Pontifícia Universidade Católica
do Rio de Janeiro

Wesley Souza Leite

Sparse subarrays for direction of arrival estimation: algorithms and geometries

Tese de Doutorado

Thesis presented to the Programa de Pós-Graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica.

Advisor: Prof. Rodrigo Caiado de Lamare

Rio de Janeiro
December 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Wesley Souza Leite

Sparse subarrays for direction of arrival estimation: algorithms and geometries

Thesis presented to the Programa de Pós-Graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica. Approved by the Examination Committee:

Prof. Rodrigo Caiado de Lamare
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Yuriy Zakharov
University of York

Prof. Wei Liu
The Hong Kong Polytechnic University

Prof. Martin Haardt
Technische Universität Ilmenau

Prof. Felix Dieter Antreich
Instituto Tecnológico de Aeronáutica

Prof. Lukas Tobias Nepomuk Landau
Departamento de Engenharia Elétrica – PUC-Rio

Rio de Janeiro, December 4th, 2024

Wesley Souza Leite

The author graduated with a degree in Electrical Engineering from the Federal University of Maranhão, Brazil, in 2014 and obtained a Master's degree in Electrical Engineering from the Pontifical Catholic University of Rio de Janeiro in 2020.

Bibliographic data

Leite, Wesley Souza

Sparse subarrays for direction of arrival estimation: algorithms and geometries / Wesley Souza Leite; advisor: Rodrigo Caiado de Lamare. – 2024.

156 f. : il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2024.
Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Arranjos esparsos. 3. Arranjos parcialmente calibrados. 4. Estimação de direção de chegada. 5. Suavização espacial. I. Lamare, Rodrigo Caiado de. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

To my parents, for their unrestricted love, loyalty and continuing support.

Acknowledgments

First of all, I would like to thank my advisor, Professor Rodrigo C. de Lamare, for his always accurate and high-quality guidance, setting my sights higher and expanding my skills in order to develop this work seamlessly.

Then, I wish to acknowledge the support and great love of my family: Antonio Leite, Regina Leite, Regilaine Leite and Ana Caldas. My mother, Regina, deserves all the gratitude I can offer. To my father, Antonio, thank you so much for fulfilling me with courage and fearlessness. To my sister, Regilaine, my deepest gratitude for encouraging me. To my wife, Ana Caldas, thank you for the patience and your love, kindly supporting me during challenging periods.

I would like to thank the members of my defense committee: Professor Yuriy Zakharov, Professor Wei Liu, Professor Martin Haardt, Professor Felix Dieter Antreich, and Professor Lukas Tobias Nepomuk Landau.

Thank you for those I did not mention explicitly (PUC colleagues and PUC professors) and helped me to accomplish this work.

Then, I wish to thank PUC-Rio for the aids granted.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

This study was also financed by PUC-Rio.

Abstract

Leite, Wesley Souza; Lamare, Rodrigo Caiado de (Advisor). **Sparse subarrays for direction of arrival estimation: algorithms and geometries**. Rio de Janeiro, 2024. 156p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

This thesis explores advanced array signal processing techniques for both fully and partially calibrated arrays. We introduce novel sparse array geometries based on sparse linear subarrays and develop new direction-of-arrival (DOA) estimation algorithms for narrowband electromagnetic signals, framed within statistical signal processing principles. The proposed algorithms, named Generalized Coarray MUSIC (GCA-MUSIC) and Generalized Coarray Root MUSIC (GCA-rMUSIC), extend the classical Multiple Signal Classification (MUSIC) framework to sparse subarrays configurations. Sparse linear subarray design techniques were proposed, as well as an analysis of the degrees of freedom of subarrays (sDoF) as a function of degrees of freedom of the whole array (DoF). Additionally, we develop Variable Window Size (VWS) versions of these algorithms, which incorporate flexible spatial smoothing apertures. These methods provide high-accuracy DoA estimates and offer the key advantage of resolving more sources than the number of physical sensors in each subarray by exploiting coarray structures. Performance analysis demonstrates that GCA-MUSIC and GCA-rMUSIC, along with its VWS variants, improve accuracy in the context of partially-calibrated arrays, where calibration uncertainties may exist. Furthermore, VWS variants of the Coarray MUSIC (CA-MUSIC) algorithm are presented for fully calibrated (coherent) arrays, enabling adaptable smoothing strategies for enhanced performance. In addition to algorithmic development, we compute the Fisher Information Matrices (FIMs) for the complete set of parameters in this generalized data model, including both self and cross-coupled parameter relationships. These matrices account for source directions, source powers, noise power, and the real and imaginary components of all calibration parameters, representing both correlated and uncorrelated source scenarios. This work significantly advances the theoretical understanding of DoA estimation performance limits by providing a more rigorous quantification of the Cramér-Rao bounds. These bounds are particularly relevant in scenarios with partially calibrated arrays and uncorrelated sources, as demonstrated using the Khatri-Rao product-based data model.

Keywords

Sparse arrays; Partially calibrated arrays; Direction of arrival estimation; Spatial smoothing.

Resumo

Leite, Wesley Souza; Lamare, Rodrigo Caiado de. **Subarranjos esparsos para estimação de direção de chegada: algoritmos e geometrias**. Rio de Janeiro, 2024. 156p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Esta tese desenvolve técnicas avançadas de processamento de sinais com arranjos de sensores, tanto para arranjos completamente calibrados quanto parcialmente calibrados. São propostas novas geometrias de arranjos esparsos baseadas em subarranjos lineares esparsos, bem como são desenvolvidos novos algoritmos de estimativa de direção de chegada (DOA) para sinais eletromagnéticos de banda estreita, utilizando-se a teoria de processamento estatístico. Os algoritmos propostos, denominados Generalized Coarray MUSIC (GCA-MUSIC) e Generalized Coarray Root MUSIC (GCA-rMUSIC), expandem a técnica clássica denominada Multiple Signal Classification (MUSIC) para configurações de subarranjos esparsos. Técnicas de projeto de subarranjos lineares esparsos foram propostas, assim como uma análise dos graus de liberdade dos subarranjos (sDoF) em função dos graus de liberdade do arranjo completo (DoF). Além disso, desenvolvem-se versões com tamanho de Janela Variável (VWS) desses algoritmos, que incorporam técnicas de suavização espacial com abertura variável. Esses métodos proporcionam estimativas de direção de alta precisão e conseguem estimar um número maior de fontes do que o número de sensores físicos em cada subarranjo, explorando estruturas de coarranjo específicas. A análise de desempenho demonstra que o GCA-MUSIC e o GCA-rMUSIC, juntamente com suas variantes VWS, melhoram a precisão no contexto de arranjos parcialmente calibrados, onde podem existir incertezas de calibração. Além disso, são apresentadas variantes VWS do algoritmo Coarray MUSIC (CA-MUSIC) para arranjos totalmente calibrados (coerentes), permitindo estratégias de suavização adaptáveis para um desempenho aprimorado. Além do desenvolvimento algorítmico, foram derivadas as Matrizes de Informação de Fisher (FIMs) para o conjunto completo de parâmetros deste modelo de dados generalizado, incluindo tanto as relações de parâmetros consigo próprios quanto cruzados. Essas matrizes levam em consideração as direções das fontes, potências das fontes, potência do ruído e as componentes reais e imaginárias de todos os parâmetros de calibração, representando cenários com fontes correlacionadas e descorrelacionadas. Este trabalho avança significativamente a compreensão teórica dos limites de desempenho da estimativa de direções, fornecendo uma quantificação mais rigorosa dos limitantes de Cramér-Rao. Esses limitantes são particularmente relevantes em cenários com arranjos parcialmente calibrados e fontes descorrelacionadas, conforme demonstrado utilizando-se modelos de dados baseados no produto de Khatri-Rao.

Palavras-chave

Arranjos esparsos; Arranjos parcialmente calibrados; Estimação de direção de chegada; Suavização espacial.

Table of contents

List of Abreviations

A/D – Analog-to-Digital Converters

AML – Asymptotic Maximum Likelihood

ASNR – Array Signal-to-Noise Ratio

BP – Basis Pursuit

CA-MUSIC – Coarray Multiple Signal Classification

CPA – Coprime Array

CPU – Central Processing Unit

CRLB – Cramér-Rao Lower Bound

CS – Compressive Sensing

DAQ – Data Acquisition

DCTM – Difference Coarray Transformation Model

DDC – Digital Down-Converter

DF – Direction Finding

DOA – Direction of Arrival

DoF – Degrees of Freedom

DSP – Digital Signal Processing/Processor

ESPRIT – Estimation of Signal Parameters via Rotational Invariance Techniques

ESS – Enhanced Spatial Smoothing

EVD – Eigenvalue Decomposition

FBA – Forward-Backward Averaging

FBSS – Forward-Backward Spatial Smoothing

FIM – Fisher Information Matrix

FSS – Forward Spatial Smoothing

GCA-MUSIC – Generalized Coarray Multiple Signal Classification

GCA-rMUSIC – Generalized Coarray Root Multiple Signal Classification

HPBW – Half Power Bandwidth

i.i.d – Independent and Identically Distributed

IF – Intermediate Frequency

IHT – Iterative Hard Thresholding

LBML-OMP – List-Based Maximum Likelihood Orthogonal Matching Pursuit

LO – Local Oscillator

LS – Least Squares

MHA – Minimum Hole Array

ML – Maximum Likelihood

MLE – Maximum Likelihood Estimator/Estimation

MMV – Multiple Measurements Vector

MRA – Minimum Redundancy Array

MUSIC – Multiple Signal Classification

MVDR – Minimum Variance Distortionless Response

NAQn – n th-Level Nested Array

NLA – Non-Uniform Linear Array

NPA – Non-Uniform Planar Array

NVA – Non-Uniform Volumetric Array

OMP – Orthogonal Matching Pursuit

PD – Positive Definite

PDF – Probability Density Function

POCS – Projection onto Convex Sets

PSD – Positive Semidefinite

QCBP – Quadratically Constrained Basis Pursuit

RARE – Rank-Reduction Estimator

RF – Radio Frequency

RMSE – Root Mean Squared Error

SCA – Subcoarray

SLA – Sparse Linear Array

SMV – Single Measurement Vector

SNAQn – nth-Order Super Nested Array

SNR – Signal-to-Noise Ratio

SpSub – Sparse Subarray

SS-MUSIC – Spatial Smoothing Multiple Signal Classification

SS-SCA – Spatial Smoothing Subcoarray

SVD – Singular Value Decomposition

TLS – Total Least Squares

UA – Uniform Arrays

UDoF – Uniform Degrees of Freedom

ULA – Uniform Linear Array

UPA – Uniform Planar Array

UVA – Uniform Volumetric Array

VWS-CA-MUSIC – Variable Window Size Coaray Multiple Signal Classification

VWS-CA-rMUSIC – Variable Window Size Coaray Root Multiple Signal Classification

VWS-GCA-MUSIC – Variable Window Size Generalized Coarray Multiple Signal Classification

VWS-GCA-rMUSIC – Variable Window Size Generalized Coarray Root Multiple Signal Classification

WSS – Weighted Spatial Smoothing

List of figures

List of tables

List of Algorithms

List of Symbols

$\mathbb{N}$ - the set of natural numbers

$\mathbb{Z}$ - the set of integers

$\mathbb{C}$ - the set of complex numbers

$[N]$ - the set of natural numbers excluding the zero element and not exceeding N, i.e, $[N] \triangleq \{1, 2, \ldots, N\}$

n - a natural number indicating the physical positioning of a sensor as multiple of d (minimum inter-element spacing), unless otherwise specified. It is referred as *normalized position.*

$\dim(\mathbf{x})$ - the dimension of a vector $\mathbf{x}$

$\mathrm{supp}\,(\mathbf{x})$ - the support of a vector $\mathbf{x} \in \mathbb{C}^N$ (index set of its non-zero elements)

$|\mathbb{L}|$ - the cardinality of a set $\mathbb{L}$ (number of elements/objects in the set $\mathbb{L}$)

$\boldsymbol{\theta}^g$ - the angle grid, in terms of sines of directions (dimensionless), over which the search procedure is performed

$\mathbb{S}$ - the subset of natural numbers denoting the physical placement of sensors, e.g. $\mathbb{S} = \{0, n_1, \ldots, n_{N-1}\}$ for an array with N sensors

$\mathbb{D}$ - the subset of integers denoting the difference coarray associated with the physical array defined by $\mathbb{S}$

