S_{ij}} = 0, \quad i, j = 1, 2, \ldots, N, \tag{2.20}$$

where S_{ij} denotes the ijth element of $\boldsymbol{S}$. This gives

$$\widehat{\boldsymbol{S}}\left(\boldsymbol{\alpha}\right) = \left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\boldsymbol{H}^H\left(\widehat{\boldsymbol{R}} - \nu\boldsymbol{I}\right)\boldsymbol{H}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}. \tag{2.21}$$

Inserting Eq. (2.21) back into Eq. (2.20) and dropping the constant terms, one has the UML estimator after some manipulation as [19]:

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}}\left\{-\ln\det[\boldsymbol{P}_{\boldsymbol{H}}\widehat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{H}} + \nu\boldsymbol{P}_{\boldsymbol{H}}^{\perp}] - \frac{1}{\nu}\operatorname{tr}[\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\widehat{\boldsymbol{R}}]\right\}, \tag{2.22}$$

where $\boldsymbol{P}_{\boldsymbol{H}} = \boldsymbol{H}(\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H$ is the projection matrix of $\boldsymbol{H}$, and $\boldsymbol{P}_{\boldsymbol{H}}^{\perp} = \boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{H}}$ is its orthogonal complement.

If the noise power ν is unknown, we set

$$\frac{\partial L}{\partial \nu} = 0,$$

thus

$$\hat{\nu} = \frac{\mathrm{tr}[\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\widehat{\boldsymbol{R}}]}{N-M}. \tag{2.23}$$

Substituting (2.23) into (2.22), we obtain

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} \left\{ -\ln\det[\boldsymbol{P}_{\boldsymbol{H}}\widehat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{H}} + \frac{\mathrm{tr}[\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\widehat{\boldsymbol{R}}]\boldsymbol{P}_{\boldsymbol{H}}^{\perp}}{N-M}] \right\}. \tag{2.24}$$

A pure algebraic derivation can be found in [20].

2.2.2 Subspace methods

A typical subspace method is multiple signal classification (MUSIC). In this method, the sample covariance matrix

$$\boldsymbol{R} = \boldsymbol{H}\left(\boldsymbol{\alpha}\right)\boldsymbol{S}\boldsymbol{H}^{H}\left(\boldsymbol{\alpha}\right) + \nu\boldsymbol{I} \tag{2.25}$$

is decomposed into

$$\boldsymbol{R} = \sum_{n=1}^{N} \lambda_n \boldsymbol{q}_n \boldsymbol{q}_n^{H},$$

where λ_n and $\boldsymbol{q}_n$ are the eigenvalues and eigenvectors of $\boldsymbol{R}$, respectively.

Organizing $\{\lambda_n\}_{n=1}^{N}$ in descending order,

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_M > \lambda_{M+1} = \lambda_{M+2} = \cdots = \lambda_N = \nu,$$

the first M eigenvalues correspond to the M signals and the associated M eigenvectors are referred to as the signal-subspace eigenvectors. The eigenvectors associated with the last $N-M$ eigenvectors equal to ν are referred to as the noise-subspace eigenvectors.

Arrange the signal-subspace eigenvectors and the noise-subspace eigenvectors into matrices:

$$\boldsymbol{Q}_s = \left[\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_M\right],$$

$$\boldsymbol{Q}_n = \left[\boldsymbol{q}_{M+1}, \boldsymbol{q}_{M+2}, \ldots, \boldsymbol{q}_N\right].$$

The signal subspace spanned by the signal-subspace eigenvectors is orthogonal to the noise subspace spanned by the noise-subspace eigenvectors

$$\text{span}\left\{\boldsymbol{Q}_s\right\} \perp \text{span}\left\{\boldsymbol{Q}_n\right\},$$

or in other words

$$\boldsymbol{Q}_s^H \boldsymbol{Q}_n = \mathbf{0}.$$

Moreover, the signal subspace is also spanned by the steering vectors,

$$\text{span}\left\{\boldsymbol{Q}_s\right\} = \text{span}\left\{\boldsymbol{H}\right\},$$

that is to say

$$\boldsymbol{H}^H \boldsymbol{Q}_n = \mathbf{0} \tag{2.26}$$

In practice, the eigenvalues and eigenvectors are estimated from the sample covariance matrix $\widehat{\boldsymbol{R}}$ using eigenvalue decomposition or singular value decomposition, then a MUSIC pseudo-spectrum is composed:

$$p\left(\boldsymbol{\alpha}\right) = \frac{1}{\boldsymbol{h}^H\left(\boldsymbol{\alpha}\right)\widehat{\boldsymbol{Q}}_n\widehat{\boldsymbol{Q}}_n^H \boldsymbol{h}\left(\boldsymbol{\alpha}\right)}, \tag{2.27}$$

where $\widehat{\boldsymbol{Q}}_n$ is the estimate of $\boldsymbol{Q}_n$ and $\boldsymbol{h}\left(\boldsymbol{\alpha}\right)$is the steering vector. A search of the M peaks over the pseudo-spectrum gives the parameters associated with the M signals.

The relation between CML and UML is given in [19] and [21]. The relation between CML and MUSIC is given in [12] and [19].

2.3 Impact of model uncertainties

It is clear from the signal model and the estimator expression that to achieve an accurate parameter estimation, the array manifold $\boldsymbol{H}$ must be perfectly known and the noise must be perfectly spatially white. However, in reality only partial knowledge is known about $\boldsymbol{H}$ due to imperfections such as

- sensor location error
- sensor gain and phase error
- mutual coupling between sensors
- statistical modeling errors of noise

Recall the data covariance matrix

$$\boldsymbol{R} = \boldsymbol{H}\boldsymbol{S}\boldsymbol{H}^H + \nu\boldsymbol{N}. \tag{2.28}$$

For a spatially white noise field, $\boldsymbol{N} = \boldsymbol{I}$, and hence Eq. (2.28) degenerates into (2.25).

A general perturbation model that incorporates the aforementioned array imperfections is

$$\widetilde{\boldsymbol{R}} = (\boldsymbol{H} + \Delta\boldsymbol{H})\,\boldsymbol{S}\,(\boldsymbol{H} + \Delta\boldsymbol{H})^H + \nu\,(\boldsymbol{N} + \Delta\boldsymbol{N})\,,$$

where the matrices $\Delta\boldsymbol{H}$ and $\Delta\boldsymbol{N}$ result from various types of model perturbations. The matrix $\Delta\boldsymbol{H}$ represents the deviation from the nominal array manifold $\boldsymbol{H}$, which incorporates the effects due to sensor location error, sensor gain and phase error, and signal-only mutual coupling. The Hermitian matrix $\Delta\boldsymbol{N}$ represents the deviation from the nominal noise covariance $\boldsymbol{N}$.

An immediate question is how these errors propagate through an estimator to the final estimate, and a straightforward answer is provided by a Taylor expansion of the objective function with respect to the modeling errors, although the derivation usually requires considerable matrix manipulation. A number of researchers have carried out the error analysis using a first and second order Taylor expansion. In particular, B. Friedlander [22] performed a first-order analysis of CML with respect to general array manifold perturbation, including sensor gain, phase and location errors. He [23] also conducted a first-order analysis of MUSIC regarding sensor gain , phase and location errors with an emphasis on the degradation of resolution ability.

A. Lee Swindlehurst and T. Kailath [24] conducted first-order analysis of MUSIC with respect to array manifold errors, such as sensor gain and phase error, sensor location error and mutual coupling, and with respect to noise and channel model errors. The emphasis was on the degradation of accuracy. They [25] also conducted a first-order analysis of the class of LS subspace fitting algorithms, including CML, multidimensional MUSIC, weighted subspace fitting (WSF) and total least squares – estimation of signal parameters via rotational invariance techniques (TLS-ESPRIT) with respect to both array and noise model errors. Li and Vaccaro [26] carried out first-order analysis of least squares – estimation of signal parameters via rotational invariance techniques (LS-ESPRIT) with respect to array manifold errors and noise model errors.

Anne Ferréol *et al.* conducted a second-order analysis of MUSIC [27] [28] and CML [29] with respect to general array manifold error that incorporates mutual coupling and sensor location error, etc. The second-order analysis accommodates errors larger than that first-order analysis can handle and reaches an error magnitude where the algorithms break down.

These analyses conclude that even a small perturbation on the data model can result in significant estimation error, and sometimes can lead to algorithm breakdown. Thus in real applications the modeling errors must be compensated for. One remedy is to modify the algorithms such that they are less sensitive to modeling errors [24] [25]. However, this method can result in sub-optimal performance [24]. The other remedy is to directly calibrate the errors. This method can completely remove the model error under suitable conditions and consequently the result is optimal. In this thesis we shall concentrate on the latter method, in particular, the calibration techniques for sensor location error. So firstly a review of the calibration techniques is given, with an emphasis on the calibration of sensor location error.

Last but not least, an example of far-field source localization is used to demonstrate the effect of array shape error on the performance of MUSIC. The nominal array shape is perturbed by Gaussian displacement, with σ_p designating the standard deviation in fraction of intersensor spacing. It is seen from Fig. 2.4 that sensor location error causes bias, peak fading and peak merging in the MUSIC pseudo-spectrum. When sensor location error is absent, two peaks appear around the true DOA's, -3° and 3°. As the error grows, the two DOA's become biased toward larger values, to different extents. In addition the two peaks fade to the floor and merge together. When the standard deviation (STD) of perturbation grows to 2% of the intersensor spacing, the two peaks are merged to the extent that they cannot be clearly distinguished. This example shows that even a small perturbation (STD 2% intersensor spacing) can severely degrade the performance of MUSIC. The impact of array shape error on ML methods is difficult to visualize as the objective function is 2-dimensional for 2-source localization, however it can be seen from the simulation results in Chapter 3.

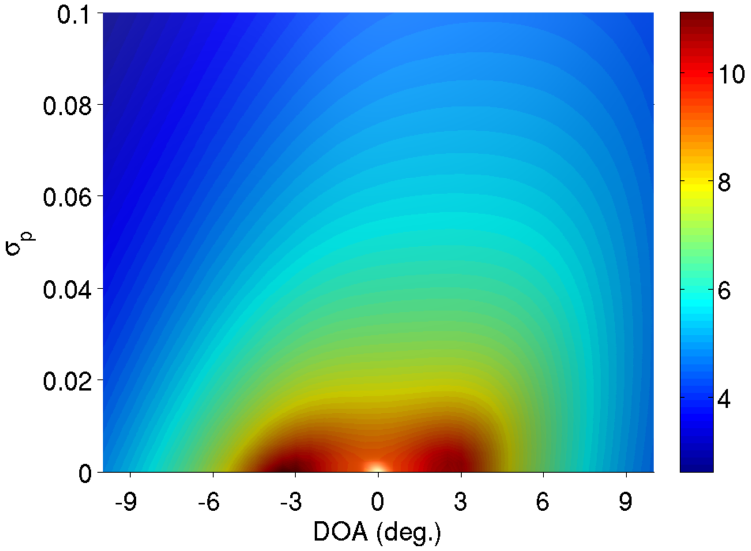

Figure 2.4: *The impact of array shape perturbation on the MUSIC pseudo-spectrum. The nominal array is a 5-sensor UCA with $\lambda/2$ intersensor spacing, perturbed by Gaussian displacement. 2 equi-power far-field sources radiate from $-3°$ and $3°$ DOA.* 1000 *snapshots, 20 dB signal-to-noise ratio (SNR).*

2.4 Calibration techniques

Calibration techniques are categorized into offline and online techniques, dependant on whether calibration sources are at known or unknown positions. Offline calibration (often termed calibration for short) uses sources at known positions to calculate the array specifications, under either a parametric or non-parametric model. This straightforward configuration can be regarded as the inverse problem of source localization. Online calibration (or termed self-calibration, auto-calibration, blind calibration) uses sources at unknown positions to jointly calculate source locations and array parameters. Owing to the interplay of the large number of unknown parameters, certain identifiability conditions must be satisfied to avoid ambiguity.

2.4.1 Offline calibration

Offline calibration uses sources at known locations to calibrate an array. The conceptually simplest procedure is to measure the array response at a sufficiently dense grid of source locations and tabulate the response for look-up in array operation afterwards. This procedure, despite its ability to calibrate all types of error, is expensive in data collection. If the array response is constrained by a certain structure, the number of unknowns can be reduced and more advanced procedures can be applied. The conventional offline calibration methods with interpolation based on a smoothness constraint of array response is reviewed in section 2.4.1.1, followed by CML/LS and subspace methods based on the physical modeling of the array response in sections 2.4.1.2 and 2.4.1.3, respectively.

2.4.1.1 Conventional offline calibration

The most common offline calibration uses one signal source placed at multiple known locations. Then the array response vector (steering vector) at these known locations can be calculated, for example for far-field calibration, as follows. From Eq. (2.4), the array output for a calibration signal coming from DOA θ_m is given by

$$\boldsymbol{x}_m(k) = \boldsymbol{h}(\theta_m)\, s_m(k) + \boldsymbol{n}_m(k) \quad m = 1, 2, \ldots M \quad k = 1, 2, \ldots, K.$$

If the transmitted signal $s_m(k)$ is known, the steering vector can be estimated by the LS solution

$$\hat{\boldsymbol{h}}(\theta_m) = \frac{\sum_{k=1}^{K} \boldsymbol{x}_m(k)\, s_m^*(k)}{\sum_{k=1}^{K} |s_m(k)|^2}.$$

If $s_m(k)$ is unknown, then the steering vector has to be estimated from the principal eigenvector of the sample covariance matrix

$$\widehat{\boldsymbol{R}}_m = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{x}_m(k)\, \boldsymbol{x}_m^H(k) = \sum_{n=1}^{N} \hat{\lambda}_n \hat{\boldsymbol{q}}_n \hat{\boldsymbol{q}}_n^H,$$

and

$$\hat{\boldsymbol{h}}(\theta_m) \propto \hat{\boldsymbol{e}}_1,$$

where $\hat{\boldsymbol{e}}_1$ is the principal eigenvector associated with the largest eigenvalue $\hat{\lambda}_1$. In this case, a gain and phase ambiguity remains and should be determined by other means.

A major advantage of this procedure is that any types of error, for instance sensor gain, phase and position error, and mutual coupling, are included, and all the errors are calibrated together. A drawback of the calibration procedure is the data collection of a large number of calibration points that is needed for an accurate estimation of the steering vector within the bearing range of interest. This drawback is even more severe for near-field calibration, where the data has to be collected from a two-dimensional calibration grid of known bearings and ranges. In practice, the data collection can be a costly and laborious task.

The reason for the requirement of a large amount of data is that the steering vector is assumed with no structure, hence the large number of unknowns – the steering vector itself at various focusing locations have to be estimated. To counteract this drawback, one may model the steering vector, resulting in much lesser number of unknown parameters. Consequently much less data is needed to identify the parameters.

A natural structure is the assumption that the steering vector is smooth with respect to the source location. Then a sparse calibration grid can be interpolated to a dense one to save data collection efforts. A more sophisticated interpolation takes effect on the correction matrix instead of the array manifold itself: the correction matrix $\boldsymbol{Q}$ performs a linear transform on the nominal array manifold $\boldsymbol{H}$ to yield the actual array manifold $\widetilde{\boldsymbol{H}}$:

$$\widetilde{\boldsymbol{H}} = \boldsymbol{Q}\boldsymbol{H}.$$

Although $\boldsymbol{Q}$ is assumed as independent of source location in [30] [31] [32] for a "global calibration", it is assumed as dependent on source location in [33] [34] [35] for a "local calibration". The local calibration is then interpolated from a sparse calibration grid to a dense one regarding the calibration matrix, as the calibration matrix is smoother than the array manifold itself with respect to source location.

Another structure is based on the spatial Fourier expansion of the steering vector [36] [37] [38] [39]. Through Fourier expansion, a steering vector of an arbitrary array can be approximated by the sum of a number of uniform linear arrays (ULA's). The calibration method then aims at estimating the Fourier coefficients. The most precise structure is the physical model of the sensors in Eq. (2.6), as usually used in the parametric array processing estimators. Fistas and Manikas [40] used the physical model to parameterize the steering vector in terms of sensor gain, phase and location. The actual values of them are then calculated from the differences among three measurements at different DOA's. If multiple measurements are available, more sophisticated estimators can be used with the parametric model. These methods fall into two categories based on the CML/LS principle or orthogonal subspace principle, in line with the source localization problem.

2.4.1.2 Offline calibration based on CML/LS principle

Based on the CML cost function Eq. (2.18),

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \left\{ \mathrm{tr} \left[\boldsymbol{P}^{\perp}_{\boldsymbol{H}(\boldsymbol{\alpha})} \widehat{\boldsymbol{R}} \right] \right\},$$

letting $\boldsymbol{\alpha}$ be the array parameters such as sensor gain, phase or location, and using data collected from sources at known locations, an estimator for offline calibration is constructed similarly to that for sources localization. Ng and Nehorai [41] implemented this estimator for calibration of sensor location errors using gradient optimization.

It is clear from the derivation of Eq. (2.18) in section 2.2.1.1 that the CML estimation under Gaussian assumption is equivalent to an LS estimation, see Eq. (2.12). An example of offline calibration based on LS formulation can be found in [42], where a single source is used to calibrate the sensor gain, phase and noise power of a radio telescope array. The proposed method is verified by experimental data.

In offline calibration, the calibration sources are placed at known locations. An important question is: given the requirement of localization accuracy, how many calibration sources are needed to meet the accuracy requirement? Another important question is: given the number of calibration sources, what is the optimal source placement? Porat and Friedlander [5] relate the off-line calibration accuracy to the DOA estimation errors for a CML estimator, and further propose a criterion for the minimal calibration resources to meet requirements of DOA estimation. Ng and Nehorai [43] [44] conducted research on the optimal placement of far-field sources for best calibration accuracy. They give the optimal source placement by deriving the CRB under the model of deterministic signals of known waveform or waveform with an unknown magnitude, and analyze the performances of the estimators based on CML and orthogonal subspace principles, concluding that the CRB can be asymptotically achieved by CML estimator. They also give Newton-type algorithms for implementation.

2.4.1.3 Offline calibration based on orthogonal subspace principle

Recall the fundamental orthogonality of subspaces, Eq. (2.26):

$$\boldsymbol{H}^{H} \boldsymbol{Q}_{n} = \mathbf{0},$$

where the array manifold $\boldsymbol{H}$ is a function of both array parameters and source locations. In a localization problem, one assumes that the array parameters are known and estimate the source locations. Conversely, in an offline calibration problem, one can assume that the source locations are known and estimate the array parameters.

Seymour *et al.* [45] deal with sensor location errors. They use the orthogonality of the signal subspace and noise subspace to compose a cost function of sensor locations. A gradient search is then used to perform the optimization for parameter estimation. The multi-dimensional search is needed because the objective function is multi-dimensional, rather than one-dimensional in the MUSIC algorithm. Similar to MUSIC, the cost function does not require temporal separation of the source as that of section 2.4.1.1, and in fact takes advantage of the presence of multiple sources to identify the parameters.

Ng *et al.* deal with sensor gain and phase error [46], and sensor gain, phase and location error [47]. They assume that the errors are small and a first-order Taylor expansion is applied upon the errors to yield an analytical solution. This technique is similar to that of [1] and [2]. In [47], they carry out experimental measurements to verify the proposed method.

2.4.2 Online calibration

Online calibration uses sources at unknown locations, and the unknown source locations have to be estimated along with the array parameters. As a result the number of unknowns for online calibration is more than that of a corresponding offline calibration. The large number of unknowns cause difficulty to optimization algorithms, and may cause ambiguity to the estimator. To reduce the number of unknown parameters, certain constraints must be imposed. Usually the physical modeling Eq. (2.6) is adopted, in contrast to offline calibration where more flexible structures are acceptable, as mentioned in section 2.4.1.1. Additionally, certain identifiability conditions must be satisfied [6] [48] [49] [50]. These parametric methods based on the physical modeling can be categorized into CML/LS methods and Subspace methods, in line with the parametric source localization and offline calibration.

2.4.2.1 CML/LS methods

Paulraj and Kailath [51] use an LS formulation for sensor gain and phase self-calibration by exploiting the Toeplitz structure of the correction matrix of a uniform linear array (ULA). An analytical solution is obtained for the LS problem, but with a phase ambiguity. This phase ambiguity can be resolved by additional measurement of the relative phase difference between any two sensors in the array, or more elegantly, by imposing a constraint that the sum of phase errors is zero [52] [53]. Reference [52] has a Newton-iterative solution and [53] has an analytical solution. For this problem, Li and Er [54] analyse the effects of the unknown elements in the multiplicative correction matrix on the performance and determine the optimal structure of the correction matrix. Yip and Zhou [55] consider gain and phase self-calibration by exploiting the cyclostationarity of the signals. The LS objective function is optimized by alternating minimization. Selone and Serra [56] use an LS formulation for mutual coupling self-calibration by exploiting the structure of ULA, and obtain an analytical solution.

Böhme and Kraus [57] consider unknown noise field self-calibration. The unknown covariance matrix of the noise field is approximated by the weighted sum of known matrices, and the weights are then estimated from an LS formulation. The entire algorithm iterates between estimating noise parameters and source DOA's until convergence is achieved. Friedlander and Weiss [58] further parameterize the unknown noise covariance by a Fourier expansion, then jointly estimate the DOA's and Fourier coefficients by optimizing a UML or LS objective function. The latter objective function is further concentrated with respect to the Fourier coefficients to reduce the dimension of the objective function.

See and Poh [59] use a multiplicative calibration matrix to account for general steering vector errors, such as sensor gain, phase and location errors, and mutual coupling. The calibration matrix is independent of source location, and the calibration sources are temporally and spatially disjoint. The concentrated LS objective function is eventually optimized using a Newton-Genetic algorithm. Mir *et al.* [60] conduct an experimental study of steering vector estimation using one source scanning at unknown directions (or in other words temporally and spatially disjoint sources at unknown directions). The LS objective function is optimized by iterative algorithm.

Chen *et al.* [61] consider wideband near-field source localization using a CML estimator with gradient search. The method is then extended to simultaneously estimate source location and one sensor location (partial self-calibration). The near-field processing is not based on a Fresnel approximation. Chen *et al.* [62] also consider self-calibration of non-uniform noise using wideband far-field sources. The noise covariance matrix is modeled as a diagonal matrix with unknown unequal diagonal elements, and a CML cost function is formed to jointly estimate the DOA's and non-uniform noise powers using iterative numerical concentration of the cost function. The derivation of wideband CRB is included.

Finally, Weiss and Friedlander's self-calibration method [1] for sensor location error is reviewed below with some mathematical details, as it is closely related to the work in this thesis. Weiss and Friedlander algorithm 1 [1] (WF1) is an array shape self-calibration algorithm using far-field sources. The parameters to be estimated are $\boldsymbol{\alpha} = \left[\boldsymbol{\theta}^T, \boldsymbol{\delta}^T\right]^T$, where $\boldsymbol{\theta}$ is the DOA vector and $\boldsymbol{\delta}$ is the shape perturbation vector.

The objective function is based on CML, and the algorithm is essentially a block-alternating optimization, iterating between the DOA parameters $\boldsymbol{\theta}$ and the perturbation parameters $\boldsymbol{\delta}$. Taking advantage of the assumption that the perturbations are small, a first-order Taylor expansion is applied with regard to $\boldsymbol{\delta}$, and the $\boldsymbol{\delta}$-block optimization is carried out analytically.

The algorithm is sketched as follows. It starts with the CML objective function Eq. (2.18). To maximize the likelihood, it suffices to minimize the last term of Eq. (2.12). Thus the cost function is

$$F_c = \sum_{k=1}^{K} \|\boldsymbol{x}(k) - \boldsymbol{H}(\boldsymbol{\theta}, \boldsymbol{\delta})\, \boldsymbol{s}(k)\|^2,$$

which is also the LS cost function due to the assumption that the noise is Gaussian and spatially white.

For the DOA parameters $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_M]$, an alternate minimization is applied, i.e. for θ_m, fix other θ's and minimize with respect to θ_m only, and then repeat for θ_{m+1}. After every θ has been optimized once, update $\{\boldsymbol{s}(k)\}_{k=1}^{K}$ according to Eq. (2.13). Repeat the minimization regarding $\boldsymbol{\theta}$ and the update of $\{\boldsymbol{s}(k)\}_{k=1}^{K}$until convergence is reached.

For the perturbation parameters $\boldsymbol{\delta}$, notice that the cost function Eq. (2.4.2.1) is a sum over the sensors,

$$F_c = \sum_{n=1}^{N} \sum_{k=1}^{K} \|x_n(k) - \sum_{m=1}^{M} H(\theta_m, \boldsymbol{\delta}_n)\, s_m(k)\|^2 = \sum_{n=1}^{N} F_{cn},$$

therefore the minimization can be split sensor-wise, i.e. for $n = 1, 2, \ldots N$, minimize

$$F_{cn} = \sum_{k=1}^{K} \|x_n(k) - \sum_{m=1}^{M} H(\theta_m, \boldsymbol{\delta}_n)\, s_m(k)\|^2, \tag{2.29}$$

in which

$$H\left(\theta_m, \boldsymbol{\delta}_n\right) = \exp\left\{j\frac{2\pi}{\lambda}\left[\left(x_n + \Delta x_n\right)\sin\theta_m + \left(y_n + \Delta y_n\right)\cos\theta_m\right]\right\} \tag{2.30}$$

is the nmth element of the array manifold $\boldsymbol{H}$.

Under the small perturbation assumption that

$$\Delta x_n \sin\theta_m + \Delta y_n \cos\theta_m \ll \frac{\lambda}{2},$$

$H\left(\theta_m, \boldsymbol{\delta}_n\right)$ can be linearized with respect to $\boldsymbol{\delta}_n = \left[\Delta x_n, \Delta y_n\right]^K$ using a first-order Taylor approximation

$$\exp\left[j\frac{2\pi}{\lambda}\left(\Delta x_n \sin\theta_m + \Delta y_n \cos\theta_m\right)\right] \approx 1 + j\frac{2\pi}{\lambda}\left(\Delta x_n \sin\theta_m + \Delta y_n \cos\theta_m\right). \tag{2.31}$$

By substituting Eqs. (2.31) and (2.30) back to Eq. (2.29), the minimization of F_n is converted to an LS problem, thus $\boldsymbol{\delta}_n$ can be solved analytically. This analytical procedure is repeated for every sensor. The overall algorithm then starts from a nominal value of $\boldsymbol{\delta}$ and alternates between optimizing $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ until convergence is reached.

2.4.2.2 Subspace methods

Friedlander and Weiss [63] conduct self-calibration of sensor gain and phase errors using far-field sources based on a subspace formulation. The sensor gain and phase errors are solved analytically, and the entire algorithm iterates between DOA and error estimations until convergence is achieved. Friedlander and Weiss [64] also conduct gain, phase and mutual coupling self-calibration using far-field sources. The sensor gain and phase errors, and mutual coupling are solved analytically, and the entire algorithm iterates among DOA, gain and phase error and mutual coupling estimations until convergence is achieved. A related work of gain and direction-dependent phase self-calibration is in [65].

Flieller *et al.* [66] perform self-calibration of general direction-dependent modelling errors. The MUSIC-type cost function is penalized with the prior of the modelling errors. The penalization is similar to that in [67] of a maximum *a posteriori* (MAP) principal. Note that subspace fitting is used to approximate the likelihood function and gradient algorithm is used to optimize the MAP objective function in [67]. Jansson *et al.* extend [67] to the case when both model error and finite sample error are present. Wan and Chia [68] perform array shape self-calibration using far-field sources. The cost function is composed from inverse subspace fitting and is optimized by alternate minimization similar to that in [1].

Finally, Weiss and Friedlander's self-calibration method [2] for sensor location error is reviewed below with some mathematical details, as it is closely related to the work in this thesis. Weiss and Friedlander algorithm 2 [2] (WF2) is a self-calibration algorithm using far-field sources. It incorporates a block-alternating technique to optimize the objective function, the objective function is based on MUSIC. The algorithm is sketched as follows.

It starts from Eq. (2.26), and the cost function is

$$F_M\left(\boldsymbol{\alpha}\right) = \|\boldsymbol{Q}_n^H \boldsymbol{H}\left(\boldsymbol{\alpha}\right)\|^2 = \sum_{m=1}^{M} \|\boldsymbol{Q}_n^H \boldsymbol{h}_m\left(\theta_m, \boldsymbol{\delta}\right)\|^2,$$

where $\boldsymbol{h}_m(\theta_m, \boldsymbol{\delta})$ is the mth column of $\boldsymbol{H}(\boldsymbol{\alpha})$.

For the DOA parameters $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_M]$, a standard MUSIC algorithm is applied. Assuming nominal sensor positions or their latest estimate, the M peaks of the pseudo spectrum Eq. (2.27) gives $\boldsymbol{\theta}$. This is also the minimizer of F since every $\|\boldsymbol{Q}_n^H \boldsymbol{h}_m\left(\theta_m, \boldsymbol{\delta}\right)\|^2$ is minimized.

For the perturbation parameters $\boldsymbol{\delta}$, the same Taylor approximation as Eq. (2.31) is applied. F is then converted to an LS cost function and thus $\boldsymbol{\delta}$ can be solved analytically. The overall algorithm starts from a nominal value of $\boldsymbol{\delta}$ and alternates between optimizing $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ until convergence is reached.

Compared with offline calibration, online calibration trades computational complexity for simplicity of data collection. It is worth mentioning some of the real-world applications of the online calibration techniques. Steinberg [69] conducted pioneering work of array shape self-calibration, Sahr [60] performed self-calibration of airborne radar array, Wijnholds [70] carried out self-calibration of astronomical radio arrays.

2.5 Identifiability in calibration

Certain geometrical conditions must be satisfied to identify the parameters in a calibration process. For instance, if in a self-calibration both the array and sources are rotated relative to the center of the array by a certain angle, the data collected after would be the same as before the rotation, and consequently no calibration process can tell the absolute positions of the sensors and sources without an angular reference. Rockah *et al.* investigate the identifiability issue in self-calibration using the CRB. The rationale is that if the CRB goes to zero as the observation time or SNR goes to infinity, the array is calibratable with the sources, and vice versa. The scenarios investigated are far-field [6] and near-field [48] array shape self-calibration and sensor phase self-calibration [49]. However, due to analytical complexity the investigation is limited to sources disjoint in the time or frequency domain despite numerical results showing minor difference for sources overlapping in the time or frequency domain.

Levi and Messer [50] investigate the identifiability in calibration using pure geometry for near and far field sources in known and unknown locations. They also state that the results have been obtained using statistical and numerical tools. The investigation is applicable to sources overlapping in the time or frequency domain, and the result is in agreement with [6] [48]. For the two physical scenarios considered in Chapters 3, 4 and 5 of this thesis, the identifiability conditions are cited below.

2.5.1 Far-field online calibration

- The nominal array coordinates do not lie along a straight line (i.e. not a linear array).
- One sensor's location and its direction to a second sensor are known.
- There are at least 3 far-field sources at unknown but distinct directions.

2.5.2 Near-field offline calibration

For an array of 6 sensors or more, a sufficient condition of full calibration is that there are 3 or more near-field sources at known locations [50] (Tab. 2).

2.6 Conclusion

In this chapter, parametric array processing methods have been introduced. While featuring high resolution, these methods are known to be sensitive to model errors. The model errors have to be calibrated to prevent performance degradation. Among the model errors, array shape error is difficult to deal with as it is tightly associated with source location and is nonlinearly expressed in the array manifold. Linearization by Taylor series expansion simplifies computation but causes accuracy and robustness problems [1] [2]. Among the calibration methods, CML/LS and subspace methods have received extensive research. Although the UML estimator is superior to CML/LS and subspace estimators for DOA estimation [21] [71] and gain-and-phase offline calibration [72], and this superiority is conjectured to be general, it is seldom used in array calibration, due to its complicated objective function that is difficult to optimize. Finally, array calibration using near-field sources is computationally more difficult than using far-field sources because the far-field approximation is invalid in the near field. By using novel techniques, this thesis attempts to resolve these difficulties for performance improvement.

Chapter 3
Far-field online calibration I

This chapter addresses the problem of array shape online calibration (self-calibration) using narrowband far-field sources at unknown directions. The unknown source directions of arrival (DOA's) and sensor locations are jointly estimated by a conditional maximum likelihood (CML) estimator. The resulting objective function, unfortunately, is complicated to optimize, due partly to the additional parameters of sensor location, compared with a DOA estimation problem. Previous efforts have been to linearize the location error parameters by first-order Taylor approximation, based on the small error assumption. Although the computational complexity is reduced, the Taylor approximation introduces bias to the estimate and renders the self-calibration inapplicable to large sensor location errors. To overcome these difficulties, the well-known space-alternating generalized expectation-maximization (SAGE) algorithm is applied to the multidimensional nonlinear optimization for the CML estimator. It breaks the multi-dimensional optimization down into multiple one or two dimensional optimizations, thus significantly reduces the computational cost. And as no approximation is resorted to, the new algorithm outperforms existing ones in critical scenarios such as large sensor location errors and closely-located sources. Moreover, it is statistically efficient.

The structure of the chapter is as follows. It starts with formulating the CML objective function in section 3.1, followed by introducing expectation-maximization (EM) and SAGE algorithms in section 3.2, the SAGE algorithm for self-calibration in section 3.3 and the expression of the Cramér-Rao bound (CRB) in subsection 3.4. It ends with simulation results in section 3.5.

3.1 Problem formulation

For far-field sources, the time delay in Eq. (2.5) is expressed by

$$\tau_{nm} = d_{nm}/c, \tag{3.1}$$

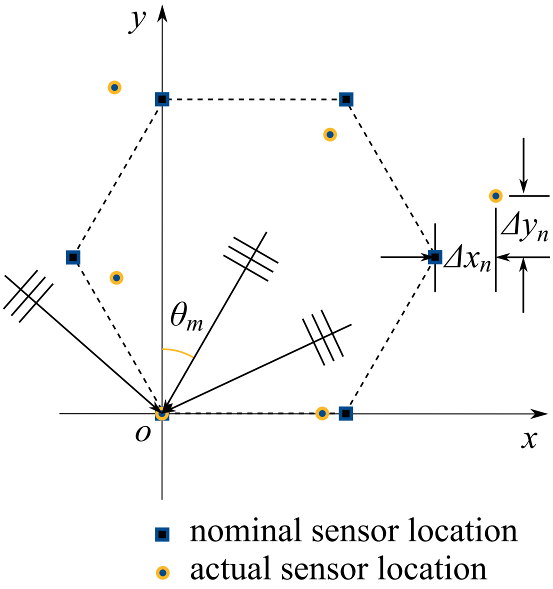

Figure 3.1: *The array-source geometry of the far-field case.*

where c is the propagation speed and d_{nm} is the distance of sensor n to the origin projected onto the source m's DOA θ_m,

$$d_{nm} = [\sin\theta_m \ \cos\theta_m] \begin{bmatrix} x_n + \Delta x_n \\ y_n + \Delta y_n \end{bmatrix}.$$

Noting that $c = \lambda\omega/(2\pi)$, λ being the wavelength, the n,mth element of the array manifold $\boldsymbol{H}(\boldsymbol{\theta}, \boldsymbol{\delta})$ can be expressed as

$$H_{nm}(\theta_m, \Delta x_n, \Delta y_n) = \exp\left\{-j\frac{2\pi}{\lambda}\left[(x_n + \Delta x_n)\sin\theta_m + (y_n + \Delta y_n)\cos\theta_m\right]\right\}, \quad n = 1, \ldots, N, \ m = 1, \ldots, M. \tag{3.2}$$

Compared with the near-field case, Eq. (2.6), the magnitude factor $\frac{1}{d_{nm}}$ is omitted in the far-field case because the array dimension is small compared with the distance between the source and the array, thus the magnitude change of a signal traversing the array can be ignored. The magnitude difference due to different path length can be expressed by the power of the signals.

The signal vector $\boldsymbol{s}(k)$ is considered to be stochastic. The noise vector $\boldsymbol{n}(k)$ is independent, identically complex normally distributed with zero mean and covariance matrix $\nu\boldsymbol{I}$, where ν is an unknown noise spectral parameter and $\boldsymbol{I}$ is an $N \times N$ identity matrix. Given the observations

$\{\boldsymbol{y}(k)\}_1^K$, the problem of central interest is to estimate the DOA parameters $\boldsymbol{\theta}$ and perturbation parameters $\boldsymbol{\delta}$.

Following [19], the CML estimator for $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ can be expressed as

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\delta}}) = \arg \max_{(\boldsymbol{\theta}, \boldsymbol{\delta})} \underbrace{\left\{ \operatorname{tr}\left(\boldsymbol{P}_{\boldsymbol{H}(\boldsymbol{\theta}, \boldsymbol{\delta})} \widehat{\boldsymbol{R}} \right) \right\}}_{F_c(\boldsymbol{\theta}, \boldsymbol{\delta})}, \tag{3.3}$$

where $\widehat{\boldsymbol{R}} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{x}(k) \boldsymbol{x}^H(k)$ is the sample covariance matrix, $\boldsymbol{P}_{\boldsymbol{H}} = \boldsymbol{H}(\boldsymbol{H}^H \boldsymbol{H})^{-1} \boldsymbol{H}^H$ is the projection matrix of $\boldsymbol{H}$. The dependence of $\boldsymbol{H}$ on $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ is suppressed for notational simplicity, and $F_c(\boldsymbol{\theta}, \boldsymbol{\delta})$ denotes the CML objective function.