$\mathbf{x}_{\mathbb{S}}$ - received signal from an array with sensors' locations defined by $\mathbb{S}$

$\mathbf{x}_{\mathbb{D}}$ - received signal in a difference coarray domain defined by $\mathbb{D}$

$\mathbf{R}_{\mathbb{L}}$ - received signal covariance matrix

$\hat{\mathbf{R}}_{\mathbb{L}}$ - sample received signal covariance matrix estimate

$\mathbf{R}_s$ - source signal covariance matrix

$\mathbf{A}_{\mathbb{L}}(\boldsymbol{\theta})$ - array manifold matrix defined over the angle elements in $\boldsymbol{\theta}$ and the sensors positions given by the set $\mathbb{L}$, which is usually assumed to be either $\mathbb{S}$ or $\mathbb{D}$

$\mathbf{a}_{\mathbb{L}}(\theta_i)$ - steering vector defined for the angle θ_i and the sensors' positions given by the set $\mathbb{L}$, which is assumed to be either $\mathbb{S}$ or $\mathbb{D}$

$[\mathbf{A}]_{i,j}$ - (i,j)-th element of matrix $\mathbf{A}$

$[\mathbf{a}]_i$ - i-th element of vector $\mathbf{a}$

$\mathrm{vec}\left(\mathbf{A}\right)$ - operator that produces a column vector in $\mathbb{C}^{mN}$ by stacking the N columns of the matrix $\mathbf{A} \in \mathbb{C}^{m\times N}$ below one another

$\mathrm{vecd}\left(\mathbf{A}\right)$ - diagonal extraction operator. Produces a column vector in $\mathbb{C}^{N}$ which consists of the diagonal elements of $\mathbf{A} \in \mathbb{C}^{N\times N}$

$\|\cdot\|_p$ - the l_p-quasinorm $(p = 0)$ or l_p-norm of a vector

$\mathrm{E}\{\cdot\}$ - the expectation operator

$\mathbf{I}$ - The identity matrix of appropriate dimensions

$\mathbf{AB}$ - standard matrix product between matrices $\mathbf{A}$ and $\mathbf{B}$

$\mathbf{A} \circ \mathbf{B}$ - Column-wise Khatri-Rao product between matrices $\mathbf{A}$ and $\mathbf{B}$

$\mathbf{A} \otimes \mathbf{B}$ - Kronecker product between matrices $\mathbf{A}$ and $\mathbf{B}$

$\mathbf{A}^H$ - the conjugate transpose or Hermitian transpose of a matrix $\mathbf{A} \in \mathbb{C}^{m\times N}$

$\mathbf{A}^*$ - the conjugate of a matrix $\mathbf{A}$

$\mathbf{A}^T$ - the transpose of a matrix $\mathbf{A}$

$\mathbf{P}_{\mathbf{A}}$ - the orthogonal projection matrix onto the spanning of the columns of the full column rank (tall) matrix $\mathbf{A}$

$\mathrm{tr}(\mathbf{A})$ - the trace of a matrix $\mathbf{A}$

$\det\left(\mathbf{A}\right)$ - the determinant of a matrix $\mathbf{A}$

$\mathrm{diag}\left(\mathbf{x}\right)$ - the diagonal matrix with main diagonal elements equal to the ones in the vector $\mathbf{x}$

$\mathrm{blkdiag}\left(\mathbf{A}_1, \ldots, \mathbf{A}_N\right)$ - the block-diagonal matrix with the matrices $\mathbf{A}_n$ on the block-diagonal structure

$\mathrm{colspan}\left(\mathbf{A}\right)$ - the column space of $\mathbf{A}$

$\mathcal{N}\left(\mathbf{A}\right)$ - the null space of $\mathbf{A}$

$\mathcal{R}\left(\mathbf{A}\right)$ - the range space of $\mathbf{A}$

$\mathrm{BW}_{\mathrm{HPBW}}$ - half-power beamwidth

$\mathrm{BW}_{\mathrm{NN}}$ - notch-notch beamwidth

κ_l - aperture of the l-th subarray

$a(n) \circledast b(n)$ - represents the discrete convolution between the sequences $a(n)$ and $b(n)$

Notice that blackboard boldface characters ($\mathbb{S}$) are sets; lowercase normal (a) and boldface characters ($\mathbf{a}$) indicate scalars, and column vectors, respectively. Uppercase boldface characters represent matrices ($\mathbf{A}$). A detailed explanation for some of the presented symbols is given appropriately along with the text.

From the definitions of Kronecker and column-wise Khatri-Rao products, it should be noticed that, for $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N] \in \mathbb{C}^{m \times N}$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_K] \in \mathbb{C}^{p \times K}$,

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1N}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mN}\mathbf{B} \end{bmatrix} \in \mathbb{C}^{mp \times NK}$$

and

$$\mathbf{A} \circ \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \mid \ldots \mid \mathbf{a}_N \otimes \mathbf{b}_N] \in \mathbb{C}^{mp \times N}$$

where the column-wise Khatri-Rao product is defined only for $N = K$.

One must still have chaos in oneself to be able
to give birth to a dancing star.

Friedrich Nietzsche, *Thus Spoke Zarathustra.*

1
Introduction

There has been a great deal of research in the field of direction of arrival (DOA) estimation using arrays of sensors [3–5]. In this context, many techniques have been used to perform this parameter estimation task [6–9]. Associated to those techniques, there are plenty of applications in the design of sonar systems, radar detection systems, communications equipments, electronic warfare systems, seismology and astronomy [10, 11].

In this work, the directions to be estimated are parameters of interest associated to signal sources and consist of angular measures between the straight line connecting the emitting source with the array and the array broadside axis. In general, there are many other parameters of interest to be estimated in the fields cited above (polarization, power level, and velocity, to name a few), but in our case we are only interested in the electromagnetic signal directions [12].

1.1
Motivation and prior works

In order to perform DOA estimation with array signal processing, we need to bring together two slightly distinct areas that must act with synergy in order to provide estimates that meet some requirement specification. Those two areas are hardware and software. In this case, by hardware we mean an array of sensors, along with some RF chains, mixers, local oscillators (LO), filters, A/D converters, digital down-converters (DDC) and some digital signal processor (DSP). The software part contains the computational implementation of the mathematical methods developed to extract the DOAs from the baseband-complex raw data that was acquired by the hardware.

Many advancements have been made in the two areas to optimize the estimation in many senses. As examples, we can cite the increased number of parameters to be estimated with a given number of physical sensors, as well as a reduction in hardware complexity (reduction of hardware acquisition costs and energy-saving capabilities, related to system operational cost and portability). The number of DOAs that can be estimated is highly dependent on the two aspects of the estimation system: the array must itself be designed so that it possesses enough degrees of freedom[1] and the algorithm being used should be capable of efficiently employ those degrees of freedom to estimate the maximum possible number of directions.

Those capabilities are maximized in scenarios with coherent sensors along the whole array, i.e., the RF channel associated to each sensor is connected to the same local oscillator and the intersensor distance between all the elements is perfectly known (the whole array manifold is known). This demands a higher hardware complexity and is very challenging to maintain for many practical scenarios, especially for large aperture arrays [13, 14].

A possible solution to this problem is to divide the whole array into subarrays and approach the task in one of two different ways: i) pre-process the data separately at some degree and forward some information to a central processing unit to finish the estimation (non-coherent processing); or ii) compensate for the unknown parameters that are needed for coherent processing by estimating them. For example, one can develop routines to estimate the phase shifts between the steering vectors of each subarray. Those phase shifts can be due to at least one of the following conditions [15]:

a) Unknown intersubarray displacements;
b) Lack of proper synchronization in a time basis (non-coherent arrays), i.e., the subarrays are linked to different local oscillators with no phase synchronization mechanism; and
c) Unknown propagation channel mismatches (fades) for the signals that arrive in each subarray.

Conventional approaches face many challenges in dealing with such situations, because of their large sensitivity to array manifold modelling errors. As mentioned earlier, we emphasize that the hardware infrastructure required to keep large arrays fully synchronized or to instantly determine inter-subarray displacements (for example, in subarrays located on different platforms) is significant, not to mention the propagation channel mismatches, which become even more pronounced as the distance between the subarrays increases. Additionally, array processing with subarrays possesses larger aperture properties when compared to coherent arrays and then can be advantageous in the resolution of directional ambiguities [13].

Many methods have been designed to deal with DOA estimation using partly calibrated arrays. We describe a few in what follows. In [16, 17], a new search-free polynomial rooting/eigenstructure-based approach was developed. This approach, termed rank-reduction (RARE) algorithm fills some gaps in

[1] The concept of degrees of freedom will be formally established in Chapter 2.

DOA estimation with partially calibrated arrays: it is capable of dealing with sparse subarrays and does not need additional knowledge from inter-subarray displacements. However, it lacks the capabilities to identify more sources than the number of physical sensors and can only be employed if the uncertainties are restricted to the intersubarray displacements. Additionally, it requires all the data to be transmitted to a central processing unit, which increases the bandwidth constraints of the subarrays communication links.

Since the subarrays are considered to be fully calibrated (only the whole array is partially calibrated), this means we can develop a method that considers only the covariance matrices of each individual subarray. A method that exploits this fact was developed in [13], where the authors use an intersection of subspaces derived from the covariance matrices to estimate a composed pseudo-spectrum. The method is termed Generalized MUSIC (G-MUSIC). The pseudo-spectrum is used to perform a 1-D search analogous to MUSIC [6] to find the DOAs. Indeed, this algorithm inspired us to develop our pseudospectrum-based GCA-MUSIC, that will be introduced in Chapter 3.

More recently, a method based on an approximate maximum likelihood estimator has been proposed [14]. This method estimates the phase shifts of each subarray and allows us to utilize the full array aperture. However, despite its outstanding performance, the associated computational complexity is extremely high, as it relies on a non-convex optimization that requires a multidimensional grid search as well as a good initial estimate of the parameter vector. Furthermore, it assumes that the local oscillators (LOs) are unsynchronized, with irregular time delays between the snapshots.

Although this can happens in practical scenarios, it is a quite extreme condition and requires a special hardware to run the associated estimation algorithms. We do not account for this scenario. The subarrays we consider are linked to different LO that are connected in a synchronized fashion to each of the RF chains within the subarray, i.e., for the problem we deal with, the subarrays are perfectly coherent and only the whole array is partially calibrated.

In [18], a new method based on a convex optimization formulation derived from a bilinear arrangement and a convex relaxation procedure was developed. In this work, the authors estimate the phase shifts between the subarrays and then use the new approximately corrected data model to estimate the DOAs using the well-known sparse signal processing formulation presented in [3]. However, this method was developed for a single snapshot scenario and also has a computational complexity prohibitive for most practical online processing scenarios. Moreover, it is not capable of estimating more sources than physical sensors, in general.

Moving on to coherent processing using more sources than the number of physical sensors, many advancements have been made in terms of array design and algorithms. In the next chapter, we discuss some sparse array design techniques that allow us to recover more sources than sensors. By now, we focus on the estimation techniques.

In [19], an approach for quasi-stationary signals was proposed to estimate more sources than sensors, relying on the vectorization operator to build-up a full-rank effective array manifold. This method was modified to deal with stationary signals through the introduction of a subspace-based algorithm known as Spatial Smoothing Multiple Signal Classification (SS-MUSIC) [1,20] that performs spatial smoothing on the vectorized model, in the context of coherent (fully calibrated) arrays. It assumes uncorrelated sources and is used throughout the next chapters in the development of our proposed algorithms for partially calibrated array processing.

We conclude this section by stating that the goal of this work is to fill the gap between the partially-calibrated array processing for DOA estimation and a technique designed specifically to deal with sparse arrays, since with some Khatri-Rao based (vectorization) signal processing derivations, it allows us to estimate more sources than the number of physical sensors and possesses a much larger aperture than uniform linear arrays (ULAs), leading to increased estimation accuracy. Additionally, we derive some results that relate the number of degrees of freedom provided by a special geometrical structure termed type-II sparse linear arrays and the degrees of freedom of its respective subarrays.

1.2 Contributions

In this work, we present a subarray-based design approach, as well as six algorithms to tackle the problem of DOA estimation using partially-calibrated arrays. The subarray-design approach we propose allows us to theoretically predict the number of degrees of freedom (DoF) of the whole array as a function of the DoF of the subarrays (sDoF). Regarding the algorithms, we developed spectrum and polynomial rooting based algorithms that are capable of estimating the DOAs in partially calibrated scenarios and coherent scenarios by using only the second-order statistics of each subarray (non-coherent processing) or the fully coherent array.

The algorithms exhibit low to moderate complexity as they rely solely on eigenvalue or singular value decomposition of covariance matrices, avoiding the need for computationally expensive convex or non-convex optimizations and multidimensional grid-search routines. Additionally, the proposed algorithms enable the estimation of more sources than the number of physical sensors in each subarray or, for some cases, even the fully coherent array. Following the derivation of the algorithms, we investigate the related linear algebra properties and computational complexity. We also present numerical results to support the conclusions drawn in the theoretical sections of this thesis.

Furthermore, we computed Fisher Information Matrices (FIM) corresponding to all parameters of the data model. This computation is an important step in determining the Cramér-Rao Lower Bound (CRLB) for the partially-calibrated data model. The process of calculating these matrices is detailed explicitly, allowing for the quantification of the amount of information each parameter contributes to the estimation of the sources' directions.

1.3 Notation

See List of Symbols on page 20.

1.4 Outline

For what follows, we exhibit the thesis outline:

- In Chapter 2, we provide a comprehensive introduction to the fundamentals of array signal processing, focusing on key principles such as the exploitation of particular models, spatial sampling configurations (uniform and sparse array geometries), and data preprocessing techniques. Additionally, we explore the essential algorithms to perform one of the most important tasks in contemporary engineering systems: DOA estimation.
- In Chapter 3, we explore the development of type-I and type-II geometries, as well as its properties in the context of array signal processing. Furthermore, we also develop a detailed analysis regarding the degrees of freedom of the parent array and its relation with the degrees of freedom of the subarrays. Moreover, we develop two algorithms to tackle the problem of DOA estimation with partially calibrated linear sparse arrays. We start by describing our stochastic system model, as well as defining our estimation problem. Following that, we derive our Generalized Coarray MUSIC (GCA-MUSIC) algorithm and presents the

calculus of its computational complexity. Similarly, after that we also present the derivation of a Generalized Coarray Root-MUSIC that is not only a simple root version of GCA-MUSIC, but introduces a subspace averaging technique that leads us to increased accuracy. We also develop the CRLB associated to the problem at hand.

- In Chapter 4, we address the problem of DOA estimation with partially-calibrated sparse linear arrays by employing coarrays with variable window sizes. Unlike the previous chapter, which focused on combining statistics computed at the subarray level, this chapter emphasizes enhancing the spatial smoothing performance in the coarray domain for each subarray. This approach mitigates one of the core limitations of coarray MUSIC: its fixed subcoarray window length. The use of a variable window length has been shown to yield better results than maintaining a fixed window size.
- Chapter 5 presents the conclusions of this thesis, highlighting its contributions to the field of research and suggesting potential directions for future work.

1.5 Publication list

In what follows, we present five papers that are derived from the results of this work:

- LEITE, W. S.; DE LAMARE, R. C. , **Noncoherent sparse subarrays for DOA estimation based on low-rank and sparse recovery**. 30th European Signal Processing Conference (EUSIPCO), Set 2022. (PUBLISHED)
- LEITE, W. S.; DE LAMARE, R. C., **Partially-Calibrated Sparse Subarrays for Direction-of-Arrival Estimation with Increased Degrees of Freedom**. 56th Annual Asilomar Conference on Signals, Systems, and Computers (ASILOMAR), nov 2022. (PUBLISHED)
- LEITE, W. S.; DE LAMARE, R. C.; ZAKHAROV, Yuriy; LIU, W.; HAARDT, Martin, **Direction Finding with Sparse Subarrays: Design, Algorithms, and Analysis**. IEEE Transactions on Aerospace and Electronic Systems (TAES), jul 2024. (PUBLISHED).
- LEITE, W. S.; DE LAMARE, R. C.; ZAKHAROV, Yuriy; LIU, W.; HAARDT, Martin. **Variable Window Size Spatial Smoothing for Coarray DOA Estimation with Sparse Arrays**. Journal Title (Submitted for Publication).

- LEITE, W. S.; DE LAMARE, R. C.; ZAKHAROV, Yuriy; LIU, W.; HAARDT, Martin. **Sparse Subarrays with Variable Window Sizes.** Conference Proceedings Title (MANUSCRIPT IN PREPARATION).

2
Fundamentals of sensor array signal processing

In this chapter, we establish the fundamentals of sensor array signal processing, along with the core concepts associated with the exploitation of model structures, spatial sampling patterns (array geometries), preprocessing data schemes, and algorithms used to perform one of the most important tasks in modern sensor systems: the so-called DOA estimation.

2.1
Key considerations in array processing

The problem of estimating the directions of arrival of incoming sources is of fundamental importance in various fields, including mobile communications, military applications (electronic warfare), automotive radar, sonar detection, astronomy, and drone surveillance, among others. Accurately determining the DOA of impinging signals can provide critical information about target or source properties, and can also be used to optimize beamforming in communication systems, enhancing data rates for specific users. As technology continues to advance and the availability of microprocessor-based devices integrated with multiple-sensor arrangements increases, these systems are becoming more powerful and cost-efficient, enabling widespread use of DOA estimation in a growing range of applications.

As engineering equipment incorporates an increasing number of sensors, their resolution capabilities improve, enabling tasks that were previously impossible to be executed within few milliseconds or less. The processing power required to run such algorithms has drastically decreased, allowing this technology to be embedded in a wide range of devices, from cutting-edge technological systems to everyday consumer products. This advancement has profoundly changed our understanding of the world, enhancing our ability to communicate and offering a much clearer visualization of the propagation of electromagnetic and acoustic signals in the spatial domain.

In the historical development of signal processing, the primary focus was on time-domain signals and their associated processing techniques. However, in many modern applications, the spatial dimension is increasingly incorporated, leveraging specialized sensor arrays to more effectively sample the spatial field or the space of traveling waves. This enables the estimation of a wide range of parameters from the impinging source signals, such as polarization, velocity, and, most importantly, their directions of arrival, which is the central focus of this thesis.

In the context of *Statistical Signal Processing*, many different approaches can be taken: we can focus on the detection, estimation or modulation theories.

Detection theory addresses the problem of determining the number of sources immersed in noise, answering questions such as: are there sources present? If so, how many are there? Although this information is essential and used in all the algorithms and developments in this thesis, it is not our primary focus. Although specific methods exist to solve this problem and constitute a preliminary step in the DOA estimation task, we assume that the number of sources is known.

Modulation theory, on the other hand, focuses on the challenge of estimating a waveform, a complex task that is also beyond the scope of this thesis, as solving it is not required to estimate the DOAs. Instead, we are concerned with estimating a particular parameter — the DOA — of a waveform immersed in Gaussian noise. More specifically, we focus on a narrowband waveform propagating through an isotropic physical medium.

We highlight that this chapter does not aim to be rigorous or exhaustive, but rather to provide the reader with the foundational concepts on which this thesis is built. It clarifies and standardizes the core aspects of sensor array signal processing, with a particular emphasis on the spatial characteristics of signals.

In order to perform sensor array signal processing, the following aspects must be considered:

a) What is the radiation pattern of each individual sensor in space?

b) What is the geometric configuration of the sensors in space?

c) What are the spatial-temporal characteristics of the source signals and the noise in which they are immersed?

d) What task do I want to perform with the sensors and the incoming signals?

To maintain simplicity, we assume that the radiation pattern of each sensor is *isotropic*, which means that each sensor exhibits the same response in all directions. Although specific radiation patterns can be incorporated into the model with additional processing, this is not considered in our analysis.

Regarding geometric configuration, sensor arrays can be classified as linear (all sensors are collinear), planar (all sensors lie within a single 2D plane), or volumetric (sensors are distributed throughout the 3D space, not confined to a plane). These three primary configurations can be further categorized into:

a) Uniform Arrays (UA): sensors are uniformly spaced, forming lines in 1D, square lattices in 2D, or cubic lattices in 3D space; or

b) Non-Uniform (or Sparse) Arrays (NA): the spacing between neighboring sensors is arbitrary.

Moving on to signal models, we adopt a spatio-temporal structure that consists of random source signals, with waves arriving from unknown DOAs. Although the source signals are temporally random, they follow a known family of probability density functions (PDFs) with unknown parameters. Specifically, we assume that the signals belong to the complex multivariate Gaussian distribution. Regarding the noise, we assume that it is temporally and spatially white, which means that it lacks temporal structure and is uniformly present at all sensors. This represents the measurement or intrinsic noise.

The task for the signal processor is to estimate the DOAs of multiple source signals in the presence of measurement noise. Naturally, the processor will have performance metrics associated with this task. Some of these performance metrics are linked to:

a) What is the maximum number of sources the array processing system can handle?

b) How accurate is the array processing system in performing the estimation task?

c) How robust is the system when dealing with modeling imperfections?

d) How flexible is the system in handling array geometries that are not fully calibrated (i.e., non-ideal or partially coherent geometries)?

e) How robust is the system when some sensors fail or are lost?

All of the aforementioned tasks are crucial for the system designer. We emphasize that addressing these tasks is directly tied to the geometry chosen during the array design process and the algorithm implemented in the signal processor. These two aspects are often interdependent in many applications, as will be discussed further in this work. Specifically, apart from handling modeling imperfections and sensor failures, we address all other mentioned aspects with our proposed array design techniques and algorithms.

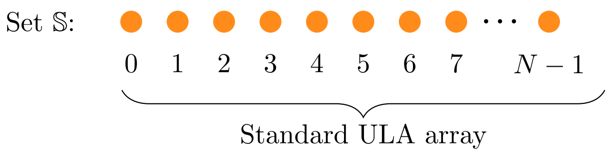


Figure 2.1: Standard ULA with sensors placed in normalized positions $\mathbb{S} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

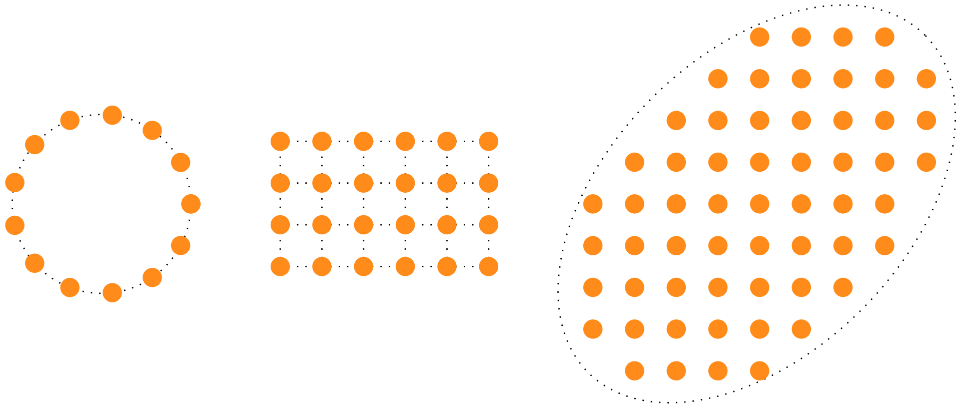


Figure 2.2: Examples of UPAs: Uniform Circular Arrays (UCAs), Uniform Rectangular Arrays (URAs), and Uniform Ellipsoidal Arrays (AEAs).

2.2 Array geometries

In the context of array geometries, Uniform Arrays (UAs) have sensors that are uniformly spaced, as illustrated in Figure 2.1 for the linear case (Uniform Linear Array - ULA). Uniform Planar Arrays (UPAs) and Uniform Volumetric Arrays (UVAs) configurations are defined similarly, as shown in Figure 2.2.

On the other hand, Non-Uniform Linear Arrays (NLAs), also called Sparse Linear Arrays (SLAs), Non-Uniform Planar Arrays (NPAs) and Non-Uniform Volumetric Arrays (NVAs) can be defined for one, two and three-dimensional configurations, respectively, when the spacing between sensors is not uniform.

Regarding the spatial-temporal characteristics of the source signals and noise, we can categorize the information with respect to temporal and spatial structures as follows:

– Temporal structure:

  a) Known source signals, meaning the signal is perfectly known over time;
  b) Source signals with a deterministic parametric model that is known, but depending on unknown parameters;
  c) Source signals with a structure that is completely known; and
  d) Source signals without any temporal structure and random.

- Spatial structure:

  a) Waves impinging from known DoAs;
  b) Waves from unknown DoAs; and
  c) Spatially-spread waves.