3.2 The EM and SAGE algorithms

Although the objective function $F_c(\boldsymbol{\theta}, \boldsymbol{\delta})$ has been concentrated regarding signal and noise parameters, it is multi-modal, multi-dimensional and nonlinear. Direct application of the maximum likelihood principle would lead to a computationally expensive optimization procedure. By using a data augmentation scheme, EM/SAGE algorithm can decompose the multi-dimensional optimization problem into multiple one or two dimensional ones, thus dramatically reducing the computational complexity, at the expense of iterations.

The SAGE algorithm is a fast variant of the widely used EM algorithm for ML estimation. Like the EM algorithm, SAGE algorithm breaks down the difficult multi-dimensional optimization into simpler, lower dimensional optimizations by data augmentation; Unlike the EM algorithm, SAGE algorithm incorporates a flexible data augmentation scheme that enables a sequential update of partitions of the parameters, leading to faster a convergence than for the EM algorithm [73].

3.2.1 EM algorithm

To properly introduce the SAGE algorithm, it is best to build it on the foundation of the EM algorithm [74]. The EM algorithm is a general iterative approach for solving ML estimation problems given observed (incomplete) data. It employs a certain data augmentation scheme to generate the augmented (complete) data from the observed data. The augmented data, if properly generated, has a simpler relation with the parameters than the observed data, therefore simplifies the ML estimation of the parameters. The principle of the EM algorithm is given below.

Suppose the complete data is $\boldsymbol{Y}$. When observed, only the incomplete data $\boldsymbol{X}$ is output by the observation mechanism. The complete data $\boldsymbol{Y}$ is related to the incomplete data by a many-to-one mapping that is non-invertible,

$$\boldsymbol{T}(\boldsymbol{Y}) = \boldsymbol{X}.$$

Let $f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\vartheta})$ and $f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\vartheta})$ denote the probability densities of $\boldsymbol{Y}$ and $\boldsymbol{X}$, respectively, where $\boldsymbol{\vartheta}$ is the parameter vector to be estimated. Their relation can be expressed by

$$f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\vartheta}) = f_{\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\vartheta}) \cdot f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\vartheta}), \tag{3.4}$$

where $f_{\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\vartheta})$ is the conditional probability density of $\boldsymbol{Y}$ given $\boldsymbol{X} = \boldsymbol{x}$.

Take logarithm on Eq. (3.4):

$$\log f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\vartheta}) = \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\vartheta}) - \log f_{\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\vartheta}), \tag{3.5}$$

and take conditional expectation on Eq. (3.5) given $\boldsymbol{X} = \boldsymbol{x}$ at a parameter value $\hat{\boldsymbol{\vartheta}}$, Eq. (3.5) becomes

$$\underbrace{\log f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\vartheta})}_{L(\boldsymbol{\vartheta})} = \underbrace{\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x};\hat{\boldsymbol{\vartheta}}}\left\{\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\vartheta})\right\}}_{Q(\boldsymbol{\vartheta},\hat{\boldsymbol{\vartheta}})} - \underbrace{\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x};\hat{\boldsymbol{\vartheta}}}\left\{\log f_{\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}}(\boldsymbol{y};\boldsymbol{\vartheta})\right\}}_{U(\boldsymbol{\vartheta},\hat{\boldsymbol{\vartheta}})},$$

where $L(\boldsymbol{\vartheta})$, $Q(\boldsymbol{\vartheta},\hat{\boldsymbol{\vartheta}})$ and $U(\boldsymbol{\vartheta},\hat{\boldsymbol{\vartheta}})$ are defined for notational convenience.

By Jensen's inequality [75]

$$U\left(\boldsymbol{\vartheta},\hat{\boldsymbol{\vartheta}}\right) \leqslant U\left(\hat{\boldsymbol{\vartheta}},\hat{\boldsymbol{\vartheta}}\right), \tag{3.6}$$

therefore, to maximize the log-likelihood of the observed data, $L(\boldsymbol{\vartheta})$, it suffices to maximize the conditional expectation of the log-likelihood of the augmented data, $Q(\boldsymbol{\vartheta},\hat{\boldsymbol{\vartheta}})$. The maximization is with respect to $\boldsymbol{\vartheta}$.

The EM algorithms is thus formulated as below:

An EM iteration consists of an expectation (E) step and a maximization (M) step. In the ith iteration, $Q\left(\boldsymbol{\vartheta},\hat{\boldsymbol{\vartheta}}^{[i]}\right)$ is computed using the current parameter estimate $\hat{\boldsymbol{\vartheta}}^{[i]}$, then it is maximized

Algorithm 1: EM algorithm

```
for $i = 0, 1, \ldots$ do
    E step: compute
        $Q\left(\boldsymbol{\vartheta}, \hat{\boldsymbol{\vartheta}}^{[i]}\right)$
    M step
        $\hat{\boldsymbol{\vartheta}}^{[i+1]} = \arg\max_{\boldsymbol{\vartheta}} Q\left(\boldsymbol{\vartheta}, \hat{\boldsymbol{\vartheta}}^{[i]}\right),$
end
```

regarding $\boldsymbol{\vartheta}$ to yield the updated estimate $\hat{\boldsymbol{\vartheta}}^{[i+1]}$. The algorithm is initialized with an arbitrary guess of the parameter, $\hat{\boldsymbol{\vartheta}}^{[0]}$, and terminated when two consecutive estimates are closer than a threshold or two consecutive Q values are closer than a threshold.

Although in general a closed-form analytical expression of $Q\left(\boldsymbol{\vartheta}, \hat{\boldsymbol{\vartheta}}\right)$ cannot be found, for the case that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are jointly Gaussian related by a linear transformation

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{Y},$$

such an expression of $Q(\boldsymbol{\vartheta}, \hat{\boldsymbol{\vartheta}})$ can be derived. An example derivation of the EM algorithm for DOA estimation is given below, on which the derivation of the SAGE algorithm for self-calibration will be based.

Let the augmented data

$$\boldsymbol{y}(k) = \begin{bmatrix} \boldsymbol{y}_1(k) \\ \boldsymbol{y}_2(k) \\ \vdots \\ \boldsymbol{y}_M(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1(\theta_1)\, s_1(k) \\ \boldsymbol{h}_2(\theta_2)\, s_2(k) \\ \vdots \\ \boldsymbol{h}_M(\theta_M)\, s_M(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{n}_1(k) \\ \boldsymbol{n}_2(k) \\ \vdots \\ \boldsymbol{n}_M(k) \end{bmatrix},$$

where $\boldsymbol{n}_m(k)$ are decomposed from $\boldsymbol{n}(k)$ under an independence constraint, i.e.

$$\sum_{m=1}^{M} \boldsymbol{n}_m(k) = \boldsymbol{n}(k)$$

and

$$\mathrm{E}\left\{\boldsymbol{n}_m(k)\, \boldsymbol{n}_{m'}(k)\right\} = \nu_m \boldsymbol{I} \delta\left(m - m'\right),$$

where $\delta\left(m-m'\right)$ is the Kronecker delta function, and

$$\sum_{m=1}^{M}\nu_m=\nu.$$

A convenient choice is $\nu_m=\frac{1}{M}\nu$.

The relation between the augmented data and the observed data is

$$\boldsymbol{x}\left(k\right)=\boldsymbol{T}\boldsymbol{y}\left(k\right),$$

where

$$\boldsymbol{T}=\underbrace{\begin{bmatrix}\boldsymbol{I} & \cdots & \boldsymbol{I}\end{bmatrix}}_{M\text{ items}}.$$

Such an augmentation scheme leads to

$$\boldsymbol{y}\left(k\right)\sim\mathcal{G}\left(\boldsymbol{s}_h\left(k;\boldsymbol{\vartheta}\right),\boldsymbol{\Lambda}\right),$$

where

$$\boldsymbol{s}_h\left(k;\boldsymbol{\vartheta}\right)=\begin{bmatrix}\boldsymbol{h}_1\left(\theta_1\right)s_1\left(k\right)\\ \boldsymbol{h}_2\left(\theta_2\right)s_2\left(k\right)\\ \vdots\\ \boldsymbol{h}_M\left(\theta_M\right)s_M\left(k\right)\end{bmatrix}$$

and

$$\boldsymbol{\Lambda}=\begin{bmatrix}\nu_1\boldsymbol{I} & & & \mathbf{0}\\ & \nu_2\boldsymbol{I} & & \\ & & \ddots & \\ \mathbf{0} & & & \nu_M\boldsymbol{I}\end{bmatrix}. \tag{3.7}$$

The log-likelihood of $\boldsymbol{y}(k)$ for K snapshots is

$$\begin{aligned}
&\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\vartheta}) \\
&= \sum_{m=1}^{M}\sum_{k=1}^{K}\left\{\frac{1}{\nu_m}\left[\boldsymbol{y}_m(k)-\boldsymbol{h}_m(\theta_m)\,s_m(k)\right]^H\left[\boldsymbol{y}_m(k)-\boldsymbol{h}_m(\theta_m)\,s_m(k)\right]+N\log\pi+\log\nu_m\right\} \\
&= \sum_{m=1}^{M}\log f_{\boldsymbol{Y}_m}(\boldsymbol{y}_m;\boldsymbol{\vartheta}_m).
\end{aligned} \tag{3.8}$$

It is observed from Eqs. (3.7) and (3.8) that the independence of $\boldsymbol{n}_m(k)$ results in the decomposition of the log-likelihood of the augmented data into the sum of its components $\log f_{\boldsymbol{Y}_m}$ that are associated with the individual signals. Therefore the parameter $\boldsymbol{\vartheta}_m$ associated with each signal can be optimized individually.

The conditional expectation of $\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\vartheta})$, $Q\left(\boldsymbol{\vartheta},\hat{\boldsymbol{\vartheta}}\right)$, is equivalent to the computation [76]

$$\hat{\boldsymbol{y}}\left(k;\hat{\boldsymbol{\vartheta}}\right) = \mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x};\hat{\boldsymbol{\vartheta}}}\left\{\boldsymbol{y}(k)\right\}.$$

Using standard properties of the joint Gaussian distribution [77] [78], one has

$$\hat{\boldsymbol{y}}\left(k;\hat{\boldsymbol{\vartheta}}\right) = \hat{\boldsymbol{s}}_h\left(k;\hat{\boldsymbol{\vartheta}}\right) + \boldsymbol{\Lambda}\boldsymbol{T}^H\left(\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^H\right)^{-1}\left[\boldsymbol{x}(k) - \boldsymbol{T}\hat{\boldsymbol{s}}_h\left(k;\hat{\boldsymbol{\vartheta}}\right)\right],$$

which yields

$$\hat{\boldsymbol{y}}_m\left(k;\hat{\boldsymbol{\vartheta}}\right) = \boldsymbol{h}_m\left(\hat{\theta}_m\right)\hat{s}_m(k) + \frac{\nu_m}{\nu}\left[\boldsymbol{x}(k) - \boldsymbol{H}\left(\hat{\boldsymbol{\theta}}\right)\hat{\boldsymbol{s}}(k)\right].$$

Following the same technique for the derivation of Eq. (3.3), the maximization of $Q\left(\boldsymbol{\vartheta},\hat{\boldsymbol{\vartheta}}\right)$ is concentrated to

$$\hat{\theta}_m = \arg\max_{\theta_m}\left\{\frac{\boldsymbol{h}_m^H(\theta_m)\,\widehat{\boldsymbol{R}}_{\hat{\boldsymbol{y}}_m}\left(\hat{\boldsymbol{\theta}}\right)\boldsymbol{h}_m(\theta_m)}{\|\boldsymbol{h}_m(\theta_m)\|^2}\right\} \quad m = 1,2,\ldots,M,$$

where

$$\widehat{\boldsymbol{R}}_{\hat{\boldsymbol{y}}_m}\left(\hat{\boldsymbol{\vartheta}}\right) = \frac{1}{K}\sum_{k=1}^{K}\hat{\boldsymbol{y}}_m\left(k;\hat{\boldsymbol{\vartheta}}\right)\hat{\boldsymbol{y}}_m^H\left(k;\hat{\boldsymbol{\vartheta}}\right),$$

and the signals are updated by

$$\hat{s}_m(k) = \frac{1}{\|\boldsymbol{h}_m\left(\hat{\theta}_m\right)\|^2}\boldsymbol{h}_m^H\left(\hat{\theta}_m\right)\hat{\boldsymbol{y}}_m\left(k;\hat{\boldsymbol{\vartheta}}\right) \quad k = 1, 2, \ldots, K,\ m = 1, 2, \ldots, M.$$

3.2.2 SAGE algorithm

Since the convergence rate of an EM-type algorithm is inversely related to the Fisher information of its complete data space [74], it is desirable to devise less-informative complete data spaces for accelerating the algorithm. In the standard EM algorithm, one M step updates all the parameters, which leads to overly-informative complete data spaces and as a result the slow convergence rate of the EM algorithm [73]. A general framework for accelerating the EM algorithm is the SAGE algorithm. A SAGE algorithm sequentially updates small groups of the parameters, which leads to less-informative complete data spaces and consequently fast convergence rate [73]. As a variant of the classical EM algorithm, the SAGE algorithm not only accelerates the EM algorithm under mild conditions [79], but also the maintains EM algorithm's property of monotonic increase of the likelihood function.

Specifically, let S denote a subset of the parameter vector $\boldsymbol{\vartheta}$'s index set $\{1, 2, \ldots, D\}$, and $\tilde{S}$ denote the complement of S intersected with $\{1, 2, \ldots, D\}$. Then let $\boldsymbol{\vartheta}_S$ denote the sub-vector indexed out by S from $\boldsymbol{\vartheta}$, and $\boldsymbol{\vartheta}_{\tilde{S}}$ the complementary sub-vector. For example, if $D = 6$ and $S = \{1, 3\}$, then $\boldsymbol{\vartheta}_S = [\vartheta_1, \vartheta_3]^T$ and $\boldsymbol{\vartheta}_{\tilde{S}} = [\vartheta_2, \vartheta_4, \vartheta_5, \vartheta_6]^T$.

The random vector $\boldsymbol{Z}^S$ associated with S is an admissible hidden data space with respect to $\boldsymbol{\vartheta}_S$ for $f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{\vartheta})$ if the joint density of $\boldsymbol{Z}^S$ and $\boldsymbol{X}$ satisfies

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\vartheta}\right) = f_{\boldsymbol{X}|\boldsymbol{Z}=\boldsymbol{z}}\left(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\vartheta}_{\tilde{S}}\right) f_{\boldsymbol{Z}}\left(\boldsymbol{z}; \boldsymbol{\vartheta}\right),$$

where $f_{\boldsymbol{Z}}\left(\boldsymbol{z}; \boldsymbol{\vartheta}\right)$ is the probability density of $\boldsymbol{Z}^S$. In words, the conditional distribution $f_{\boldsymbol{X}|\boldsymbol{Z}=\boldsymbol{z}}$ $\left(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\vartheta}_{\tilde{S}}\right)$ must be independent of $\boldsymbol{\vartheta}_S$, or in other words $\boldsymbol{Z}^S$ must be a complete data space for $\boldsymbol{\vartheta}_S$ given that $\boldsymbol{\vartheta}_{\tilde{S}}$ is known.

Dividing the parameter vector $\boldsymbol{\vartheta}$ into a sub-vector $\boldsymbol{\vartheta}_S$ and its complementary sub-vector $\boldsymbol{\vartheta}_{\tilde{S}}$, the conditional expectation of the likelihood of the complete data $\boldsymbol{Z}^S$ is calculated as

$$\begin{aligned}Q^S\left(\boldsymbol{\vartheta}_S;\hat{\boldsymbol{\vartheta}}\right) = Q^S\left(\boldsymbol{\vartheta}_S;\hat{\boldsymbol{\vartheta}}_S,\hat{\boldsymbol{\vartheta}}_{\tilde{S}}\right) &\triangleq \mathrm{E}_{\boldsymbol{Z}^S|\boldsymbol{X}=\boldsymbol{x};\hat{\boldsymbol{\vartheta}}}\left\{\log f_{\boldsymbol{Z}^S}\left(\boldsymbol{z}^S;\boldsymbol{\vartheta}_S,\hat{\boldsymbol{\vartheta}}_{\tilde{S}}\right)\right\}\\ &= \int\left[\log f_{\boldsymbol{Z}^S}\left(\boldsymbol{z}^S;\boldsymbol{\vartheta}_S,\hat{\boldsymbol{\vartheta}}_{\tilde{S}}\right)\right] f_{\boldsymbol{Z}^S}\left(\boldsymbol{z}^S|\boldsymbol{X}=\boldsymbol{x};\hat{\boldsymbol{\vartheta}}\right)d\boldsymbol{z}^S.\end{aligned} \tag{3.9}$$

The SAGE algorithm can be summarized as in Alg. 2.

Algorithm 2: SAGE algorithm

```
for $i = 0, 1, \ldots$ do
    Choose an index subset $S = S^{[i]}$
    Choose an admissible hidden data space $\boldsymbol{Z}^{S^{[i]}}$ for $\boldsymbol{\vartheta}_{S^{[i]}}$
    E-step: compute according to Eq. (3.9)
        $Q^{S^{[i]}}\left(\boldsymbol{\vartheta}_{S^{[i]}};\hat{\boldsymbol{\vartheta}}^{[i]}\right)$
    M-step:
        $\hat{\boldsymbol{\vartheta}}^{[i+1]}_{S^{[i]}} = \arg\max_{\boldsymbol{\vartheta}_{S^{[i]}}} Q^{S^{[i]}}\left(\boldsymbol{\vartheta}_{S^{[i]}};\hat{\boldsymbol{\vartheta}}^{[i]}\right),$
        $\hat{\boldsymbol{\vartheta}}^{[i+1]}_{\tilde{S}^{[i]}} = \hat{\boldsymbol{\vartheta}}^{[i]}_{\tilde{S}^{[i]}}$
end
```

The proof of monotonicity is given in [73]. In the following section a SAGE-based algorithm for array shape self-calibration is constructed.

3.3 A SAGE-based algorithm for array shape self-calibration

First of all, as the parameter vector $\boldsymbol{\vartheta}$ is divided into sub-vectors that are estimated sequentially, it is useful to introduce an indexing system that involves a *cycle* in addition to an *iteration* [79] [80]. A cycle consists of an E-step followed by an M-step, which will be followed immediately by a new E-step. The new E-step is the beginning of the next cycle. An iteration consists of one or more cycles.

A subset $\boldsymbol{\vartheta}_c$ of all unknown parameters is updated at one cycle. All subsets are updated once in one iteration. In the following, a SAGE based algorithm is derived which updates DOA parameters and perturbation parameters sequentially.

Let $\boldsymbol{\vartheta}$ summarize all unknown parameters

$$\boldsymbol{\vartheta} = \left[\boldsymbol{\vartheta}_1^T, \boldsymbol{\vartheta}_2^T, \ldots \boldsymbol{\vartheta}_M^T, \boldsymbol{\vartheta}_{M+1}^T, \boldsymbol{\vartheta}_{M+2}^T, \ldots, \boldsymbol{\vartheta}_{M+N}^T, \vartheta_{M+N+1}\right]^T,$$

where $\boldsymbol{\vartheta}_c = [\theta_c, s_c(k), k = 1, 2, \ldots, K]^T$, $(c = 1, 2, \ldots, M)$ includes the DOA and signal parameters associated with the cth source. The sensor perturbation parameters are divided into N subsets with $\boldsymbol{\vartheta}_{M+n} = [\Delta x_n, \Delta y_n]^T$, $n = 1, 2, \ldots, N$. The last subset $\vartheta_{M+n+1} = \nu$ contains the noise parameter.

As mentioned previously, SAGE updates a subset of parameters at each cycle. It is natural to choose different augmented data $\boldsymbol{z}_c$ for each cycle. For DOA estimation, an augmentation scheme similar to that in [79] is used:

$$\boldsymbol{z}_c(k) = \boldsymbol{h}_c(\theta_c, \boldsymbol{\delta}) s_c(k) + \boldsymbol{n}(k), \quad c = 1, 2, \ldots, M. \tag{3.10}$$

Under this augmentation scheme, the conditional expectation of $\boldsymbol{z}_c$ is carried out as follows: As

$$\boldsymbol{n} \sim \mathcal{G}(\boldsymbol{0}, \nu \boldsymbol{I}),$$

it follows that

$$\boldsymbol{z}_c \sim \mathcal{G}(\boldsymbol{h}_c(\theta_c, \boldsymbol{\delta}) s_c, \nu \boldsymbol{I}).$$

The dependence on k has been suppressed for notational simplicity. The covariance matrix between $\boldsymbol{z}_c$ and $\boldsymbol{x}$ is

$$\text{Cov}\left(\boldsymbol{z}_c \boldsymbol{x}^H\right) = \text{E}\left(\boldsymbol{n}\boldsymbol{n}^H\right) = \nu \boldsymbol{I},$$

therefore

$$\begin{bmatrix} \boldsymbol{z}_c \\ \boldsymbol{x} \end{bmatrix} \sim \mathcal{G}\left(\begin{pmatrix} \boldsymbol{h}_c(\theta_c, \boldsymbol{\delta}) s_c \\ \boldsymbol{H}(\boldsymbol{\theta}, \boldsymbol{\delta}) \boldsymbol{s} \end{pmatrix}, \nu \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I} \end{bmatrix}\right).$$

Using the property of joint Gaussian distribution in appendix A,

$$\begin{aligned} \hat{\boldsymbol{z}}_c &= \text{E}_{\boldsymbol{Z}_c|\boldsymbol{X};\hat{\boldsymbol{\vartheta}}}(\boldsymbol{z}_c) \\ &= \boldsymbol{h}_c\left(\hat{\theta}_c, \hat{\boldsymbol{\delta}}\right) \hat{s}_c + \nu \boldsymbol{I} (\nu \boldsymbol{I})^{-1} \left[\boldsymbol{x} - \boldsymbol{H}\left(\hat{\theta}_c, \hat{\boldsymbol{\delta}}\right) \hat{\boldsymbol{s}}\right] \\ &= \boldsymbol{h}_c\left(\hat{\theta}_c, \hat{\boldsymbol{\delta}}\right) \hat{s}_c + \left[\boldsymbol{x} - \boldsymbol{H}\left(\hat{\theta}_c, \hat{\boldsymbol{\delta}}\right) \hat{\boldsymbol{s}}\right]. \end{aligned} \tag{3.11}$$

For sensor location and noise parameters, a straightforward data augmentation is chosen:

$$\boldsymbol{z}_c\left(k\right)=\boldsymbol{x}\left(k\right),\quad c=M+1,M+2,\ldots,M+N+1, \tag{3.12}$$

or put in words, using the latest DOA estimate and the array observation, array calibration is carried out sensor by sensor. The required initial DOA estimate can be attained by subspace methods such as multiple signal classification (MUSIC) or the standard maximum likelihood (ML) approach using nominal sensor locations.

Let $\hat{\boldsymbol{\vartheta}}^{[i,c]}$ denote the estimate of the cth cycle and ith iteration, and $\hat{\boldsymbol{\vartheta}}^{[i,0]}=\hat{\boldsymbol{\vartheta}}^{[i-1,M+N]}$. Given an initial estimate $\hat{\boldsymbol{\vartheta}}^{[0,0]}$ and the augmentation scheme Eqs. (3.10) and (3.12), the ith iteration of the SAGE algorithm proceeds as follows.

I. Update DOA parameter

For $c=1,2,\ldots,M$

E-step: calculate the conditional expectation of the augmented log-likelihood function

$$Q\left(\boldsymbol{\vartheta}_c,\hat{\boldsymbol{\vartheta}}^{[i,c-1]}\right)=\mathrm{E}_{\boldsymbol{Z}_c|\boldsymbol{X}=\boldsymbol{x};\hat{\boldsymbol{\vartheta}}^{[i,c-1]}}\left[\log f_{\boldsymbol{Z}_c}\left(\boldsymbol{z}_c;\boldsymbol{\vartheta}_c\right)\right],$$

which according to Eq. (3.11) is equivalent to calculating

$$\begin{aligned}&\hat{\boldsymbol{z}}_c\left(k;\boldsymbol{\vartheta}^{[i,c-1]},\boldsymbol{\delta}^{[i-1]}\right)\\&=\boldsymbol{h}_c\left(\vartheta_c^{[i,c-1]},\boldsymbol{\delta}^{[i-1]}\right)s_c^{[i,c-1]}(k)+\boldsymbol{x}(k)-\boldsymbol{H}\left(\boldsymbol{\vartheta}^{[i,c-1]},\boldsymbol{\delta}^{[i-1]}\right)\boldsymbol{s}^{[i,c-1]}(k)\end{aligned}$$

and

$$\widehat{\boldsymbol{R}}_{\hat{\boldsymbol{z}}_c}\left(\boldsymbol{\vartheta}^{[i,c-1]},\boldsymbol{\delta}^{[i-1]}\right)=\frac{1}{K}\sum_{k=1}^{K}\hat{\boldsymbol{z}}_c\left(k;\boldsymbol{\vartheta}^{[i,c-1]},\boldsymbol{\delta}^{[i-1]}\right)\hat{\boldsymbol{z}}_c\left(k;\boldsymbol{\vartheta}^{[i,c-1]},\boldsymbol{\delta}^{[i-1]}\right)^H$$

M-step: update $\boldsymbol{\vartheta}_c=\left[\vartheta_c,s_c\left(k\right),k=1,2,\ldots,K\right]^T$

$$\theta_c^{[i,c]}=\arg\max_{\theta_c}\frac{\boldsymbol{h}_c^H(\theta_c,\boldsymbol{\delta}^{[i-1]})\widehat{\boldsymbol{R}}_{\hat{\boldsymbol{z}}_c}\left(\boldsymbol{\vartheta}^{[i,c-1]},\boldsymbol{\delta}^{[i-1]}\right)\boldsymbol{h}_c(\theta_c,\boldsymbol{\delta}^{[i-1]})}{\|\boldsymbol{h}_c(\theta_c,\boldsymbol{\delta}^{[i-1]})\|^2}$$

$$s_c(k;\theta_c^{[i,c]},\boldsymbol{\delta}^{[i-1]}) = \frac{1}{\|\boldsymbol{h}_c(\theta_c^{[i,c]},\boldsymbol{\delta}^{[i-1]})\|^2}\boldsymbol{h}_c^H(\theta_c^{[i,c]},\boldsymbol{\delta}^{[i-1]})\hat{\boldsymbol{z}}_c\left(k;\boldsymbol{\vartheta}^{[i,c-1]},\boldsymbol{\delta}^{[i-1]}\right)$$

where $\boldsymbol{\delta}^{[i-1]}$ represent the perturbation parameters from the previous iteration.

II. Update sensor location parameter

For $c = M+1, M+2, \ldots, M+N$

E-step:

$$Q\left(\boldsymbol{\vartheta}_c, \hat{\boldsymbol{\vartheta}}^{[i,c-1]}\right) = \mathrm{E}_{\boldsymbol{X}|\boldsymbol{X}=\boldsymbol{x};\hat{\boldsymbol{\vartheta}}^{[i,c-1]}}\left[\log f_{\boldsymbol{X}}\left(\boldsymbol{x};\boldsymbol{\vartheta}_c\right)\right] \tag{3.13}$$

which is equivalent to calculating

$$\widehat{\boldsymbol{R}} = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{x}\left(k\right)\boldsymbol{x}^H\left(k\right) \tag{3.14}$$

M-step: update $\boldsymbol{\vartheta}_c = \left[\Delta x_{c-M}, \Delta y_{c-M}\right]^T$

$$\left[\Delta x_{c-M}^{[i,c]}, \Delta y_{c-M}^{[i,c]}\right]^T = \arg \max_{\substack{\Delta x_{c-M}\\ \Delta y_{c-M}}} \operatorname{tr}\left[\boldsymbol{P}_{\boldsymbol{H}\left(\boldsymbol{\theta}^{[i,M]},\boldsymbol{\delta}^{[i,c-1]}\right)}\widehat{\boldsymbol{R}}\right] \tag{3.15}$$

The perturbation vector

$$\boldsymbol{\delta}^{[i,c-1]} = \left[\Delta x_1^{[i,c-1]}, \Delta y_1^{[i,c-1]}, \ldots, \Delta x_{c-M}, \Delta y_{c-M}, \ldots, \Delta x_N^{[i,c-1]}, \Delta y_N^{[i,c-1]}\right]^T$$

consists of the error parameters of the $c-M$th sensor $\Delta x_{c-M}, \Delta y_{c-M}$ and other $2N-2$ parameters fixed at the latest estimates. The sample covariance matrix is given by Eq. (3.14). Note that Eq. (3.15) is simply the concentrated log-likelihood function for array observations $\boldsymbol{x}(k)$ with the DOA parameters fixed at current estimate $\boldsymbol{\theta}^{[i,M]}$. If the first sensor position is assumed to be known, i.e. $\Delta x_1 = \Delta y_1 = 0$, the corresponding cycle can be skipped. The latest estimate is simply replaced by the known value $\mathbf{0}$. For the known direction condition, $\Delta y_2 = 0$, the perturbation parameter Δx_2 requires only a one dimensional search.

III. Update noise parameter

For $c = M+N+1$

The *E-step* is similar to Eq. (3.13). In the *M-step*, the noise parameter is updated by a closed form expression as in standard ML estimation:

$$\nu^{[i,c]} = \frac{\mathrm{tr}\left[\boldsymbol{P}_{\boldsymbol{H}\left(\boldsymbol{\theta}^{[i,M]},\boldsymbol{\delta}^{[i,M+N]}\right)}\widehat{\boldsymbol{R}}\right]}{N}$$

In summary, the SAGE algorithm decouples the DOA and perturbation parameters and updates them alternately. The multi-dimensional search procedure is greatly simplified to one and two dimensional search procedures at the expense of iterations.

3.4 CRB

In this section, the expression of CRB for self-calibration is given. The CRB defines the lowest mean squared error (MSE) that any unbiased estimator can attain asymptotically. The covariance matrix of the estimation errors is denoted by

$$\boldsymbol{C}(\boldsymbol{\vartheta}) = \mathrm{E}[(\hat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta})(\hat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta})^T]. \tag{3.16}$$

The vector-parameter CRB states that

$$\boldsymbol{C}(\boldsymbol{\vartheta}) \geqslant \boldsymbol{B}_{\mathrm{CR}}(\boldsymbol{\vartheta}) \triangleq \boldsymbol{J}^{-1},$$

for any unbiased estimate of $\boldsymbol{\vartheta}$. That $\boldsymbol{C}(\boldsymbol{\vartheta})$ is greater than or equal to CRB $\boldsymbol{B}_{\mathrm{CR}}(\boldsymbol{\vartheta})$ means that $\boldsymbol{C}(\boldsymbol{\vartheta}) - \boldsymbol{B}_{\mathrm{CR}}(\boldsymbol{\vartheta})$ is a positive semi-definite matrix. The matrix $\boldsymbol{J}$ is commonly referred to as Fisher's information matrix (FIM).

The i, jth element of $\boldsymbol{J}$ is

$$J_{ij} \triangleq \mathrm{E}\left[\frac{\partial L_{\boldsymbol{X}}(\boldsymbol{\vartheta})}{\partial \vartheta_i}\frac{\partial L_{\boldsymbol{X}}(\boldsymbol{\vartheta})}{\partial \vartheta_j}\right] = -\mathrm{E}\left[\frac{\partial^2 L_{\boldsymbol{X}}(\boldsymbol{\vartheta})}{\partial \vartheta_i \partial \vartheta_j}\right], \tag{3.17}$$

where $L_{\boldsymbol{X}}(\boldsymbol{\vartheta})$ is the log-likelihood function of random variable $\boldsymbol{X}$ parameterized by $\boldsymbol{\vartheta}$, and ϑ_i and ϑ_j are the ith and jth elements of $\boldsymbol{\vartheta}$, respectively.

Taking the second partial derivative and then an expectation on the log-likelihood function for a single snapshot k

$$\begin{aligned} L_X(\boldsymbol{\vartheta}) &= \ln p_{\boldsymbol{X}|\boldsymbol{\vartheta}}(\boldsymbol{x}(k)) \\ &= -\ln\det\left[\pi \boldsymbol{C}_{\boldsymbol{X}}(\boldsymbol{\vartheta})\right] - \left[\boldsymbol{x}^H(k) - \boldsymbol{m}^H(\boldsymbol{\vartheta})\right]{\boldsymbol{C}_{\boldsymbol{X}}}^{-1}(\boldsymbol{\vartheta})\left[\boldsymbol{x}(k) - \boldsymbol{m}(\boldsymbol{\vartheta})\right], \end{aligned}$$

where $\boldsymbol{C}_{\boldsymbol{X}}(\boldsymbol{\vartheta})$ is the covariance matrix, $\boldsymbol{x}(k)$ the kth realization and $\boldsymbol{m}(\boldsymbol{\vartheta})$ the mean of $\boldsymbol{X}$, by definition (3.17) one has

$$J_{ij} = -\text{tr}\left[\frac{\partial \boldsymbol{C}_{\boldsymbol{X}}^{-1}(\boldsymbol{\vartheta})}{\partial \vartheta_i}\frac{\partial \boldsymbol{C}_{\boldsymbol{X}}^{-1}(\boldsymbol{\vartheta})}{\partial \vartheta_j}\right] + 2\text{Re}\left[\frac{\partial \boldsymbol{m}^H(\boldsymbol{\vartheta})}{\partial \vartheta_i}\boldsymbol{C}_{\boldsymbol{X}}^{-1}(\boldsymbol{\vartheta})\frac{\partial \boldsymbol{m}(\boldsymbol{\vartheta})}{\partial \vartheta_j}\right]. \tag{3.18}$$

In some cases the mean is either zero or not a function of $\boldsymbol{\vartheta}$ so that the second term of Eq. (3.18) can be omitted, then

$$J_{ij} = -\text{tr}\left[\frac{\partial \boldsymbol{C}_{\boldsymbol{X}}^{-1}(\boldsymbol{\vartheta})}{\partial \vartheta_i}\frac{\partial \boldsymbol{C}_{\boldsymbol{X}}^{-1}(\boldsymbol{\vartheta})}{\partial \vartheta_j}\right]. \tag{3.19}$$

The expression of Eq. (3.19) for self-calibration is given in [19] as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{D}_{\boldsymbol{\theta}}^H \boldsymbol{P}_{\boldsymbol{H}}^{\perp} \boldsymbol{D}_{\boldsymbol{\theta}} \odot \boldsymbol{\Sigma}^T & \boldsymbol{D}_{\boldsymbol{\theta}}^H \boldsymbol{P}_{\boldsymbol{H}}^{\perp} \odot (\boldsymbol{D}_{\boldsymbol{x}}\boldsymbol{\Sigma})^T & \boldsymbol{D}_{\boldsymbol{\theta}}^H \boldsymbol{P}_{\boldsymbol{H}}^{\perp} \odot (\boldsymbol{D}_{\boldsymbol{y}}\boldsymbol{\Sigma})^T \\ (\boldsymbol{D}_{\boldsymbol{x}}\boldsymbol{\Sigma}) \odot \left(\boldsymbol{D}_{\boldsymbol{\theta}}^H \boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right)^T & \left(\boldsymbol{D}_{\boldsymbol{x}}\boldsymbol{\Sigma}\boldsymbol{D}_{\boldsymbol{x}}^H\right) \odot \left(\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right)^T & \left(\boldsymbol{D}_{\boldsymbol{x}}\boldsymbol{\Sigma}\boldsymbol{D}_{\boldsymbol{y}}^H\right) \odot \left(\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right)^T \\ (\boldsymbol{D}_{\boldsymbol{y}}\boldsymbol{\Sigma}) \odot \left(\boldsymbol{D}_{\boldsymbol{\theta}}^H \boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right)^T & \left(\boldsymbol{D}_{\boldsymbol{y}}\boldsymbol{\Sigma}\boldsymbol{D}_{\boldsymbol{x}}^H\right) \odot \left(\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right)^T & \left(\boldsymbol{D}_{\boldsymbol{y}}\boldsymbol{\Sigma}\boldsymbol{D}_{\boldsymbol{y}}^H\right) \odot \left(\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right)^T \end{bmatrix}, \tag{3.20}$$

in which

$$\boldsymbol{\Sigma} = \boldsymbol{S}\boldsymbol{H}^H\boldsymbol{R}^{-1}\boldsymbol{H}\boldsymbol{S},$$

$$\boldsymbol{D}_{\boldsymbol{x}} = \begin{pmatrix} \frac{\partial \boldsymbol{h}_1}{\partial \Delta x_1} \\ \frac{\partial \boldsymbol{h}_2}{\partial \Delta x_2} \\ \vdots \\ \frac{\partial \boldsymbol{h}_N}{\partial \Delta x_N} \end{pmatrix},$$

where $\boldsymbol{h}_n$ is the nth row of $\boldsymbol{H}$, and

$$\boldsymbol{D}_{\boldsymbol{y}} = \begin{pmatrix} \frac{\partial \boldsymbol{h}_1}{\partial \Delta y_1} \\ \frac{\partial \boldsymbol{h}_2}{\partial \Delta y_2} \\ \vdots \\ \frac{\partial \boldsymbol{h}_N}{\partial \Delta y_N} \end{pmatrix}.$$

$$\boldsymbol{D}_{\boldsymbol{\theta}} = \left(\begin{array}{cccc} \frac{\partial \boldsymbol{g}_1}{\partial \theta_1} & \frac{\partial \boldsymbol{g}_2}{\partial \theta_2} & \cdots & \frac{\partial \boldsymbol{g}_M}{\partial \theta_M} \end{array} \right),$$

where $\boldsymbol{g}_m$ is the mth column of $\boldsymbol{H}$.

For the far-field self-calibration case,

$$\boldsymbol{D}_{\boldsymbol{\theta}} = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\delta}=0} = j\pi \boldsymbol{H} \odot (\boldsymbol{P}\boldsymbol{\Theta})\,|_{\boldsymbol{\delta}=0},$$

where

$$\boldsymbol{P} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix},$$

$$\boldsymbol{\Theta} = \begin{bmatrix} \cos\theta_1 & \cos\theta_2 & \cdots & \cos\theta_M \\ -\sin\theta_1 & -\sin\theta_2 & \cdots & -\sin\theta_M \end{bmatrix},$$

$$\boldsymbol{D}_{\boldsymbol{x}} = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{\delta}=0} = j\pi \boldsymbol{H} \begin{bmatrix} \sin\theta_1 & & & \mathbf{0} \\ & \sin\theta_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \sin\theta_M \end{bmatrix} \Bigg|_{\boldsymbol{\delta}=\mathbf{0}},$$

and

$$\boldsymbol{D}_{\boldsymbol{y}} = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{y}}\bigg|_{\boldsymbol{\delta}=0} = j\pi \boldsymbol{H} \begin{bmatrix} \cos\theta_1 & & & \mathbf{0} \\ & \cos\theta_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \cos\theta_M \end{bmatrix} \Bigg|_{\boldsymbol{\delta}=\mathbf{0}}.$$

Note that $\Delta x_1, \Delta y_1$ and Δy_2 have been set to zero to fulfil the identifiability condition: as a result the related elements in $\boldsymbol{J}$ should be deleted since they are not parameters any more. More specifically, if Eq. (3.20) is notationally simplified as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_{\boldsymbol{\theta}\boldsymbol{\theta}} & \boldsymbol{J}_{\boldsymbol{\theta}\boldsymbol{x}} & \boldsymbol{J}_{\boldsymbol{\theta}\boldsymbol{y}} \\ \boldsymbol{J}_{\boldsymbol{\theta}\boldsymbol{x}}^H & \boldsymbol{J}_{\boldsymbol{x}\boldsymbol{x}} & \boldsymbol{J}_{\boldsymbol{x}\boldsymbol{y}} \\ \boldsymbol{J}_{\boldsymbol{\theta}\boldsymbol{y}}^H & \boldsymbol{J}_{\boldsymbol{x}\boldsymbol{y}}^H & \boldsymbol{J}_{\boldsymbol{y}\boldsymbol{y}} \end{bmatrix}, \tag{3.21}$$

where the nine sub-matrices in Eq. (3.21) correspond to the nine sub-matrices in Eq. (3.20), for example,

$$\boldsymbol{J_{\theta\theta}} = \boldsymbol{D}_{\boldsymbol{\theta}}^{H}\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\boldsymbol{D}_{\boldsymbol{\theta}} \odot \boldsymbol{\Sigma}^{T}.$$

Then the following should be deleted: the first column in $\boldsymbol{J_{\theta x}}$ and $\boldsymbol{J_{xx}}$, the first and second columns in $\boldsymbol{J_{\theta y}}$, $\boldsymbol{J_{xy}}$ and $\boldsymbol{J_{yy}}$; the first row in $\boldsymbol{J_{\theta x}}$ and $\boldsymbol{J_{xy}}$, the first and second rows in $\boldsymbol{J_{yy}}$.

3.5 Simulation results

In the simulation, a nominally uniform circular array of 6 sensors is considered. The inter-sensor spacing is half a wavelength $\lambda/2$. The sensor's nominal positions are perturbed by a Gaussian random displacement with zero mean and a standard deviation σ_p. The random displacement is generated once and kept fixed throughout each experiment. The random displacements at $\sigma_p = \lambda/2$ are collected in Tab. 3.1. Δx_1, Δy_1 and Δy_2 are set to zero to fulfil the identifiability condition.

n	Δx_n $(\lambda/2)$	Δy_n $(\lambda/2)$
1	0	0
2	−0.3775	0
3	−0.2959	0.9409
4	−1.4751	−0.9921
5	−0.2340	0.2120
6	0.1184	0.2379

Table 3.1: *The random displacements at* $\sigma_p = \lambda/2$.

3.5.1 An example of self-calibration

In this section, a typical self-calibration process is presented to illustrate the behavior of the algorithm. In the self-calibration setup, the narrowband signals are emitted by three sources of equal power located at $\boldsymbol{\theta}_0 = [-35°, 0°, 35°]^T$. The standard deviation of sensor position errors is $\sigma_p = 0.25\lambda$ which corresponds to 50% inter-sensor spacing. The number of snapshots $K = 50$. The signal-to-noise ratio (SNR), defined as $10\log[\mathrm{E}_k[|s_m(k)|^2]/\nu]$, is 12 dB. The nominal sensor positions are used to initialize the algorithm, i.e. $\boldsymbol{\delta}^{[0]} = \mathbf{0}$. The initial DOA estimate is chosen as $\boldsymbol{\theta}^{[0]} = [-34°, 4.1°, 41.6°]$. The maximum number of iterations is 200.

The algorithm is terminated if the relative change of parameters is less than 0.01% between two consecutive iterations or the maximum number of iterations is reached.

Iteration	$\hat{\theta}_1$ (deg.)	$\hat{\theta}_2$ (deg.)	$\hat{\theta}_3$ (deg.)	Iteration	$\hat{\theta}_1$ (deg.)	$\hat{\theta}_2$ (deg.)	$\hat{\theta}_3$ (deg.)
0	-34.80	4.10	41.60	11	-31.58	10.99	40.97
1	-31.01	7.83	40.85	13	-32.20	9.92	40.82
2	-26.44	13.80	40.03	15	-32.68	8.77	40.61
3	-26.26	13.81	39.93	17	-33.09	7.63	40.38
4	-26.39	13.45	40.17	19	-33.44	6.55	40.14
5	-27.25	13.12	40.48	21	-33.75	5.57	39.89
6	-28.34	12.86	40.73	23	-34.02	4.72	39.65
7	-29.30	12.58	40.89	25	-34.26	3.99	39.42
8	-30.08	12.26	40.98	79	-35.61	0.10	36.98
9	-30.69	11.89	41.02	80	-35.61	0.10	36.98

Table 3.2: *Estimated DOA's of selected iterations.*

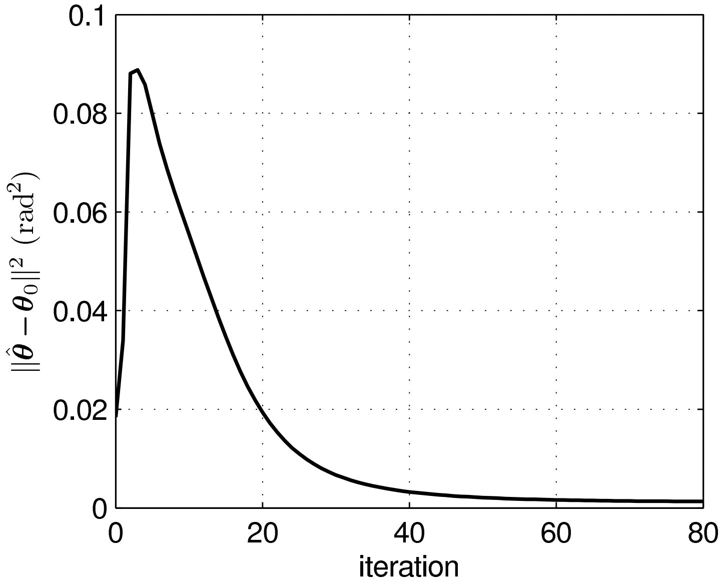

Figure 3.2: *DOA errors vs. iteration number*

Table 3.2 shows that the SAGE algorithm converged in 80 iterations, terminating when the relative change of $\boldsymbol{\theta}$ in the last two iterations is less than the threshold. The squared error (SE) sum of DOA, $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\|^2$, is reduced from 1.84×10^{-2} rad^2 at initialization to 0.13×10^{-2} rad^2 at termination, which is a 93% reduction. Fig. 3.2 shows the SE sum of DOA as a function of iterations. It is observed that the first 3 iterations increased the DOA error, but increased the objective function significantly as shown in Fig. 3.4. The monotonic increase of objective function is guaranteed by the characteristic of the SAGE algorithm.

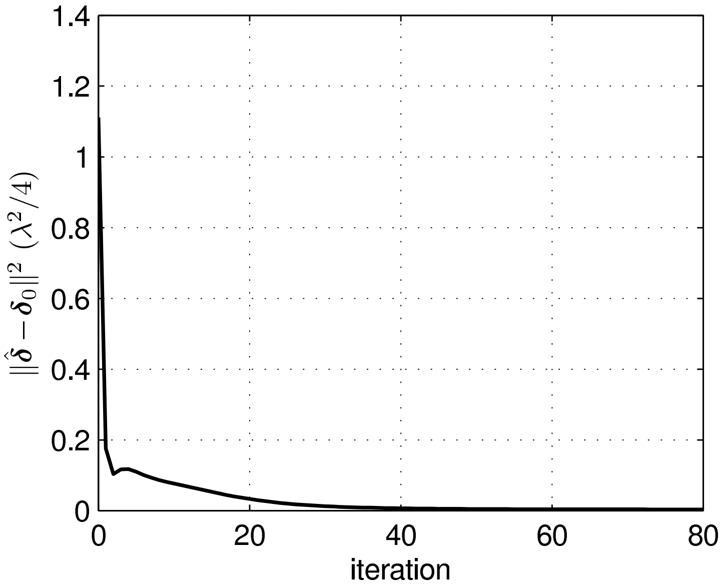

Figure 3.3: *Sensor location errors vs. iteration number.*

The SE sum of sensor locations, $\|\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}_0\|^2$, is dramatically reduced by 99.7%, from 1.1115 $\lambda^2/4$ of nominal sensor locations to 0.0033 $\lambda^2/4$ of estimated sensor locations. Fig. 3.3 shows it as a function of iterations. A fluctuation of sensor location error is also observed between the second and the third iterations, corresponding to the fluctuation of DOA error as shown in Fig. 3.2. However the SE sum of sensor locations decreases monotonically after that.

3.5.2 Monte Carlo experiments

The purpose of the Monte Carlo (MC) experiments is twofold. Firstly, to compare the performance of the proposed procedure with the existing procedures, i.e. the standard SAGE algorithm [79] and the WF1 procedure. All three procedures optimize the same CML objective function in Eq. (3.3). Secondly, to demonstrate the statistical efficiency of the proposed procedure by comparing it with the CRB. The MC experiments are carried out in the following three cases.

3.5.2.1 Well-separated sources

In the first case, the narrowband signals are generated by three widely separated sources of equal power located at $\boldsymbol{\theta}_0 = [-35°, 0°, 35°]$. The deviation of sensor position errors is $\sigma_p = 0.1\lambda$ which corresponds to 20% intersensor spacing. The number of snapshots $K = 50$. The SNR

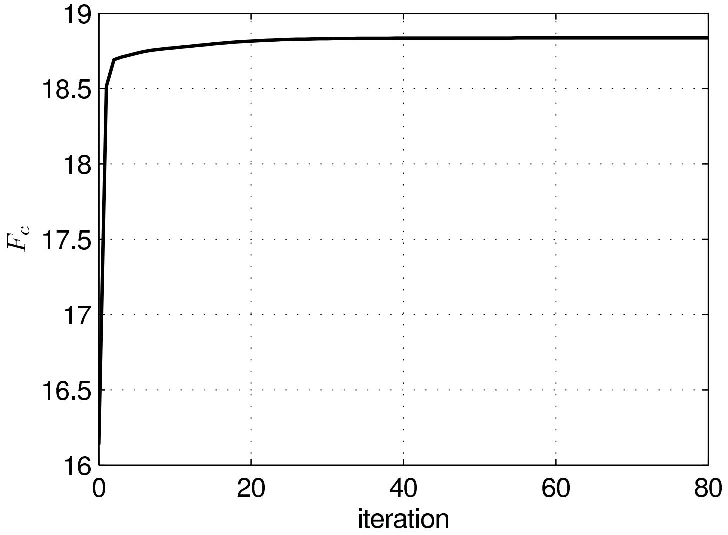

Figure 3.4: *Objective function value convergence.*

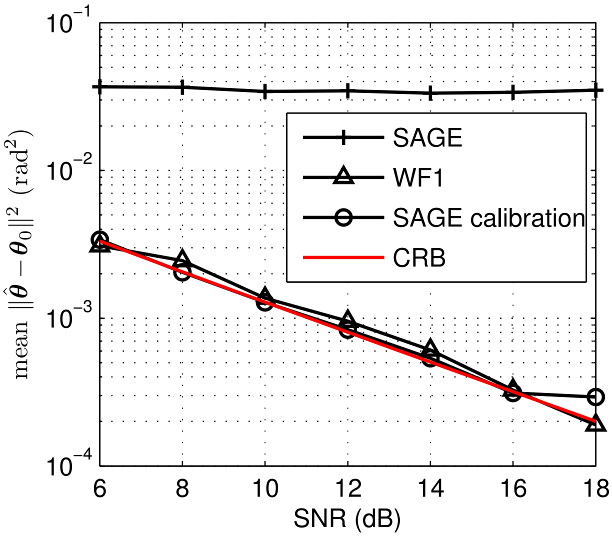

Figure 3.5: *Mean* $\|\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}_0\|^2$ *vs. SNR. True DOA parameters* $\boldsymbol{\theta}_0 = [-35^\circ, 0^\circ, 35^\circ]$, $\sigma_p = 0.1\lambda$.

ranges from 8 dB to 18 dB with a 2 dB step. For each SNR, 100 MC trials are performed. The nominal sensor positions are used to initialize the algorithm, i.e. $\boldsymbol{\delta}^{[0]} = \mathbf{0}$. For a fair comparison, a fixed initial DOA estimate $\boldsymbol{\theta}^{[0]} = [-34°, 4.1°, 41.6°]$ is chosen. The maximum number of iterations is 200. The algorithm is terminated if the relative change of parameters is less than 0.01% between two consecutive iterations or the maximum number of iterations is reached.

Fig. 3.5 shows the MSE sum of DOA estimation obtained from averaging the distance between the estimated and the true DOA parameters, $\|\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}_0\|^2$. One can easily observe that the standard conditional maximum likelihood – space generalized expectation-maximization (CML-SAGE) algorithm fails to provide any useful information. Both self-calibration techniques lead to significant improvement in estimation accuracy. The SAGE algorithm and WF1 method perform similarly over the entire SNR range, their MSE sums being close to the CRB.

3.5.2.2 Closely-located sources

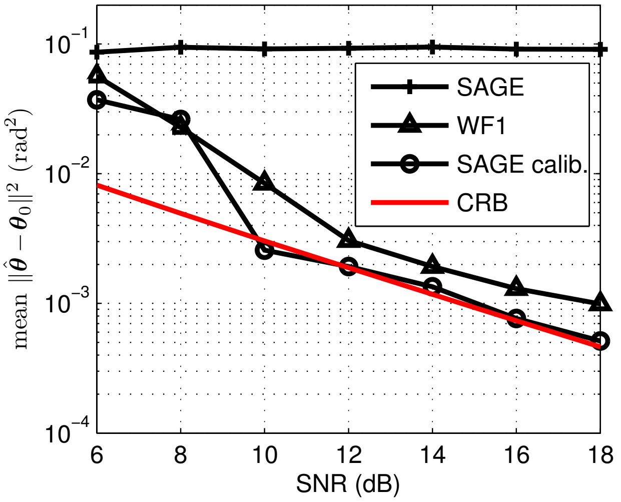

Figure 3.6: *Mean* $\|\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}_0\|^2$ *vs. SNR. True DOA parameters* $\boldsymbol{\theta}_0 = [-35°, 12°, 30°]$, $\sigma_p = 0.1\lambda$.

In the second case, closely located sources with $\boldsymbol{\theta}_0 = [-35°, 12°, 30°]^T$ are considered. The initial value is given by $\boldsymbol{\theta}^{[0]} = [-29.8°, 16.1°, 33.6°]$. Other parameters are kept the same. The MSE's presented in Fig. 3.6 show that the SAGE algorithm has an overall higher estimation accuracy than the WF1 method in this more critical situation. The MSE sum of WF1 approaches the CRB when SNR is greater than or equal to 10 dB. However, both methods require a higher

SNR than in the first experiment to provide reasonable estimates. Similarly to Fig. 3.5, the standard CML-SAGE approach fails totally when sensor positions deviate from the nominal values.

3.5.2.3 Large sensor location errors

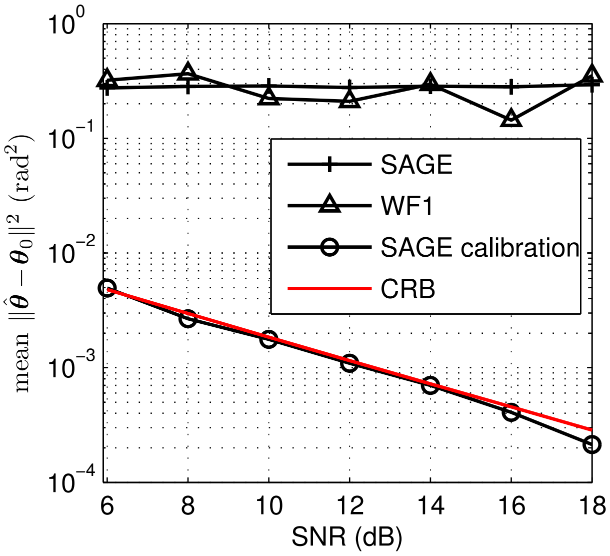

Figure 3.7: *Mean* $\|\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}_0\|^2$ *vs. SNR. True DOA parameters* $\boldsymbol{\theta}_0 = [-35°, 0°, 35°]$, $\sigma_p = 0.25\lambda$.

In the third case, widely separated signals as in the first experiment are considered, i.e. $\boldsymbol{\theta}_0 = [-35°, 0°, 35°]$. The standard deviation of sensor position error is increased to $\sigma_p = 0.25\lambda$ which corresponds to 50% of inter-sensor spacing. Fig. 3.7 shows that over the entire SNR range, both the standard CML-SAGE and the WF1 algorithms fail to estimate DOA parameters, whereas the SAGE-based self-calibration method still provides optimal results that approach the CRB. This shows that the proposed algorithm has a much higher robustness against large sensor position errors than the WF1 method and retains its optimal MSE performance at large large sensor position errors.

3.5.2.4 Computational cost

Finally, the average numbers of iterations-to-converge for the three Monte Carlo (MC) cases are illustrated against SNR in Fig. 3.8. The number of iterations is averaged over the 100 MC trials at each SNR. As expected, the case of well-separated sources requires the least average

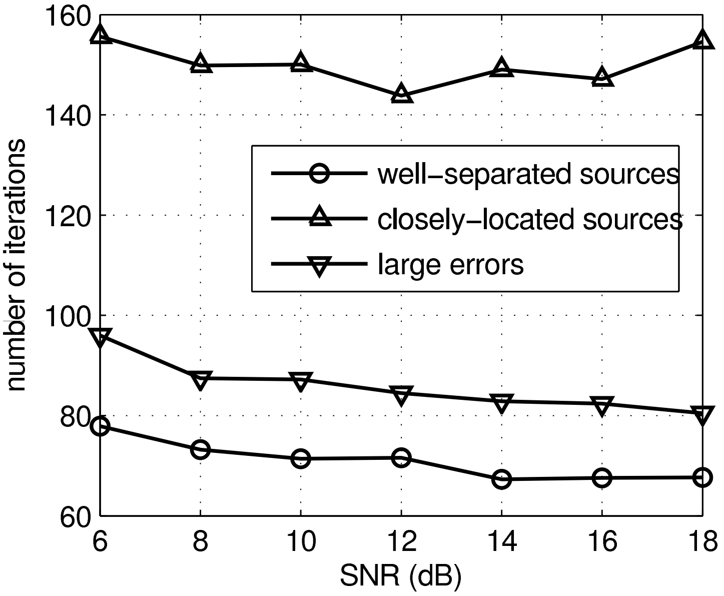

Figure 3.8: *Average number of iterations for convergence vs. SNR.*

number of iterations to converge, whereas the more challenging cases of closely-located sources and large errors require more average number of iterations to converge. In the more challenging cases the case of closely-located sources demands the most average number of iterations.

The cases of well-separated sources and large errors feature a generally decreasing trend against increasing SNR in average number of iterations to converge. Specifically, in the former case the average number decreases from 77.9 at 6 dB to 67.7 at 18 dB, and in the latter case the average number decreases from 96.0 at 6 dB to 80.5 at 18 dB. However in the case of closely-located sources the average number of iterations to converge does not decrease as the SNR increases, instead it fluctuates between the minimum of 143.8 at 12 dB and the maximum of 155.6 at 6 dB.

The reason why the number of iterations is the major concern is that simulations show that in each iteration the computational cost is the virtually same. Using MUSIC for DOA estimation as a reference, one may infer the absolute computational cost of SAGE self-calibration. For example, for the case of large sensor location errors in section 3.5.2.3, the computation time for each self-calibration process is 1.03 second at SNR $=12$ dB, and the time for MUSIC DOA estimation is 0.006 second at the same SNR. MUSIC uses the nominal array to localize the three sources as in section 3.5.2.3, assuming no array perturbation. The experiments are conducted with Matlab 7.8.0.347 (R2009a) on a PC with Intel Xeon CPU E5420 at 2.50 GHz (8 core), and

8 GB RAM. It is noted that array shape self-calibration based on the MUSIC objective function is also a nonlinear multidimensional optimization problem [2], which is computationally much more costly than MUSIC DOA estimation, and the computational cost is dependent on the optimization algorithm employed. Therefore, a thorough comparison of the computational costs of the self-calibration techniques requires a dedicated work, which is out of the scope of this thesis.

3.6 Conclusion

A SAGE algorithm based self-calibration procedure is derived for arrays with sensor position uncertainties. Through data augmentation, DOA estimation and array calibration can be carried out by a computationally simple search procedure. Numerical experiments show that the proposed method outperforms the existing method for closely located signal sources and is robust to large sensor errors. Moreover, the accuracy of the proposed procedure approaches the CRB. Therefore, the SAGE-based procedure provides an attractive alternative to current self-calibration techniques.

Chapter 4
Far-field online calibration II

In this chapter, the same far-field self-calibration problem as that of Chapter 3 is dealt with by another novel formulation. First, an unconditional maximum likelihood (UML) estimator, rather than the conditional maximum likelihood (CML) estimator[1] in Chapter 3, is used for jointly estimating the source directions of arrival (DOA's) and sensor locations. Although the UML estimator is a standard estimator for direction finding (DF), it has not yet been used for array-shape self-calibration, due to its complicated objective function that requires much optimization effort. The optimization complexity is overcome by particle swarm optimization (PSO) with decaying diagonal loading (DDL), which is the second novelty of the formulation.

In the particle swarm optimization – decaying diagonal loading (PSO-DDL) paradigm, the PSO is a recently-developed stochastic global optimization algorithm, inspired by social intelligence in biology. Being "global", PSO does not need accurate initialization, which is a common problem for all local optimization algorithms. As no approximation is involved, the self-calibration is applicable to large error and the result is statistically efficient. DDL is developed to take advantage of the unexpected behavior that the UML objective function is more difficult to globally optimize at high signal-to-noise ratio (SNR) than at low SNR. The reason is that the global minimum of the UML cost function is more prominent at low SNR than at high SNR. An exponentially decaying diagonal loading (DL) is introduced, to *temporarily* lower the SNR for exploitation of the global minimum prominence. DDL is coupled with the progress of PSO to guide the optimization algorithm towards the target, and in the end is reduced to zero to preserve the quality of the UML estimator. The DDL enhances the PSO's search ability without sacrificing the estimator's performance, and requires negligible computation. Since its effect is on the objective function, DDL can be coupled with any global optimizer for performance enhancement, which is supported by simulation results of the differential evolution (DE) algorithm.

[1] A comparison between the CML-SAGE in Chapter 3 and the UML-PSO in Chapter 4 seems infeasible because 1) both the estimators and the optimizers are different, and 2) SAGE is a local optimizer whereas PSO is a global optimizer. Fair comparisons are feasible if CML-PSO and UML-SAGE procedures for array shape calibration are developed in the future.

In summary, the novel formulation of UML optimized by PSO-DDL provides a practicable solution for array shape self-calibration. It is free of the initialization problem, applicable to large sensor location errors, with optimal accuracy, and computationally feasible. Moreover, the application of DDL presents a new perspective of the UML objective function: contrary to the intuition that high SNR is better than low SNR, the shape of the cost function is easier for a global optimizer at low SNR than at high SNR. In other words, the prominence of the global minimum is different from its precision.

This chapter starts with formulating the UML objective function in section 4.1, followed by introducing PSO and DDL in sections 4.2 and 4.3, respectively, and ends with presenting simulation results in section 4.4. The simulation results include PSO-DDL results and differential evolution – decaying diagonal loading (DE-DDL) results.

4.1 Problem formulation

The physical scenario considered is the same as that in chapter 3. Following [19], the UML estimator for $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ can be expressed as

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\delta}}) = \arg\min_{(\boldsymbol{\theta},\boldsymbol{\delta})} \underbrace{\left\{ \det \left[\boldsymbol{P_H} \widehat{\boldsymbol{R}} \boldsymbol{P_H} + \frac{\mathrm{tr}(\boldsymbol{P_H^\perp} \widehat{\boldsymbol{R}}) \boldsymbol{P_H^\perp}}{N - M} \right] \right\}}_{F(\boldsymbol{\theta},\boldsymbol{\delta})}, \tag{4.1}$$

where $\widehat{\boldsymbol{R}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{x}(k)\boldsymbol{x}^H(k)$ is the sample covariance matrix, $\boldsymbol{P_H} = \boldsymbol{H}(\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H$ is the projection matrix of $\boldsymbol{H}$, and $\boldsymbol{P_H^\perp} = \boldsymbol{I} - \boldsymbol{P_H}$ is its orthogonal complement. The dependence of $\boldsymbol{H}$ on $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ is suppressed for notational simplicity, and $F(\boldsymbol{\theta}, \boldsymbol{\delta})$ denotes the UML objective function.

4.2 PSO

The UML objective function in Eq. (2.24) is multi-modal, multi-variate and non-linear, which poses a challenge to optimization. Local optimization techniques such as the various gradient algorithms, the expectation-maximization (EM)/space-alternating generalized expectation-maximization (SAGE) algorithm and the simplex algorithm can be applied to conduct the optimization, however the initialization problem remains, as common to all local optimization techniques. An alternative is to apply global optimization techniques, such as the numerous stochastic optimization algorithms including genetic algorithm (GA), simulated annealing (SA) etc., to conduct the optimization. One of the stochastic optimization algorithms is PSO, whose good performance, simple implementation and robust control parameters distinguish it from others that are also potentially applicable to the problem.

4.2.1 PSO introduction

PSO [81] is a stochastic global optimisation technique that is simple in kernel and robust in its control parameters. It is reported that PSO is computationally less intensive than the genetic algorithm (GA) for a number of array processing and other problems [82]. A relatively simple version of PSO is introduced, which is adequate for the current problem. Other PSO variants that are also applicable can be found in the relevant literature such as [83]. Firstly the problem-independent PSO kernel is described, then the problem-specific designs follow.

The problem-independent PSO kernel is a constriction PSO [84]. In it, a swarm particle is a point in the D-dimensional solution space of the optimisation problem, whose coordinate is $\boldsymbol{\xi}_i \in \mathbb{R}^D$, where i denotes the ith particle, $i = 1, 2, \ldots, P$. The P particles constitute a swarm, wherein each particle is propelled by random movement in the solution space but is guided towards the global optimum of the objective function by swarm intelligence – the processing of shared information of the objective function.

At the tth time step, the velocity of the ith particle is updated according to

$$\boldsymbol{\psi}_i(t+1) = \chi \big\{ \boldsymbol{\psi}_i(t) + \varphi_1 \boldsymbol{r_1} \odot [\boldsymbol{p}_i(t) - \boldsymbol{\xi}_i(t)] + \varphi_2 \boldsymbol{r_2} \odot [\boldsymbol{g}_i(t) - \boldsymbol{\xi}_i(t)] \big\},\ i = 1, \ldots, P, \tag{4.2}$$

where $\odot$ is the element-wise product operator. The three terms in the brackets of Eq. (4.2) account for inertia, individual experience and group information, sequentially. Specifically, $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are random vectors uniformly distributed in $[0, 1]$ that propel the particles, $\boldsymbol{p}_i(t)$ is the particle best location of its own experience from time step 1 to t, $\boldsymbol{g}_i(t)$ is the group best location at current time step t. "Best" is in the sense of the lowest value of the cost function to be minimized. The scalar $\chi = 2/|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|$ is the constriction factor that progressively constricts the harmful particle oscillation resulting from the random movement. Here $\varphi = \varphi_1 + \varphi_2$, in which φ_1 is the acceleration constant toward the particle best location and thus a large value of it encourages exploration of the solution space; and φ_2 is the acceleration constant toward the group best location and thus a large value of it encourages exploitation of potential districts of global optimum. The value of φ should be greater than 4 to prevent particle oscillation [84], and in practice $\varphi = 4.1$ is a viable value [83].

For a unit time step, the position of the ith particle is updated as

$$\boldsymbol{\xi}_i\left(t+1\right) = \boldsymbol{\xi}_i\left(t\right) + \boldsymbol{\psi}_i\left(t+1\right),\; i = 1, \ldots, P. \tag{4.3}$$

In order to prevent the particle from flying too fast and thus overshooting the target or exceeding the boundary, the particle velocity is limited by a maximum velocity $\mathit{\Psi}$. If the velocity in any dimension d exceeds $\mathit{\Psi}$, it is clamped by

$$\psi_d(t+1) = \mathit{\Psi}\frac{\psi_d(t)}{|\psi_d(t)|},\; d = \mathit{1}, \ldots, D, \tag{4.4}$$

where ψ_d is the dth component of $\boldsymbol{\psi}_i$, and i has been suppressed for notational convenience. If a particle flies out of the boundary of the solution space in any dimension, it is reset to a random position in that dimension. In the case that the parameters to be optimised have different boundaries, one may map the feasible ranges into $[0, 1]$. As a result the velocity can be updated uniformly, and $\mathit{\Psi}$ is the same for all the dimensions.

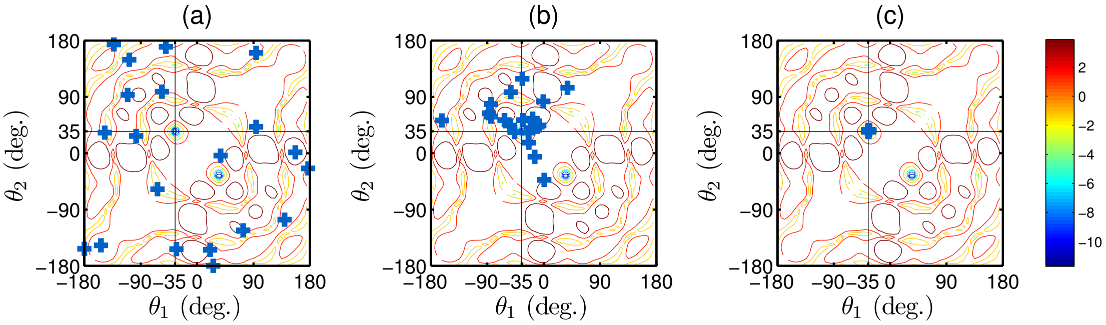

Figure 4.1: *Illustration of PSO progressive behavior. The blue crosses represent the* 20 *particles. (a)* $t = 1$, *(b)* $t = 15$, *(c)* $t = 45$. *The objective function minimized is* $F(\theta_1, \theta_2)|_{\boldsymbol{\delta}=\mathbf{0}}$, $M = 2$ *far-field sources reside at* $[\theta_1, \theta_2]_0 = [-35\ 35]^\circ$. *The array is an* $N = 6$ *sensor uniform circular array (UCA),* $a = \lambda/2$. $K = 256$ *snapshots. To reveal the shape of the objective function,* $\log(F)$ *is contoured as the background with linear color map.*

An example of the progressive behavior of PSO is illustrated in Fig. 4.1. In this illustration, the PSO is employed to optimize the objective function of a DOA estimation problem. In the beginning, Fig. 4.1(a), the particles are initialized to random positions in the solution space, uniformly occupying the plane. Fig. 4.1(b) shows that at the 15th PSO iteration the particles have largely concentrated towards the target which lies at $[-35\ 35]^\circ$. In the end, Fig. 4.1(c), all the particles arrive at the target, and the optimization is completed.

4.2.2 DOA collision avoidance and string topology in PSO

When PSO is applied to the objective function in Eq. (2.24), two features of the optimization problem have to be considered. The first arises from the structure of F, which involves matrix inversion. If $\boldsymbol{H}$ is rank-deficient, then $\boldsymbol{H}^H\boldsymbol{H}$ is singular and not invertible, which happens when two DOA's are identical. In practice, when two DOA parameters are too close together ("collision") during the PSO iterations, numerical instability emerges and as a result PSO is not able to separate the collided DOA parameters. The second lies in the aim of the optimization. For some optimization problems where the final aim is to reduce the cost indexed by the objective function for example, a local minimum is acceptable if it is not too much higher than the global minimum. However, for an estimation problem where the final aim is to find the global minimum that is associated with the parameter estimate, a local minimum is not acceptable if the associated parameter value is far from that associated with the global minimum, even if the objective function value of the local minimum is very close to that of the global minimum. Therefore for the estimation problem it is desirable to recruit an optimizer that reaches the global minimum of F with as high probability as possible, yielding accurate estimates. The solutions to the two problems are given below.

The first strategy deals with identical DOA's. As the entries of $\boldsymbol{\xi}_i(t)$ are randomly and independently updated, there is a possibility that two or more DOA entries are adjusted to an identical value. This renders the array manifold $\boldsymbol{H}$ rank-deficient and consequently makes $(\boldsymbol{H}^H\boldsymbol{H})^{-1}$ non-existent in Eq. (2.24), which causes numerical instability. In practice, if two or more DOA parameters are close to each other, PSO would be trapped in these false values and often cannot escape despite the random particle position update. A simple and effective solution is to assign a random value to one of the parameters if at any PSO iteration t, two DOA parameters happen to be closer than a pre-specified value δ [82]. If δ is smaller than the the sources' angular separation, it does not affect the self-calibration.