In this work, we focus on SLAs or NLAs, where the sensors are spaced by integer multiples of the minimum intersensor distance. This represents a more restricted class of NLAs, as it imposes an additional structure on the sensor array geometry.

Sparse arrays have drawn significant attention from the community as a result of their ability to achieve high-resolution spatial sensing with reduced hardware complexity. Unlike traditional ULAs, which rely on uniformly spaced sensors, sparse arrays strategically place their elements with variable inter-element spacing, as discussed in Section 2.1.

The strategic physical placement of sensors in sparse arrays not only improves resolution performance, particularly in terms of reducing the root mean square error (RMSE), but also significantly optimizes cost-efficiency, system complexity, and the available degrees of freedom (DoF). The DoFs, which refer to the number of independent source signal parameters that can be estimated, are crucial to improving the system's capacity to resolve multiple sources in challenging environments.

This section introduces the fundamental principles of sparse array design, providing a basis for understanding their advantages in sensor array signal processing. In particular, we will explore their applications to improve statistical estimation performance, with a primary focus on DoA estimation.

Sensor array signal processing has been widely utilized in recent decades for its ability to detect, localize, and estimate the parameters of multiple signal sources by leveraging the spatial diversity offered by an array of sensors. Traditional uniform (non-sparse) array geometries often require a large number of sensors to achieve the desired resolution and accuracy, which increases system complexity and cost. In contrast, sparse arrays offer a more efficient approach, delivering similar or even superior performance with fewer sensors.

At the core of sparse array theory is the concept of spatial sparsity, which capitalizes on the fact that many signal environments are inherently sparse. This sparsity implies that, at any given time, only a few signal sources are active within the field of view of the sensor array. By intelligently designing the sensor array geometry, sparse arrays can effectively sample the spatial domain[1], allowing for accurate signal reconstruction and parameter estimation while significantly reducing the number of required sensors.

The development of sparse arrays involves a detailed study of mathematical models and signal processing techniques. Methods such as compressive sensing, subspace decomposition, beamforming, array interpolation, and grid-based or gridless approaches are essential to their design and implementation. The performance of sparse arrays is closely tied to their geometry, which must be optimized to balance resolution, reduce sidelobe levels, and maintain robustness against noise and interference. In this context, we also examine the signal processing techniques, as certain algorithms can benefit from specific array geometries. In fact, some signal processing algorithms perform better with sparse arrays of particular structures.

In this section, we explore the foundational concepts of sparse sensor arrays, starting with an overview of traditional sensor array theory and its inherent limitations. We then examine the theoretical principles underlying sparse arrays, focusing on key design strategies and the mathematical models that describe their behavior. Finally, we discuss various optimization techniques used in the design of sparse sensor arrays, emphasizing their impact on practical applications.

The most standard and intuitive array design involves constructing a uniform grid, with sensors (filled orange circles) placed at each grid position, as shown in Figure 2.1. In this case, the sensors are physically positioned at intervals of $0, d, 2d, 3d, \ldots, 9d$, where d represents the minimum spacing between sensors and we take the first sensor as the reference. To simplify, we normalize all positions by d. The resulting set of normalized positions defines the array geometry, denoted as $\mathbb{S}$. Note that we focus on arrays where the sensor positions are integer multiples of d, which represent a specific type of array geometry.

The array aperture, N_a, represents the array length in terms of the minimum intersensor spacings. Specifically, it is defined as the maximum absolute value among all the elements in $\mathbb{S}$, such that:

$$N_a \triangleq \max_i n_i \quad \forall n_i \in \mathbb{S} \tag{2-1}$$

[1] We say that the array sample the space, in contrast to time sampling, which is performed at the sensor level.

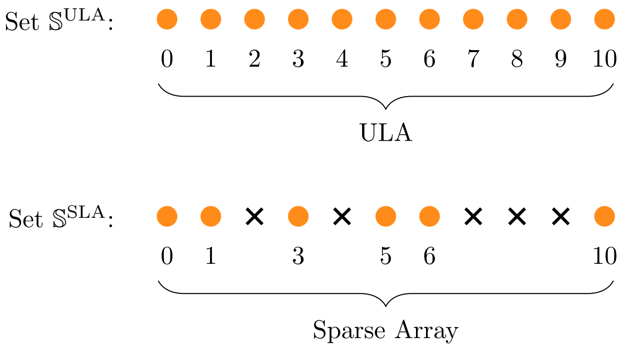

Set $\mathbb{S}^{\mathrm{ULA}}$:
0 1 2 3 4 5 6 7 8 9 10
ULA
Set $\mathbb{S}^{\mathrm{SLA}}$:
0 1 3 5 6 10
Sparse Array

Figure 2.3: Sparse linear array generated by means of the removal of sensors from a ULA with the same aperture. In this case, we say that the sparse array is a ULA with "missing" sensors at positions $\mathbb{S}^{\mathrm{missing}} = \{2, 4, 7, 8, 9\}$.

Sparse Linear Arrays (SLAs), also known as Non-Uniform Linear Arrays (NLAs), can be constructed from two different perspectives:

a) By removing certain sensors from a ULA with the same aperture; or

b) By redistributing the sensors of a ULA with a reduced aperture, while maintaining the same number of physical sensors as the sparse array.

For example, consider the configuration shown in Figure 2.3. The crosses in the sparse array indicate positions where sensors are absent, illustrating the structure of an SLA. This configuration is derived from a ULA with $N = 11$ sensors (aperture $N_a = 10$) by removing the sensors at positions $\{2, 4, 7, 8, 9\}$.

Alternatively, as shown in Figure 2.4, the SLA can be interpreted as a redistribution of the sensors from a ULA with $N = 6$ elements onto a grid with the same minimum intersensor spacing, as described previously in alinea b). This approach can be viewed as an expansion of a ULA with a smaller aperture ($N_a = 5$), following specific design rules or criteria to achieve desired characteristics in the array response, which is a key aspect of sparse sensor array design.

Before we introduce some specific sparse sensor array geometries, let us establish some core definitions in sparse sensor array design theory, starting by the *difference coarray set*, as follows:

Definição 2.1 *(Difference Coarray Set - $\mathbb{D}$) The difference coarray indicated by the set $\mathbb{D}$ is a set associated to the sensors' positions $\mathbb{S}$ through the following relation*

$$\mathbb{D} \triangleq \left\{ n_1 - n_2 \mid (n_1, n_2) \in \mathbb{S}^2 \right\} \tag{2-2}$$

arranged in ascending order and without any repetition of elements.

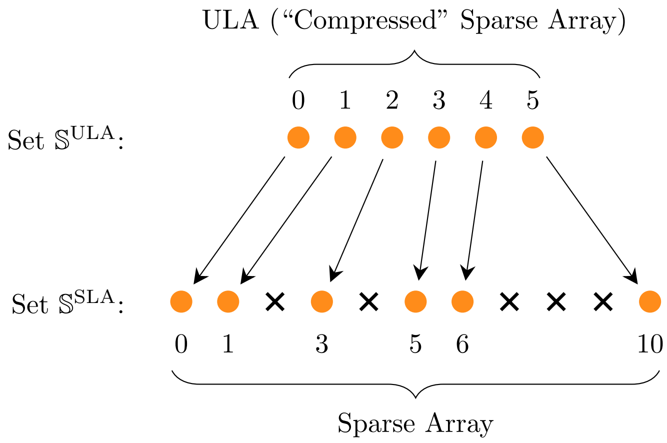

ULA ("Compressed" Sparse Array)
0 1 2 3 4 5
Set $\mathbb{S}^{\mathrm{ULA}}$:
Set $\mathbb{S}^{\mathrm{SLA}}$:
0 1 3 5 6 10
Sparse Array

Figure 2.4: Sparse linear array generated by means of a redistribution of ULA sensors increasing the spacing (inserting "holes") between elements. In this case, the ULA can be seen as a "compressed" sparse array.

Next, we define the concept of degrees of freedom.

Definição 2.2 *(Degrees of Freedom -* DoF*) The number of Degrees of Freedom, denoted by DoF, of an array geometry specified by* $\mathbb{S}$ *is the cardinality of its difference coarray set. Then,*

$$\mathrm{DoF} \triangleq |\mathbb{D}| \tag{2-3}$$

In what follows, we introduce the concept of the central difference coarray consecutive set, a subset of $\mathbb{D}$, which is crucial for the upcoming discussion on spatial smoothing algorithms.

Definição 2.3 *(Central Difference Coarray Consecutive Set -* $\mathbb{U}$*) The central difference coarray consecutive set, denoted by* $\mathbb{U}$*, associated with a difference coarray set* $\mathbb{D}$*, is the set*

$$\mathbb{U} = \{0\} \cup \{\pm [m]\} \tag{2-4}$$

arranged in ascending order and without any repetition of elements, such that m *is chosen to maximize the cardinality of* $\mathbb{U}$ *with* $\mathbb{U} \subseteq \mathbb{D}$*, i.e., it corresponds to the central contiguous part of* $\mathbb{D}$*. The symbol* $[m]$ *means the set* $\{1, 2, \ldots, m\}$*.*

As with the difference coarray set, the cardinality of $\mathbb{U}$ is also of significant interest.

Definição 2.4 *(Uniform Degrees of Freedom -* UDoF*) The number of Uniform* DoF*, denoted by* UDoF*, of a geometry specified by* $\mathbb{S}$*, is the cardinality of its central difference coarray consecutive set* ($\mathbb{U}$)*, i.e.,*

$$\mathrm{UDoF} \triangleq |\mathbb{U}| \tag{2-5}$$

Lastly, we state also the formal definition of the so-called *weight function*, presented in Definition 2.5. It is a core definition for further discussions.

Definição 2.5 (Weight Function - $w(m)$) *The weight function is an application*

$$w : \mathbb{Z} \to \{0\} \cup \{\pm [N]\}$$

such that

$$w(m) = \begin{cases} |\mathbb{M}(m)| & \text{if } m \in \mathbb{D} \\ 0 & \text{otherwise} \end{cases} \tag{2-6}$$

where $\mathbb{M}(m) = \{(n_1, n_2) \in \mathbb{S}^2 \mid n_1 - n_2 = m\}$*, i.e.,* $\mathbb{M}(m)$ *is a function that counts the number of sensor pairs with separation m (covariance lag).*

This function is crucial for evaluating of the effects of electromagnetic coupling between sensors and serves as an indicator of the degree of array redundancy [21].

Next, we introduce some specific sparse array geometries. Let us start by the well known Minimum Redundancy Arrays (MRA) [22]. They are designed to minimize the number of pairs of physical sensors with the same position difference, under the restriction that the associated difference coarray set is *hole-free*, which means that there are no missing lags in its covariance matrix.

Since many DOA estimation algorithms are based on the signal covariance matrix, MRA structures can be useful to them because of their ability to estimate second-order statistics with a minimum number of physical sensors. However, one of the drawbacks of MRA is the lack of a simple formation rule for the set $\mathbb{S}$. This set is defined through exhaustive search routines and consists of an intricate combinatorial optimization problem that increases its complexity with the number of physical sensors. For MRAs, it can be shown that [10]:

$$N_a = \frac{N(N-1) - N_R + N_H}{2} \tag{2-7}$$

where N_a, N_R and N_H correspond to the aperture length, number of redundancies (undesired feature) and number of holes[2]. It is required that $N_H = 0$ and N_a is maximized in MRA design. Some of the known MRAs presented in [10] are exhibited in Table 2.1.

Before proceeding, it is important to introduce two additional sparse linear geometries, as they will be referenced later in the text: the minimum hole array (MHA) and the two-level nested array (NAQ2). The so-called MHA, non-redundant arrays or Golomb rulers, are designed in such a way that the number of missing spatial covariance lags is minimized with the restriction

[2] The number of redundancies and holes is computed considering both sides of the weight function, which will be defined later on. The original approach in [10] considers only positive coarray locations.

Table 2.1: Minimum Redundancy Arrays

N	N_R	N_a	$\mathbb{S}$
3	0	3	$\{0, 1, 3\}$
4	0	6	$\{0, 1, 4, 6\}$
5	2	9	$\{0, 1, 4, 7, 9\}$
6	4	13	$\{0, 1, 4, 5, 11, 13\}$
7	8	17	$\{0, 1, 4, 10, 12, 15, 17\}$
8	10	23	$\{0, 1, 4, 10, 16, 18, 21, 23\}$
9	14	29	$\{0, 1, 4, 10, 16, 22, 24, 27, 29\}$
10	18	36	$\{0, 1, 3, 6, 13, 20, 27, 31, 35, 36\}$
11	24	43	$\{0, 1, 3, 6, 13, 20, 27, 34, 38, 42, 43\}$
12	32	50	$\{0, 1, 3, 6, 13, 20, 27, 34, 41, 45, 49, 50\}$
13	40	58	$\{0, 1, 2, 3, 27, 32, 36, 40, 44, 48, 52, 55, 58\}$

Table 2.2: Minimum Hole Arrays

N	N_H	N_a	$\mathbb{S}$
2	0	1	$\{0, 1\}$
3	0	3	$\{0, 1, 3\}$
4	0	6	$\{0, 1, 4, 6\}$
5	2	11	$\{0, 1, 4, 9, 11\}$
6	4	17	$\{0, 1, 4, 10, 12, 17\}$
7	8	25	$\{0, 1, 4, 10, 18, 23, 25\}$
8	12	34	$\{0, 1, 4, 9, 15, 22, 32, 34\}$
9	16	44	$\{0, 1, 5, 12, 25, 27, 35, 41, 44\}$
10	20	55	$\{0, 1, 6, 10, 23, 26, 34, 41, 53, 55\}$

that there are no redundancies in its covariance matrix, except for the spatial lag of zero [23]. That said, we minimize N_H subject to $N_R = 0$. This sensor array forces a non-redundant acquisition system and generates a large aperture array compared to the other geometries discussed in this text [24].

These geometries are obtained from tabulated entries, as in the case of MRA. Some MHA described in [10] can be seen in Table 2.2. One of the drawbacks of this kind of structure is the nonexistence of closed-form solutions for the design of $\mathbb{S}$ [23, 25].

The so-called n th-level nested array (NAQn) was first introduced in [20]. The discussion is restricted here to the two-level nested array (NAQ2). The formation rule (positioning procedure) described by $\mathbb{S}$ is given by (2-8). The total number of sensors is $N = N_1 + N_2$. It is straightforward to notice that NAQ2 arrangements consist of two ULA segments. The first one with N_1 sensors and unitary spacing and the second with N_2 sensors and spacing of magnitude $N_1 + 1$.

$$\mathbb{S} = \{n,\ n = 0, 1, \ldots, N_1 - 1\} \cup \{n(N_1 + 1) - 1,\ n = 1, 2, \ldots, N_2\} \tag{2-8}$$

That said, the first ULA is a dense structure, and the second one is a sparse one. Two of its fundamental characteristics are commonly highlighted:

i) Filled diffrence coarray (hole free), i.e., $\mathbb{D} = \mathbb{U}$;
ii) The DoF/UDoF are obtained through (2-9). From that, notice that there are $\mathcal{O}(N_1 N_2)$ available DoF for arrays with only $N_1 + N_2$ sensors. The fact that NAQ2 structures have no holes in its coarray is a desired property for covariance matrix based algorithms, especially when spatial smoothing techniques are employed.

$$\text{DoF} = \text{UDoF} = 2N_2(N_1 + 1) - 1 \tag{2-9}$$

Other recently reported array geometries include the maximum inter-element spacing (MISC) criterion [26–28], which offer high degrees of freedom and excellent performance in the presence of mutual coupling.

2.3 Direction of arrival estimation techniques

In this section, we briefly discuss several key approaches to DoA estimation, grouping them into the following categories: beamforming, maximum-likelihood, optimization-based, compressive sensing, and subspace methods. Before we go into each family of algorithms, we give an overview in the following alineas:

a) **Beamforming** methods are among the earliest techniques, relying on the constructive and destructive interference of signals to estimate source directions through a process known as beam steering.

b) **Maximum-likelihood** methods provide statistically optimal estimates by maximizing the likelihood function, often requiring multidimensional grid searches. Although computationally intensive when the number of sources exceeds $D = 3$, these methods serve as a benchmark due to their efficiency and ability to perform with as few as one snapshot.

c) **Compressive sensing** algorithms exploit the sparsity of source signals on a predefined grid to recover DoA information using techniques such as fixed-point equations, l_1-norm minimization, and residue-measurement vector correlations. They are well-suited to challenging scenarios with few snapshots and high signal-to-noise ratio (SNR) environments.

d) **Subspace-based** methods are widely used due to their asymptotic efficiency and the ability to achieve super-resolution. These algorithms offer a reasonable trade-off between accuracy and computational complexity, often relying on eigenvalue or singular value decompositions and one-dimensional searches.

In the following subsections, we delve deeper into each family of algorithms.

2.3.1 Signal model

In what follows, we establish our data acquisition model for fully calibrated arrays[3]:

$$\mathbf{x}_{\mathbb{S}}(t) = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t) \tag{2-10}$$

where $t \in [T]$, $[T]$ is the set $\{1, 2, \ldots, T\}$, $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}) \in \mathbb{C}^{N\times D}$ is the array manifold, N is the number of physical sensors, D is the number of impinging sources with normalized directions given by $\boldsymbol{\theta} \in [-1, 1)^D$ (sine of DOAs), $\mathbf{s}(t) \in \mathbb{C}^D$ is the source signal at time instant t, $\mathbf{x}_{\mathbb{S}}(t) \in \mathbb{C}^N$ is the array received signal snapshot at time instant t, and $\mathbf{n}_{\mathbb{S}}(t) \in \mathbb{C}^N$ is the noise vector measurement. The noise and the source signals are drawn from a circularly complex multivariate Gaussian distribution. The noise is spatially and temporally white and the sources are uncorrelated. Sources and noise are uncorrelated between themselves.

All the techniques we use in this section are based on the exploitation of the second-order statistics associated with the random quantities defined in (2-10). The covariance matrix of $\mathbf{x}(t)$ is written as

$$\begin{aligned}\mathbf{R}_{\mathbb{S}} &= \mathrm{E}\{\mathbf{x}_{\mathbb{S}}(t)\mathbf{x}_{\mathbb{S}}(t)^H\}\\ &= \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathrm{E}\{\mathbf{s}(t)\mathbf{s}^H(t)\}\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta}) + \mathrm{E}\{\mathbf{n}_{\mathbb{S}}(t)\mathbf{n}_{\mathbb{S}}(t)^H\}\\ &= \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I}\end{aligned} \tag{2-11}$$

where $\mathbf{R}_s \in \mathbb{C}^{D\times D}$ is the source covariance matrix and σ_n^2 is the noise variance at each sensor.

2.3.2 Beamforming methods

Beamforming algorithms [10, 29–31] are primarily used to perform waveform estimation. They are also used to detect waveforms incoming into the sensor array, not to mention their capabilities to estimate the spatial spectrum of the space-time process. However, here we use them to find the directions of incoming waves, which is also one of their key applications.

[3] We leave the discussion of the partially-calibrated array scenario to Chapter 3.

They are based on the core idea that when the electronic array steering points at the directions of the incoming wave, the power received by the array achieves a maximum. Then, they spatially filter the data to determine the directions of the incoming waves. The beamforming steering is done through the adjusting of the beamforming weights or taps, which consists of complex scalars associated to the output of each sensor.

The beamforming output can be written as

$$y(t) = \mathbf{w}^H \mathbf{x}_{\mathbb{S}}(t) \tag{2-12}$$

i.e., a weighted sum of the sensor outputs. The beamforming output power is defined as:

$$\begin{aligned} P(\mathbf{w}, \theta) &= \mathrm{E}\{|y(t)|^2\} \\ &= \mathrm{E}\{y(t)y^*(t)\} \\ &= \mathbf{w}^H \mathrm{E}\{\mathbf{x}_{\mathbb{S}}(t)\mathbf{x}_{\mathbb{S}}^H(t)\}\mathbf{w} \\ &= \mathbf{w}^H \mathbf{R}_{\mathbb{S}} \mathbf{w} \end{aligned} \tag{2-13}$$

i.e., the beamforming output power depends only on the weights vector and the signal covariance matrix. Each of the beamforming techniques has a different method to define the weight vector $\mathbf{w}$. Next, we present two methods: the conventional Bartlett and Capon beamforming techniques.

The conventional beamformer solve the following optimization problem

$$\begin{aligned} \underset{\mathbf{w}}{\text{maximize}} \quad & \mathbf{w}^H \mathbf{R}_{\mathbb{S}} \mathbf{w} \\ \text{subject to} \quad & |\mathbf{w}| = 1 \end{aligned} \tag{2-14}$$

whose solution (optimum point) is

$$\mathbf{w}_{\mathrm{bl}} = \frac{\mathbf{a}_{\mathbb{S}}(\theta)}{\sqrt{\mathbf{a}_{\mathbb{S}}^H(\theta)\mathbf{a}_{\mathbb{S}}(\theta)}} \tag{2-15}$$

where $\mathbf{a}_{\mathbb{S}}(\theta)$ is the array steering vector. The optimum value of this problem is given by

$$P_{\mathrm{bl}}(\theta) = \frac{\mathbf{a}_{\mathbb{S}}^H(\theta)\mathbf{R}_{\mathbb{S}}\mathbf{a}_{\mathbb{S}}(\theta)}{\mathbf{a}_{\mathbb{S}}^H(\theta)\mathbf{a}_{\mathbb{S}}(\theta)} \tag{2-16}$$

In this case, the output power depends only on the DoA value. By performing a 1-D search over all possible values of θ and finding the maximum, we can determine the DoA of the impinging signal. The algorithm is summarized in Algorithm 2.1.

Another very popular beamforming technique is called Capon

Algorithm 2.1: Bartlett beamformer

```
Input  : Array geometry $\mathbb{S}$
         Data matrix $\hat{\mathbf{X}}_{\mathbb{S}}$
         Search grid $\boldsymbol{\theta}^g$
         Number of sources $D$
$\hat{\mathbf{R}}_{\mathbb{S}} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}}\hat{\mathbf{X}}_{\mathbb{S}}^H$
for $i \leftarrow 1$ to $g$ do
    $P_{\mathrm{bl}}(\theta_i) \leftarrow \left(\mathbf{a}_{\mathbb{S}}^H(\theta_i)\hat{\mathbf{R}}_{\mathbb{S}}\mathbf{a}_{\mathbb{S}}(\theta_i)\right) / \left(\mathbf{a}_{\mathbb{S}}^H(\theta_i)\mathbf{a}_{\mathbb{S}}(\theta_i)\right)$

Find the $D$ largest peaks of $P_{\mathrm{bl}}(\theta)$
Retain the corresponding DoAs of the peaks as the DoAs estimates
  $\hat{\boldsymbol{\theta}}$
Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

beamfoming. It solves the following problem

$$\begin{aligned} \underset{\mathbf{w}}{\text{maximize}} \quad & \mathbf{w}^H\mathbf{R}_{\mathbb{S}}\mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^H\mathbf{a}_{\mathbb{S}}(\theta) = 1 \end{aligned} \tag{2-17}$$

It is also known as the *Minimum Variance Distortionless Response (MVDR)* filter and attempts to minimize the noise and any signals power coming from directions different from θ, keeping the looking direction with unitary gain. By resorting to Lagrange multipliers, we can solve this optimization problem and write the solution (optimum point) given by

$$\mathbf{w}_{\mathrm{cp}} = \frac{\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{a}_{\mathbb{S}}(\theta)}{\mathbf{a}_{\mathbb{S}}^H(\theta)\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{a}_{\mathbb{S}}(\theta)} \tag{2-18}$$

By plugging (2-18) into the objective function of (2-17), we find that the optimum value of the Capon beamforming problem is

$$P_{\mathrm{cp}}(\theta) = \frac{1}{\mathbf{a}_{\mathbb{S}}^H(\theta)\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{a}(\theta)} \tag{2-19}$$

Again, it suffices to evaluate (2-19) over all possible values of θ and choose the local maxima as the DoAs, i.e., it suffices to perform a 1-D search. The algorithm is summarized in Algorithm 2.2.

One of the biggest limitations of using beamforming to find the DoAs is its insufficient resolution when the sources are close (less than one beamwidth apart) [10], no matter how many snapshots or how favorable the SNR may be.

To improve beamforming resolution, the number of sensors must be increased, which is a significant limitation of this method. However, due to its simplicity and low computational complexity, it remains suitable for

```
Algorithm 2.2: Capon beamformer
    Input  : Array geometry $\mathbb{S}$
             Data matrix $\hat{\mathbf{X}}_{\mathbb{S}}$
             Search grid $\boldsymbol{\theta}^g$
             Number of sources $D$
    $\hat{\mathbf{R}}_{\mathbb{S}} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}}\hat{\mathbf{X}}_{\mathbb{S}}^H$
    for $i \leftarrow 1$ to $g$ do
        $P_{\text{cp}}(\theta_i) \leftarrow 1/(\mathbf{a}_{\mathbb{S}}^H(\theta_i)\hat{\mathbf{R}}_{\mathbb{S}}^{-1}\mathbf{a}(\theta_i))$

    Find the $D$ largest peaks of $P_{\text{cp}}(\theta)$
    Retain the corresponding DoAs of the peaks as the DoAs estimates
      $\hat{\boldsymbol{\theta}}$
    Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

applications with lower performance requirements.