The second strategy deals with swarm topology. The swarm topology determines the neighbourhood of a swarm particle, from which the group best location $\boldsymbol{g}_i(t)$ in Eq. (4.2) is chosen:

the ith particle compares the objective function value at its own location with those at its neighbours' in its group. The location associated with the best objective function value is chosen as $\boldsymbol{g}_i(t)$. If every particle's neighbours are defined as all the other particles in the swarm, the swarm is said to have a "global best" topology; if a particle's neighbours are defined as part of the other particles in the swarm, the swarm is said to have a "local best" topology. A general trend is that swarm topologies with more connections between the particles tend to converge faster, but have a lower success probability of reaching the global optimum, and vice versa [85]. This tendency is also observed in the trials of topologies for the self-calibration problem considered in this work. For the parameter estimation problem it is desirable to maximize PSO's ability of reaching the global optimum, therefore a "string" topology with least connection between the particles is defined for the PSO:

$$\text{Particles of the } i\text{th group} \begin{cases} i, i+1 & i = 1 \\ i-1, i, i+1 & i = 2, \ldots, P-1 \\ i-1, i & i = P. \end{cases}$$

4.2.3 PSO performance

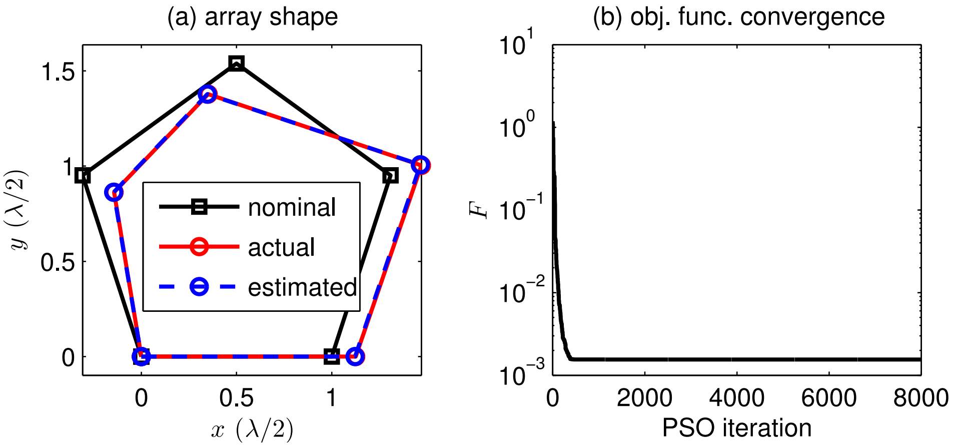

Figure 4.2: *A successful self-calibration performed by PSO. (a) The calibrated (estimated) array shape compared with the actual and nominal array shapes. (b) The convergence pattern of the objective function value as optimized by PSO.*

Equipped with the strategies in section 4.2.2, PSO is applied to optimize F. The nominal array is a 5-sensor UCA with inter-sensor spacing $a = \lambda/2$, perturbed to actual array shape by uniformly distributed random displacements with boundary $b = 0.2a$. Three sources of opportunity reside at DOA $\boldsymbol{\theta} = [-35° \ 0° \ 35°]^T$, which are unknown *a priori* to the estimator.

parameter	true	estimated	parameter	true	estimated
θ_1 (rad)	-0.6109	-0.6143	x_4 $(\lambda/2)$	0.3508	0.3466
θ_2 (rad)	0.0000	-0.0031	x_5 $(\lambda/2)$	-0.1437	-0.1461
θ_3 (rad)	0.6109	0.6125	y_3 $(\lambda/2)$	1.004	1.007
x_2 $(\lambda/2)$	1.1259	1.1227	y_4 $(\lambda/2)$	1.3779	1.3783
x_3 $(\lambda/2)$	1.4713	1.4660	y_5 $(\lambda/2)$	0.8625	0.8626

Table 4.1: *Estimated parameters compared with true values.*

SNR$=22$ dB. The PSO is configured by the same PSO parameters as that in Tab. 4.3. A typical successful PSO operation on the array self-calibration problem is shown in Fig. 4.2. It is seen from Fig. 4.2(a) that the calibrated array shape is close to the actual shape, and from Tab. 4.1 that the DOA estimate is close to the true DOA. The calibrated and actual sensor coordinates are also in Tab. 4.1.

Surprisingly, however, when applied over a range of SNR values, PSO starts to fail when the SNR is *higher* than a certain threshhold, and the proportion of failure increases as the SNR increases. An example of the threshold and failure increase is shown in Fig. 4.3. The left hand column shows the F convergence of the cost function F over 200 trial runs at 4 values of SNR. The right hand column shows the final converged value of the angle estimates for each of the trials. It can be seen that at SNR $=10$ dB all the 200 DOA estimates are close to the true value, but from SNR $=14$ dB on, spiky estimates appear indicating that some runs produce poor estimates. The number of spikes increases as the SNR increases.

The PSO performance at SNR $=26$ dB is shown in Fig. 4.4 with more detail. Fig. 4.4(b) shows that a significant portion of the optimizations failed to reach the global minimum of F, yielding erroneous array-shape estimates, the first of which is shown Fig. 4.4(a) as an example; and irrelevant DOA estimates, shown in Fig. 4.4(c). In addition, the optimized value of F in Fig. 4.4(b) shows an apparent separation between success and failure. Using $F=10^{-3}$ as a criterion, 32 out of the 200 optimizations failed to reach the global minimum. The error of the corresponding DOA estimates shows a similar separation. This separation suggests that the failures are caused by the PSO being trapped in local minima.

It is noted that even a few spiky estimates can result in a mean squared error (MSE) of the 200 trials much higher than the optimal, and worse is that a threshhold at 10 dB poses a serious problem in applications where SNR is traded for calibration accuracy. These are shown by the MSE of PSO in Fig. 4.11 to avoid duplication.

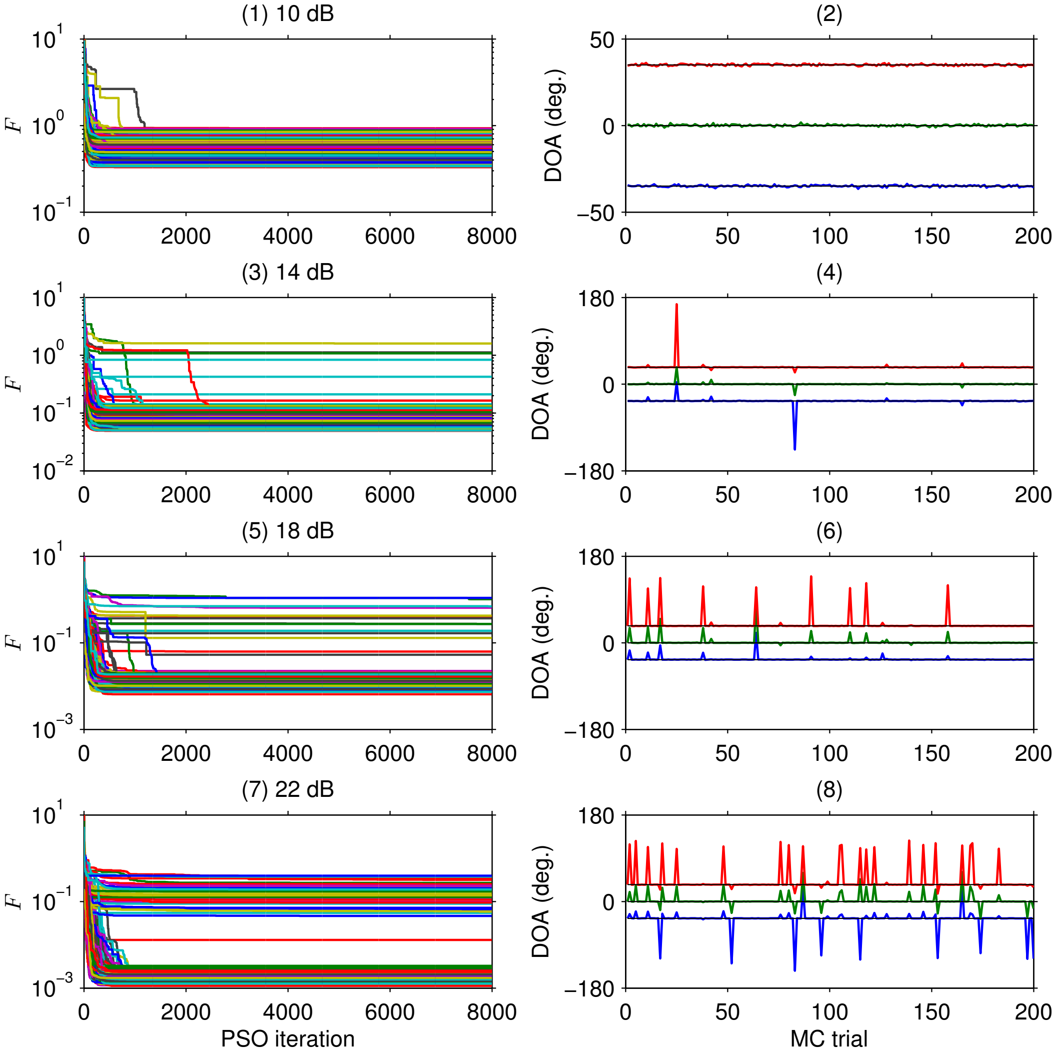

Figure 4.3: *PSO performance at critical SNR's. In the left column are the F convergence over 200 MC trials at 4 values of SNR. In the right column are the final converged value of the DOA estimates for each of the trials.*

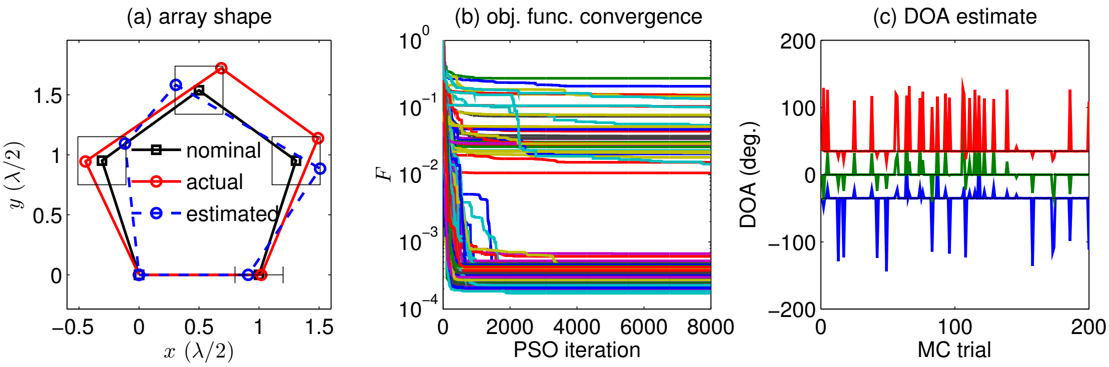

Figure 4.4: *The failure of PSO at SNR* $= 26$ *dB. (a) The first failed array shape estimation compared with nominal and actual shapes. The thin-line black squares and line segment indicate the perturbation boundaries. (b) The pattern of* F *in the process of* 200 *optimizations. (c) The* 200 *DOA estimates.*

4.3 DDL

In this section the reason for PSO's failure at high SNR is analyzed, and then by noticing that reducing SNR is asymptotically equivalent to diagonal loading, a DDL technique is proposed to solve the problem at high SNR.

4.3.1 DL effect on F

The reason for PSO's failure at high SNR is that the global minimum of F is less prominent at higher SNR than at lower SNR. In the following this phenomenon is firstly demonstrated by visualisation of F and then further explained by using the first derivative of F.

To begin with, the asymptotic equivalence between DL and SNR reduction is established, and the parameters involved are defined. Asymptotically ($K \to \infty$), lowering the SNR is equivalent to diagonally loading the covariance matrix $\boldsymbol{R}$. Suppose additional noise $\boldsymbol{n}_l$ is loaded to the array output,

$$\tilde{\boldsymbol{x}}(k) = \boldsymbol{H}\boldsymbol{s}\,(k) + \boldsymbol{n}\,(k) + \boldsymbol{n}_l\,(k)\,,\ k = 1, \ldots, K, \tag{4.5}$$

where $\boldsymbol{n}_l(k)$ is independent, identically complex normally distributed with zero mean and covariance matrix $\nu_l\boldsymbol{I}$, and ν_l is the power of the loaded noise, then asymptotically

$$\widetilde{\boldsymbol{R}} = \boldsymbol{R} + \nu_l\boldsymbol{I}, \tag{4.6}$$

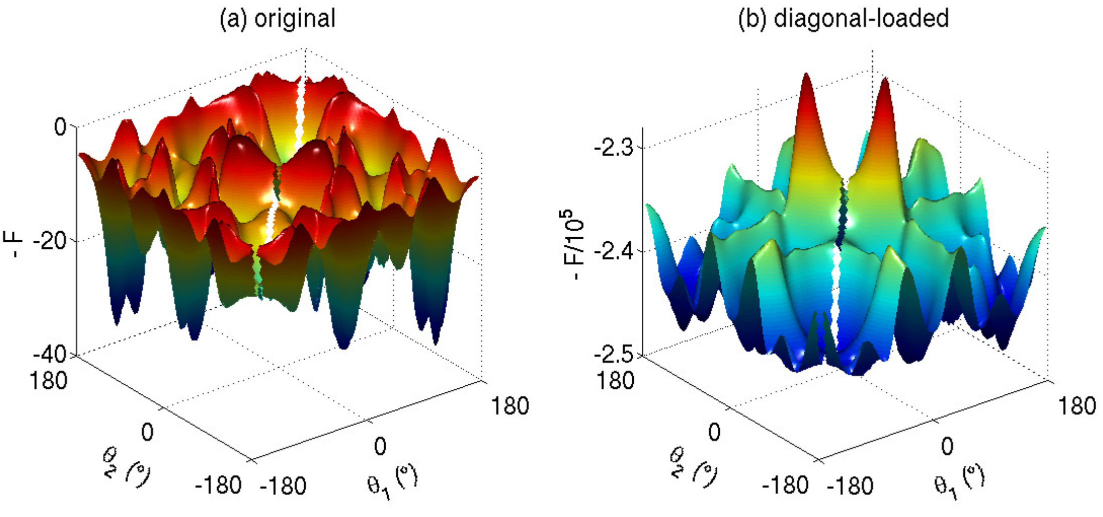

Figure 4.5: *The* negative *objective function before (a) and after (b) diagonal loading. 5-sensor uniform circular array, inter-sensor spacing* $\frac{\lambda}{2}$*; 2 signals at* $[-35\ 35]^\circ$, $K = 1000$*; SNR*$=$ 14 *dB, SNR*$_a$ $= -10$ *dB.*

where $\widetilde{\boldsymbol{R}} = \lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \tilde{\boldsymbol{x}}(k)\tilde{\boldsymbol{x}}^H(k)$ and $\boldsymbol{R} = \lim_{K\to\infty} \widehat{\boldsymbol{R}}$ with probability 1. The additional noise appears in the covariance matrix as diagonal loading.

Assuming normalised signal power, the original signal-to-noise ratio can be expressed as $\text{SNR} = -10\log\nu$. The effective signal-to-noise ratio after DL is defined as

$$\text{SNR}_a = -10\log\left(\nu + \nu_l\right), \tag{4.7}$$

and consequently the loaded signal-to-noise ratio is $\text{SNR}_l = \text{SNR}_a - \text{SNR}$.

Since it is impossible to visualize the high-dimensional objective function of the self-calibration problem, the effect of DL has to be demonstrated by a corresponding two-dimensional DOA estimation problem, which is visualised in Fig. 4.5. The objective function is the degenerated F by setting $\boldsymbol{\delta} = \boldsymbol{0}$. Comparing Fig. 4.5(a) and Fig. 4.5(b) it is seen that diagonal loading of the sample covariance matrix $\widehat{\boldsymbol{R}}$, which is asymptotically equivalent to reducing the SNR, makes prominent the global minimum of the objective function, therefore eases the searching of it for a global optimiser. The objective function by directly adding noise, Eq. (4.5), closely resembles that by DL, Fig. 4.5(b), therefore is not shown. This phenomenon is observed independently from the number of sensors for the DOA estimation problem, and is confirmed by the simulations in section 4.4.1 for the self-calibration problem.

The fact that F at higher SNR is more difficult to optimise might seem counter-intuitive at first sight, however it is worth pointing out that the *precise* global minimum at high SNR is not necessarily *prominent* at the same time, as shown by Fig. 4.5(a). The reason that the prominence of the global minimum of an objective function received little attention in the existing array processing literature is that, a local optimization algorithm descends to the nearest minimum, and if it is initialised in the basin of the global minimum it reaches the global minimum whether it is prominent or not; whereas for a global optimization algorithm that searches the entire parameter space without an accurate initialisation, the prominence of the global minimum is important, particularly at the early stages when the individual particles have not gathered around the global minimum.

The analytical explanation of the DL effect on F is carried out as follows. Jacobi's formula [86] (Part Three, Section 8.3) states that

$$(\det \boldsymbol{X})' = (\det \boldsymbol{X}) \cdot \mathrm{tr}\left(\boldsymbol{X}^{-1}\boldsymbol{X}'\right), \tag{4.8}$$

where $\boldsymbol{X}$ is an invertible matrix.

Let

$$\boldsymbol{X} = \boldsymbol{P_H}\widehat{\boldsymbol{R}}\boldsymbol{P_H} + \frac{\mathrm{tr}(\boldsymbol{P}^{\perp}_{\boldsymbol{H}}\widehat{\boldsymbol{R}})\boldsymbol{P}^{\perp}_{\boldsymbol{H}}}{N-M} \tag{4.9}$$

and use (B.3) and (B.15) in [21], it follows that

$$\mathrm{tr}\left(\boldsymbol{X}^{-1}\boldsymbol{X}'\right) = 2\mathrm{Re}\left(\mathrm{tr}\left\{\left[\left(\boldsymbol{H}^H\widehat{\boldsymbol{R}}\boldsymbol{H}\right)^{-1} - \frac{1}{\hat{\nu}}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\right]\boldsymbol{H}^H\widehat{\boldsymbol{R}}\boldsymbol{P}^{\perp}_{\boldsymbol{H}}\boldsymbol{H}'\right\}\right), \tag{4.10}$$

where

$$\hat{\nu} = \frac{\mathrm{tr}(\boldsymbol{P}^{\perp}_{\boldsymbol{H}}\widehat{\boldsymbol{R}})}{N-M}. \tag{4.11}$$

Inserting (4.9) and (4.10) into (4.8) , the first derivative of F is obtained as

$$\begin{aligned} F' = 2\det&\left[\boldsymbol{P_H}\widehat{\boldsymbol{R}}\boldsymbol{P_H} + \hat{\nu}\boldsymbol{P}^{\perp}_{\boldsymbol{H}}\right] \times \\ &\mathrm{Re}\left(\mathrm{tr}\left\{\left[\left(\boldsymbol{H}^H\widehat{\boldsymbol{R}}\boldsymbol{H}\right)^{-1} - \frac{1}{\hat{\nu}}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\right]\boldsymbol{H}^H\widehat{\boldsymbol{R}}\boldsymbol{P}^{\perp}_{\boldsymbol{H}}\boldsymbol{H}'\right\}\right). \end{aligned} \tag{4.12}$$

Replace $\widehat{\boldsymbol{R}}$ with $\widehat{\boldsymbol{R}} + \nu_l\boldsymbol{I}$ in (4.11) and (4.12), note that tr $\left(\boldsymbol{P}^{\perp}_{\boldsymbol{H}}\right) = N - M$ and $\boldsymbol{H}^H\boldsymbol{P}^{\perp}_{\boldsymbol{H}} = \boldsymbol{0}$, then the first derivative after diagonal loading, F_l', is given by

$$F_l' = 2\underbrace{\det\left[\boldsymbol{P_H}\left(\widehat{\boldsymbol{R}}+\nu_l\boldsymbol{I}\right)\boldsymbol{P_H}+(\hat{\nu}+\nu_l)\,\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right]}_{\alpha(\nu_l)}\times$$
$$\underbrace{\mathrm{Re}\left(\mathrm{tr}\left\{\left[\left(\boldsymbol{H}^H(\widehat{\boldsymbol{R}}+\nu_l\boldsymbol{I})\boldsymbol{H}\right)^{-1}-\frac{1}{\hat{\nu}+\nu_l}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\right]\boldsymbol{H}^H\widehat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\boldsymbol{H}'\right\}\right)}_{\beta(\nu_l)}. \quad (4.13)$$

Two asymptotic ($K \to \infty$) properties of the DL can be derived from (4.12) and (4.13):

1) *An extremum of F remains at $\boldsymbol{\theta}_0, \boldsymbol{\delta}_0$ after DL.*

Proof. At the true parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ and $\boldsymbol{\delta} = \boldsymbol{\delta}_0$,

$$\boldsymbol{R} = \boldsymbol{H}_0\boldsymbol{S}\boldsymbol{H}_0^H + \nu\boldsymbol{I},$$

where $\boldsymbol{H}_0$ stands for $\boldsymbol{H}(\boldsymbol{\theta}_0, \boldsymbol{\delta}_0)$ and $\boldsymbol{S}$ the signal covariance matrix $\mathrm{E}\left(\boldsymbol{s}\boldsymbol{s}^H\right)$. The factor $\boldsymbol{H}^H\boldsymbol{R}\boldsymbol{P}_{\boldsymbol{H}}^{\perp}$ in (4.12) and (4.13) satisfies

$$\boldsymbol{H}_0^H\boldsymbol{R}\boldsymbol{P}_{\boldsymbol{H}_0}^{\perp} = \boldsymbol{H}_0^H\left(\boldsymbol{H}_0\boldsymbol{S}\boldsymbol{H}_0^H + \nu\boldsymbol{I}\right)\boldsymbol{P}_{\boldsymbol{H}_0}^{\perp} = \boldsymbol{0}.$$

Therefore $F_l'|_{\boldsymbol{\theta}_0,\boldsymbol{\delta}_0} = F'|_{\boldsymbol{\theta}_0,\boldsymbol{\delta}_0} = 0$, an extremum remains at $\boldsymbol{\theta}_0, \boldsymbol{\delta}_0$ after DL. □

2) $|F_l'| > |F'|$ *when ν_l is large.*

Proof. When ν_l is large, $\boldsymbol{P_H}\boldsymbol{R}\boldsymbol{P_H} + \hat{\nu}\boldsymbol{P}_{\boldsymbol{H}}^{\perp} + \nu_l\boldsymbol{I} \approx \nu_l\boldsymbol{I}$, therefore

$$\det\left[\boldsymbol{P_H}\boldsymbol{R}\boldsymbol{P_H} + \hat{\nu}\boldsymbol{P}_{\boldsymbol{H}}^{\perp} + \nu_l\boldsymbol{I}\right] \approx \nu_l^N;$$

and

$$\left[\boldsymbol{H}^H\left(\boldsymbol{R}+\nu_l\boldsymbol{I}\right)\boldsymbol{H}\right]^{-1} = \left(\nu_l^{-1} - \nu\nu_l^{-2}\right)\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1} - \nu_l^{-2}\boldsymbol{H}^{\dagger}\boldsymbol{H_0}\boldsymbol{S}\boldsymbol{H_0}^H\boldsymbol{H}^{\dagger H} + O\left(\nu_l^{-3}\right),$$

where

$$\boldsymbol{H}^{\dagger} = \left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\boldsymbol{H}^H$$

$\boldsymbol{\theta}$ (rad)	$(1.3536; 0.0709)$	$(-1.8344; 1.5837)$	$(-2.3799; -2.7661)$
F'	1.4486	31.8563	10.0029
$\Delta F/\Delta\theta_1$	1.4482	31.8549	10.0029
F'_l	1.1520×10^5	2.0535×10^4	6.9100×10^4
$\Delta F_l/\Delta\theta_1$	1.1520×10^5	2.0530×10^4	6.9101×10^4

Table 4.2: *Comparison of the first derivatives calculated by Eq.* (4.12) *and numerical difference at random points on F.*

is the Moore-Penrose pseudoinverse of $\boldsymbol{H}$. Therefore

$$\left(\boldsymbol{H}^H\left(\boldsymbol{R}+\nu_l\boldsymbol{I}\right)\boldsymbol{H}\right)^{-1}-\frac{1}{\hat{\nu}+\nu_l}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\approx-\nu_l^{-2}\boldsymbol{H}^\dagger\boldsymbol{H_0}\boldsymbol{S}\boldsymbol{H_0}^H\boldsymbol{H}^{\dagger H}.$$

As a result

$$F'_l=O\left(\nu_l^{N-2}\right).$$

Because F' is independent of ν_l, $|F'_l|>|F'|$ when ν_l is large. □

Assuming F is sufficiently smooth in the vicinity of the global minimum, properties 1) and 2) show that DL makes F more pointed when ν_l is large, which suggests that the global minimum is made more prominent. A complete proof of the prominence of the global minimum requires analytical information on the local minima of F, and remains an open problem.

It is remarked that DL takes effect on the objective function, thus its effect is independent of the specific global optimizer, and during the derivation no assumption was made on the array-source geometry or the types of parameter, thus the effect is independent of them, too.

The validity of Eq. (4.12) is verified in Tab. 4.2. The first derivative is calculated by both Eq. (4.12) and numerical difference at random values of $\boldsymbol{\theta}$, and Tab. 4.2 shows three of them. In the latter case the first derivative is calculated as

$$\frac{\Delta F}{\Delta\theta_1}=\frac{F(\theta_1+\Delta\theta,\theta_2)-F(\theta_1-\Delta\theta,\theta_2)}{2\Delta\theta},$$

in which $\Delta\theta=0.1°$. The array-source geometry is: 5-sensor circular array (CA), half-wavelength intersensor spacing, two signals at $[-35\ 35]°$; SNR$=20$ dB, SNR$_a=-15$ dB, $K=2\times10^5$. It is seen from Tab. 4.2 that the results of the numerical difference are very close to those of Eq. (4.12).

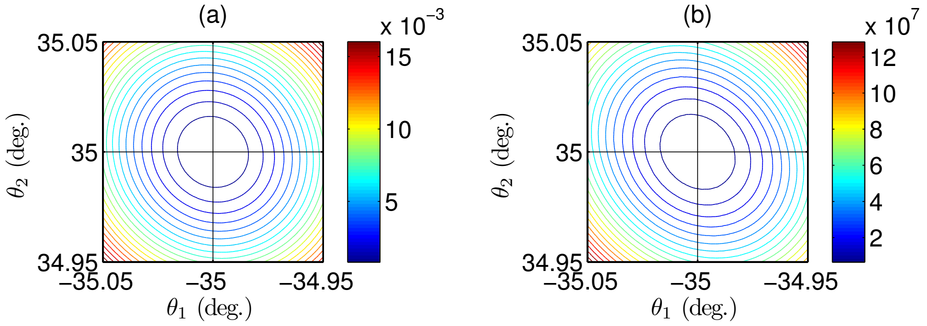

Figure 4.6: *The objective function in the vicinity of the true parameter. The objective function is for* 5*-sensor UCA with* $\lambda/2$ *inter-sensor spacing,* 2 *sources at* $\boldsymbol{\theta} = [-35° \; 35°]^T$, $K = 1000$. *(a) SNR* $= 30$ *dB, (b) SNR*$_a$ $= -40$ *dB.*

To verify property 1), the objective function in the vicinity of the true parameter is illustrated in Fig. 4.6. It is seen from Fig. 4.6(a) and (b) that the increased noise power only increases the ellipticity of the contour but keeps the global minimum of F on the true parameter.

Property 1) is also suggested by the Cramér-Rao bound (CRB) expression [21] [87]:

$$\boldsymbol{B}_{\text{CR}} = \frac{\nu}{2K} \{\cdot\} .$$

In reality the noise power ν is finite, therefore $\boldsymbol{B}_{\text{CR}} \to \mathbf{0}$ asymptotically ($K \to \infty$), which suggests that when $K \to \infty$ the global minimum of F resides on the true parameter no matter what the finite value of ν is. This translates into that DL does not shift away the global minimum when $K \to \infty$.

To verify property 2), $F_l'(\nu_l)$ and its two factors $\alpha(\nu_l)$ and $\beta(\nu_l)$ in Eq. (4.13) are plotted in Fig. 4.7. The derivative of F_l' is with respect to θ_1. Derivatives with respect to other parameters behave in the same fashion. It is seen that as ν_l increases, $\beta(\nu_l)$ increases much faster than $\alpha(\nu_l)$ decreases, as a result $F_l'(\nu_l)$ increases as ν_l increases.

4.3.2 The DDL design

Inspired by the above effect a DDL technique is devised. In this, the amount of DL is sufficiently large in the beginning to make the global minimum prominent and thus guides the PSO.

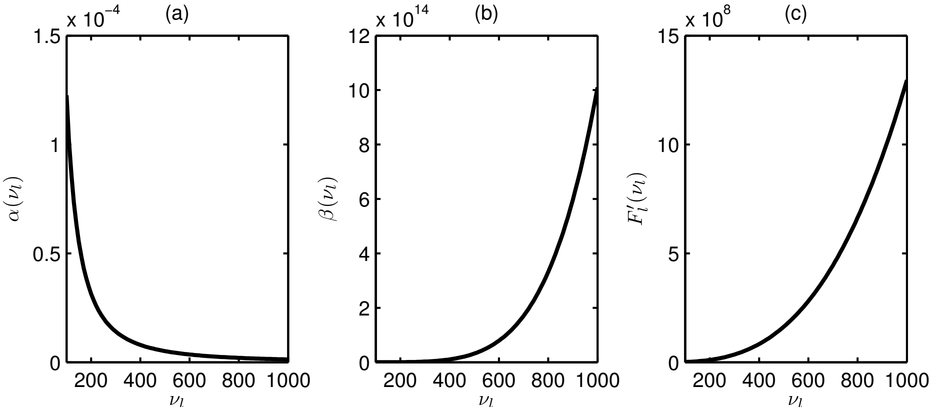

Figure 4.7: *Verification of the expressions of F_l' and its two factors in Eq.* (4.13) *at* $\boldsymbol{\theta} = [-34.5°, -35°]^T$ *as functions of* ν_l. *(a)* $\alpha(\nu_l)$, *(b)* $\beta(\nu_l)$, *(c)* $F_l' = 2\alpha(\nu_l)\beta(\nu_l)$. *The array-source geometry is 5-sensor CA, half-wavelength intersensor spacing, two signals at* $[-35\ 35]°$. *SNR=* 20 *dB.* ν_l *varies from 100 to 1000, which is approximately SNR*$_a$ *from* -20 *to* -30 *dB.*

It reduces to zero in the end and thus completely recovers the original F. DDL differs from existing DL techniques for improving performance in that it does not permanently alter the objective function. The objective function eventually optimized is the original one, therefore the precision of the original is preserved.

The decaying DL schedule is coupled to the progress of PSO: initially the prominent global minimum made by diagonal loading attracts the particles; then as the PSO iterates, the global minimum becomes less and less prominent, but meanwhile the swarm has largely concentrated in the global minimum area thanks to the constriction factor; eventually the loading reduces to zero and the original objective function gives out the precise location of the global minimum to the swarm.

The decaying loading schedule is defined as follows. The amount of loading is exponentially reduced from SNR$_l$ to zero during PSO iterations $t = 1$ to $t = [r_l T] + 1$, where $r_l \in (0, 1)$ is the ratio of loaded iteration and T is the maximum iteration. Here $[\cdot]$ stands for the rounding operator.

$$\nu_l(t) = \begin{cases} 10^{-\frac{\text{SNR}}{10}} \left\{ 10^{-\frac{\text{SNR}_l}{10}\left(1 - \frac{t-1}{[r_l T]}\right)} - 1 \right\}, & t = 1, \ldots, [r_l T] \\ 0, & t = [r_l T] + 1, \ldots, T \end{cases} . \tag{4.14}$$

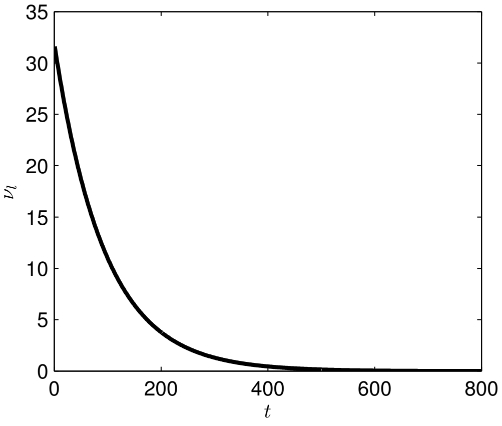

Figure 4.8: *Plot of $\nu_l\,(t)$ in loaded iterations. $r_l = 0.1$, $T = 8000$, SNR $= 22$ dB, SNR$_a = -15$ dB.*

An illustration of Eq. (4.14) is given in Fig. 4.8. The loaded sample covariance $\widetilde{\widehat{\boldsymbol{R}}}(t)$, fed to Eq. (4.1) for objective function evaluation, is then

$$\widetilde{\widehat{\boldsymbol{R}}}(t) = \widehat{\boldsymbol{R}} + \nu_l\,(t)\,\boldsymbol{I}. \tag{4.15}$$

It is noted that defining an initial effective signal-to-noise ratio SNR$_a$, Eq. (4.7), assumes a known original SNR, which may pose a problem if the original SNR is unknown. However, simulation indicates that the PSO-DDL is not sensitive to the amount of loading, therefore in the case that the original SNR is unknown, one may load $\boldsymbol{R}$ by a large amount without affecting the results.

There also exist other transformations of objective function to enhance optimization results, such as eliminating local minima to make the global minimum prominent [88], and partially convexifying the objective function [89], which differ from DDL in terms of technique and effect on the objective function. In addition, they both permanently change the objective function.

To conclude sections 4.2 and 4.3, the PSO-DDL technique is summarised in Alg. 3.

4.4 Self-calibration results

In this section, the PSO-DDL technique is applied to self-calibration. Its performance is compared with existing techniques. For comparison, the array-source is the same as that in section

Algorithm 3: PSO-DDL

Input: PSO and DDL parameters, objective function, solution space definition

Output: DOA and sensor perturbation estimates

```
Swarm initialisation: random normalised locations and random velocities;
for each iteration do
    Load diagonally the sample covariance matrix according to Eqs. (4.14) and (4.15);
    for each particle do
        Map particle location to solution vector in solution space;
        Evaluate the objective function of current iteration according to Eq. (2.24) with $\widetilde{\widehat{\boldsymbol{R}}}(t)$
        in Eq. (4.15);
        Update particle best location $\boldsymbol{p}_i$ and group best location $\boldsymbol{g}_i$;
        Update particle velocity according to Eq. (4.2);
        if velocity exceeds maximum then
            Clamp particle velocity according to Eq. (4.4);
        end
        Update particle location according to Eq. (4.3);
        if particle location out of boundary or collision at DOA dimensions then
            Set random location;
        end
    end
    Check termination criterion;
end
return final global best location;
```

4.2 where PSO alone is applied. Two sets of simulation cases are presented in subsection 4.4.1, the large number of snapshots K and small K; and in each set both large and small perturbations are considered. The simulation results are also compared with the CRB, whose expression is given in subsection 3.4. Finally the DE algorithm and its results are briefly presented in subsection 4.4.2 to demonstrate the generality of DDL.

4.4.1 Simulation cases

A 5-sensor 3-signal geometry is considered. The nominal array is a uniform circular array with inter-sensor spacing $a = \frac{\lambda}{2}$, whose sensor locations are perturbed by Δx_n and Δy_n that are independently, uniformly distributed in $[-b, b]$, in which b is the sensor location tolerance. The three sources reside at $\boldsymbol{\theta}_0 = [-35\ 0\ 35]^\circ$, emitting equi-power, uncorrelated signals. For each observation 1000 snapshots are taken. The SNR simulated ranges from -18 dB to 38 dB with 4 dB step size, and 200 Monte Carlo trials are performed for each SNR. The array shape is the same for the entire SNR range but is different in each Monte Carlo trial. The error measurement

PSO parameters						DDL parameters	
φ_1	φ_2	δ	P	Ψ	T	SNR$_a$	r_l
2.4	1.7	10^{-4} rad	30	0.5	3000	-15 dB	0.13

Table 4.3: *PSO and DDL control parameters*

is the sum of squared error of the three signals $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\|^2$ averaged over the 200 Monte Carlo trials.

According to [6], for the far-field case a nominally circular array shape can be calibrated if i) there are at least three sources and ii) one sensor location and its direction to a second are known. Therefore $\Delta x_1, \Delta y_1$ and Δy_2 are set to zero, fulfilling the identifiability condition. In consequence the dimension of the solution space is $D = 10$.

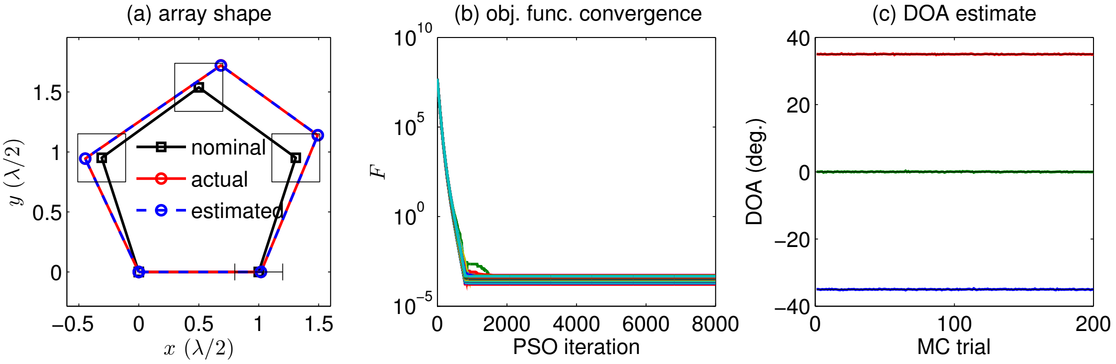

Figure 4.9: *The success of PSO-DDL at SNR = 26 dB. (a) The array shape estimation compared with nominal and actual shapes. The thin-line black squares and line segment indicate the perturbation boundaries. (b) The pattern of F in the process of 200 optimizations. (c) The 200 DOA estimates.*

First of all, the same experiment as that in Fig. 4.4 is performed by PSO-DDL. The control parameters of PSO and DDL are specified in Tab. 4.3. PSO is terminated after the maximum number of iterations T is reached, which amounts to PT objective function evaluations per optimisation. The result is shown in Fig. 4.9. Compared with Fig. 4.4, Fig. 4.9(a) shows as an example that the array shape which cannot be calibrated by PSO alone is now calibrated by PSO-DDL; Fig. 4.9(b) shows that the optimized values of F are well grouped in one region instead of separated in two bundles. In addition, for the first 800 PSO iterations, which are diagonal loaded ones, the optimized F value follows an approximately exponential decline that expresses the effect of DDL; Fig. 4.9(c) shows that all the calibrated DOA estimates are close to true values rather than many an outlier in Fig. 4.4(c) straying far away. The statistical performance is then presented as follows.