2.3.3 Maximum likelihood estimation

The Maximum Likelihood Estimator (MLE) is widely adopted for benchmarking purposes. It is an unbiased estimator under specific assumptions, meaning that the expected value of the parameter vector estimate equals its true value.

In scenarios with infinite data availability, the MLE achieves the best possible performance in terms of root mean square error (RMSE), namely the square root of the corresponding Cramér-Rao Lower Bound (CRLB). Even with limited data, it demonstrates satisfactory performance. However, one of its major drawbacks is the extremely high computational complexity, as it requires a D-dimensional search during the optimization process [7, 10].

There are two main versions of the algorithm, depending on the signal model used: deterministic or stochastic sources. Since our source signals are random, we use the Stochastic Maximum Likelihood Estimator. Proceeding to the algorithm description, we can express the single-snapshot probability density function (pdf) as follows [10]:

$$p_{\mathbf{x}_{\mathbb{S}}(t)|\mathbf{u}}\left(\dot{\mathbf{x}}_{\mathbb{S}}(t)\right) = \frac{1}{|\pi \mathbf{R}_{\mathbb{S}}|} \exp\left\{-\dot{\mathbf{x}}_{\mathbb{S}}^H(t)\mathbf{R}_{\mathbb{S}}^{\mathbf{-1}}\dot{\mathbf{x}}_{\mathbb{S}}(t)\right\} \tag{2-20}$$

where $\mathbf{x}_{\mathbb{S}}(t) \in \mathbb{C}^N$ is the array received signal (a random variable), and $\dot{\mathbf{x}}_{\mathbb{S}}(t) \in \mathbb{C}^N$ is its realization (the *dotted version*, i.e., a snapshot taken at time t). The vector $\mathbf{u} = \begin{pmatrix}\boldsymbol{\theta}^T & \mathbf{p}^T\end{pmatrix}^T \in \mathbb{R}^{D\times 2}$ represents the nonrandom unknown parameter vector to be estimated, which includes the true source directions and powers.

In this scenario, we assume the source powers are also unknown. Note that the received signal covariance matrix in (2-11) depends on the DoA angles $\boldsymbol{\theta}$ and the source covariance matrix $\mathbf{R}_s$, leading to a total of $2D$ parameters that must be estimated, under the assumption that the noise power σ_n^2 is known *a priori*. For the Multiple Measurements Vector (MMV) model, we assume i.i.d snapshots. Then, the joint PDF can be written as

$$p_{\mathbf{x}_{\mathbb{S}}(1),\ldots,\mathbf{x}_{\mathbb{S}}(T)|\mathbf{u}}\left(\dot{\mathbf{x}}_{\mathbb{S}}\right) = \prod_{t=1}^{T} \frac{1}{|\pi \mathbf{R}_{\mathbb{S}}|} \exp\left\{-\dot{\mathbf{x}}_{\mathbb{S}}^{H}(t)\mathbf{R}_{\mathbb{S}}^{-\mathbf{1}}\dot{\mathbf{x}}_{\mathbb{S}}(t)\right\} \tag{2-21}$$

which results in the log-likelihood function

$$\begin{aligned}\mathcal{L}_{\mathbf{x}_{\mathbb{S}}}(\mathbf{u}) &= \ln\left(p_{\mathbf{x}_{\mathbb{S}}(1),\ldots,\mathbf{x}_{\mathbb{S}}(T)|\mathbf{u}}\left(\dot{\mathbf{x}}_{\mathbb{S}}\right)\right) \\ &= -T\ln\left(|\mathbf{R}_{\mathbb{S}}|\right) - \sum_{t=1}^{T} \dot{\mathbf{x}}_{\mathbb{S}}^{H}(t)\mathbf{R}_{\mathbb{S}}^{-1}\dot{\mathbf{x}}_{\mathbb{S}}(t) - TN\ln\left(\pi\right)\end{aligned} \tag{2-22}$$

If we pre-multiply the expression in (2-22) by the positive constant $1/T$, omit the constant term $TN\ln(\pi)$ (as it does not affect the optimal point or value of the function) and apply the cyclic property of the trace, we obtain

$$\begin{aligned}\mathcal{L}_{\mathbf{x}_{\mathbb{S}}}(\mathbf{u}) &= -\ln\left(|\mathbf{R}_{\mathbb{S}}|\right) - \frac{1}{T}\sum_{t=1}^{T} \dot{\mathbf{x}}^{H}(t)\mathbf{R}_{\mathbb{S}}^{-1}\dot{\mathbf{x}}(t) \\ &= -\left[\ln\left(|\mathbf{R}_{\mathbb{S}}|\right) + \operatorname{tr}\left(\frac{1}{T}\sum_{t=1}^{T} \dot{\mathbf{x}}^{H}(t)\mathbf{R}_{\mathbb{S}}^{-1}\dot{\mathbf{x}}(t)\right)\right] \\ &= -\left[\ln\left(|\mathbf{R}_{\mathbb{S}}|\right) + \operatorname{tr}\left(\mathbf{R}_{\mathbb{S}}^{-1}\left(\frac{1}{T}\sum_{t=1}^{T} \dot{\mathbf{x}}(t)\dot{\mathbf{x}}^{H}(t)\right)\right)\right] \\ &= -\left[\ln\left(|\mathbf{R}_{\mathbb{S}}|\right) + \operatorname{tr}\left(\mathbf{R}_{\mathbb{S}}^{-1}\hat{\mathbf{R}}_{\mathbb{S}}\right)\right]\end{aligned} \tag{2-23}$$

where

$$\hat{\mathbf{R}}_{\mathbb{S}} = \frac{1}{T}\sum_{t=1}^{T} \dot{\mathbf{x}}(t)\dot{\mathbf{x}}^{H}(t) \tag{2-24}$$

is the sample covariance estimator of $\mathbf{R}_{\mathbb{S}}$. It coincides with the MLE estimator for the covariance matrix of Gaussian random vectors.

The MLE for $\mathbf{u}$, the parameter vector of interest, is the value of $\mathbf{u}$ that maximizes (2-23). To solve this optimization problem, since $\mathbf{u}$ is a partitioned vector (partitions $\boldsymbol{\theta}$ and $\mathbf{p}$) and the problem formulation allows it, the first step is to maximize over $\mathbf{p}$ (or $\mathbf{R}_s = \operatorname{diag}(\mathbf{p})$), in such a way we obtain an explicit function of $\boldsymbol{\theta}$. After that, we maximize over the remaining D variables (directions), according to [7,10].

The MLE for the source covariance matrix $\mathbf{R}_s$, which is denoted as $\mathbf{R}_{s,\mathrm{ML}}$, is given by

$$\mathbf{R}_{\mathbf{s},\mathrm{ML}} = \mathbf{A}_{\mathbb{S}}^{\dagger}(\boldsymbol{\theta}) \left(\hat{\mathbf{R}}_{\mathbb{S}} - \sigma_n^2 \mathbf{I}\right) \left[\mathbf{A}_{\mathbb{S}}^{\dagger}(\boldsymbol{\theta})\right]^H \tag{2-25}$$

where $\mathbf{A}_{\mathbb{S}}^{\dagger}(\boldsymbol{\theta}) = \left(\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\right)^{-1} \mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})$ is the Moore-Penrose pseudoinverse associated with the matrix $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})$. This assumes that $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})$ is of full column rank (i.e., more physical sensors than DoAs, and no ambiguities are present). Using (2-11), we find the ML estimate for $\mathbf{R}_{\mathbb{S}}$, which is given by

$$\begin{aligned} \mathbf{R}_{\mathbb{S},\mathrm{ML}} &= \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{R}_{\mathbf{s},\mathrm{ML}}\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I} \\ &= \mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})} \left(\hat{\mathbf{R}}_{\mathbb{S}} - \sigma_n^2 \mathbf{I}\right) \mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})} + \sigma_n^2 \mathbf{I} \end{aligned} \tag{2-26}$$

where $\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})} = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}) \left(\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\right)^{-1} \mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})$ is the orthogonal projection matrix onto the subspace spanned by the columns of $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})$. Substituting (2-26) into (2-23) and manipulating the expression to obtain a more tractable form of the final optimization problem results in

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\mathrm{AML}} =& \underset{\boldsymbol{\theta}}{\mathrm{argmax}} \Big\{ -\ln\left[\det\left(\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})}\hat{\mathbf{R}}_{\mathbb{S}}\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})} + \sigma_n^2 \mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})}^T\right)\right] - \\ & \mathrm{tr}\left(\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})}^T \hat{\mathbf{R}}_{\mathbb{S}}\right) / \sigma_n^2 \Big\} \end{aligned} \tag{2-27}$$

Here, $\det(\cdot)$ denotes the determinant operator, $\ln(\cdot)$ represents the natural logarithm, AML refers to *Asymptotic Maximum Likelihood*, and $\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})}^T = \mathbf{I} - \mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})}$ is the orthogonal projection matrix onto the null space of $\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})$. Given that one of the model assumptions is that the noise variance σ_n^2 is known, in the practical scenario where this quantity is unavailable, it can be consistently estimated using the method outlined in [10] as

$$\sigma_{n,\mathrm{ML}}^2 = \frac{\mathrm{tr}\left(\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})}^T \hat{\mathbf{R}}_{\mathbb{S}}\right)}{N - D}, \tag{2-28}$$

which allows us to estimate the final directions as

$$\hat{\boldsymbol{\theta}}_{\mathrm{AML}} = \underset{\boldsymbol{\theta}}{\mathrm{argmin}} \left\{ \det\left(\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})}\hat{\mathbf{R}}_{\mathbb{S}}\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})} + \frac{\mathrm{tr}\left(\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})}^T \hat{\mathbf{R}}_{\mathbb{S}}\right)\mathbf{P}_{\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})}^T}{N - D} \right) \right\} \tag{2-29}$$

Therefore, the DoAs $\boldsymbol{\theta}$ can be estimated by optimizing (2-29), by using the array manifold matrix $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})$, the number of sensors N, the number of sources D, and the sample received signal covariance matrix, denoted as $\hat{\mathbf{R}}_{\mathbb{S}}$, derived in (2-24). The D-dimensional optimization required to determine the solution of (2-29) imposes a substantial computational burden, despite the method's excellent accuracy. This becomes particularly important in real-time engineering applications, where the extended computation time would severely impact system responsiveness.

Finally, the objective function in (2-29) is highly non-convex, exhibiting multiple local minima. This prevents the use of convex optimization techniques such as interior-point methods for solving the problem. In practice, a grid of dimensions $D \times g$ candidates is constructed to evaluate the function, where g is the number of discretization points in the field of view Θ. After identifying the global minimum by exhaustively evaluating all grid points, gradient descent-based methods are applied to refine the initial solution, thereby improving the overall accuracy of the method.

2.3.4 Compressive sensing

Compressive Sensing (CS) algorithms aim to find the sparsest solution to an underdetermined system of equations. The way we formulate the DoA problem to allow the use of those algorithms to find the DoAs is explained in what follows. Consider the data acquisition in (2-10). If we replace the vector of true DoAs $\boldsymbol{\theta}$ by a discretized grid $\boldsymbol{\theta}^g \in \mathbb{R}^{(-1,1]^g}$ over the entire field of view Θ such that $g \gg N$, then we have much more unknowns than equations. The problem can be written as:

$$\mathbf{x}_{\mathbb{S}}(t) = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g)\mathbf{s}^g(t) + \mathbf{n}_{\mathbb{S}}(t) \tag{2-30}$$

In this case, we change the terminology to be adherent to the CS Theory: $\mathbf{x}_{\mathbb{S}}(t) \in \mathbb{C}^N$ becomes the measurements vector, $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g) \in \mathbb{C}^{g \times D}$ becomes the measurement matrix or dictionary, $\mathbf{s}^g(t) \in \mathbb{C}^g$ is called the sparse vector to be recovered and $\mathbf{n}_{\mathbb{S}}(t)$ is still called measurement noise. The sparse recovery algorithms try to find the solution of (2-30) with maximum sparsity. Another key concept in CS theory is the one of support set. The index set of non-zero entries of a sparse vector is called its support, according to the following definition:

Definição 2.6 (Support set) *The support set of a vector* $\mathbf{x} \in \mathbb{C}^N$ *corresponds to the index set of its nonzero entries, according to*

$$\operatorname{supp}(\mathbf{x}) \triangleq \{j \in [N] | x_j \neq 0\} \tag{2-31}$$

Moreover, we also need to explore the concept of K-Sparse Vector.

Definição 2.7 (K-Sparse Vector) *The vector* $\mathbf{x} \in \mathbb{C}^N$ *is said to be* K*-sparse if at most* K *of its entries are non-zero, which is mathematically represented by the following statement*

$$\|\mathbf{x}\|_0 \triangleq |\operatorname{supp}(\mathbf{x})| \leq K \tag{2-32}$$

Notice that $\|\cdot\|_0$ denotes the number of non-zero elements in $\mathbf{x}$. It is referred to as a pseudo-norm because it does not satisfy the homogeneity property, i.e., multiplying by a scalar does not result in the norm being scaled by that same scalar. If we consider all the T snapshots and perform a sparse recovery for each of it, then the resulting recovered matrix of source signals would be represented by

$$\hat{\mathbf{S}} = \left[\hat{\mathbf{s}}^g(1) \quad \hat{\mathbf{s}}^g(2) \quad \dots \quad \hat{\mathbf{s}}^g(T)\right] \tag{2-33}$$

The l_2-norm of the rows of $\hat{\mathbf{S}} \in \mathbb{C}^{g \times T}$ is a g-dimensional column vector that is ideally D-sparse. However, since we seek approximations, we relax the problem in this case and use the l_2-norm of the rows as a pseudo-spectrum $\hat{\mathbf{q}}_{\text{ps}}$, from which we search for peaks.

Naturally, due to noise and imperfect recovery conditions, this pseudo-spectrum is not strictly D-sparse, but it can be approximated by a D-sparse signal if a sufficient number of measurements (i.e., number of sensors, N) and snapshots are provided. The support set of $\hat{\mathbf{q}}_{\text{ps}}$, denoted by $\mathbb{L} = \text{supp}\left(\hat{\mathbf{q}}_{\text{ps}}\right)$ is then used to index the grid to recover the DoAs as $\boldsymbol{\theta} = [\boldsymbol{\theta}^g]_{\mathbb{L}}$.

Many types of algorithms are used for sparse recovery, each with its own strengths and weaknesses. As examples, we can cite l_1-norm minimization algorithms (Basis Pursuit (BP) and Quadratically Constrained Basis Pursuit (QCBP)), Greedy algorithms (Orthogonal Matching Pursuit (OMP)), and Thresholding-based algorithms (Iterative Hard Thresholding (IHT)).

The l_1-norm minimization algorithms or Basis Pursuit solve the sparse recovery problem by minimizing the l_1-norm of the sparse signal (in this case, $\mathbf{s}^g(t)$) under the constraint that the reconstruction error is small. These algorithms are especially effective at promoting sparsity and can provide an exact solution when the problem is well-posed, enough measurements are available and the SNR conditions are not that challenging. This is achievable through convex relaxation of the original NP-hard l_0-norm minimization problem into a more tractable l_1-norm minimization, which significantly reduces the computational burden of the initial problem.

Despite this relaxation, BP are still recognized for being computationally intensive, as they involve solving a convex optimization problem. While they offer a good balance between accuracy and complexity, their relatively high computational cost makes them more suitable for scenarios where high accuracy is prioritized. They are less computationally demanding than Maximum Likelihood methods, but are still heavier compared to other alternatives, as will be discussed in what follows. The l_1-minimization algorithms attempt to solve the following problem:

$$\begin{aligned} \underset{\mathbf{s}^g(t)}{\text{minimize}} \quad & \|\mathbf{s}^g(t)\|_1 \\ \text{subject to} \quad & \mathbf{x}_{\mathbb{S}}(t) = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g)\mathbf{s}^g(t), \end{aligned} \tag{2-34}$$

where the measurement noise is ignored even though there are variations that account for it, such as the Quadratically Constrained Basis Pursuit (QCBP), according to

$$\begin{aligned} \underset{\mathbf{s}^g(t)}{\text{minimize}} \quad & \|\mathbf{s}^g(t)\|_1 \\ \text{subject to} \quad & \|\mathbf{x}_{\mathbb{S}}(t) - \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g)\mathbf{s}^g(t)\|_2 \leq \eta \end{aligned} \tag{2-35}$$

where the hyperparameter η is closely related to the sensor noise.

The Greedy algorithms iteratively select the dictionary elements (columns or atoms) most correlated with the residual signal one at a time. Those algorithms are in general much faster than BP due to its iterative nature, but they are prone to suboptimal solutions when the signal is not perfectly sparse or when a substantial amount of noise is present. Nevertheless, its speed and simplicity make them popular in real-time applications, such as DoA estimation, where computational efficiency is critical. The OMP algorithm is a remarkable algorithm of this class.

The OMP algorithm [32,33] is an iterative method designed to minimize the Euclidean residual norm $\|\mathbf{x}_{\mathbb{S}}(t) - \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g)\mathbf{s}^g(t)\|_2$ at each iteration. A fundamental assumption, as in the other cases, is that the number of sources D, corresponding to the source signal sparsity level, is known *a priori*. The algorithm is formally defined in Algorithm 2.3.

It should be noted that the algorithm performs exactly D iterations, incrementally building up the support set $\mathbb{L} = \operatorname{supp}(\mathbf{x}_{\mathbb{S}}(t))$ with cardinality $|\mathbb{L}| = D$. Each iteration selects the column of the dictionary $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g)$ that is most correlated with the residual, refining the sparse signal estimate. However, its performance may degrade under low SNR or when the columns of the dictionary exhibit high mutual correlation, which is influenced by the array geometry and the proximity of the sources in the field of view Θ.

It comprises three fundamental stages. In line 3, the correlation-maximization step is carried out, wherein the atom from the dictionary that is most correlated with the current residual is selected. This step identifies an index to be added to the support set $\mathbb{L}$. This addition is performed in line 4. Subsequently, an orthogonal projection of the received signal onto the subspace spanned by the selected atoms so far is performed in line 5. At the end, the DoAs are estimated by indexing the grid $\boldsymbol{\theta}^g$ with the estimated support set $\mathbb{L}^{(D)}$.

Algorithm 2.3: Orthogonal Matching Pursuit - OMP

```
**Input** : Steering matrix or dictionary $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})$
           Array received signal $\mathbf{x}_{\mathbb{S}}(t)$
           Number of sources $D$
1 $\mathbb{L}^{(0)} = \varnothing$, $\mathbf{s}^g(t)^{(0)} = \mathbf{0}$
2 for $i \leftarrow 0$ to $D-1$ do
3     $j \leftarrow \underset{\hat{j} \in [N]}{\operatorname{argmax}} \left\{ \left| \left[ \mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta}^g) \left( \mathbf{x}_{\mathbb{S}}(t) - \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g)\mathbf{s}^g(t)^{(i)} \right) \right]_{\hat{j}} \right| \right\}$
4     $\mathbb{L}^{(i+1)} = \mathbb{L}^i \cup \{j\}$
5     $\mathbf{s}^g(t)^{(i+1)} \leftarrow \underset{\mathbf{z} \in \mathbb{C}^N}{\operatorname{argmin}} \left\{ \|\mathbf{x}_{\mathbb{S}}(t) - \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g)\mathbf{z}\|_2,\ \operatorname{supp}(\mathbf{z}) \subset \mathbb{L}^{(i+1)} \right\}$
6 $\hat{\boldsymbol{\theta}} = [\boldsymbol{\theta}^g]_{\mathbb{L}^{(D)}}$
**Output:** Estimated DoAs $\hat{\boldsymbol{\theta}} = [\boldsymbol{\theta}^g]_{\mathbb{L}^{(D)}}$
```

Another significant class of CS algorithms consists of thresholding-based methods, with Iterative Hard Thresholding (IHT) being an important example [34]. IHT combines gradient-based updates with a hard thresholding step to enforce sparsity. This algorithm is valued for its simplicity and relatively low computational complexity, making it suitable for large-scale problems, such as those involving a high number of sources and a large array of sensors. However, it can suffer from convergence issues and may require careful tuning of its parameters to achieve optimal performance.

It solves the following optimization problem

$$\underset{\mathbf{s}^g(t)}{\text{minimize}} \qquad \|\mathbf{x}_{\mathbb{S}}(t) - \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g)\mathbf{s}^g(t)\|_2^2 \tag{2-36}$$

$$\text{subject to}\, |\mathbb{L}| \leq D, \tag{2-37}$$

i.e., it iteratively seeks the solution that minimizes the least squares error under the constraint of a D-sparse signal recovered signal. The formal definition of the algorithm is provided in Algorithm 2.4. The $H_D(\cdot)$ operator retains the D largest magnitude elements of a vector, settings the remaining elements to zero, thereby ensuring the signal remains D-sparse, once we target D-sparse vectors [33, 35]. It is important to note that IHT does not compute an orthogonal projection directly in either of its two steps.

Among these, several variations have been specifically tailored for DoA estimation, taking into account the unique characteristics of the array manifold and the spatial sparsity of the sources [36]. These adaptations often exploit

Algorithm 2.4: Iterative Hard Thresholding - IHT

```
Input : Steering matrix or dictionary $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})$
        Array received signal $\mathbf{x}_{\mathbb{S}}(t)$
        Number of sources $D$
$\mathbf{s}^g(t)^{(0)} = \mathbf{0}$, $i = 0$
while some stopping criterion is not met do
    $\mathbf{u}^{(i)} \leftarrow \mathbf{s}^g(t)^{(i)} + \left(\mathbf{A}^H \left(\mathbf{x}_{\mathbb{S}}(t) - \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}^g)\mathbf{s}^g(t)^{(i)}\right)\right)$
    $\mathbf{s}^g(t)^{(i+1)} \leftarrow H_D\left(\mathbf{u}^{(i)}\right)$
    $i = i + 1$
$\mathbb{L} = \text{supp}\left(\mathbf{s}^g(t)^{(i+1)}\right)$
$\hat{\boldsymbol{\theta}} = [\boldsymbol{\theta}^g]_{\mathbb{L}}$
Output: Estimated DoAs $\hat{\boldsymbol{\theta}} = [\boldsymbol{\theta}^g]_{\mathbb{L}}$
```

the inherent structure of the DoA problem to improve the accuracy and efficiency of the estimation process, even under challenging conditions such as low signal-to-noise ratios or limited snapshots.