The result of PSO-DDL is compared with WF1 and WF2 algorithms and the CRB which is based on the stochastic signal assumption (section 8.11.2 of [19]). The Flanagan algorithm [90] is not compared with as its final step is WF2; the algorithm in [91] is not compared to as it lacks the initialization step. Although the coarse calibration steps of [90] can be applied to PSO-DDL for larger b, and existing DOA estimation algorithms can be used to initialise [91], a thorough investigation is outwith the scope of this work as initializers differ in sensitivity to array shape perturbation and consequently affect the self-calibration result.

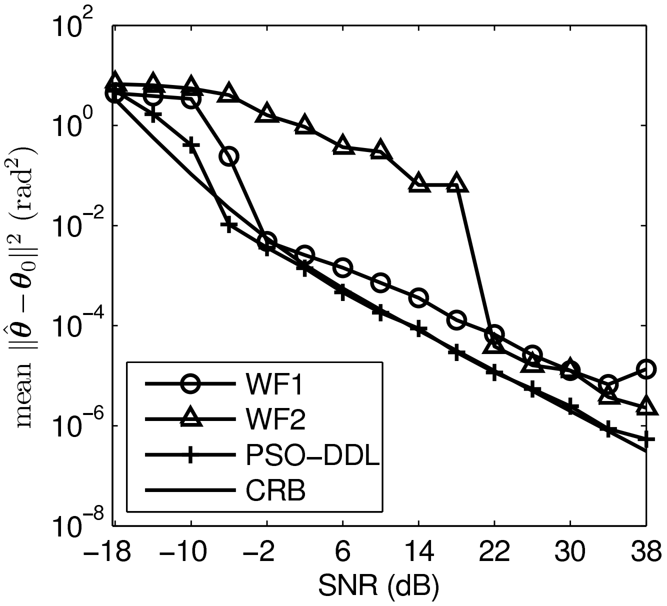

Figure 4.10: *Comparison of calibration accuracy under small sensor location errors, $b = 0.05a$. $K = 1000$.*

In the first experiment the perturbation boundary $b = 0.05a$ is considered. The DOA ranges from -180° to 180°. One observes from Fig. 4.10 that both WF1 and WF2 succeed in calibrating the array shape at SNR's higher than 22 dB, with a similar MSE sum. Due to the fact that WF2 is based on the eigen-structure objective function, it shows a threshold at 22 dB, below which SNR the MSE sum increases dramatically, whereas WF1 shows a much lower threshold at -2 dB, due to its CML objective function. Nevertheless neither of the two algorithms approaches the CRB at SNR's higher than their thresholds as a result of the Taylor approximation that renders the bias the same order of magnitude as the standard deviation (STD), making the MSE considerably higher than the CRB despite the fact that the STD approaches the CRB [1, 2, 90]. PSO-DDL shows a threshold at -6 dB that is lower than both WF1 and WF2, and above that SNR its MSE sum approaches the CRB. This MSE sum is apparently lower than that of both WF1 and WF2. This experiment shows that under small perturbations PSO-DDL has an optimal accuracy that is lower than WF1 and WF2 and it has a lower threshold SNR.

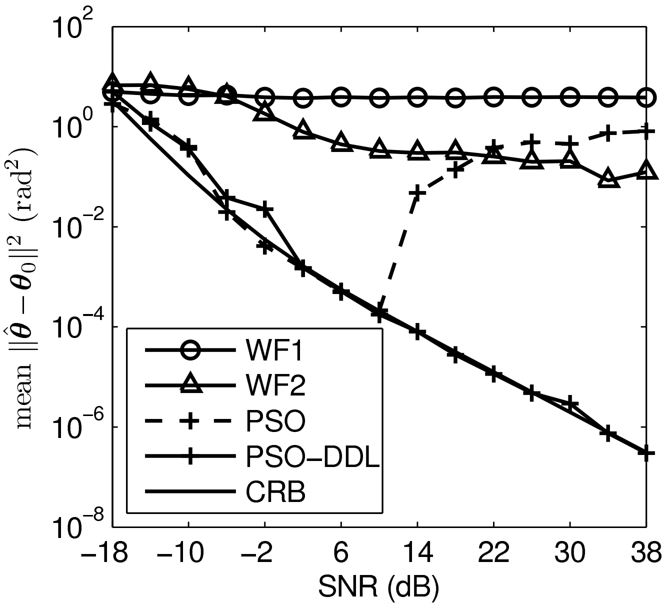

Figure 4.11: *Comparison of calibration accuracy under large sensor location errors, $b = 0.2a$. $K = 1000$.*

In the second experiment the perturbation boundary b is enlarged to $0.2a$. Two of the PSO and DDL parameters are adjusted to $T = 8000$ and $r_l = 0.1$ for the more challenging case. Other settings are the same as in the first experiment. It is observed from Fig. 4.11 that PSO-DDL succeeds in calibrating the array with a CRB-approaching MSE sum of DOA parameters at SNR's greater than 2 dB, whereas both WF1 and WF2 fail with estimation errors much higher than PSO-DDL and the CRB, and the errors of WF1 and WF2 do not decrease as the SNR increases. We also note that without DDL, PSO alone would fail with large estimation errors, comparable to WF1 and WF2 when the SNR is greater than 18 dB. This experiment shows that PSO-DDL has much higher robustness against large sensor location errors than both WF1 and WF2, and it retains optimal accuracy at large perturbations.

In the third and fourth experiments the self-calibration is performed at $K = 20$. Figs. 4.12 and 4.13 show the results at small and large perturbations, respectively. The other array-source settings are the same as the $K = 1000$ case. PSO and WF1 are the same, too, except that WF2 is now initialized by the multiple signal classification (MUSIC) algorithm, which is the initialization in the original literature.

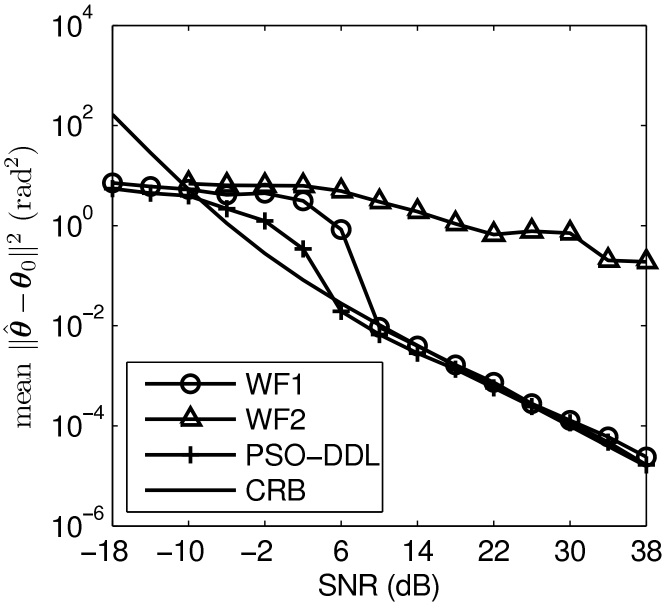

Figure 4.12: *Comparison of calibration accuracy under small sensor location errors, $b = 0.05a$. $K = 20$.*

It is seen again that PSO-DDL outperforms WF1 and WF2 in terms of robustness and accuracy. Fig. 4.12 shows that under small perturbations, PSO-DDL calibrates the arrays with CRB-approaching accuracy at SNR greater than 6 dB. WF1 performs the self-calibration with an MSE slightly higher than PSO-DDL, and a threshold SNR at 10 dB, which is higher than PSO-DDL. WF2 failed to calibrate the arrays for the entire SNR range due to the low snapshot number. Fig. 4.13 shows that under large perturbations, both WF1 and WF2 failed, but PSO-DDL succeeded with CRB-approaching accuracy at SNR greater than 10 dB. In addition, it also shows that without DDL, PSO alone failed with a large error that never approached the CRB. Compared with the large number of snapshots, the small number of snapshots affects PSO-DDL only in that the threshold SNR is higher. It affects WF1 in the same manner, but affects WF2 more seriously.

The third and fourth experiments indicate that although the analysis of DL is carried out asymptotically, the DL still takes effect at very small number of snapshots, and the PSO-DDL technique performs well, too.

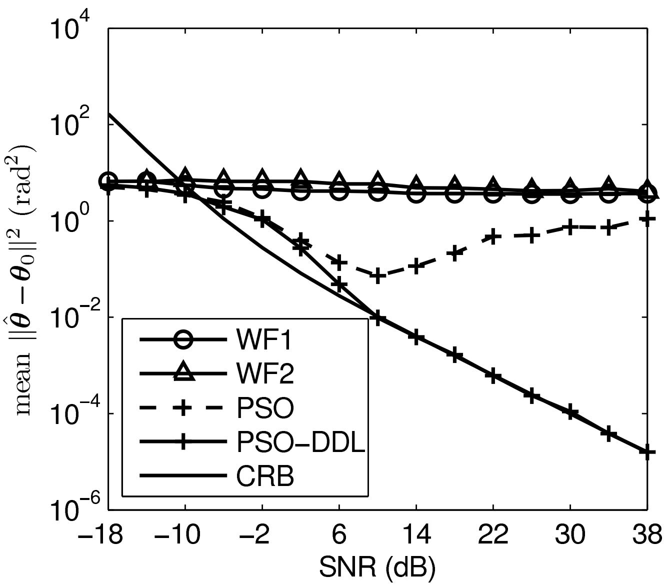

Figure 4.13: *Comparison of calibration accuracy under large sensor location errors,* $b = 0.2a$. $K = 20$.

4.4.2 DE results

In support of the remark that DDL is a universal technique that can be coupled with other algorithms to improve performance, a different stochastic global optimization algorithm, DE [92] has also been employed to optimize F. In the following, firstly the algorithm and its coupling to DDL are introduced, and then its performances with and without the aid of DDL are compared. However the comparison of PSO-DDL and DE-DDL is beyond the scope of this study.

DE is a stochastic algorithm method that updates its population in each generation by differential perturbation and evolutionary crossover. The population consists of P members, each of which, $\boldsymbol{x}^{(p)}$, is a parameter vector in the D-dimensional solution space. For an estimation problem, D is the number of the parameters to be estimated.

The pth member of the population, $\boldsymbol{x}^{(p)}$, in the $(g+1)$st generation is updated from the gth generation by the following two operations:

1) Difference: Three members of the gth generation are randomly chosen, their indices r_1, r_2 and r_3 being generated by a random number generator with uniform distribution, and are mutually different and different from p. The weighted difference between $\boldsymbol{x}^{(r_2,g)}$ and $\boldsymbol{x}^{(r_3,g)}$ is

added to $\boldsymbol{x}^{(r_1,g)}$ to form a new parameter vector $\boldsymbol{v}^{(p,g)}$:

$$\boldsymbol{v}^{(p,g)} = \boldsymbol{x}^{(r_1,g)} + w\left[\boldsymbol{x}^{(r_2,g)} - \boldsymbol{x}^{(r_3,g)}\right],$$

where $w \in [0, 2]$ is the weight of the added difference.

2) Evolution: The new parameter vector $\boldsymbol{v}^{(p,g)}$ is then randomly crossed over with $\boldsymbol{x}^{(p,g)}$ to form another new parameter vector $\boldsymbol{u}^{(p,g)}$. If $\boldsymbol{u}^{(p,g)}$ is written out in full form as,

$$\boldsymbol{u}^{(p,g)} = \left[u_1^{(p,g)}, \cdots, u_d^{(p,g)}, \cdots, u_D^{(p,g)}\right]^T,$$

then the random crossover can be expressed as

$$u_d^{(p,g)} = \begin{cases} v_d^{(p,g)} & d = \langle r\rangle_D, \langle r+1\rangle_D, \cdots \langle r+R-1\rangle_D \\ x_d^{(p,g)} & \text{other } d \end{cases}$$

for $d = 1, 2, \cdots, D$, where $\langle\rangle_D$ stands for the modulo operation with modulus D, r is a random integer uniformly drawn from the interval $[1, D]$ and R is a random number generated as in Alg. 4 so that $P(R \geqslant N_L) = P_c^{N_L-1}$, $N_L > 0$. Here P_c is the crossover probability, and is a control parameter to be set. In Alg. 4, rand() generates a random number in $[0, 1)$.

```
Algorithm 4: $R$-gen
Input: $P_c$
Output: $R$
Initialization: $R = 1$
while rand()< $P_c$ and $R < D$ do
    $R = R + 1$
end
return $R$
```

Finally $\boldsymbol{u}^{(p,g)}$ is compared with $\boldsymbol{x}^{(p,g)}$ in terms of the objective function value, and subsequently the pth member in the $g+1$th generation $\boldsymbol{x}^{(p,g+1)}$ is updated as

$$\boldsymbol{x}^{(p,g+1)} = \begin{cases} \boldsymbol{u}^{(p,g)} & F\left(\boldsymbol{u}^{(p,g)}\right) < F\left(\boldsymbol{x}^{(p,g)}\right) \\ \boldsymbol{x}^{(p,g)} & \text{otherwise} \end{cases}.$$

The operations of 1) difference and 2) evolution are repeated for $p = 1, 2, \cdots, P$.

In addition to the basic DE algorithm, the boundary control, parameter mapping and DOA parameter collision avoidance are also needed for the array shape self-calibration problem. The same techniques used for PSO are employed. As will be shown later, the problem of obscure global minimum hinders DE the same way as PSO, and once again the DDL guides the optimization algorithm through. To sum up, the DE-DDL is tabulated in Alg. 5.

Algorithm 5: DE-DDL

Input: DE and DDL parameters, objective function, solution space definition

Output: DOA and sensor perturbation estimates

```
Population initialization: random normalized locations;
for each generation do
    Load diagonally the sample covariance matrix according to Eqs. (4.14) and (4.15);
    for each member do
        Evaluate the objective function of current generation according to Eq. (2.24) with
        $\widetilde{\widetilde{\boldsymbol{R}}}(t)$ in Eq. (4.15);
        Update particle location according to 1) difference and 2) evolution;
        if member location out of boundary or collision at DOA dimensions then
            Set random location;
        end
    end
    Check termination criterion;
end
return final best member's location;
```

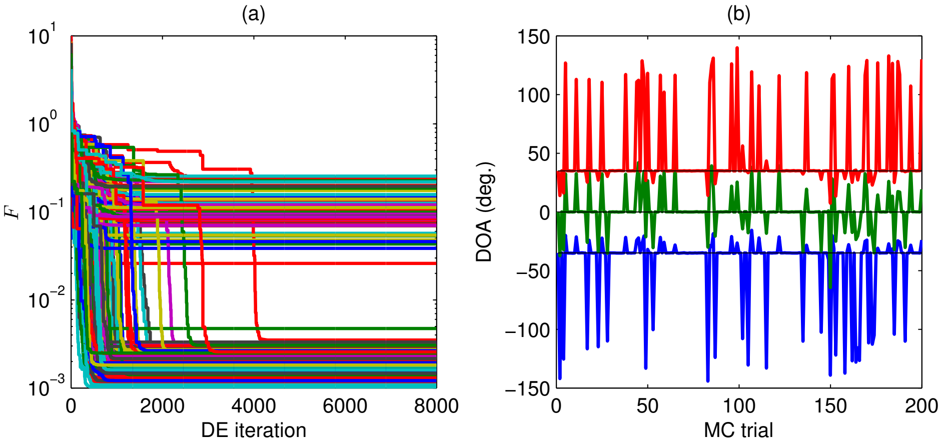

Figure 4.14: *DE performance at SNR* $=$ 22 *dB. Far-field self-calibration.*

Figs. 4.14 and 4.15 show the simulation results of the same DE with and without DDL. The array-source settings are exactly the same as that in the second experiment of section 4.4.1, or in other words F is the same. The SNR shown is 22 dB. Fig. 4.14(a) shows that a significant

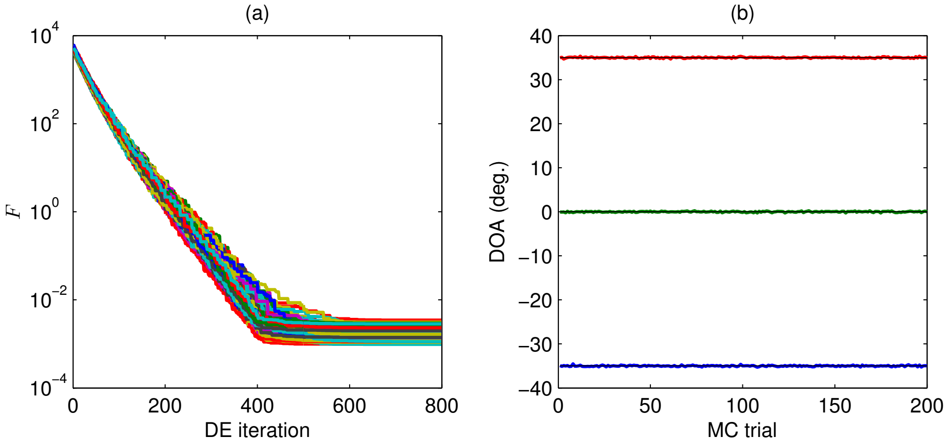

Figure 4.15: *DE-DDL performance at SNR* $= 22$ *dB. Far-field self-calibration.*

proportion of the 200 MC trials performed by DE alone are trapped in local minima, ending up with optimized F values grouped higher than the global minimum bundle, and correspondingly Fig. 4.14(b) shows a large proportion of spiky false DOA estimates. In contrast, Fig. 4.15(a) shows that DE-DDL optimizes F all to the global minimum, the straight slope between the 1st and the 400th DE iterations reflecting the effect of DDL, and correspondingly Fig. 4.15(b) shows that all the DOA estimates are close to the true values. Note that this is achieved in one tenth of the number of iterations of lone DE.

4.5 Conclusion

In this chapter, the PSO-DDL paradigm is proposed to optimize the UML objective function for the array shape self-calibration problem. Compared with the existing WF1 and WF2 algorithms, the proposed method is more robust to large shape perturbation and has lower calibration error due to the direct optimization on the UML objective function. Its accuracy approaches the CRB at an SNR higher than the threshold, and the threshold is lower than that of WF1 and WF2. The CRB-approaching accuracy is retained even for large perturbations.

Furthermore, PSO does not need initialization close to the true parameters. The DDL technique is of general value in its transformation of the objective function and thus can be coupled with other global optimization algorithms to improve performance. This is supported by simulation results of the DE algorithm. The PSO-DDL method is simple to implement and computationally feasible.

Chapter 5
Near-field offline calibration

5.1 Introduction

The scenario considered in chapters 3 and 4 is far field. Another important scenario is near field. As shown in section 2.1.1 of chapter 2, the fundamental difference between near field and far field is that the first-order Taylor approximation in the far field is invalid in the near field, rendering the steering vector more complicated. Naturally, a second-order Taylor approximation, namely the Fresnel approximation, accommodates sources closer to the array, and indeed this approximation has been adopted by a large number of papers in the literature on the problem of near-field source localization, see [93–100] for example. Even a third-order Taylor approximation has been proposed for near-field source localization [101]. However, the Taylor approximations introduce errors that grow as the source comes closer to the array, and thus are not applicable to the entire near field region [93]. There are also near-field source localization methods that are independent of the Taylor approximation, where an alternate gradient search is involved for optimization [102–104], or the constraint of uniform distribution of the sensors is imposed for the application of a rooting technique [105]. For the calibration problem, He *et al.* [35] recently proposed a non-parametric approach for near-field offline calibration, based on calibration matrix interpolation, however it requires a single source at a large number of known positions that form a two-dimensional grid.

In this chapter, the near-field calibration is carried out using the method of the unconditional maximum likelihood (UML) estimator optimized by the particle swarm optimization – decaying diagonal loading (PSO-DDL) technique. Specifically, array shape offline calibration is carried out using narrowband near-field sources at known positions. The unknown sensor locations are estimated by a UML estimator whose objective function is optimized by PSO-DDL. To the best of the author's knowledge, the UML estimator has not been applied in near-field source localization or near-field array calibration before, due to the complexity of the cost function optimization. Nevertheless PSO-DDL is suited for the optimization of the UML cost function because it does not rely on the structure of the cost function as do other local optimization algorithms. In addition, it does not need accurate initialization as required by the local optimization algorithms. Since the UML estimator is employed, and no approximation is involved, the method is applicable to the entire near field region and the calibration result is optimal, approaching the Cramér-Rao bound (CRB). Compared with the non-parametric approach in [35] that requires data from a large number of known source positions, the proposed method needs as few as 2 sources at known positions to calibrate an array. The structure of this chapter is as follows. It starts with formulating the UML objective function in section 5.2, followed by configuring PSO-DDL in section 5.3. The simulation results are presented in section 5.4, including the differential evolution (DE) results. Finally, section 5.5 concludes the chapter.

5.2 Problem formulation

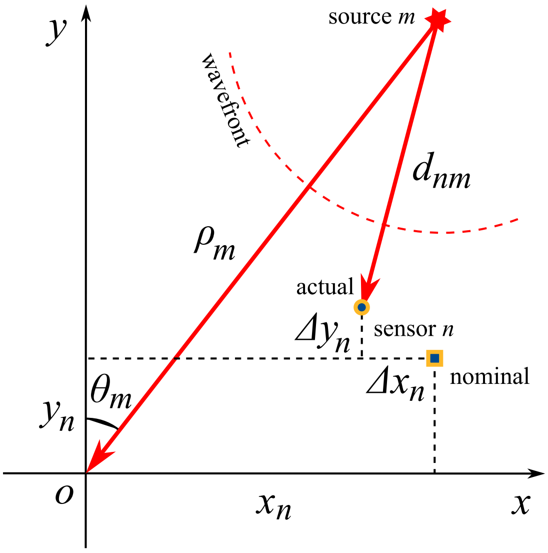

Figure 5.1: *The array-source geometry of near-field case.*

For the near-field case, the n,mth element of the array manifold $\boldsymbol{H}\left(\boldsymbol{\theta},\boldsymbol{\rho},\boldsymbol{\delta}\right)$ is given by

$$\begin{aligned} &H_{nm}\left(\theta_m,\rho_m,\Delta x_n,\Delta y_n\right)=\frac{1}{d_{nm}}\exp\left\{j\frac{2\pi}{\lambda}(\rho_m-d_{nm})\right\}, \\ &n=1,\ldots,N,\ m=1,\ldots,M, \end{aligned} \tag{5.1}$$

where θ_m is the DOA of the mth source and ρ_m its range.

$$d_{nm}=\sqrt{\left[\rho_m\sin\theta_m-(x_n+\Delta x_n)\right]^2+\left[\rho_m\cos\theta_m-(y_n+\Delta y_n)\right]^2}$$

is the distance between the nth sensor and the mth source. Without loss of generality, the reference point of zero phase is chosen as the origin of the coordinates.

The H_{nm} for near-field, Eq. (5.1), differs from that for far-field, Eq. (3.2) in two aspects: 1) The magnitude factor $\frac{1}{d_{nm}}$, which represents the signal strength loss in free space. 2) The phase factor $\frac{2\pi}{\lambda}(\rho_m - d_{nm})$ which denotes the phase delay from the source to the sensor. They are both functions of source bearing and range, resulting from the proximity of the source to the array.

Following [19], the UML estimator for $\boldsymbol{\delta}$ can be expressed as

$$\hat{\boldsymbol{\delta}} = \arg\min_{\boldsymbol{\delta}} \underbrace{\left\{ \det \left[\boldsymbol{P_H} \widehat{\boldsymbol{R}} \boldsymbol{P_H} + \frac{\operatorname{tr}(\boldsymbol{P}^{\perp}_{\boldsymbol{H}} \widehat{\boldsymbol{R}}) \boldsymbol{P}^{\perp}_{\boldsymbol{H}}}{N - M} \right] \right\}}_{F(\boldsymbol{\delta})}, \tag{5.2}$$

where $\widehat{\boldsymbol{R}} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{x}(k) \boldsymbol{x}^H(k)$ is the sample covariance matrix, $\boldsymbol{P_H} = \boldsymbol{H}(\boldsymbol{H}^H \boldsymbol{H})^{-1} \boldsymbol{H}^H$ is the projection matrix of $\boldsymbol{H}$, and $\boldsymbol{P}^{\perp}_{\boldsymbol{H}} = \boldsymbol{I} - \boldsymbol{P_H}$ is its orthogonal complement. The dependence of $\boldsymbol{H}$ on $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ is suppressed for notational simplicity, and $F(\boldsymbol{\delta})$ denotes the objective function.

5.3 PSO-DDL

5.3.1 PSO

The PSO detailed in section 4.2 is employed to optimize the objective function $F(\boldsymbol{\delta})$ in Eq. (5.2). Its control parameters are given in Tab. 5.1. Compared with the parameters for far-field self-calibration in Tab. 4.3, the acceleration constants φ_1 and φ_2 are adjusted towards more exploration ability, while their sum is kept as 4.1 to balance the oscillation avoidance and convergence speed [83]. The number of particles P is increased to 100 and the maximum velocity $\varPsi$ is reduced to 0.01 to facilitate a finer search of the solution space. Accordingly the maximum number of PSO iterations is increased to 16000 for more search time. Overall the PSO is enhanced in search ability to accommodate the solution space of higher dimensions and a larger range in each dimension, at the cost of more computation load which is proportional to PT. The storage space needed is proportional to P, which is a minor concern since the parameter of each particle is simple and thus requires little storage space.

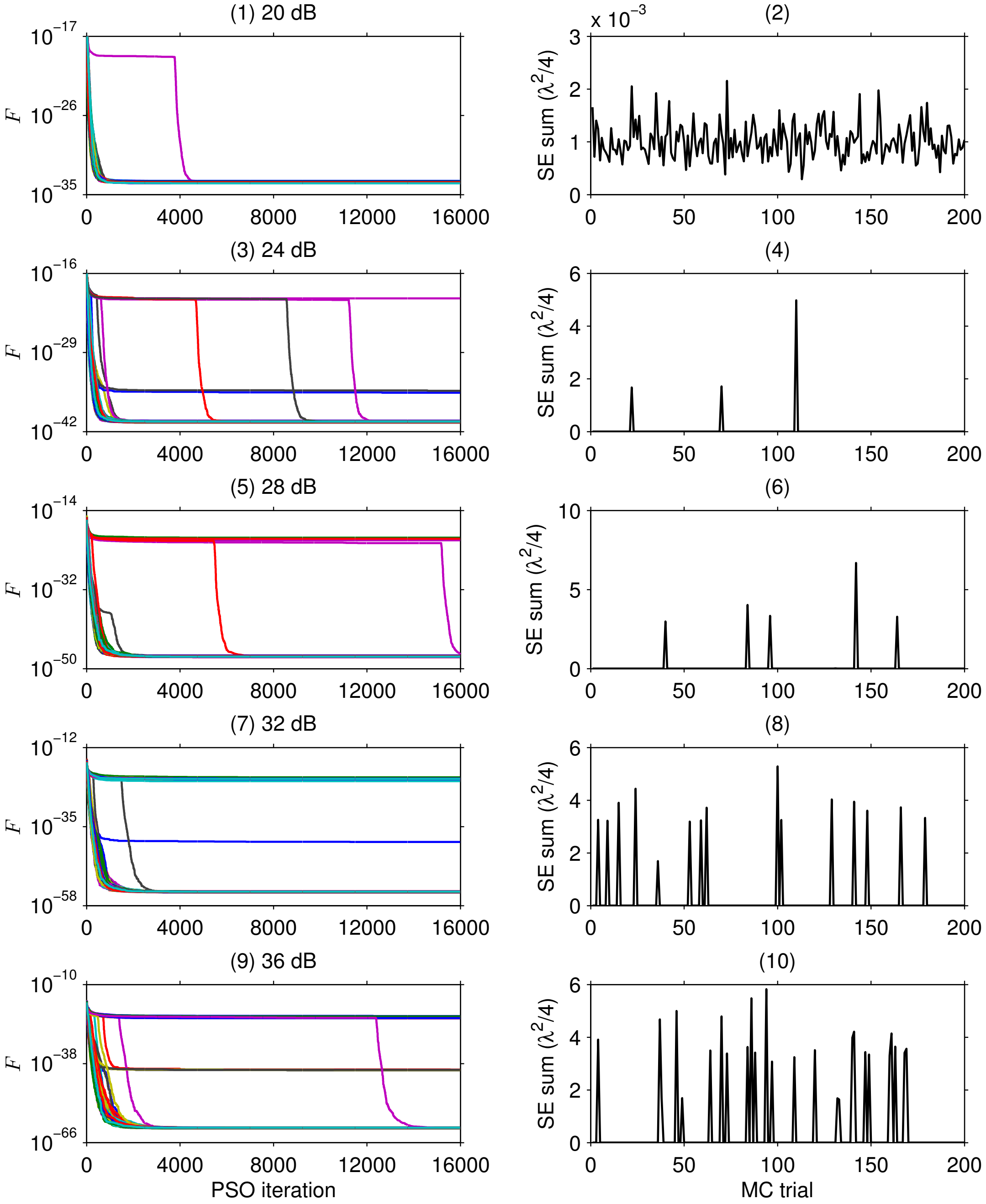

Figure 5.2: *PSO performance at critical SNR's. In the left column are the F convergence over 200 MC trials at 5 values of SNR. In the right column are the squared error (SE) sum of the final converged value of the perturbation estimates for each of the trials. The simulation environment is the same as that in section 5.4.1.*

φ_1	φ_2	P	Ψ	T
2.7	1.4	100	0.01	16000

Table 5.1: *PSO control parameters*

PSO encounters the same problem as the far-field self-calibration case in section 4.2: although it performs well when the signal-to-noise ratio (SNR) is low, it starts to fail when the SNR grows high. Fig. 5.2 demonstrates the failing process as SNR grows. The left column shows the F convergence over 200 MC trials at 5 values of SNR, and the right column shows the SE sum, defined as $\|\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}_0\|$, of the final converged value of the perturbation estimates for each of the trials. The simulation scenario is described in section 5.4.1 to avoid duplication.

Figs. 5.2(1) and (2) show the highest SNR simulated where PSO succeeded in all the 200 Monte Carlo (MC) trials. Fig. 5.2(1) shows that all the optimized F values are of the same order of magnitude, the difference resulting from the finite number of snapshots for the estimator. Accordingly the SE sums of sensor location estimates are similar to each other, as shown in Fig. 5.2(2). The concentration of optimized F values and SE sums indicate that all 200 PSO operations arrived at the global minimum of F.

Figs. 5.2(3) and (4) show that at the next higher SNR simulated, 24 dB, 3 out of the 200 PSO operations failed. Fig. 5.2(3) shows that although the majority of the optimized F values group about 10^{-41}, two float up around 10^{-36}, and one around 10^{-21}. In Fig. 5.2(4), 3 spiky SE sums show up, corresponding to the 3 high optimized F values previously mentioned. The extraordinarily high final optimized F values and SE sums indicate that PSO failed to reach the global minimum of F in the 3 operations.

In the following rows of Fig. 5.2, the proportion of failed PSO increases as the SNR increases, from 5 failures out of 200 at 28 dB to 15 failures at 32 dB and 26 failures at 36 dB. The entire Fig. 5.2 shows that PSO encounters a threshold SNR between 20 dB and 24 dB, greater than which PSO starts to fail in achieving the global minimum of F.

An example of the failure of a single PSO operation is shown in Fig. 5.3. It is one of the MC trials at SNR $=$ 36 dB, i.e. the highest SNR shown in Fig. 5.2. Fig. 5.3(a) shows a substantial difference between the estimated array shape and the actual one. The perturbation area enclosed by the dashed orange square of the sensor in the upper-left corner in Fig. 5.3(a) is magnified in Fig. 5.3(b) to show the trace of the stray PSO over its iterations for this sensor.

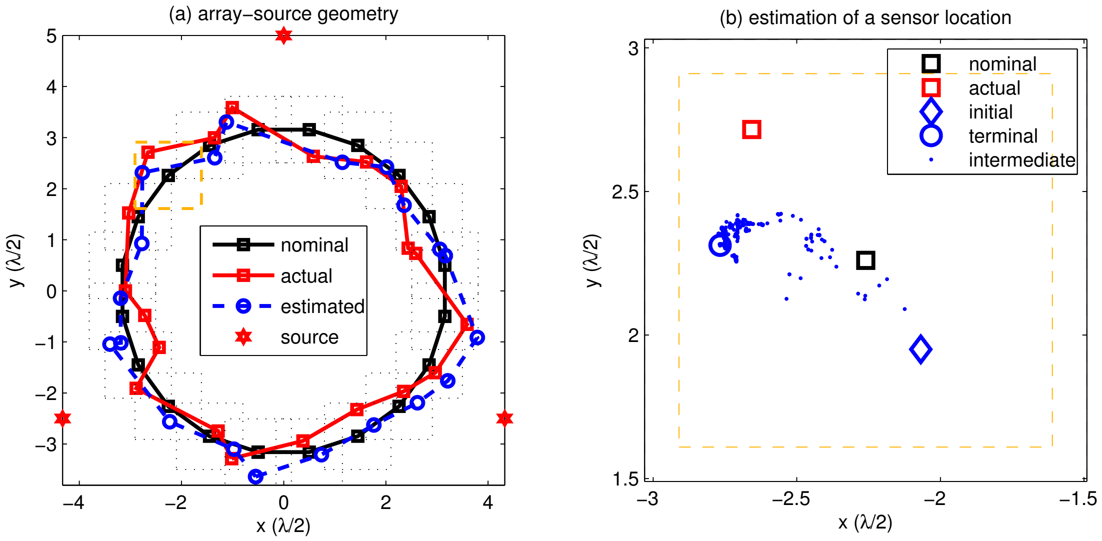

Figure 5.3: *Failure of PSO at SNR* $= 36$ *dB. (a) The calibrated array shape compared with the nominal and actual shapes. The dotted/dashed squares represent the error boundary. (b) A magnified portion of (a), encompassed by the yellow dashed square, showing the estimation process of one sensor position.*

5.3.2 DL effect on F

The cause of PSO failure at high SNR is again attributed to F's lack of a prominent global minimum at high SNR, as indicated by the mathematical analysis in section 4.3.1, and visualized in Fig. 5.4 for an example. In Fig. 5.4(a) F is degenerated to a function of the location of the first sensor, while all the other sensor locations are set at the actual ones. More specifically, $(\delta x_1, \delta y_1)$ is the deviation from the actual sensor location $(x_1 + \Delta x_1, y_1 + \Delta y_1)$. The same applies to F_l. Comparing Figs. 5.4(a) and (b) one finds that diagonal loading (DL) makes the global minimum, which lies at $(\delta x_1, \delta y_1) = (0, 0)$, more prominent.

At this stage it is clear that F in the case of near-field off-line calibration behaves in the same fashion as that in the case of far-field on-line calibration, in terms of the prominence of the global minimum at various SNR. Indeed, the analysis in section 4.3.1 is independent of the specific array manifold. Thus DDL again can be adopted to ease the search of the global minimum for PSO. The DDL control parameters for this case are in Tab.5.2. The larger r_l provides a slower reduction of the DL amount, which is in accordance with the lower maximum velocity Ψ in Tab. 5.1.

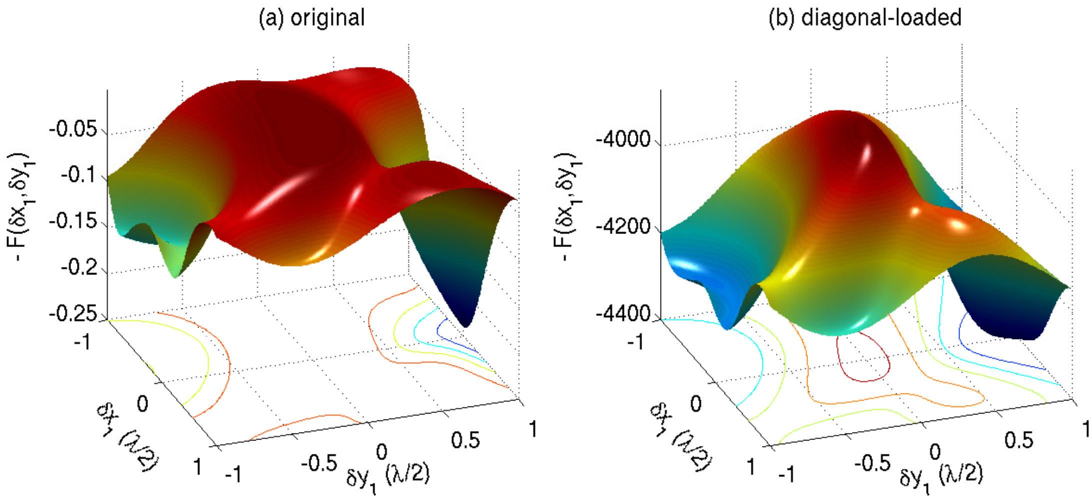

Figure 5.4: *Comparison of F before and after DL. The F is for: 6-sensor UCA with $a = \lambda/2$ inter-sensor spacing, perturbed by uniform random displacements with boundary $b = 0.1a$. 3 sources at bearings $-120°$, $0°$ and $120°$, all with the same range $r = 2a$ relative to the center of the nominal array. SNR $= 14$ dB, SNRa $= -5$ dB.*

5.4 Calibration results

A large number of experiments for various numbers of sensors and sources have been carried out. Three representative cases are presented in sections 5.4.1, 5.4.3 and 5.4.4, respectively. The first one is a 20-sensor circular array (CA), with 40 parameters to be simultaneously optimized, which demonstrates PSO-DDL's ability to handle high-dimensional parameter spaces. The second one is a 20-sensor linear array (LA), which demonstrates PSO-DDL's versatility when dealing with different array geometries. The third one is a 2-source geometry which highlights the advantage of near-field calibration that needs fewer sources than far-field calibration does. To demonstrate that the PSO-DDL can be combined with efficient local optimization algorithms after diagonal-loaded iterations for computation reduction, an example of the simplex algorithm succeeding PSO-DDL for the same simulation case in section 5.4.1 is presented in section 5.4.2. In support of the remark that the DL effect is independent of the specific global optimizer, DDL is also coupled with the DE algorithm to perform the same calibration task in section 5.4.1, the simulation results of which are presented in section 5.4.5.

SNR_a	r_l
-15 dB	0.4

Table 5.2: *DDL control parameters*

5.4.1 Calibration of a 20-sensor circular array

The case considered is the calibration of a 20-sensor array using 3 near-field sources. Specifically, the nominal array is a 20-sensor uniform circular array (UCA) with $a = \lambda/2$ inter-sensor spacing, perturbed by random displacements uniformly distributed in $[-b,\ b]$, where $b = 0.65a$. 3 sources are located at bearings $-120°$, $0°$ and $120°$, all with the same range $5a$ relative to the center of the nominal array. The geometry of the nominal array, actual array and sources is shown in Fig. 5.6(a), among calibration results. The identifiability condition is given in [50], and is satisfied here. The near field [14] is defined as range $\rho < 2D_a^2/\lambda$, where D_a is the dimension of the array. In this case $D_a = 3.20\lambda$ is the diameter of the array, hence $\rho < 20.43\lambda$ is the near field. The Fresnel region [14] is defined as $0.62\sqrt{D_a^3/\lambda} < \rho < 2D_a^2/\lambda$, which is $3.54\lambda < \rho < 20.43\lambda$ in this case. The range of the sources $\rho_m = 2.5\lambda$ falls in the inner part of the near field, beyond the Fresnel region. The sources are emitting equi-power, independent random signals, the additive Gaussian random noise are independent from sensor to sensor and independent from the signals. The PSO parameters are the same as those of Tab. 5.1, and the DL parameters are specified in Tab. 5.2.

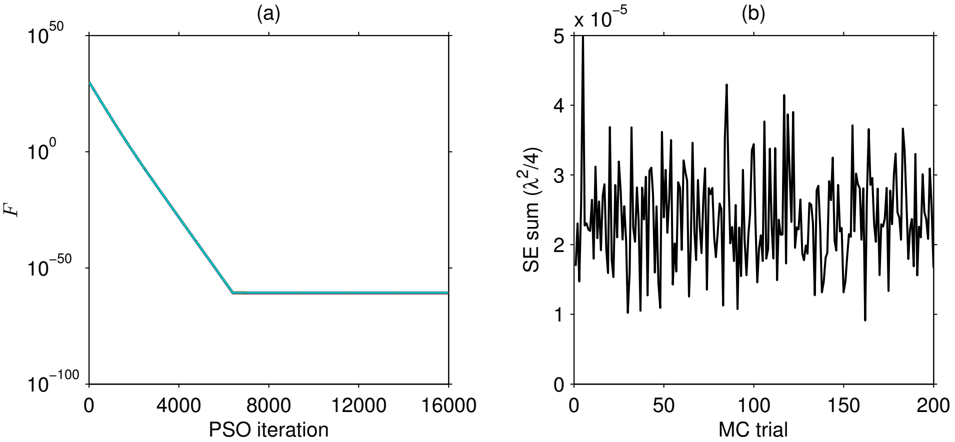

Figure 5.5: *PSO-DDL performance at SNR* $= 36$ *dB. (a) The* F *convergence pattern in the* 200 *MC trials. (b) The SE sum in the* 200 *MC trials.*

Fig. 5.5 illustrates the PSO-DDL in 200 MC trials at SNR $= 36$ dB. It is observed from Fig.

5.5(a) that all the optimized F values are so close to each other that the 200 curves overlap to the extent that only one can be seen. Correspondingly Fig. 5.5(b) shows that the calibration errors are all of the same order of magnitude. Compared with the PSO performance for the same task that is illustrated in plots (9) and (10) in Fig. 5.2, DDL helps PSO overcome the trap of local minima; as a result all the 200 optimizations arrive at the global minimum of F, yielding accurate estimates.

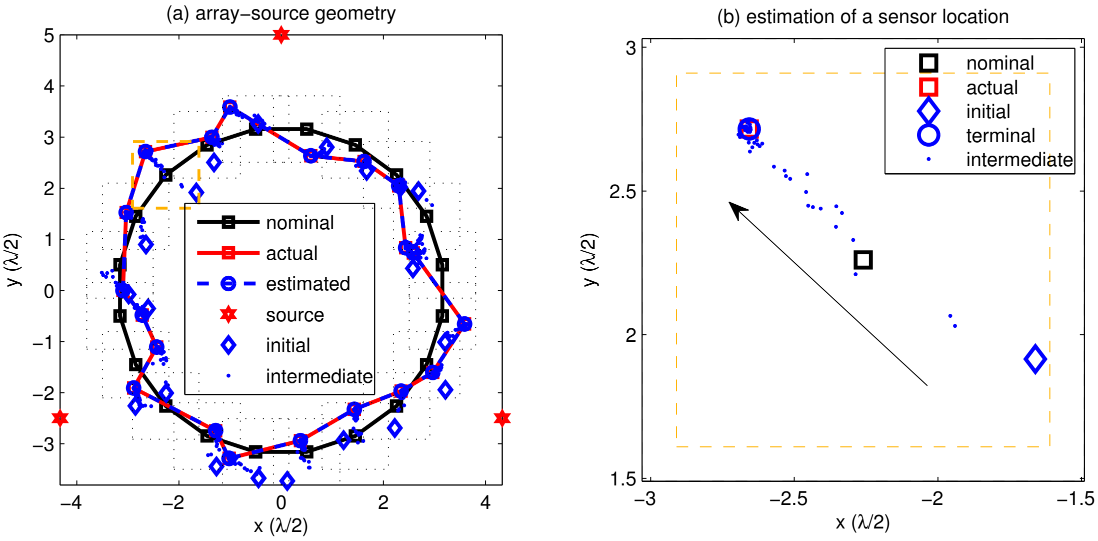

Figure 5.6: *A typical PSO-DDL operation at SNR* $=$ 36 *dB. (a) The calibrated array shape compared with the nominal and actual shapes. The dotted/dashed squares represent the error boundary. (b) A magnified portion of (a), encompassed by the yellow dashed square, showing the estimation process of one sensor position. The arrow indicates the direction of PSO estimation progress.*

Fig. 5.6 depicts a typical calibration process at SNR $=$ 36 dB, in which (b) is a magnified portion of (a). In Fig. 5.6(a), the dashed squares denote the boundaries of displacements. It is seen that the calibrated array shape, whose sensor positions are denoted by the blue circles, coincides with the actual one.

Fig. 5.6(b) depicts the calibration process of one sensor. The initial estimate of sensor position is a random guess within the perturbation boundary. As PSO iterates on, the intermediate estimates denoted by the blue dots, progress towards the actual sensor position. Every two consecutive dots are separated by 20 PSO iterations in the optimization. It is remarkable that although there appears to be a certain degree of random movements, the overall trajectory of the intermediate estimates aims for the actual sensor position in a straightforward pattern. This pattern can also be found in Fig. 5.6(a) for other sensors, which reflects the effect of swarm intelligence.

It is also noted that the majority of the journey towards the actual position is accomplished in the initial PSO iterations, and most of the PSO iterations are spent on later estimate refinement for achieving the maximal accuracy. There are 799 dots evenly dividing the 16000 PSO iterations, while only about two dozen of them can be seen outside the circle that represent the terminal estimate in Fig. 5.6(b). This slow convergence of PSO at later stages is further addressed in section 5.4.2.

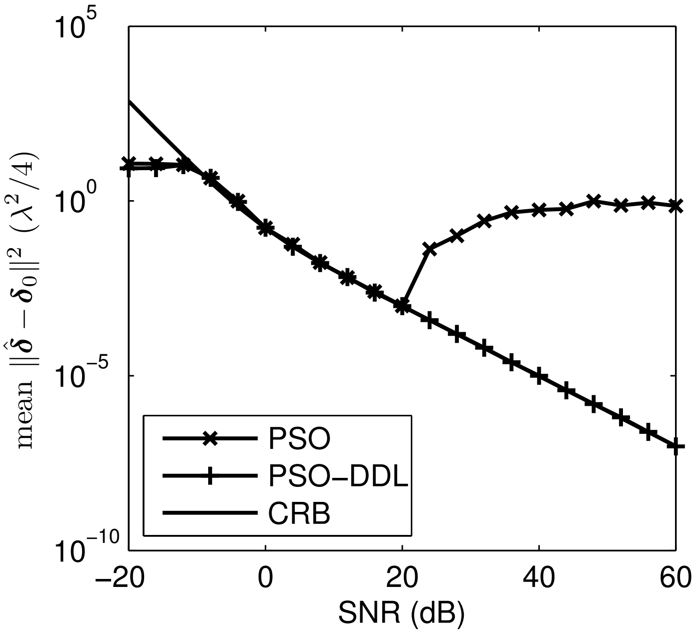

Figure 5.7: *The comparison of PSO and PSO-DDL's MSE's and the CRB. Circular array calibration.*

The calibration results are summarized in 5.7. The mean squared errors (MSE's) of PSO and PSO-DDL are compared to each other and to the CRB. The CRB expression can be found in Appendix B. For SNR equal to or lower than 20 dB, PSO yields accurate results that approach the CRB. The MSE of PSO results shows a threshold at SNR $=$ 20 dB; above that SNR the MSE increases and eventually levels off at a high value that is comparable to that of SNR $= -4$ dB. This threshold has also been seen in Fig. 5.2. From Figs. 5.2 and 5.7 it is clear that even a few failures can significantly increase the MSE, because the local minima are far away from the global minimum. In contrast, the MSE of PSO-DDL results approaches the CRB in the entire SNR range simulated. Its level-off in -12 to -20 dB is caused by the assumption that

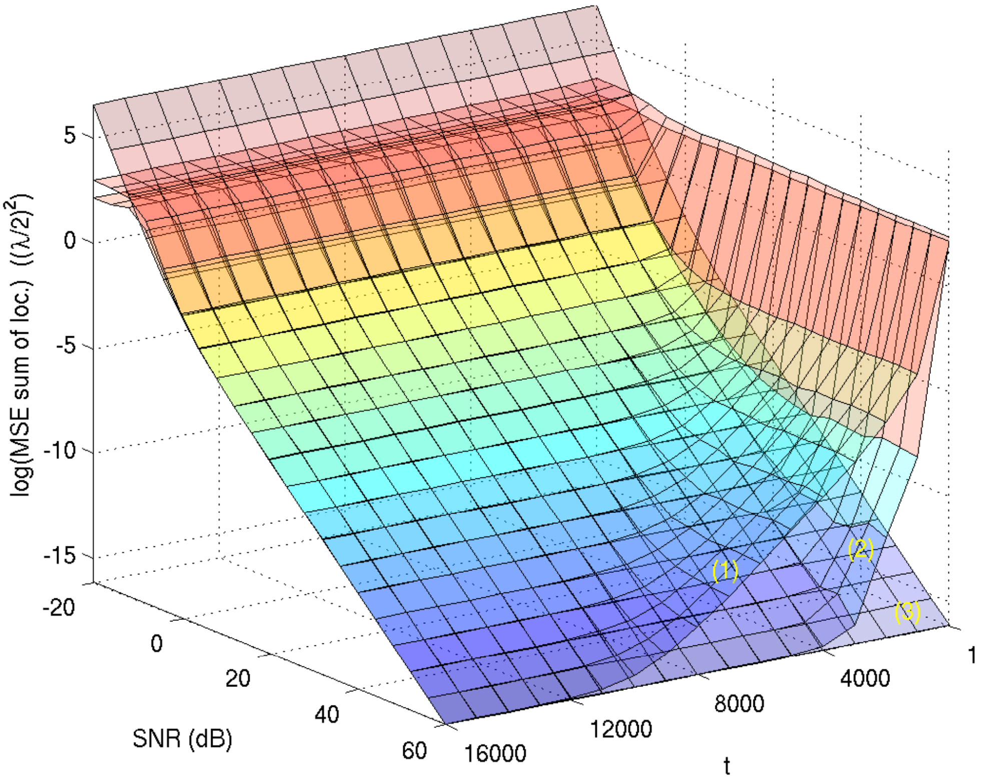

Figure 5.8: *The MSE sum surface of (1) PSO-DDL and (2) PSO-DDL-simplex compared with (3) the CRB.*

the sensor location error is bounded. Another observation is that the MSE's of neither PSO nor PSO-DDL show a threshold at low SNR, typically about 0 dB, where the MSE of DOA estimation and self-calibration usually abruptly jumps above the CRB.

5.4.2 The efficiency of local optimization algorithm for refinement

The slow convergence of PSO has been pointed out in section 5.4.1 by an example of a single calibration process. In the following, this issue is further investigated by an MC experiment in order to gain an averaged perspective. Then a local optimization technique is introduced to significantly reduce the computational burden incurred by the slow convergence.

The MC experiment is a detailed investigation of the slow convergence problem pointed out in section 5.4.1. In particular, the optimization results of every 10 PSO iterations are recorded to produce the MSE reduction as a function of increasing PSO iterations. The result is presented by surface (1) of Fig. 5.8. The PSO results at the recorded iterations are then used to initiate the simplex algorithm, whose final optimization results are used to produce surface (2) of Fig. 5.8.

From surface (1) of Fig. 5.8 one observes that for a fixed SNR, initially the MSE decreases exponentially as t increases linearly, then the MSE gradually levels off. This is more evident when the SNR is large; the higher the SNR, the more PSO iterations are needed to approach the CRB. As the termination criterion of PSO is set as a fixed maximal number of iterations, the number has to be sufficiently large for PSO to set on the global minimum of the highest SNR. As a result many of the PSO iterations at lower SNR are wasted after the CRB has already been achieved.

Another observation is that the local minima are considerably higher than the global minimum, as shown by the left column of Fig. 5.2. The difference is more than several orders of magnitude for the SNR range illustrated. If the PSO can escape the trap of the local minima, it plunges to the basin of the global minimum quickly, and the rest of the iterations are spent on the refinement of the estimate, in a slow exponential fashion as shown in Fig. 5.8, while the values of F hardly change during this process, as shown by the left column of Fig. 5.2. To the contrary, in Fig. 5.5 the F values optimized by PSO-DDL follow the exponential decrease induced by the exponential DL schedule, and after the diagonal loaded iterations the F values arrive at the order of magnitude of the global minimum. A reasonable inference follows that after the diagonal loaded iterations, PSO has arrived at the basin of the global minimum that contains no local minima. Therefore a local optimization algorithm is applicable immediately after or even before the end of the diagonal loaded PSO iterations to save the large number of PSO iterations after DL for estimate refinement.

The simplex algorithm is used as an example to demonstrate the applicability and efficiency of the local optimization algorithm. [106] is referred to for more details of the algorithm. It succeeds PSO-DDL to perform the local optimization. The succession point are PSO iterations No. $1, 1000, 2000, \cdots, 16000$. The PSO iteration No. 1 is in fact the random initialization step, hence it serves as the random initialization of simplex for comparison with other PSO initializations.

Surface (2) of Fig. 5.8 shows the simplex performance at varied succession point t. Take the curve of SNR $= 60$ dB for example: the later the succession point, the lower the MSE of simplex optimization results, which indicates that the more PSO iterations taken, the more PSO's of the 200 trials have entered the global minimum basin. It can be seen that 5000 PSO iterations with the aid of DDL guarantees entrance to the global minimum basin and consequently the MSE of simplex results approaches the CRB. It is suggested by surface (1) that the lower the SNR, the fewer iterations are needed to arrive at the global minimum basin, and this is evidenced by surface (2): for SNR $= 52$ dB, 4000 PSO iterations suffice for simplex to approach the CRB; for SNR $= 44$ dB, 3000 suffice and the trend goes on. Even if the succession point is fixed as the maximum needed, 5000 PSO iterations, and the PSO-DDL is terminated at 12000 iterations which is shown by surface (1) to be the minimum for CRB approach, simplex saves 58.3% of the PSO iterations while its own computational cost is extremely small.

5.4.3 Linear array calibration using near-field sources

Besides the circular array (CA), another practically important array geometry is the linear array (LA). Unlike the case of far-field self-calibration, the LA can be calibrated with near-field sources [50]. In this sub-section the LA calibration is carried out using the same PSO-DDL as that in section 5.4.1 to show PSO-DDL's versatility in dealing with different array shapes. The calibration again supports the remark that the DL effect is independent of the specific array manifold.

In the simulation, apart from the array geometry and the SNR range, all settings of the LA calibration are exactly the same as those of the CA calibration in section 5.4.1. The array is the CA in Fig. 5.6(a) “linearized”, as shown in Fig. 5.9. The nominal LA has 20 sensors, with half-wavelength inter-sensor spacing. It lies on the x-axis of the Cartesian coordinates and its centroid coincides with the origin. The sensor locations are displaced to the actual locations by the same perturbation as for the CA in section 5.4.1. All the other simulation settings, such as calibration source placement, signal and noise environment, are also the same as those of section 5.4.1.

The calibration result is illustrated in Fig. 5.10. The MSE sums of PSO and PSO-DDL are compared with each other and with the CRB. The SNR simulated varies from -20 dB to 40 dB with 4 dB step size. For SNR $\leqslant 16$ dB, PSO yields accurate results that approach the CRB. However it shows a threshold at SNR $= 16$ dB; above that SNR the MSE sum increases and eventually levels off at a high value that is comparable to that of SNR $= -4$ dB. In contrast,

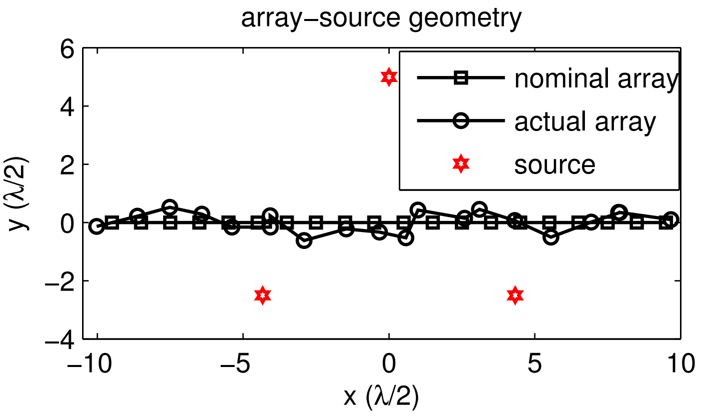

Figure 5.9: *Array-source geometry for nominally linear array calibration.*

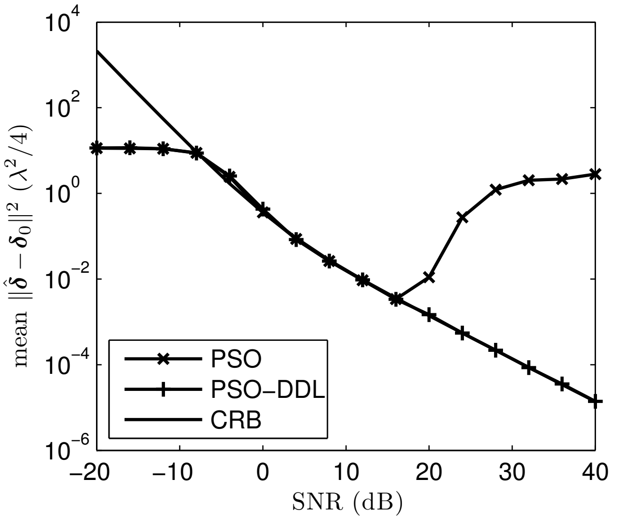

Figure 5.10: *The comparison of PSO and PSO-DDL's MSE's and the CRB. Linear array calibration.*

the MSE sum of PSO-DDL result approaches the CRB in the entire SNR range simulated. Its level-off in -8 to -20 dB is a result of the assumption that the sensor location error is limited to $[-0.65a, 0.65a]$. As the CRB goes to ∞ when the SNR falls to $-\infty$, the MSE sum is bound to be lower than the CRB at low SNR. Another observation is that the MSE sums of neither PSO nor PSO-DDL demonstrate a threshold at low SNR, typically about 0 dB, where usually the MSE of DOA estimation and self-calibration sharply rises above the CRB.

Compared with the CA calibration in section 5.4.1, PSO for LA calibration shows a threshold 4 dB lower than that for CA, and the start of the level-off at low SNR for both PSO and PSO-DDL is 4 dB higher for LA calibration than CA calibration. These indicate that the calibration of LA is slightly more difficult than CA in this configuration.

5.4.4 Two-source calibration

In a calibration process, it is an advantage to use the least number of sources to calibrate the most sensors. The significance of two-source calibration lies in that two near-field sources are able to calibrate an array but two far-field sources are not. Levi and Messer [50] conducted a geometrical analysis of the sufficient conditions obtained by statistical and numerical means for array shape calibration using near or far field sources or both in known or unknown positions or both. The result of the geometrical analysis is in accordance with the numerical results of the simulation and CRB evaluation in sections 5.4.4.1 and 5.4.4.2 that two far-field sources are incapable of eliminating the translational residual. The numerical results also suggest that the sufficient condition obtained from geometrical analysis for two far-field sources calibration is also the necessary condition, in agreement with the claim “array position calibration cannot be obtained under any circumstances” (for far-field calibrating sources) in [50]. However, the capability of two far-field sources in fully calibrating an array demonstrates that the sufficient condition obtained from geometrical analysis in [50] is not the necessary condition. In particular, the extra near-field source other than the two calibrating sources (case 13, Table 2 in [50]) is not necessary for a full array shape calibration.

5.4.4.1 The advantage of near-field calibration over far-field calibration

First of all, a simulation case is used to show the difference between near-field and far-field calibration, and then a numerical evaluation of the CRB is used to provide more information

PSO parameters					DDL parameters	
φ_1	φ_2	P	Ψ	T	SNR$_a$	r_l
2.7	1.4	100	0.01	2000	-15 dB	0.7

Table 5.3: *PSO and DDL control parameters*

from another perspective.

In the simulation, the nominal array is a 6-sensor UCA with $a = \lambda/2$ inter-sensor spacing, perturbed by random displacements uniformly distributed in $[-b,\ b]$, where $b = 0.4a$. For the near-field case, the 2 sources are positioned at bearings 0° and 90°, both with the same range $2a$ relative to the center of the nominal array. For the far-field case, the 2 sources are located at bearings 0° and 90°. The sources are emitting equi-power, independent random signals, the additive Gaussian random noise are independent from sensor to sensor and independent from the signals.

The objective function is the UML of Eq. (5.2). It is first coarsely optimized by PSO-DDL and then refined by simplex. The control parameters of PSO and DDL are tabulated in Tab. 5.3. The major change is that the number of PSO iterations is dramatically reduced to 2000 compared with 16000 in Tab. 5.1, because here PSO-DDL only serves as an initializer for simplex. In the mean time r_l is increased to 0.7 so that the majority of the PSO iterations are diagonal loaded iterations to reserve PSO-DDL's ability of arriving at the global minimum basin. The bulk of PSO iterations for estimate refinement is replaced by simplex.

The calibration result is shown by the MSE comparison in Fig. 5.11. It is clearly seen that the MSE of 2-source near-field calibration approaches the CRB, whereas far-field MSE is extremely high. The simplex refinement effect is shown by the contrast between the MSE of PSO-DDL with and without simplex in the near-field case, whereas for the far-field case the optimization of PSO-DDL is so erroneous that the simplex is unable to make any improvement. In the following the cause of the error of far-field calibration is found to be the far-field geometry rather than the UML estimator or the optimization technique.

Fig. 5.12(a) shows the far-field calibration result by the comparison of array shapes. The simplex-refined result is visually the same as PSO-DDL's therefore the former is not shown. It is seen that the estimated array shape is visually identical to the actual one, but a translational error displaces it. If the translational error is subtracted from the the estimated sensor position,

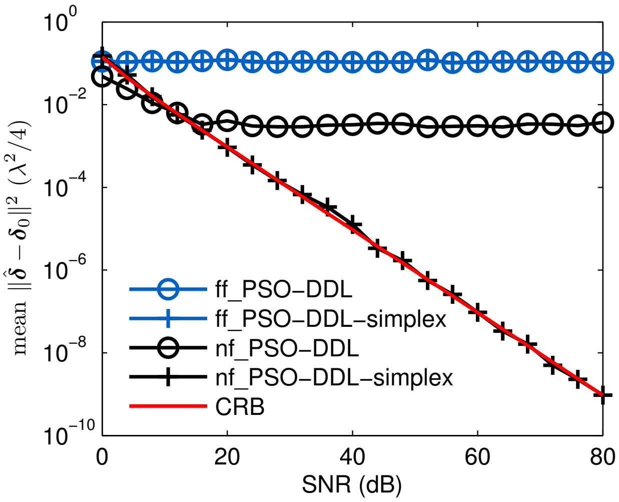

Figure 5.11: *Comparison of near-field and far-field calibration conducted by PSO-DDL and PSO-DDL-simplex, and the CRB. In the legend, ff stands for far-field and nf for near-field.*

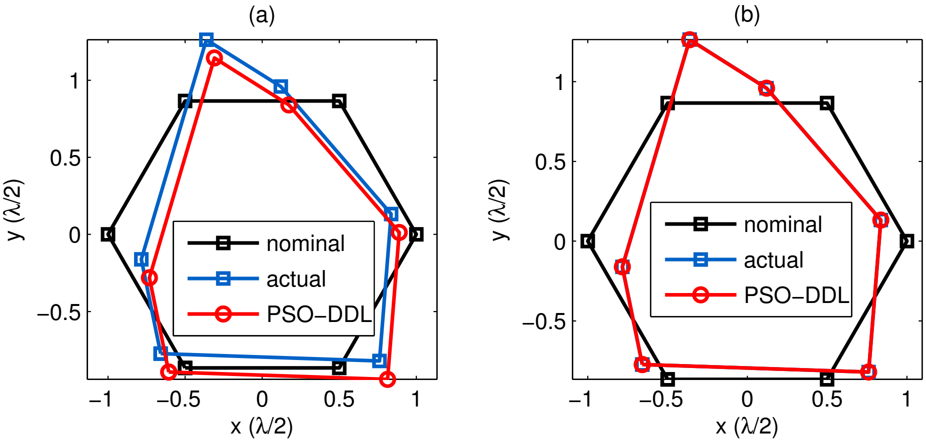

Figure 5.12: *Array shape calibrated by 2 far-field sources. (a) Before translational error removal. (b) After translational error removal.*

the estimated array coincides with the actual one, as shown in Fig. 5.12(b).

Enlightened by the above observation the removal of the translational error is applied to the entire simulation of the far-field case. This is to show that the translational error is the only residual. There are two ways of removing this error. The first is imposing a restriction on the optimization algorithm, that is, assuming the location of one sensor is known and commanding PSO to only optimize the other sensor locations. The second is optimizing all the sensor locations first, and afterwards using the assumed known sensor location to calculate the translational error and remove it from all the sensors, as done in Fig. 5.12(b). The second approach is adopted since the existing all-sensor calibration results can be readily used.

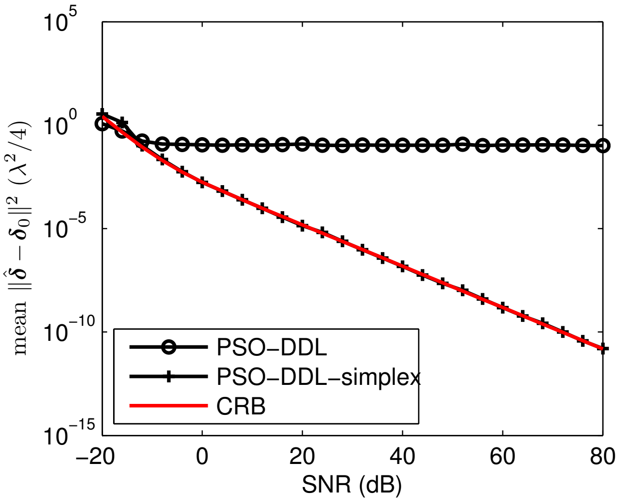

Figure 5.13: *MSE sum of 2 far-field source calibration after removal of translational error.*

Without loss of generality the first sensor in the lower left in Fig. 5.12(a) is chosen as the anchor. Its displacement is calculated from the actual sensor location, then the displacement is subtracted from every sensor. The MSE is then calculated from the aligned shape estimates. Fig. 5.13 illustrates its comparison with the CRB. It can be seen that after the removal of the translational error the MSE of 2-source far-field calibration approaches the CRB. Note that the CRB is for 5-sensor calibration since one sensor position is assumed known.

5.4.4.2 Indications from CRB

It has been shown by simulations in section 5.4.4.1 that 2 near-field sources are able to calibrate an array but 2 far-field are not. This is because the far-field calibration produces a translational error that can only be removed by providing an anchor sensor's accurate position. This geometrical problem is also manifested by the comparison of the CRB's for far-field and near-field calibration.

In particular, the CRB for 2 far-field source calibration does not exist for any of the DOA's evaluated, for the Fisher information matrix Eq. (B.1) is singular as indicated by numerical evaluations. And as shown in Fig. 5.13, after providing a sensor position the CRB for the far-field case approaches zero as SNR increases, which means that the removal of the translational error renders the array calibratable. In contrast, Fig. 5.14 shows that 2 near-field sources calibrate an array, despite the fact that the specific CRB varies with the source position. Fig. 5.14(a) shows the CRB over a range of SNR at 4 DOA pairs. Although there are differences among the CRB's of different DOA's, all the CRB's approach zero for high SNR. Fig. 5.14(b) shows that at a fixed SNR the CRB is a function of source bearing with multiple local minima.

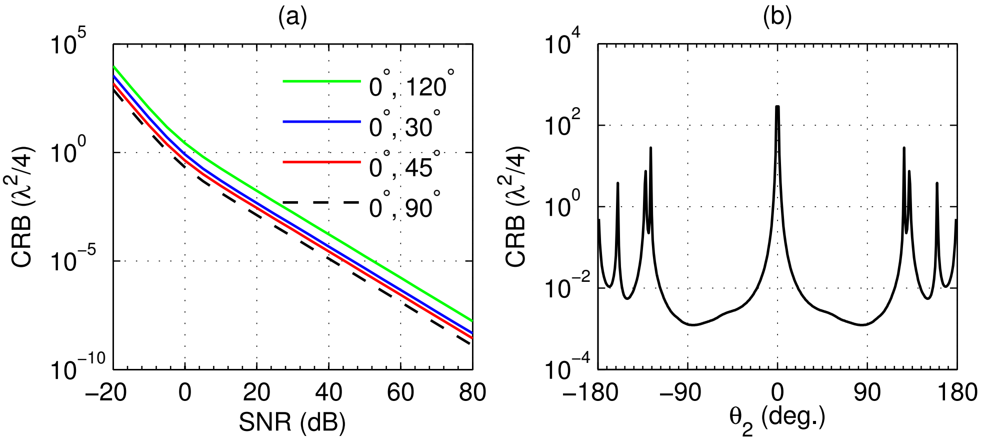

Figure 5.14: *CRB as a function of source position. (a) CRB over a range of SNR for 4 pairs of θ_1, θ_2. (b) CRB as a function of θ_2 at SNR $= 20$ dB, $\theta_1 = 0°$.*

5.4.5 DE results

In support of the remark that DDL is a general technique that can be coupled with other algorithms to improve performance, DE [92] has also been employed to optimize the F for the near-field case. However the comparison of PSO-DDL and DE-DDL is beyond the scope of this study.

Figs. 5.15 and 5.16 show the simulation results of the same DE with and without DDL. The

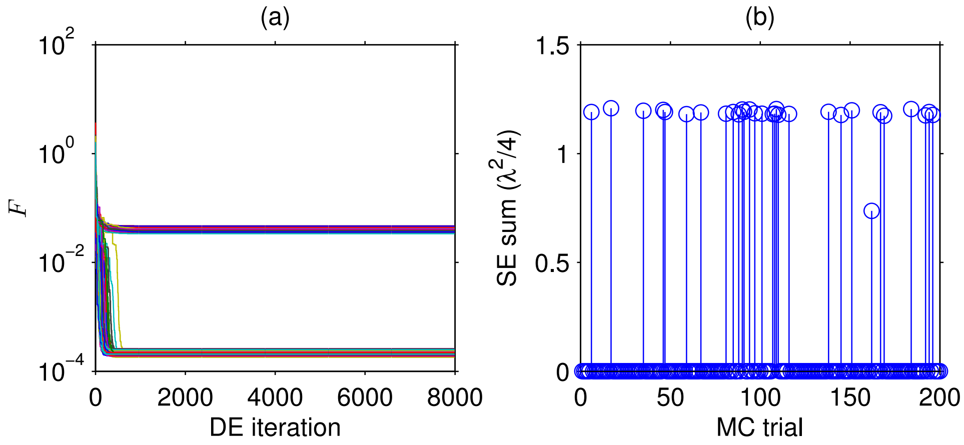

Figure 5.15: *DE performance at SNR* $= 22$ *dB. Near-field calibration.*

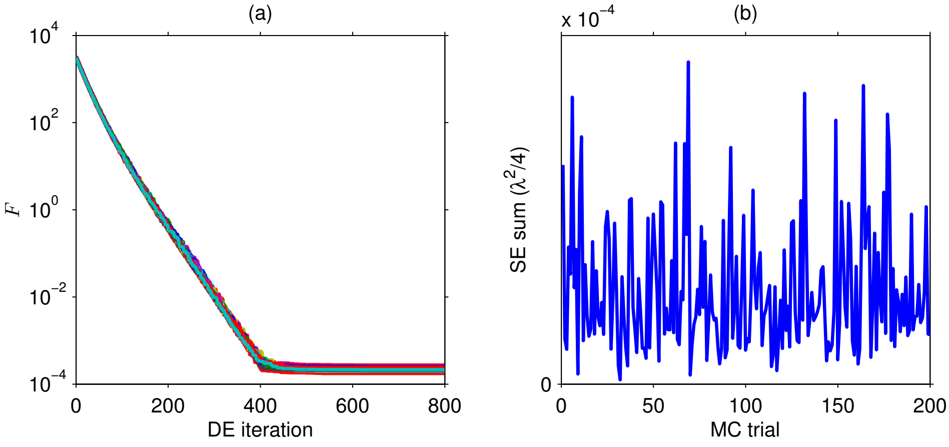

Figure 5.16: *DE-DDL performance at SNR* $= 22$ *dB. Near-field calibration.*

array-source settings are exactly the same as those of section 5.4.1, or in other words F is the same. The SNR shown is 26 dB. Fig. 5.15(a) shows that a significant proportion of the 200 MC trials performed by DE alone are trapped in local minima, ending up with optimized F values grouped higher than the global minimum bundle, and correspondingly Fig. 5.15(b) shows a large proportion of spiky false DOA estimates. In contrast, Fig. 5.16(a) shows that DE-DDL optimizes F all to the global minimum, the slope between the 1st and the 400th DE iterations reflecting the effect of DDL, and correspondingly Fig. 5.16(b) shows that all the DOA estimates are close to the true values. Note that this is achieved in one tenth of the number of iterations of DE alone.

5.5 Conclusion

This chapter is primarily concerned with the application of the PSO-DDL technique for optimization of the UML objective function to the near-field off-line calibration, which is more complicated than the corresponding far-field case due to the range effect. The simulation results show that the method is applicable to the entire near field region, free of the initialization problem and capable of achieving the optimal accuracy that approaches the CRB. Equipped with the UML-PSO-DDL method, it is found by simulation that two near-field sources are capable of calibrating an array, relaxing the requirement of the sufficient condition in [50].

As for the optimization technique PSO-DDL, the 20-sensor calibrations show PSO's ability to handle a large number of parameters. The simulation of simplex succeeding PSO-DDL shows that the problem of slow convergence of PSO can be resolved by the succession of efficient local optimization algorithms, at the expense of implementation complexity. The application of DDL to the near-field case supports the remark that the DL effect is independent of the specific array manifold; the simulation results of DE coupled with DDL support the remark that the DL effect is independent of the specific optimization algorithm.

Chapter 6
Conclusions and suggestions for future work

6.1 Conclusions

In this thesis, calibration methods have been proposed to combat the performance degradation of array processing incurred by array shape errors. Calibration methods can be classified as online and offline methods, dependent on whether calibration sources are at known or unknown positions. The field region of an array can be divided into far field and near field, based on the distance from it. Two physical cases are considered, the online calibration (self-calibration) using far-field sources and the offline calibration using near-field sources. The calibration employs parametric methods. Specifically, the maximum likelihood (ML) estimators have been employed to estimate the errors, for their superior performance over subspace estimators. However, the well-known computational complexities in optimizing the objective functions of the ML estimators demands effective and efficient optimization algorithms. Rather than the Taylor expansions that induce approximation errors, the space-alternating generalized expectation-maximization (SAGE) algorithm and particle swarm optimization (PSO) algorithm have been recruited to optimize objective functions of the conditional maximum likelihood (CML) and unconditional maximum likelihood (UML) estimators, respectively.

In chapter 3, the far-field online calibration is carried out using CML estimator with a SAGE-based optimization. Through data augmentation, joint direction of arrival (DOA) estimation and array calibration can be carried out by a computationally simple search procedure. Numerical experiments show that the proposed method outperforms the existing method for closely located signal sources and is robust to large sensor errors. Moreover, the accuracy of the proposed procedure approaches the Cramér-Rao bound (CRB). Therefore, the SAGE-based procedure provides an attractive alternative to current self-calibration techniques.

In chapter 4, the same calibration as in chapter 3 is conducted using UML estimator optimized by PSO. A new technique, decaying diagonal loading (DDL) is developed to enhance the performance of PSO at high signal-to-noise ratio (SNR) by dynamically lowering it. Compared with the existing WF1 and WF2 algorithms, the proposed method is more robust to large shape perturbation and has lower calibration error due to its direct optimization on the UML objective function without approximation. Its accuracy approaches the CRB at an SNR higher than the threshold, and the threshold is lower than that of WF1 and WF2. The CRB-approaching accuracy is retained even at large perturbations. Furthermore, PSO does not need initialization close to the true parameters. The DDL technique is of general value in its transformation of the objective function and thus can be coupled with other global optimization algorithms to improve performance. This is supported by simulation results of the differential evolution (DE) algorithm. The PSO-DDL method is simple to implement and computationally feasible.

In chapter 5, the near-field offline calibration is performed by the UML estimator optimized by PSO-DDL. The method is free of the initialization problem, applicable to the entire near-field region, and the calibration accuracy approaches the CRB. Numerical simulations show that the method can calibrate an array of 20 sensors perturbed by large shape errors with only 3 sources. Furthermore, the development process of the DDL technique introduces the “prominence” concept of the global optimum. This is different from the precision of it, which improves as the SNR increases. The counter-intuitive behavior of the UML objective function that its global optimum stands more prominent at lower SNR is harnessed by the DDL to enhance the performance of the stochastic global optimization algorithms.

6.2 Suggestions for future work

- For the SAGE-based algorithm, its convergence acceleration reduces the computational load. In addition, it should be also applicable to the UML estimator for the calibration problem.

- For the DDL technique, a complete analysis would elucidate the diagonal loading (DL) effect on the shape of the UML objective function. It would also be interesting to investigate the DL effect on other objective functions, such as the CML and sub-space objective functions. Thirdly, what is the DL effect on the location of the global optimum in the finite-sample case? Fourthly, is there an optimal DDL schedule? If so, how to design it?

- It has been shown that when the global optimizer converges very slowly in the later stages, it is beneficial to succeed it with an efficient local optimization algorithm. A SAGE-based algorithm is a possible choice to succeed PSO-DDL for computation reduction.

- Although the calibration methods are developed in the context of array shape error in this thesis, they are equally applicable to other types of array response errors such as sensor gain and phase errors, and mutual coupling.

Appendix A
Conditional expectation for joint Gaussian distribution

For $\boldsymbol{X} \sim \mathcal{G}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$, partition $\boldsymbol{X}$ into

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{pmatrix},$$

and correspondingly

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \tag{A.1}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \tag{A.2}$$

the conditional distribution of $\boldsymbol{X}_1$ given $\boldsymbol{X}_2 = \boldsymbol{x}_2$ is [77][78] Gaussian with mean $\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\boldsymbol{x}_2 - \boldsymbol{\mu}_2\right)$ and covariance matrix $\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$.

Appendix B
The CRB for near-field offline calibration

The CRB for near-field offline calibration is a degenerate form of (8.686) in [19]. The Fisher information matrix $\boldsymbol{J}(\boldsymbol{\delta})$, which is the inverse of the CRB, is given by

$$\boldsymbol{J}(\boldsymbol{\delta}) = \begin{bmatrix} \left(\boldsymbol{D_x}\boldsymbol{\Sigma}\boldsymbol{D_x^H}\right) \odot \left(\boldsymbol{P_H^\perp}\right)^T & \left(\boldsymbol{D_x}\boldsymbol{\Sigma}\boldsymbol{D_y^H}\right) \odot \left(\boldsymbol{P_H^\perp}\right)^T \\ \left(\boldsymbol{D_y}\boldsymbol{\Sigma}\boldsymbol{D_x^H}\right) \odot \left(\boldsymbol{P_H^\perp}\right)^T & \left(\boldsymbol{D_y}\boldsymbol{\Sigma}\boldsymbol{D_y^H}\right) \odot \left(\boldsymbol{P_H^\perp}\right)^T \end{bmatrix}. \tag{B.1}$$

The notation is the same as that of section 3.4.

For the near-field case,

$$\frac{\partial H_{nm}}{\partial \Delta x_n} = d_{nm}^{-1}\left(d_{nm}^{-1} + j\pi\right)\left[r_m \sin\theta_m - (x_n + \Delta x_n)\right] H_{nm},$$

and

$$\frac{\partial H_{nm}}{\partial \Delta y_n} = d_{nm}^{-1}\left(d_{nm}^{-1} + j\pi\right)\left[r_m \cos\theta_m - (y_n + \Delta y_n)\right] H_{nm}$$

in $\lambda/2$ unit.

Appendix C
Publications

C.1 Conference papers

1. P.-J. Chung and S. Wan, “Array self-calibration using sage algorithm,” in proc. *5th IEEE Sensor Array and Multichannel Signal Processing Workshop, SAM 2008*, Darmstadt, Germany, July 2008, pp. 165 – 169.

2. S. Wan, P.-J. Chung, and B. Mulgrew, “Array shape self-calibration using particle swarm optimization and decaying diagonal loading,” in Proc. *Sensor Signal Process. for Defence*, London, UK, 2010, in press.

3. S. Wan, P.-J. Chung, and B. Mulgrew, “Near-field array shape calibration,” *ICASSP 2011*, accepted.

C.2 Journal paper

S. Wan, P.-J. Chung, and B. Mulgrew, “Maximum likelihood array calibration using particle swarm optimization,” to be submitted to *IET Signal Processing*, 2011, invited paper.

ARRAY SELF-CALIBRATION USING SAGE ALGORITHM

Pei-Jung Chung and Shuang Wan

Institute for Digital Communications
School of Engineering and Electronics, The University of Edinburgh, UK
p.chung@ed.ac.uk, s.wan@ed.ac.uk

ABSTRACT

The performance of most existing array processing algorithms relies heavily on the precise knowledge of array manifold, which is decided by individual sensor characteristics and array configuration. A major challenge for self-calibration techniques is the increased computational burden due to additional perturbation parameters. In this contribution, a novel procedure for array self-calibration is presented. We apply the well known numerical method, the Space Alternating Generalized EM algorithm (SAGE), to simplify the multi-dimensional search procedure required for finding maximum likelihood (ML) estimates. Simulation shows that the proposed algorithm outperforms existing methods that are based on the small perturbation assumption. Furthermore, the proposed algorithm remain robust in critical scenarios including large sensor position errors and closely located signals.

1. INTRODUCTION

The problem of estimating direction of arrival (DOA) plays a key role in array processing and related applications such as radar, wireless communications, seismology and sonar. Most direction finding algorithms are very sensitive to model uncertainties caused by sensor position errors, phase and gain deviations, etc [5]. To mitigate this problem, various calibration methods have been proposed. They can generally be categorized into off-line and on-line calibration. The latter one is also known as self-calibration for it does not need the known directions of arrival of the calibrating sources as off-line calibration does. Self-calibration techniques are of particular importance when cooperative sources are not available.

In this contribution, we focus on sensor position perturbations and far-field signals. The Space Alternating Generalized Expectation-Maximization (SAGE) algorithm [4] is applied to jointly estimate DOA parameters and sensor position errors. The SAGE algorithm is a variate of the widely used Expectation and Maximization (EM) algorithm [2]. Because of its flexible augmentation scheme, SAGE converges faster than EM while retaining the advantages of numerical simplicity and stability. Rather than computing all parameters simultaneously as the standard EM algorithm does, the SAGE algorithm updates subsets of parameters sequentially in one iteration. It was proved in [1] that SAGE based direction finding algorithm achieves a faster convergence rate than EM under mild conditions.

Since the perturbation parameters and DOA parameters can not be estimated independently, the identifiability issue is fundamental to calibration problems. According to Rockah and Schultheiss [6], an array is calibratable if the Cramér-Rao bound approaches zero for large sample sizes. The conditions derived from this criterion in the far-field case are that one sensor position and its direction to another sensor should be known and there are at least three calibrating sources. These requirements are assumed to be fulfilled in this work.

This paper is outlined as follows. The data model is described in Section 2. The SAGE algorithm based self-calibration procedure is developed in Section 3. Numerical results are presented and discussed in Section 4. Section 5 concludes this work.

2. PROBLEM FORMULATION

Consider an array of N sensors receiving signals emitted by M far-field sources located at $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_M]^T$. The nominal position of the ith sensor is given by $\boldsymbol{p}_i = (x_i, y_i)$. The sensor position error $\Delta\boldsymbol{p}_i = (\Delta x_i, \Delta y_i)$ is small compared to array dimension and considered as constant within one observation interval.

For narrow band signals, the array output can be ex-

P.-J. Chung acknowledges support of her position from the Scottish Funding Council and their support of the Joint Research Institute with the Heriot-Watt University as a component part of the Edinburgh Research Partnership.

pressed as

$$\boldsymbol{X}(t) = \boldsymbol{H}(\boldsymbol{\theta}, \boldsymbol{\Delta})\boldsymbol{s}(t) + \boldsymbol{N}(t),\ t = 1, \ldots, T \qquad (1)$$

where the mth column of the transfer matrix

$$\boldsymbol{H}(\boldsymbol{\theta}, \boldsymbol{\Delta}) = [\boldsymbol{d}(\theta_1, \boldsymbol{\Delta}), \cdots, \boldsymbol{d}(\theta_m, \boldsymbol{\Delta}), \ldots, \boldsymbol{d}(\theta_M, \boldsymbol{\Delta})] \qquad (2)$$

$\boldsymbol{d}(\theta_m, \boldsymbol{\Delta})$ is the steering vector associated with the direction θ_m evaluated at perturbed sensor positions $\tilde{\boldsymbol{p}}_i = \boldsymbol{p}_i + \Delta\boldsymbol{p}_i$, $(i = 1, \cdots, N)$. For notational convenience, we summarize all sensor displacement parameters in $\boldsymbol{\Delta}$

$$\boldsymbol{\Delta} = [\Delta x_1, \Delta y_1, \ldots, \Delta x_i, \Delta y_i, \ldots, \Delta x_N, \Delta y_N]. \qquad (3)$$

Without loss of generality, the position of the first sensor is assumed to be known, i.e. $\Delta x_1 = 0, \Delta y_1 = 0$. By setting $\Delta y_2 = 0$, the direction of the first sensor to the second sensor is known.

The signal vector $\boldsymbol{s}(t)$ is considered as deterministic and unknown. The noise vector $\boldsymbol{N}(t)$ is independent, identically complex normally distributed with zero mean and covariance matrix $\nu\boldsymbol{I}$, where ν is an unknown noise spectral parameter and $\boldsymbol{I}$ is an $N \times N$ identity matrix. Given the observations $\boldsymbol{X} = \{\boldsymbol{X}(t)\}_{t=1}^{T}$, the problem of central interest is to estimate the DOA parameters $\boldsymbol{\theta}$ and perturbation parameters $\boldsymbol{\Delta}$.

3. SAGE ALGORITHM FOR ARRAY SELF-CALIBRATION

In the standard ML approach, the likelihood function can be concentrated with respect to signal and noise parameters. The resulting likelihood depends only on the M-dimensional DOA parameter vector $\boldsymbol{\theta}$. In the self-calibration problem, it is not possible to separate DOA and perturbation parameters. Direct application of likelihood principle would lead to a computationally expensive optimization procedure. In [7], the authors consider small perturbation, known noise level and simplify the data model by Taylor expansion so that the position errors can be obtained by least square estimates. In the following, we overcome the computational difficulty by applying the SAGE algorithm without making additional assumptions on perturbation and noise parameters.

It is well known that through proper choice of augmented data, the complicated multi-dimensional maximization associated with standard ML estimation can be simplified to one dimensional search by EM-like algorithms [3]. Because of a more flexible data augmentation scheme, the SAGE algorithm has a better convergence speed than the EM algorithm [1]. Each iteration of SAGE consists of several cycles. A subset $\boldsymbol{\vartheta}_c$ of all unknown parameters is updated at one cycle. All subsets are updated once in one iteration. In the following, we shall derive a SAGE based algorithm that updates DOA parameters and perturbation parameters sequentially.

Let $\boldsymbol{\vartheta}$ summarize all unknown parameters

$$\boldsymbol{\vartheta} = [\boldsymbol{\vartheta}_1, \ldots, \boldsymbol{\vartheta}_M, \boldsymbol{\vartheta}_{M+1}, \ldots, \boldsymbol{\vartheta}_{M+N}, \boldsymbol{\vartheta}_{M+N+1}] \qquad (4)$$

where $\boldsymbol{\vartheta}_c = [\theta_c, s_c(t), t = 1, \ldots, T]$, $(c = 1, \ldots, M)$ includes the DOA and signal parameters associated with the cth source. The sensor perturbation parameters are divided into N subsets with $\boldsymbol{\vartheta}_{M+i} = \Delta\boldsymbol{p}_i$, $(i = 1, \ldots, N)$. The last subset $\boldsymbol{\vartheta}_{M+N+1} = [\ \nu\ ]$ contains the noise parameter.

As mentioned previously, SAGE updates a subset of parameters at each cycle. It is natural to choose different augmented data $\boldsymbol{Z}_c$ for each cycle. For DOA estimation, we use a similar augmentation scheme as in [1]:

$$\boldsymbol{Z}_c(t) = \boldsymbol{d}(\theta_c, \boldsymbol{\Delta})s_c(t) + \boldsymbol{N}(t), \quad c = 1, \ldots, M. \qquad (5)$$

For sensor position and noise parameters, we take a straightforward approach by choosing

$$\boldsymbol{Z}_c(t) = \boldsymbol{X}(t), \quad c = M+1, \ldots, M+N+1. \qquad (6)$$

In other words, using the latest DOA estimate and the array observation, array calibration is carried out sensor by sensor. The required initial DOA estimate can be initialized by subspace methods such as MUSIC or the standard ML approach using nominal sensor positions.

Let $\boldsymbol{\vartheta}^{[i,c]}$ denote the estimate of the cth cycle and ith iteration and $\boldsymbol{\vartheta}^{[i,0]} = \boldsymbol{\vartheta}^{[i-1,M+N]}$. Given an initial estimate $\boldsymbol{\vartheta}^{[0,0]}$ and the augmentation scheme (5) and (6), the ith iteration of the SAGE algorithm proceeds as follows.

The ith iteration of the SAGE algorithm

I. Update DOA parameter

For $c = 1, 2, \ldots, M$

E-step : Calculate the conditional expectation of the augmented log-likelihood function.

$$Q\left(\boldsymbol{\vartheta}_c, \boldsymbol{\vartheta}^{[i,c-1]}\right) = E\left[\log f_{\boldsymbol{Z}_c}(\boldsymbol{\vartheta}_c)|\boldsymbol{X}, \boldsymbol{\vartheta}^{[i,c-1]}\right]. \qquad (7)$$

M-step : Update $\boldsymbol{\vartheta}_c = [\theta_c, s_c(t), t = 1, \ldots, T]$.

$$\theta_c^{[i,c]} = \arg\max_{\theta_c} \boldsymbol{d}_c(\theta_c, \boldsymbol{\Delta}^{[i-1]})^H \widehat{\boldsymbol{R}}_{\boldsymbol{Z}_c} \boldsymbol{d}_c(\theta_c, \boldsymbol{\Delta}^{[i-1]}) \qquad (8)$$

where $\boldsymbol{\Delta}^{[i-1]}$ represent the perturbation parameters from the previous iteration. The signal parameters $s_c(t), t = 1, \dots, T$ are updated in a similar manner as in [1].

II. Update sensor position perturbation parameters

For $c = M + 1, \dots, M + N$

E-step :

$$Q\left(\boldsymbol{\vartheta}_c, \boldsymbol{\vartheta}^{[i,c-1]}\right) = E\left[\log f_{\boldsymbol{X}}(\boldsymbol{\vartheta}_c)|\boldsymbol{X}, \boldsymbol{\vartheta}^{[i,c-1]}\right]. \quad (9)$$

M-step : Update $\boldsymbol{\vartheta}_c = \Delta \boldsymbol{p}_{c-M}$.

$$\Delta \boldsymbol{p}_{c-M}^{[i,c]} = \arg \max_{\Delta \boldsymbol{p}_{c-M}} \operatorname{tr}\left[\boldsymbol{P}(\boldsymbol{\theta}^{[i,M]}, \boldsymbol{\Delta}^{[i,c-1]})\widehat{\boldsymbol{R}}_{\boldsymbol{X}}\right] \quad (10)$$

where $\boldsymbol{P}(\boldsymbol{\theta}^{[i,M]}, \boldsymbol{\Delta}^{[i,c-1]})$ is the projection matrix onto the column space of $\boldsymbol{H}(\boldsymbol{\theta}^{[i,M]}, \boldsymbol{\Delta}^{[i,c-1]})$.

The perturbation vector

$$\boldsymbol{\Delta}^{[i,c-1]} = [\Delta \boldsymbol{p}_1^{[i,c-1]}, \dots, \Delta \boldsymbol{p}_{c-M}, \dots, \Delta \boldsymbol{p}_N^{[i,c-1]}] \quad (11)$$

consists of the error vector $\Delta \boldsymbol{p}_{c-M}$ and other $(N-1)$ vectors fixed at the latest estimates. The sample covariance matrix is given by $\widehat{\boldsymbol{R}}_{\boldsymbol{X}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{X}(t)\boldsymbol{X}(t)^H$. Note that (10) is simply the concentrated log-likelihood function for array observations $\boldsymbol{X}(t)$ with the DOA parameter fixed at current estimate $\boldsymbol{\theta}^{[i,M]}$. If the first sensor position is assumed to be known, i.e. $\Delta \boldsymbol{p}_1 = \boldsymbol{0}$, the corresponding cycle can be skipped. The latest estimate is simply replaced by the known value $\boldsymbol{0}$. For the known direction condition, $\Delta y_2 = 0$, the perturbation parameter Δx_1 requires only one dimensional search.

III. Update noise parameter

For $c = M + N + 1$, the E-step is similar to (9). In the M-step, the noise parameter is updated by a closed form expression as in standard ML estimation [1].

In summary, the SAGE algorithm decouples the DOA and perturbation parameters and updates them alternatively. The multi-dimensional search procedure is greatly simplified to one and two dimensional search procedures.

4. SIMULATION

In the simulation, we consider a nominally uniform circular array of 6 sensors with inter-element spacings of half a wavelength $\lambda/2$, see Fig 1. Their nominal positions are perturbed by a random displacement with zero mean and a standard deviation σ_p . The random displacement is generated once and kept fixed throughout each experiment. For comparison, we also apply the standard SAGE algorithm (ML-SAGE) [1] and the self-calibration method proposed by Weiss and Friedlander (WF) [7] to the same batch of data.

4.1. Well separated sources

In the first experiment, the narrowband signals are generated by three widely separated sources of equal power located at $\boldsymbol{\theta}_{\text{true}} = [-35^\circ\ 0^\circ\ 35^\circ]$. The arriving angles are measured from broadside of the array. The deviation of sensor position errors is $\sigma_p = 0.1\lambda$ which corresponds to 20% inter-sensor spacing. The number of snapshots $T = 50$. The Signal to Noise Ratio (SNR), defined as $10\log[|s_m(t)|^2/\nu]$, ranges from 8 dB to 18 dB with a 2dB step. For each SNR, 100 Monte Carlo trials are performed.

The nominal sensor positions are used to initialize the algorithm, i.e. $\boldsymbol{\Delta}^{[0]} = \boldsymbol{0}$. For a fair comparison, we choose a fixed initial DOA estimate $\boldsymbol{\theta}^{[0]} = [34^\circ\ 4.1^\circ\ 41.6^\circ]$. The maximum number of iteration is 200. The algorithm is terminated if the relative change of parameters is less than 0.1% between two consecutive iterations or the maximum number of iterations is reached.

Fig. 2 shows the Mean Squared Errors (MSE) of DOA estimation obtained from averaging the distance between the estimated and the true DOA parameters, $||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_{\text{true}}||$. One can easily observe that the standard ML-SAGE algorithm fails to provide any useful information. Both self-calibration techniques lead to significant improvement in estimation accuracy. The SAGE algorithm and WF method perform similarly over the entire SNR range.

4.2. Closely located sources

In the second experiment, we consider closely located source with $\boldsymbol{\theta}_{\text{true}} = [-35^\circ\ 12^\circ\ 30^\circ]$. The initial value is given by $\boldsymbol{\theta}^{[0]} = [-29.8^\circ\ 16.1^\circ\ 33.6^\circ]$. Other parameters are the kept the same.

The MSEs presented in Fig. 3 show that the SAGE algorithm has an overall higher estimation accuracy than the WF method in this more critical situation. However, both methods require a higher SNR than in the first experiment to provide reasonable estimates. Similarly to Fig 2, the standard ML-SAGE approach fails totally when sensor positions deviate from the nominal values.

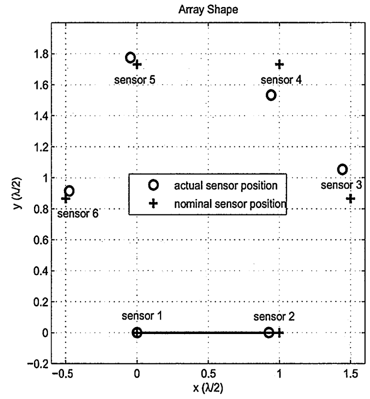

Fig. 1. Array geometry. Uniform circular array of 6 sensors with inter-element spacing $\lambda/2$.

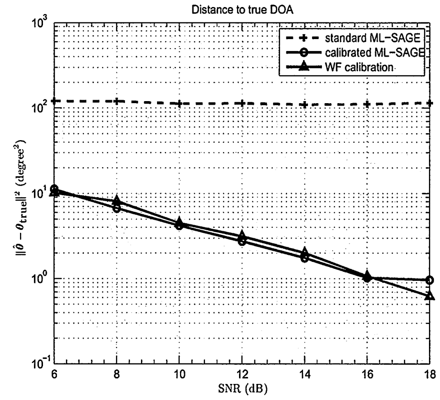

Fig. 2. Average $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_{\text{true}}\|^2$ vs SNR. True DOA parameters $\boldsymbol{\theta}_{\text{true}} = [-35° \ 0° \ 35°]$, $\sigma_p = 0.1\ \lambda$.

4.3. Large sensor position errors

In the third experiment, we consider widely separated signals as in the first experiment, i.e. $\boldsymbol{\theta}_{\text{true}} = [-35° \ 0° \ 35°]$. The standard deviation of sensor position error is increased to $\sigma_p = 0.25\lambda$ which corresponds to 50% of inter-sensor spacing. The monte Carlo trials are performed at SNR = 6, 12, 18 dB.

Fig 4 shows that over the entire SNR range, both the standard ML-SAGE and the WF algorithms fail to estimate DOA parameters, whereas the SAGE based self-calibration method still provides reasonable results. This shows that the proposed algorithm has a much higher robustness against large sensor position errors than the WF method.

5. CONCLUSION

We derived a SAGE algorithm based self-calibration procedure for arrays with sensor position uncertainties. Through data augmentation, DOA estimation and array calibration can be carried out by a computationally simple search procedure. Numerical experiments showed that the proposed method outperforms the existing method for closely located signal sources and is robust to large sensor errors. Therefore, the SAGE-based procedure provides an attractive alternative to current self-calibration techniques.

6. REFERENCES

[1] P.J. Chung and J.F. Bohme. Comparative convergence analysis of EM and SAGE algorithms in DOA estimation. *IEEE Transactions on Signal Processing*, 49(12):2940–2949, 2001.

[2] A. P. Dempster, N. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B39:1–38, 1977.

[3] M. Feder and E. Weinstein. Parameter estimation of superimposed signals using the EM algorithm. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 36(4):477–489, April 1988.

[4] J. A. Fessler and A. O. Hero. Space–alternating generalized expectation–maximization algorithm. *IEEE Trans. Signal Processing*, 42(10):2664–2677, October 1994.

[5] H. Krim and M. Viberg. Two decades of array signal processing research: the parametricapproach. *IEEE Signal Processing Magazine*, 13(4):67–94, July 1996.

[6] Y. Rockah and P. M. Schultheiss. Array shape calibration using sources in unknown locations- Part I: Farfield sources. *IEEE Trans. ASSP*, 35:286–299, 1987.

[7] B. Weiss, J. A.and Friedlander. Array shape calibration using sources in unknown locations - a maximum likelihood approach. *IEEE Trans. Acoust., Speech, Signal Processing*, 37(12), December 1989.

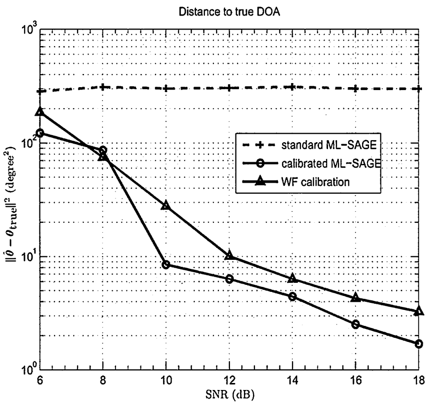

Fig. 3. Average $||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_{\text{true}}||^2$ vs SNR. True DOA parameters $\boldsymbol{\theta}_{\text{true}} = [-35° \ 12° \ 30°]$, $\sigma_p = 0.1 \ \lambda$.

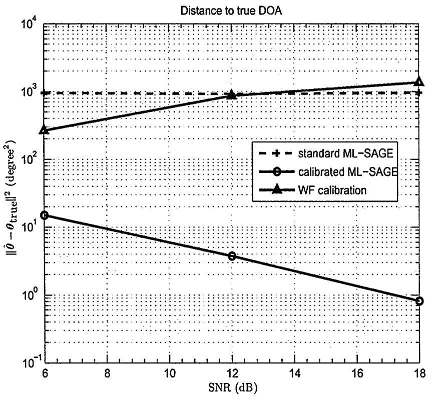

Fig. 4. Average $||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_{\text{true}}||^2$ vs SNR. True DOA parameters $\boldsymbol{\theta}_{\text{true}} = [-35° \ 0° \ 35°]$, $\sigma_p = 0.25 \ \lambda$.

Array shape self-calibration using particle swarm optimization and decaying diagonal loading

Shuang Wan, Pei-Jung Chung, Bernard Mulgrew

Institute for Digital Communications, School of Engineering, The University of Edinburgh
Edinburgh, EH9 3JL, UK
{S.Wan, P.Chung, B.Mulgrew}@ed.ac.uk

***Abstract*—Calibration of array shape error is a key issue for most existing direction finding algorithms. In this paper, we propose a new procedure that employs particle swarm optimization (PSO) and decaying diagonal loading (DDL) techniques to optimize the unconditional maximum likelihood (UML) function for array self-calibration. The proposed method is able to self-calibrate large array shape error. Moreover, it outperforms previous ones in terms of mean squared error that attains Cramér-Rao bound. An example of 5-sensor and 3-source geometry is used to demonstrate the efficacy of the proposed method.**

I. Introduction

Errors in sensor location can severely degrade the direction finding performance of a sensor array. A small perturbation in sensor location can often dominate all other causes of direction of arrival (DOA) estimation error [1]. Therefore it is crucial to calibrate the array shape. Calibration methods are categorized into off-line and on-line methods, dependant on whether calibration sources are at known or unknown positions. In this work, we concentrate on on-line calibration, namely the self-calibration problem that uses sources at unknown positions.

Rockah proposed a self-calibration method that requires sources to be separable in the time or the frequency domain [2]. Weiss and Friedlander used objective functions based on conditional maximum likelihood (CML) [3] (termed WF1) and MUSIC [4] (termed WF2) for the self-calibration problem. In there, a first-order Taylor approximation is applied to the sensor location parameters to achieve an analytical optimization which reduces the computational complexity. However, the Taylor approximation requires that the perturbation is small, and exacerbates the mean squared error (MSE) of the result. Flanagan and Bell [5] relieved the small perturbation constraint by preceding WF2 with a "coarse calibration" procedure. Chung and Wan [6] used EM algorithm to optimize the CML objective function. This led to improved calibration accuracy under conditions of large perturbation and closely-located sources, but lacks an initialization mechanism.

Unlike previous deterministic optimization methods, we propose the stochastic particle swarm optimization - decaying diagonal loading (PSO-DDL) procedure to optimize the unconditional maximum likelihood (UML) objective function for the self-calibration problem. PSO-DDL does not need accurate initialization and reaches the global optimum of the UML objective function if the control parameters are properly set. The entire paradigm offers an MSE that attains the Cramér-Rao bound (CRB) under both small and large perturbations of sensor location at a reasonable computational cost.

The rest of this paper is organized as follows. The problem is formulated in section II, the PSO-DDL paradigm is laid out in section III, followed by simulation results in section IV. Section V summarizes and concludes the paper.

II. Problem Formulation

Consider an array of N sensors receiving signals emitted by M far-field sources positioned at $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_M]^T$. The known nominal location of the nth sensor is given by $[x_n, y_n]^T$, which is perturbed by unknown error $[\Delta x_n, \Delta y_n]^T$ to the actual location $[x_n + \Delta x_n, y_n + \Delta y_n]^T$. The perturbations are summarized in vector $\boldsymbol{\delta} = [\Delta x_1, \Delta y_1, \ldots, \Delta x_N, \Delta y_N]^T$, which is considered constant in a calibration process.

For narrow band signals, the K snapshots of array output can be expressed in Fourier domain as

$$\boldsymbol{y}(k) = \boldsymbol{H}(\boldsymbol{\theta}, \boldsymbol{\delta})\, \boldsymbol{s}(k) + \boldsymbol{n}(k), \; k = 1, \ldots, K, \tag{1}$$

where the n, mth element of the array manifold $\boldsymbol{H}(\boldsymbol{\theta}, \boldsymbol{\delta})$ is

$$\begin{aligned} &H_{nm}(\theta_m, \Delta x_n, \Delta y_n) = \\ &\exp\left\{ j\frac{2\pi}{\lambda}\left[(x_n + \Delta x_n)\sin\theta_m + (y_n + \Delta y_n)\cos\theta_m \right] \right\}, \\ &n = 1, \ldots, N, \; m = 1, \ldots, M, \end{aligned}$$

in which λ is the signal wavelength.

The signal vector $\boldsymbol{s}(k)$ is considered to be stochastic. The noise vector $\boldsymbol{n}(k)$ is independent, identically complex normally distributed with zero mean and covariance matrix $\nu \boldsymbol{I}$, where ν is an unknown noise spectral parameter and $\boldsymbol{I}$ is an $N \times N$ identity matrix. Given the observations $\{\boldsymbol{y}(k)\}_1^K$, the problem of central interest is to estimate the DOA parameters $\boldsymbol{\theta}$ and perturbation parameters $\boldsymbol{\delta}$.

Following [7], the UML estimator for $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ can be expressed as

$$\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\delta}}\right) = \underset{(\boldsymbol{\theta},\boldsymbol{\delta})}{\arg\min} \underbrace{\left\{ \det\left[\boldsymbol{P_H}\hat{\boldsymbol{R}}\boldsymbol{P_H} + \frac{\operatorname{tr}(\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\hat{\boldsymbol{R}})\boldsymbol{P}_{\boldsymbol{H}}^{\perp}}{N - M} \right] \right\}}_{F(\boldsymbol{\theta},\boldsymbol{\delta})}, \tag{2}$$

where $\hat{\boldsymbol{R}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}(k)\boldsymbol{y}^H(k)$ is the sample covariance matrix, $\boldsymbol{P_H} = \boldsymbol{H}(\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H$ is the projection matrix of

$\boldsymbol{H}$, and $\boldsymbol{P}_{\boldsymbol{H}}^{\perp} = \boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{H}}$ is its orthogonal complementary. The dependence of $\boldsymbol{H}$ on $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ is suppressed for notational simplicity, and $F(\boldsymbol{\theta}, \boldsymbol{\delta})$ denotes the objective function.

III. The PSO-DDL technique

A. PSO for Array Shape Self-calibration

We briefly introduce a basic PSO by casting it into the self-calibration problem. Interested readers are referred to [8] for more information. A swarm particle (a point $\boldsymbol{x}_i \in \mathbb{R}^D$) "flies" in the D-dimensional solution space whose velocity $\boldsymbol{v}_i(t) \in \mathbb{R}^D$ at the tth time step is updated by

$$\begin{aligned} \boldsymbol{v}_i(t+1) = \chi\{&\boldsymbol{v}_i(t) + \varphi_1 \boldsymbol{r}_1 \odot [\boldsymbol{p}_i(t) - \boldsymbol{x}_i(t)] + \\ &\varphi_2 \boldsymbol{r}_2 \odot [\boldsymbol{g}_i(t) - \boldsymbol{x}_i(t)]\}, \; i = 1, \ldots, P, \end{aligned} \tag{3}$$

where $\odot$ denotes the element-wise product operator. $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are random vectors uniformly distributed in $[0,1]$, $\boldsymbol{p}_i$ the particle best location of its own experience from time step 1 to t, and $\boldsymbol{g}_i$ the group best location at current time step t. $\chi = 2/|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|$ is the constriction factor, where $\varphi = \varphi_1 + \varphi_2$. φ_1 and φ_2 are acceleration constants. P is the number of particles. $\boldsymbol{v}_i$ is clamped if it exceeds the maximum velocity V in any dimension:

$$v_{id} = V \frac{v_{id}}{|v_{id}|}, \; d = 1, \ldots, D, \tag{4}$$

where v_{id} is the dth component of $\boldsymbol{v}_i$.

For unit time step, the ith particle position is updated as

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1), \; i = 1, \ldots, P. \tag{5}$$

If $\boldsymbol{x}_i$ is beyond the solution space in any dimension, it is reset to a random position in that dimension. In the case that $\boldsymbol{x}_i$'s components have different ranges, one may map them all into $[0,1]$ by, for instance, a linear mapping.

Two strategies specific to the array self-calibration problem are stated below.

1) If at any PSO iteration, two DOA's happened to be closer than a pre-specified value ε, one of them is assigned a random value [9]. This is to prevent $\boldsymbol{H}$ from rank-deficiency.

2) The particle group [8], within which the group best location $\boldsymbol{g}_i$ is picked up, is defined to formulate a string topology for its high search ability indicated by simulation:

$$\text{Particles of the } i\text{th group} \begin{cases} i, i+1 & i = 1 \\ i-1, i, i+1 & i = 2, \ldots, P-1 \\ i-1, i & i = P. \end{cases}$$

B. Diagonal Loading

It was observed that at high signal-to-noise ratio (SNR), PSO alone often failed to reach the global minimum of F, the reason being that the global minimum of F is more prominent at lower SNR than at higher SNR. In the following we first demonstrate this phenomenon by visualizing F and then further explain it using the first derivative of F.

It is a simple matter to establish that lowering the SNR by adding spatially white noise is asymptotically equivalent to diagonally loading the covariance matrix: $\hat{\boldsymbol{R}} = \boldsymbol{R} + \nu_l \boldsymbol{I}$, where $\boldsymbol{R} = \lim_{K\to\infty} \hat{\boldsymbol{R}}$ and ν_l is the power of the loaded noise. Assuming normalized signal power, the original signal-to-noise ratio can be expressed as $\text{SNR} = -10\log\nu$. We then define the effective signal-to-noise ratio after diagonal loading (DL) as

$$\text{SNR}_a = -10\log(\nu + \nu_l), \tag{6}$$

and consequently the loaded signal-to-noise ratio as $\text{SNR}_l = \text{SNR}_a - \text{SNR}$.

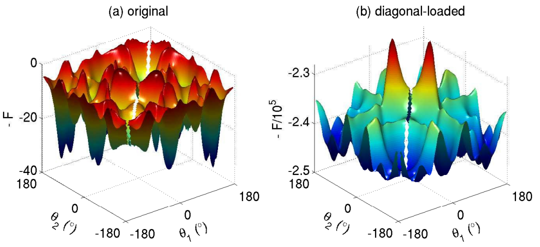

Fig. 1. The *negative* objective function before (a) and after (b) diagonal loading. 5-sensor uniform circular array, inter-sensor spacing $\frac{\lambda}{2}$; 2 signals at $[-35\ 35]^\circ$, $K = 1000$; SNR= 14 dB, $\text{SNR}_a = -10$ dB.

Since it is impossible to visualize the high-dimensional objective function of the self-calibration problem, we have to demonstrate the effect of DL using a corresponding two-dimensional DOA estimation problem, which is visualized in Fig. 1. The objective function is F degenerated by setting $\boldsymbol{\delta} = \boldsymbol{0}$. Comparing Fig. 1(a) and Fig. 1(b) we see that diagonal loading of the sample covariance matrix $\hat{\boldsymbol{R}}$, which is asymptotically equivalent to reducing the SNR, makes prominent the global minimum of the objective function. This eases searching it for a global optimizer. This phenomenon is observed independent of the number of sensors for the DOA estimation problem and is confirmed by simulations for the self-calibration problem in section IV.

We then analytically explain the DL effect on F. Jacobi's formula [10] (Part Three, Section 8.3) states that

$$(\det \boldsymbol{X})' = (\det \boldsymbol{X}) \cdot \operatorname{tr}\left(\boldsymbol{X}^{-1}\boldsymbol{X}'\right), \tag{7}$$

where $\boldsymbol{X}$ is an invertable matrix.

Denote

$$\boldsymbol{X} = \boldsymbol{P}_{\boldsymbol{H}} \hat{\boldsymbol{R}} \boldsymbol{P}_{\boldsymbol{H}} + \frac{\operatorname{tr}(\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\hat{\boldsymbol{R}})\boldsymbol{P}_{\boldsymbol{H}}^{\perp}}{N - M} \tag{8}$$

and use (B.3) and (B.15) in [11], it follows that

$$\begin{aligned} \operatorname{tr}\left(\boldsymbol{X}^{-1}\boldsymbol{X}'\right) = 2\operatorname{Re}\Bigg(\operatorname{tr}\Bigg\{ &\left[\left(\boldsymbol{H}^H \hat{\boldsymbol{R}} \boldsymbol{H}\right)^{-1} - \frac{1}{\hat{\nu}}\left(\boldsymbol{H}^H \boldsymbol{H}\right)^{-1}\right] \\ &\boldsymbol{H}^H \hat{\boldsymbol{R}} \boldsymbol{P}_{\boldsymbol{H}}^{\perp} \boldsymbol{H}' \Bigg\}\Bigg), \end{aligned} \tag{9}$$

where

$$\hat{\nu} = \frac{\operatorname{tr}(\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\hat{\boldsymbol{R}})}{N - M}. \tag{10}$$

Inserting (8) and (9) into (7) we obtain the first derivative of F as

$$F' = 2\det\left[\boldsymbol{P_H}\hat{\boldsymbol{R}}\boldsymbol{P_H} + \hat{\nu}\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right]\mathrm{Re}\left(\mathrm{tr}\left\{\left[\left(\boldsymbol{H}^H\hat{\boldsymbol{R}}\boldsymbol{H}\right)^{-1} - \frac{1}{\hat{\nu}}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\right]\underline{\boldsymbol{H}^H\hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{H}}^{\perp}}\boldsymbol{H}'\right\}\right). \tag{11}$$

Replace $\hat{\boldsymbol{R}}$ with $\hat{\boldsymbol{R}} + \nu_l\boldsymbol{I}$ in (10) and (11), note that $\mathrm{tr}\left(\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right) = N-M$ and $\boldsymbol{H}^H\boldsymbol{P}_{\boldsymbol{H}}^{\perp} = \boldsymbol{0}$, then the first derivative after diagonal loading, F_l', is given by

$$F_l' = 2\det\left[\boldsymbol{P_H}\left(\hat{\boldsymbol{R}}+\nu_l\boldsymbol{I}\right)\boldsymbol{P_H}+(\hat{\nu}+\nu_l)\boldsymbol{P}_{\boldsymbol{H}}^{\perp}\right]\mathrm{Re}\left(\mathrm{tr}\left\{\left[\left(\boldsymbol{H}^H(\hat{\boldsymbol{R}}+\nu_l\boldsymbol{I})\boldsymbol{H}\right)^{-1} - \frac{1}{\hat{\nu}+\nu_l}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\right]\underline{\boldsymbol{H}^H\hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{H}}^{\perp}}\boldsymbol{H}'\right\}\right). \tag{12}$$

Two asymptotic ($K \to \infty$) properties of DL can be derived from (11) and (12):

1) *DL does not shift the global minimum of F.*

Proof: At the true parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ and $\boldsymbol{\delta} = \boldsymbol{\delta}_0$,

$$\boldsymbol{R} = \boldsymbol{H}_0\boldsymbol{S}\boldsymbol{H}_0^H + \nu\boldsymbol{I},$$

where $\boldsymbol{H}_0$ stands for $\boldsymbol{H}(\boldsymbol{\theta}_0, \boldsymbol{\delta}_0)$ and $\boldsymbol{S}$ the signal covariance matrix $\mathrm{E}\left(\boldsymbol{s}\boldsymbol{s}^H\right)$. The underlined factor in (11) and (12) satisfies

$$\boldsymbol{H}_0^H\boldsymbol{R}\boldsymbol{P}_{\boldsymbol{H}_0}^{\perp} = \boldsymbol{H}_0^H\left(\boldsymbol{H}_0\boldsymbol{S}\boldsymbol{H}_0^H + \nu\boldsymbol{I}\right)\boldsymbol{P}_{\boldsymbol{H}_0}^{\perp} = \boldsymbol{0}.$$

Therefore $F_l'|_{\boldsymbol{\theta}_0,\boldsymbol{\delta}_0} = F'|_{\boldsymbol{\theta}_0,\boldsymbol{\delta}_0} = 0$, the global minimum stays at $\boldsymbol{\theta}_0, \boldsymbol{\delta}_0$ after DL. ■

2) $|F_l'| > |F'|$ *when ν_l is large.*

Proof: When ν_l is large, $\boldsymbol{P_H}\boldsymbol{R}\boldsymbol{P_H} + \hat{\nu}\boldsymbol{P}_{\boldsymbol{H}}^{\perp} + \nu_l\boldsymbol{I} \approx \nu_l\boldsymbol{I}$, therefore

$$\det\left[\boldsymbol{P_H}\boldsymbol{R}\boldsymbol{P_H} + \hat{\nu}\boldsymbol{P}_{\boldsymbol{H}}^{\perp} + \nu_l\boldsymbol{I}\right] \approx \nu_l^N;$$

and

$$\left[\boldsymbol{H}^H(\boldsymbol{R}+\nu_l\boldsymbol{I})\boldsymbol{H}\right]^{-1} = (\nu_l^{-1} - \nu\nu_l^{-2})\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1} - \nu_l^{-2}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\boldsymbol{H}^H\boldsymbol{H_0}\boldsymbol{S}\boldsymbol{H_0}^H\boldsymbol{H}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1} + O\left(\nu_l^{-3}\right),$$

therefore

$$\left(\boldsymbol{H}^H(\boldsymbol{R}+\nu_l\boldsymbol{I})\boldsymbol{H}\right)^{-1} - \frac{1}{\hat{\nu}+\nu_l}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1} \approx - \nu_l^{-2}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\boldsymbol{H}^H\boldsymbol{H_0}\boldsymbol{S}\boldsymbol{H_0}^H\boldsymbol{H}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}.$$

As a result

$$F_l' \propto \nu_l^{N-2}.$$

Because F' is independent of ν_l, $|F_l'| > |F'|$ when ν_l is large. ■

Assuming F is sufficiently smooth in the vicinity of the global minimum, properties 1) and 2) show that DL makes F more pointed when ν_l is large, which suggests that the global minimum is made more prominent. A complete proof of the prominence of global minimum requires analytical information of the local minima of F, and remains an open problem.

It is remarked that DL takes effect on the objective function, thus its effect is independent of the specific global optimizer; and during the derivation no assumption was made on the array-source geometry and types of parameter, thus the effect is independent of them, too.

C. DDL and its coupling to PSO

Inspired by the above effect we devise a decaying diagonal loading (DDL) technique. In this, the amount of DL is sufficiently large in the beginning to make the global minimum prominent and thus guides the PSO. It reduces to zero in the end and thus completely recovers the original F. DDL differs from existing DL techniques for improving performance in that it does not permanently alter the objective function. The objective function eventually optimized is the original one, therefore the precision of the original is preserved.

In DDL, the amount of loading exponentially decays,

$$\nu_l(t) = \begin{cases} 10^{-\frac{\mathrm{SNR}}{10}}\left\{10^{-\frac{\mathrm{SNR}_l}{10}\left(1-\frac{t-1}{[r_lT]}\right)} - 1\right\}, & t = 1,\ldots,[r_lT] \\ 0, & t = [r_lT]+1,\ldots,T, \end{cases} \tag{13}$$

where $r_l \in (0,1)$ is the ratio of loaded iteration, T is the maximum iteration and $[\cdot]$ stands for the rounding operator. PSO-DDL is not sensitive to SNR_l as indicated by simulation. Therefore, in the case that the original SNR is unknown, one may load $\hat{\boldsymbol{R}}$ in large amount without affecting the results.

There also exist other transformations of objective function to enhance optimization result, such as eliminating local minima to make global minimum prominent [12], and partially convexitizing the objective function [13], which differ from DDL in technique and effect on the objective function.

To conclude section III, the PSO-DDL technique is summarized in Alg. 1.

IV. Simulation Results

A. PSO-DDL results

A 5-sensor 3-source geometry is considered. The nominal array is a uniform circular array with inter-sensor spacing $a = \frac{\lambda}{2}$, whose sensor locations are perturbed by Δx_n and Δy_n that are independently, uniformly distributed in $[-b, b]$, in which b is the sensor location tolerance. $\Delta x_1, \Delta y_1$ and Δy_2 are set to zero according to [1] to fulfil the identifiability condition. The three sources reside at $\boldsymbol{\theta}_0 = [-35\ 0\ 35]^\circ$, emitting equi-power, uncorrelated signals. The SNR simulated ranges from -18 dB to 38 dB with 4 dB step size, and 200 Monte Carlo (MC) trials are performed for each SNR. The array shape is the same for the entire SNR range but is different in each MC trial. The error measure is the sum of squared error of the three signals, $||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0||^2$, averaged over the 200 trials.

Algorithm 1: PSO-DDL

```
Input: PSO and DDL parameters, objective function,
       solution space definition
Output: DOA and sensor perturbation estimates
Swarm initialization: random normalized locations and
random velocities;
for each iteration do
    Diagonally load $\hat{\boldsymbol{R}}$ according to (13);
    for each particle do
        Map particle location to solution vector in
        solution space;
        Evaluate the objective function of current
        iteration according to (2) with loaded $\hat{\boldsymbol{R}}$;
        Update particle best location $\boldsymbol{p}_i$ and group best
        location $\boldsymbol{g}_i$;
        Update particle velocity according to (3);
        if velocity exceeds maximum then
            Clamp particle velocity according to (4);
        end
        Update particle location according to (5);
        if particle location out of boundary or collision
        at DOA dimensions then
            Set random location;
        end
    end
    Check termination criterion;
end
return final global best location;
```

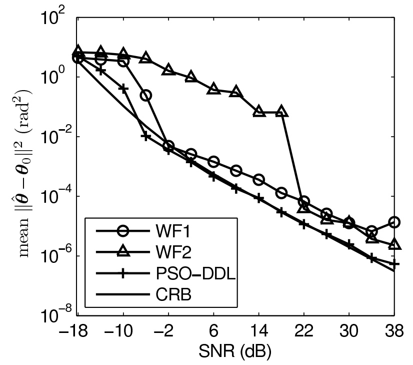

Fig. 2. Comparison of calibration accuracy under small sensor location error, $b = 0.05a$.

The results of PSO-DDL are compared with the WF1 and WF2 algorithms, and the CRB which is based on stochastic signal assumption (section 8.11.2 of [7]). Although cases with a small number of snapshots (e.g. $K = 20$) have been simulated, where WF1 and WF2 break down but PSO-DDL works, due to space limits, we present the $K = 1000$ results where WF1 and WF2 work. WF2 is initialized at true DOA for convenience. The results are not compared with [5] as its final step is WF2 and are also not compared with [6] as it lacks the initialization step. Although the coarse calibration steps of [5] can be applied to PSO-DDL for larger b, and existing DOA estimation algorithms can be used to initialize [6], a thorough investigation is out of the scope of this work as initializers differ in sensitivity to array shape perturbation and consequently affect the self-calibration result.

In the first experiment the perturbation boundary $b = 0.05a$ is considered. The DOA ranges from −180° to 180°. The control parameters for PSO and DDL are summarized in Tab. I. PSO is terminated after the maximum number of iterations, T, is reached, which amounts to PT objective function evaluations per optimization. One observes from Fig. 2 that both WF1 and WF2 succeed in calibrating the array shape at SNR's higher than 22 dB, with a similar MSE sum. Due to the fact that WF2 is based on the eigen-structure objective function, it shows a threshold at 22 dB, below which the MSE sum increases dramatically. WF1 shows a much lower threshold at −2 dB, due to its CML objective function. Nevertheless, neither of the two algorithms attains the CRB at an SNR higher than their thresholds. This results from the Taylor approximation that renders the bias the same order of magnitude as the standard deviation (STD), making the MSE considerably higher than the CRB despite the STD approaching the CRB [3], [4], [5]. PSO-DDL shows a threshold at −6 dB which is lower than both WF1 and WF2. Above this SNR, its MSE sum attains the CRB. This MSE sum is apparently lower than that of both WF1 and WF2. This experiment shows that under small perturbation, PSO-DDL has optimal accuracy that is better than WF1 and WF2, and has a lower threshold SNR.

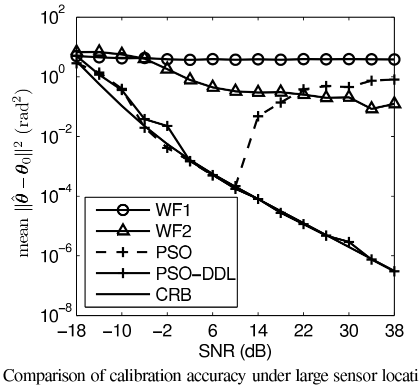

Fig. 3. Comparison of calibration accuracy under large sensor location error, $b = 0.2a$.

In the second experiment the perturbation boundary b is enlarged to $0.2a$. Two of the PSO and DDL parameters are adjusted to $T = 8000$ and $r_l = 0.1$ for the more challenging case. Other settings are the same as in the first experiment. It is observed from Fig. 3 that PSO-DDL succeeds in calibrating the array with CRB-attaining MSE sum of DOA parameters

TABLE I
PSO AND DDL CONTROL PARAMETERS

PSO parameters						DDL parameters	
φ_1	φ_2	ε	P	V	T	SNR$_a$	r_l
2.4	1.7	10^{-4} rad	30	0.5	3000	-15 dB	0.13

at SNR's greater than 2 dB. Both WF1 and WF2 fail with estimation errors much higher than PSO-DDL and the CRB. The errors of WF1 and WF2 do not decrease as the SNR increases. We also note that without DDL, PSO alone would fail with large estimation error, comparable to WF1 and WF2, when the SNR is greater than 10 dB. This experiment shows that PSO-DDL has much higher robustness against large sensor location errors than both WF1 and WF2, and it retains optimal accuracy at large perturbation.

B. DE-DDL results

In support of the remark that DDL is a universal technique that can be coupled with other algorithms to improve performance, a different stochastic global optimization algorithm, differential evolution (DE) [14] has also been employed to optimize F. In the following we compare its performances with and without the aid of DDL, but we do not detail the optimization setup due to space limits and the focus of the paper.

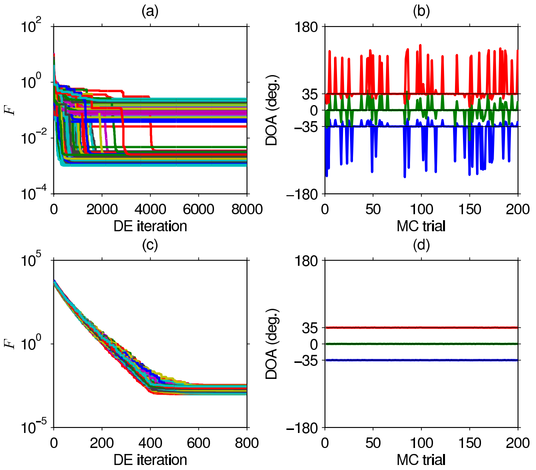

Fig. 4. Simulation results of DE with and without DDL, 200 MC trials. (a) DE, convergence pattern of F optimization. (b) DE, DOA estimates. (c) DE-DDL, convergence pattern of F optimization. (d) DE-DDL, DOA estimates.

Fig. 4 shows the simulation results of the same DE with and without DDL. The array-source settings are exactly the same as that in the second experiment of section IV-A, or in other words F is the same. The SNR shown is 22 dB. Fig. 4(a) shows that a significant proportion of the 200 MC trials performed by DE alone are trapped in local minima, ending up with optimized F values grouped higher than the global minimum bundle, and correspondingly Fig. 4(b) shows a large proportion of spiky false DOA estimates. In contrast, Fig. 4(c) shows that DE-DDL optimizes F all to the global minimum, the straight slope between the 1st and 400th DE iterations reflecting the effect of DDL, and correspondingly Fig. 4(d) shows that all the DOA estimates are close to the true values. Note that this is achieved in one tenth of the number of iterations of DE alone.

V. Conclusion

We propose the PSO-DDL paradigm to optimize the UML objective function for the array shape self-calibration problem. Compared with the existing WF1 and WF2 algorithms, the proposed method is more robust to large shape perturbation and has lower calibration error due to its direct optimization on the UML objective function. Its accuracy attains the CRB at an SNR higher than the threshold, and the threshold is lower than that of WF1 and WF2. The CRB-attaining accuracy is retained even at large perturbation.

Furthermore, PSO does not need initialization close to the true parameters. The DDL technique is of general value in its transformation of the objective function and thus can be coupled with other global optimization algorithms to improve performance. This is supported by simulation results of the DE algorithm. The PSO-DDL method is simple to implement and computationally affordable.

References

[1] Y. Rockah and P. Schultheiss, "Array shape calibration using sources in unknown locations – part I: Far-field sources," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 3, pp. 286–299, Mar. 1987.

[2] Y. Rockah, "Array processing in the presence of uncertainty," Ph.D. dissertation, Yale University, 1986.

[3] A. J. Weiss and B. Friedlander, "Array shape calibration using sources in unknown locations – a maximum likelihood approach," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 12, pp. 1958–1966, Dec. 1989.

[4] ——, "Array shape calibration using eigenstructure methods," *Signal Process.*, vol. 22, no. 3, pp. 251–258, Mar. 1991.

[5] B. P. Flanagan and K. L. Bell, "Array self-calibration with large sensor position errors," *Signal Process.*, vol. 81, no. 10, pp. 2201–2214, Oct. 2001.

[6] P.-J. Chung and S. Wan, "Array self-calibration using SAGE algorithm," *IEEE SAM 2008*, pp. 165–169, Jul. 2008.

[7] H. L. Van Trees, *Optimum Array Processing*. Wiley-Interscience, New York, 2002.

[8] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.

[9] M.-H. Li and Y.-L. Lu, "Maximum likelihood DOA estimation in unknown colored noise fields," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 3, pp. 1079–1090, Jul. 2008.

[10] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, Chichester, 1999.

[11] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 10, pp. 1783–1795, Oct. 1990.

[12] K. Parsopoulos and M. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 211–224, Jun. 2004.

[13] J. Lo, "Minimization through convexitization in training neural networks," *IEEE IJCNN 2002*, vol. 2, pp. 1889 –1894, Aug. 2002.

[14] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution," *IEEE ICEC*, pp. 842–844, May 1996.

Bibliography

[1] A. J. Weiss and B. Friedlander, "Array shape calibration using sources in unknown locations - a maximum likelihood approach," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 1958–1966, Dec. 1989.

[2] A. J. Weiss and B. Friedlander, "Array shape calibration using eigenstructure methods," *Signal Processing*, vol. 22, no. 3, pp. 251–258, Mar. 1991.

[3] M. J. Hinich and W. Rule, "Bearing estimation using a large towed array," *The Journal of the Acoustical Society of America*, vol. 58, no. 5, pp. 1023–1029, 1975.

[4] O. Besson, AA Monakov, and C. Chalus, "Signal waveform estimation in the presence of uncertainties about the steering vector," *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2432–2440, 2004.

[5] B. Porat and B. Friedlander, "Accuracy requirements in off-line array calibration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 545–556, Apr. 1997.

[6] Y. Rockah and P. Schultheiss, "Array shape calibration using sources in unknown locations–part I: Far-field sources," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 286–299, Mar. 1987.

[7] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametricapproach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, July 1996.

[8] J. Böhme, "Separated estimation of wave parameters and spectral parameters by maximum likelihood," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '86.*, Apr. 1986, vol. 11, pp. 2819–2822.

[9] A. G. Jaffer, "Maximum likelihood direction finding of stochastic sources: a separable solution," in *International Conference on Acoustics, Speech, and Signal Processing, 1988. ICASSP-88*, New York, NY, USA, Apr. 1988, pp. 2893–2896.

[10] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276 – 280, Mar. 1986.

[11] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 7, pp. 984 –995, July 1989.

[12] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and cramer-rao bound," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, pp. 720–741, May 1989.

[13] P. M. Schultheiss and J. P. Ianniello, "Optimum range and bearing estimation with randomly perturbed arrays," *The Journal of the Acoustical Society of America*, vol. 68, no. 1, pp. 167–173, 1980.

[14] W. L. Stutzman and G. A. Thiele, *Antenna theory and design*, Wiley, 1998.

[15] J. W. Goodman, *Introduction to Fourier optics*, Roberts & Company Publishers, 2005.

[16] G. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.

[17] G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," *Inverse Problems*, vol. 19, no. 2, pp. R1–R26, 2003.

[18] J. Böhme, *Advances in Spectrum Analysis and Array Processing, S. Haykin, Ed.*, chapter 1: Array processing, pp. 1–63, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[19] H. L. Van Trees, *Optimum Array Processing*, Wiley-Interscience, New York, 2002.

[20] P. Stoica and A. Nehorai, "On the concentrated stochastic likelihood function in array signal processing," *Circuits, Systems, and Signal Processing*, vol. 14, no. 5, pp. 669–674, 1995.

[21] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1783–1795, Oct. 1990.

[22] B. Friedlander, "Sensitivity analysis of the maximum likelihood direction-finding algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 6, pp. 953–968, Nov. 1990.

[23] B. Friedlander, “A sensitivity analysis of the MUSIC algorithm,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1740–1751, Oct. 1990.

[24] A. L. Swindlehurst and T. Kailath, “A performance analysis of subspace-based methods in the presence of model errors. I. the MUSIC algorithm,” *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1758–1774, July 1992.

[25] A. L. Swindlehurst and T. Kailath, “A performance analysis of subspace-based methods in the presence ofmodel error. II. multidimensional algorithms,” *IEEE Transactions on Signal Processing*, vol. 41, no. 9, pp. 2882–2890, Sept. 1993.

[26] F. Li and R. J. Vaccaro, “Sensitivity analysis of DOA estimation algorithms to sensor errors,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 708–717, July 1992.

[27] A. Ferreol, P. Larzabal, and M. Viberg, “On the asymptotic performance analysis of subspace DOA estimation in the presence of modeling errors: case of MUSIC,” *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 907–920, Mar. 2006.

[28] A. Ferreol, P. Larzabal, and M. Viberg, “On the resolution probability of MUSIC in presence of modeling errors,” *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1945–1953, May 2008.

[29] A. Ferreol, P. Larzabal, and M. Viberg, “Performance prediction of maximum-likelihood direction-of-arrival estimation in the presence of modeling errors,” *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4785 –4793, oct. 2008.

[30] J. Pierre and M. Kaveh, “Experimental performance of calibration and direction-findingalgorithms,” in *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-91*, Toronto, Ont., Canada, Apr. 1991, pp. 1365–1368.

[31] C. M. S. See, “Sensor array calibration in the presence of mutual coupling andunknown sensor gains and phases,” *Electronics Letters*, vol. 30, no. 5, pp. 373–374, Mar. 1994.

[32] B. C. Ng and C. M. S. See, “Sensor-array calibration using a maximum-likelihood approach,” *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 6, pp. 827 –835, Jun. 1996.

[33] A. Lundgren, M. Lanne, and M. Viberg, "Two-step esprit with compensation for modelling errors using a sparse calibration grid," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007*, Apr. 2007, vol. 2, pp. II–1045 –II–1048.

[34] M. Viberg and A. Lundgren, "Array interpolation based on local polynomial approximation with application to doa estimation using weighted music," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009*, Apr. 2009, pp. 2145 –2148.

[35] H. He, Y. Wang, and J. Saillard, "Interpolation-based calibration for near-field source localization," in *EUSIPCO 2009*, 2009, pp. 1131–1135.

[36] M. A. Doron and E. Doron, "Wavefield modeling and array processing. I. spatial sampling," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2549–2559, Oct. 1994.

[37] M. A. Doron, E. Doron, and H. Rafael, "Wavefield modeling and array processing. II. Algorithms," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2560–2570, 1994.

[38] M. A. Doron, E. Doron, and H. Rafael, "Wavefield modeling and array processing. III. Resolution capacity," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2571–2580, 1994.

[39] F. Belloni, A. Richter, and V. Koivunen, "DoA estimation via manifold separation for arbitrary array structures," *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 4800–4810, Oct. 2007.

[40] N. Fistas and A. Manikas, "A new general global array calibration method," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-94*, Adelaide, SA, Australia, Apr. 1994.

[41] B. C. Ng and A. Nehorai, "Active array sensor location calibration," in *Proc. 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-93*, Apr. 1993, vol. 4, pp. 21 –24 vol.4.

[42] A.-J. Boonstra and A.-J. van der Veen, "Gain calibration methods for radio telescope arrays," *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 25 – 38, Jan. 2003.

[43] B. C. Ng and A. Nehorai, "Optimum active array shape calibration," in *Twenty-Fifth Asilomar Conference on Signals, Systems and Computers*, Nov. 1991, pp. 893 –897 vol.2.

[44] B. C. Ng and A. Nehorai, "Active array sensor localization," *Signal Process.*, vol. 44, no. 3, pp. 309–327, 1995.

[45] L. Seymour, C. Cowan, and P. Grant, "Bearing estimation in the presence of sensor positioning errors," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '87*, Apr. 1987, vol. 12, pp. 2264–2267.

[46] B. P. Ng, M. H. Er, and C. Kot, "Array gain/phase calibration techniques for adaptive beamforming and direction finding," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 141, no. 1, Feb. 1994.

[47] B. P. Ng, J. P. Lie, M. H. Er, and A.-G. Feng, "A practical simple geometry and gain/phase calibration technique for antenna array processing," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 7, pp. 1963 –1972, jul. 2009.

[48] Y. Rockah and P. Schultheiss, "Array shape calibration using sources in unknown locations–Part II: Near-field sources and estimator implementation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 6, pp. 724–735, 1987.

[49] Y. Rockah, H. Messer, and P. M. Schultheiss, "Localization performance of arrays subject to phase errors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 4, pp. 402–410, July 1988.

[50] M. Levi and H. Messer, "Sufficient conditions for array calibration using sources of mixed tapes," *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-90*, pp. 2943–2946 vol.5, Apr 1990.

[51] A. Paulraj and T. Kailath, "Direction of arrival estimation by eigenstructure methods with unknown sensor gain and phase," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '85.*, Apr. 1985, vol. 10, pp. 640 – 643.

[52] M. P. Wylie, S. Roy, and R. F. Schmitt, "Self-calibration of linear equi-spaced (LES) arrays," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-93*, Apr. 1993, vol. 1, pp. 281 –284 vol.1.

[53] M. P. Wylie, S. Roy, and H. Messer, "Joint DOA estimation and phase calibration of linear equispaced (LES) arrays," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3449 –3459, Dec. 1994.

[54] Y.-M. Li and M. H. Er, "Theoretical analyses of gain and phase error calibration with optimal implementation for linear equispaced array," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 712 – 723, Feb. 2006.

[55] P. C. Yip and Y.-F. Zhou, "A self-calibration algorithm for cyclostationary signals and its uniqueness analysis," *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95*, vol. 3, pp. 1892 –1895 vol.3, May. 1995.

[56] F. Sellone and A. Serra, "A novel online mutual coupling compensation algorithm for uniform and linear arrays," *IEEE Transactions on Signal Processing*, vol. 55, no. 2, pp. 560 –573, Feb. 2007.

[57] J. F. Böhme and D. Kraus, "On least squares methods for direction of arrival estimation in the presence of unknown noise fields," in *Proc. International Conference on Acoustics, Speech, and Signal Processing, ICASSP-88*, Apr. 1988, pp. 2833 –2836 vol.5.

[58] B. Friedlander and A. J. Weiss, "Direction finding using noise covariance modeling," *IEEE Transactions on Signal Processing*, vol. 43, no. 7, pp. 1557–1567, July 1995.

[59] C.-M. S. See and B.-K. Poh, "Parametric sensor array calibration using measured steering vectors of uncertain locations," *IEEE Transactions on Signal Processing*, vol. 47, no. 4, pp. 1133 –1137, Apr. 1999.

[60] H. S. Mir, J. D. Sahr, G. F. Hatke, and C. M. Keller, "Passive source localization using an airborne sensor array in the presence of manifold perturbations," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2486–2496, June 2007.

[61] J. C. Chen, R. E. Hudson, and K. Yao, "Maximum-likelihood source localization and unknown sensor locationestimation for wideband signals in the near-field," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1843–1854, Aug. 2002.

[62] C. E. Chen, F. Lorenzelli, R. E. Hudson, and K. Yao, "Maximum Likelihood DOA Estimation of Multiple Wideband Sources in the Presence of Nonuniform Sensor Noise," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 6, 2008.

[63] B. Friedlander and A. J. Weiss, "Eigenstructure methods for direction finding with sensor gain and phase uncertainties," in *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-88*, Apr. 1988, pp. 2681–2684.

[64] B. Friedlander and A. J. Weiss, "Direction finding in the presence of mutual coupling," *IEEE Transactions on Antennas and Propagation*, vol. 39, no. 3, pp. 273–284, Mar. 1991.

[65] A. J. Weiss and B. Friedlander, "Almost blind steering vector estimation using second-order moments," *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 1024–1027, Apr. 1996.

[66] A. Flieller, A. Ferreol, P. Larzabal, and H. Clergeot, "Robust bearing estimation in the presence of direction-dependent modelling errors: identifiability and treatment," in *Proc. International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95*, May. 1995, vol. 3, pp. 1884 –1887 vol.3.

[67] M. Viberg and A. L. Swindlehurst, "A Bayesian approach to auto-calibration for parametric array signal processing," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3495–3507, Dec. 1994.

[68] C. R. Wan, J. T. Goh, and C. S. Chia, "Array calibration based on inverse subspace fitting," in *MTS/IEEE Conference and Exhibition, OCEANS*, 2001, vol. 4, pp. 2486 –2491.

[69] B. Steinberg, "Radar imaging from a distorted array: The radio camera algorithm and experiments," *IEEE Transactions on Antennas and Propagation*, vol. 29, no. 5, pp. 740 – 748, Sep. 1981.

[70] S. J. Wijnholds and A.-J. van der Veen, "Multisource self-calibration for sensor arrays," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3512 –3522, Sep. 2009.

[71] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and cramer-rao bound: further results and comparisons," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 12, pp. 2140 –2150, Dec. 1990.

[72] Q. Cheng, Y. Hua, and P. Stoica, "Asymptotic performance of optimal gain-and-phase estimators of sensor arrays," *IEEE Transactions on Signal Processing*, vol. 48, no. 12, pp. 3587–3590, Dec. 2000.

[73] J. A. Fessler and A. O. Hero, “Space-alternating generalized expectation-maximization algorithm,” *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2664–2677, Oct. 1994.

[74] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[75] C. Rao, “Estimation and tests of significance in factor analysis,” *Psychometrika*, vol. 20, no. 2, pp. 93–111, June 1955.

[76] M. Feder and E. Weinstein, “Parameter estimation of superimposed signals using the EM algorithm,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 4, pp. 477–489, Apr. 1988.

[77] B. D. O. Anderson and J. B. Moore, *Optimal filtering*, Prentice-Hall, Englewood Cliffs, 1979.

[78] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis (3rd edition)*, Wiley-Interscience, New Jersey, 2003.

[79] P.-J. Chung and J. F. Böhme, “Comparative convergence analysis of EM and SAGE algorithms in DOA estimation,” *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 2940–2949, 2001.

[80] X. L. Meng and D. van Dyk, “The EM algorithm – an old folk-song sung to a fast new tune,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 59, no. 3, pp. 511–567, 1997.

[81] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” *6th IEEE International Symposium on Micro Machine and Human Science, MHS ’95*, pp. 39–43, Oct. 1995.

[82] M.-H. Li and Y.-L. Lu, “Maximum likelihood DOA estimation in unknown colored noise fields,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1079–1090, July 2008.

[83] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, “Particle swarm optimization: Basic concepts, variants and applications in power sys-

tems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, April 2008.

[84] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. Congress on Evolutionary Computation*, 2000, vol. 1, pp. 84–88.

[85] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. Congress on Evolutionary Computation, CEC '02*, 2002, vol. 2, pp. 1671–1676.

[86] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Wiley, 1999.

[87] P. Stoica, E. G. Larsson, and A. B. Gershman, "The stochastic CRB for array processing: a textbook derivation," *IEEE Signal Processing Letters*, vol. 8, no. 5, pp. 148–150, May 2001.

[88] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 211–224, Jun. 2004.

[89] J. T. Lo, "Minimization through convexitization in training neural networks," in *Proc. International Joint Conference on Neural Networks, IJCNN '02*, Aug. 2002, vol. 2, pp. 1889 –1894.

[90] B. P. Flanagan and K. L. Bell, "Array self-calibration with large sensor position errors," *Signal Processing*, vol. 81, pp. 2201–2214, October 2001.

[91] P.-J. Chung and S. Wan, "Array self-calibration using sage algorithm," in *proc. 5th IEEE Sensor Array and Multichannel Signal Processing Workshop, SAM 2008*, Darmstadt, Germany, July 2008, pp. 165–169.

[92] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution," in *Proc. IEEE International Conference on Evolutionary Computation*, May 1996, pp. 842 –844.

[93] A. L. Swindlehurst and T. Kailath, "Passive direction-of-arrival and range estimation for near-field sources," *Fourth Annual ASSP Workshop on Spectrum Estimation and Modeling*, pp. 123–128, Aug. 1988.

[94] D. Starer and A. Nehorai, "Passive localization of near-field sources by path following," *IEEE Transactions on Signal Processing*, vol. 42, no. 3, pp. 677–680, Mar. 1994.

[95] K. Abed-Meraim, Y.-B. Hua, and A. Belouchrani, "Second-order near-field source localization: algorithm and performance analysis," *Thirtieth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 723–727, Nov. 1996.

[96] R. N. Challa and S. Shamsunder, "Passive near-field localization of multiple non-gaussian sources in 3-D using cumulants," *Signal Processing*, vol. 65, no. 1, pp. 39 – 53, 1998.

[97] W.-J. Zhi and M. Y.-W. Chia, "Near-field source localization via symmetric subarrays," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007*, vol. 2, pp. II–1121–II–1124, April 2007.

[98] R. Boyer and J. Picheral, "Second-order near-field localization with automatic paring operation," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008*, pp. 2569–2572, April 2008.

[99] H. He, Y. Wang, and J. Saillard, "A high resolution method of source localization in near-field by using focusing technique," *Proc. European Signal Processing Conference, EURASIP, Lausanne, Switzerland*, Aug. 2008.

[100] K. Deng, Q.-Y. Yin, and H.-M. Wang, "Blind ranges, frequencies and doas estimation for near field sources," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009*, pp. 2125–2128, April 2009.

[101] E. Boyer, A. Ferreol, and P. Larzabal, "Simple robust bearing-range source's localization with curved wavefronts," *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 457–460, June 2005.

[102] J. A. Cadzow, "Multiple source location-the signal subspace approach," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 7, pp. 1110–1125, Jul. 1990.

[103] Y. D. Huang and M. Barkat, "Near-field multiple source localization by passive sensor array," *IEEE Transactions on Antennas and Propagation*, vol. 39, no. 7, pp. 968–975, July 1991.

[104] R. Jeffers, K. L. Bell, and H. L. Van Trees, "Broadband passive range estimation using MUSIC," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '02*, vol. 3, pp. III–2921–III–2924, May 2002.

[105] A. J. Weiss and B. Friedlander, "Range and bearing estimation using polynomial rooting," *IEEE Journal of Oceanic Engineering*, vol. 18, no. 2, pp. 130–137, Apr. 1993.

[106] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex algorithm in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.