2.3.5 Subspace methods

Another important class of array processing algorithms rely on the subspace properties of the received signal covariance matrix $\mathbf{R}_{\mathbb{S}}$. Indeed, this matrix is a positive semidefinite (PSD) Hermitian matrix that admits an expansion using orthonormal basis of the type eigenvalue decomposition (EVD). Let us rewrite it as

$$\mathbf{R}_{\mathbb{S}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H \tag{2-38}$$

where

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \end{bmatrix} \in \mathbb{C}^{N \times N} \tag{2-39}$$

is the matrix with orthonormal columns $\mathbf{u}_i \in \mathbb{C}^N$ called eivenvectors and

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix} \in \mathbb{C}^{N \times N} \tag{2-40}$$

is the diagonal matrix with real non-negative eigenvalues λ_i. Let us consider we order the eigenvalues in decreasing size of magnitude, according to

$$\lambda_{\max} \triangleq \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \triangleq \lambda_{\min} \tag{2-41}$$

reflecting the ordering in the matrix $\mathbf{U}$. The expression in (2-38) can be alternatively rewritten as

$$\mathbf{R}_{\mathbb{S}} = \sum_{i=1}^{N} \lambda_i \mathbf{u}_i \mathbf{u}_i^H \tag{2-42}$$

which consists of a sum of rank-one matrices weighted by the corresponding eigenvalues. The spatially white measurement noise can be written as

$$\begin{aligned} \mathbf{R}_n &= \mathrm{E}\{\mathbf{n}_{\mathbb{S}} \mathbf{n}_{\mathbb{S}}^H\} \\ &= \sigma_n^2 \mathbf{I} \end{aligned} \tag{2-43}$$

Given that

$$\begin{aligned} \mathbf{R}_{\mathbb{S}} &= \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}) \mathbf{R}_s \mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta}) + \sigma_n^2 \mathbf{I} \\ &= \mathbf{R}_{\text{sig}} + \sigma_n^2 \mathbf{I} \end{aligned} \tag{2-44}$$

we can express $\mathbf{R}_{\mathbb{S}}$ alternatively as

$$\mathbf{R}_{\mathbb{S}} = \sum_{i=1}^{N} \left(\lambda_i^{\text{sig}} + \sigma_n^2 \right) \mathbf{u}_i \mathbf{u}_i^H \tag{2-45}$$

where λ_i^{sig} are the eigenvalues of $\mathbf{R}_{\text{sig}}$. In our model, since the number of sources is smaller than the number of physical sensors ($D < N$), we can write

$$\lambda_i = \begin{cases} \lambda_i^{\text{sig}} + \sigma_n^2 & \text{if } i \in \{1, 2, \ldots, D\} \\ \sigma_n^2 & \text{if } i \in \{D+1, D+2, \ldots, N\} \end{cases} \tag{2-46}$$

because $\lambda_i^{\text{sig}} = 0$ if $i > D$ (smaller number of columns than rows in $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})$, i.e., $D < N$). From this, we can conclude that there are $N - D$ eigenvalues with multiplicity $N - D$ and magnitude σ_n^2. More specifically, the measurement noise is responsible for ensuring the full rank property of $\mathbf{R}_{\mathbb{S}}$. Moreover, we add that the inverse of $\mathbf{R}_{\mathbb{S}}$ can be written as

$$\begin{aligned} \mathbf{R}_{\mathbb{S}}^{-1} &= \sum_{i=1}^{N} \frac{1}{\lambda_i^{\text{sig}} + \sigma_n^2} \mathbf{u}_i \mathbf{u}_i^H \\ &= \frac{1}{\sigma_n^2} \mathbf{I} - \sum_{i=1}^{N} \frac{\lambda_i^{\text{sig}}}{\sigma_n^2 \left(\lambda_i^{\text{sig}} + \sigma_n^2 \right)} \mathbf{u}_i \mathbf{u}_i^H \end{aligned} \tag{2-47}$$

If there are no signals that exhibit full correlation, $\mathbf{R}_s$ is obviously positive definite (PD). Retaining this, it is clear that the columns of the array steering matrix $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})$ define a D-dimensional subspace that contains the signal energy. A subset of the eigenvectors of $\mathbf{R}_{\mathbb{S}}$ indeed represent an orthogonal basis for that subspace. Then, we can write

$$\mathbf{u}_i = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}) \mathbf{c}_i, \qquad i \in [D] \tag{2-48}$$

where $\mathbf{c}_i \in \mathbb{C}^D$. Also, since $\mathbf{R}_{\text{sig}} = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta}) \mathbf{R}_s \mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})$, it can be rewritten as

$$\lambda_i^{\text{sig}}\mathbf{u}_i = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})\mathbf{u}_i \tag{2-49}$$

Substituting (2-48) into (2-49), we have

$$\lambda_i^{\text{sig}}\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{c}_i = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{c}_i \tag{2-50}$$

which is equivalent to

$$\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\left(\lambda_i^{\text{sig}} - \mathbf{R}_s\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\right)\mathbf{c}_i = 0 \tag{2-51}$$

which implies in

$$\det\left(\lambda_i^{\text{sig}} - \mathbf{R}_s\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\right) = 0 \tag{2-52}$$

with D solutions, with $i \in [D]$. Each λ_i^{sig} has a corresponding $\mathbf{c}_i$ and $\mathbf{u}_i$. From this, we form the following matrix

$$\mathbf{U}_S = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_D\end{bmatrix} \in \mathbb{C}^{N\times D} \tag{2-53}$$

The range space of (2-53) is called signal subspace. We can also express $\mathbf{U}_S$ as linear combination of the columns of the array steering matrix as

$$\mathbf{U}_S = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{C} \tag{2-54}$$

where

$$\mathbf{C} = \begin{bmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_D\end{bmatrix} \in \mathbb{C}^{D\times D} \tag{2-55}$$

It is clear from (2-44) that the noise covariance matrix spans an N-dimensional subspace, since it is a scalar multiple of the $N \times N$ identity matrix. Then, in order to represent this noise subspace, we need to add $N - D$ eigenvalues and eigenvectors. Observe that these additional eigenvectors must be orthogonal to $\mathbf{U}_N$. We choose them to be orthonormal. This matrix is defined as

$$\mathbf{U}_N = \begin{bmatrix}\mathbf{u}_{D+1} & \mathbf{u}_{D+2} & \dots & \mathbf{u}_N\end{bmatrix} \in \mathbb{C}^{N\times(N-D)} \tag{2-56}$$

whose range space is called noise subspace with eigenvalues σ_n^2 of multiplicity $N-D$, as we have stated before. Observe also that the signal subspace contains noise and signal and the noise subspace contains only noise. The received signal covariance matrix $\mathbf{R}_{\mathbb{S}}$ can also be alternatively written as

$$\mathbf{R}_{\mathbb{S}} = \mathbf{U}_S\boldsymbol{\Lambda}_S\mathbf{U}_S^H + \mathbf{U}_N\boldsymbol{\Lambda}_N\mathbf{U}_N^H \tag{2-57}$$

where

$$\boldsymbol{\Lambda}_S = \begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D\end{bmatrix} \tag{2-58}$$

and

$$\mathbf{\Lambda}_N = \begin{bmatrix} \sigma_n^2 & & & \\ & \sigma_n^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{bmatrix} \tag{2-59}$$

In summary, we can state that the decomposition of the received signal covariance matrix, $\mathbf{R}_{\mathbb{S}}$, yields two orthogonal subspaces: the signal (plus noise) subspace and the noise subspace. These subspaces share no common elements other than the zero vector, and each is characterized by a distinct set of basis vectors. These bases are constructed from specific eigenvectors obtained through the eigendecomposition of $\mathbf{R}_{\mathbb{S}}$. The signal subspace is defined by the eigenvectors corresponding to the largest eigenvalues, whereas the noise subspace is formed by the eigenvectors corresponding to the smallest eigenvalues [37].

The array manifold matrix, $\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})$, contains columns representing the steering vectors for different signal directions, and these vectors span the signal subspace. Therefore, the signal subspace can equivalently be described as the span of these steering vectors. Additionally, the signal subspace is spanned by the eigenvectors associated with the D largest eigenvalues of the covariance matrix $\mathbf{R}_{\mathbb{S}}$.

In contrast, the noise subspace is spanned by the remaining eigenvectors, corresponding to the $N - D$ smallest eigenvalues, where N represents the total number of array elements or the dimension of the observation space. These eigenvectors form the basis for the noise subspace, which remains orthogonal to the signal subspace. This orthogonality is critical for separating the signal from noise in the received data.

Some of the most widely used algorithms that rely on the subspace properties of the received signal covariance matrix are Spectral MUSIC (Multiple Signal Classification) [6] and some of its variants [38,39], root-MUSIC [40], as well as ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) [9, 41, 42].

MUSIC is one of the earliest and most widely used subspace-based methods. It exploits the fact that the noise subspace is orthogonal to the steering vectors corresponding to the true DoAs, allowing for accurate estimation of the signal directions. Root MUSIC, a variant of MUSIC, simplifies the DoA estimation process by solving for the roots of a polynomial on the unit circle, which reduces computational complexity in certain array geometries such as uniform linear arrays (ULAs).

In addition to MUSIC and root MUSIC, ESPRIT, in its Least Squares (LS) and Total Least Squares (TLS) offers another powerful high resolution approach, particularly due to its exploitation of the shift invariance property of some arrays, which allows for parameter estimation without the need for spectral search, i.e., a search-free method.

A variation of ESPRIT, denoted Unitary ESPRIT, proposed in [43], further improves on ESPRIT by utilizing the fact that the operator with the phase delays between subarrays is unitary utilizes real-valued computations, enhancing computational efficiency and accuracy. Other notable algorithms in this family include subspace fitting techniques [44], each providing unique advantages depending on the specific array geometry and signal environment.

In this discussion, we will focus on Spectral MUSIC and root-MUSIC due to their foundational role in subspace-based DoA estimation, their widespread use in practical applications, and primarily because the algorithms proposed in this work are based on these two methods. Spectral and root-MUSIC offer a reasonable balance between accuracy and computational efficiency, making them well-suited for a variety of array configurations and signal environments.

2.3.5.1
Multiple signal classification: the spectral MUSIC algorithm

One of the most important algorithms that relies on subspace theory for DoA estimation is the Multiple Signal Classification (MUSIC) algorithm. It was proposed in [6] and is widely used in array signal processing due to its high resolution performance, i.e., its ability to distinguish closely spaced sources. It operates by exploiting the eigendecomposition of the covariance matrix of the received signal data, distinguishing between the signal and noise subspaces.

The key concept behind MUSIC is the orthogonality between the noise subspace and the steering vectors associated with the signal sources, as we saw in the previous section. The algorithm projects the array steering vectors onto the noise subspace, identifying the signal directions as the points where this projection is minimized, corresponding to peaks in a spatial spectrum. This enables the precise estimation of signal directions even in the presence of substantial noise and closely spaced sources.

Unlike conventional beamforming techniques, which rely on spatial filtering as discussed in Section 2.3.2, MUSIC leverages the eigendecomposition approach to achieve superior resolution by separating the signal subspace from the noise subspace. However, the performance of the MUSIC algorithm depends on the correct estimation of the number of signals, and its computational complexity increases with the number of array elements and signals (EVD or Singular Value Decomposition (SVD) decomposition must be performed). Despite these challenges, MUSIC remains a fundamental and highly effective tool in modern array signal processing applications.

The MUSIC algorithm relies on the idea that signal and noise subspaces are mutually orthogonal. This can be mathematically written as

$$\mathbf{u}_i^H \mathbf{U}_N = \mathbf{0}^T \ \forall i \in [D] \tag{2-60}$$

which can be rewritten, using (2-56), as

$$\left(\mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{c}_i\right)^H \mathbf{U}_N = \mathbf{0}^T \ \forall \ i \in [D] \tag{2-61}$$

or equivalently

$$\mathbf{c}_i^H \mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta})\mathbf{U}_N = \mathbf{0}^T \tag{2-62}$$

$\iff$

$$\mathbf{a}_{\mathbb{S}}^H(\theta_i)\mathbf{U}_N = 0 \tag{2-63}$$

$\iff$

$$\|\mathbf{a}_{\mathbb{S}}^H(\theta_i)\mathbf{U}_N\|_2^2 = 0 \ \forall \ i \in [D] \tag{2-64}$$

From this, we compute the so-called MUSIC spectrum as

$$P_{\text{SPECT-MUSIC}}(\theta_i) = \frac{1}{\mathbf{a}_{\mathbb{S}}^H(\theta_i)\mathbf{U}_N\mathbf{U}_N^H\mathbf{a}_{\mathbb{S}}^H(\theta_i)} \tag{2-65}$$

for all $\theta_i \in \boldsymbol{\theta}^g$, plot (2-65) against $\boldsymbol{\theta}^g$ and find the largest D local maxima that correspond to the D estimated DoAs.

The MUSIC algorithm is widely recognized in array signal processing for its versatility and broad application. It can be effectively utilized with both SLAs and ULAs [10]. This versatility underscores the robustness and adaptability of the Spectrum MUSIC algorithm across different array configurations. The algorithm is summarized in Algorithm 2.5.

2.3.5.2 The search-free root-MUSIC algorithm

Based on the derivation of Section 2.3.5.1, [40] proposed a search-free directions finding algorithm that is very much similar to Spectral MUSIC, but differently relies of the polynomial rooting to find the DoAs. Although it presents superior performance in comparison to its spectral version, it possesses a great limitation in terms of array geometries, since it can only be used with ULAs. Attempts to use it with SLAs require some pre-processing using some sort of array mapping, interpolations or coarrays. In this section, we analyze the standard form of the algorithm.

The polynomial version of the steering vector of a coherent ULA can be expressed as

$$\mathbf{f}(z) = \left[1, z, z^2, \ldots, z^{N-1}\right]^T \tag{2-66}$$

which is evaluated at

```
Algorithm 2.5: Spectral MUSIC
    Input  : Array geometry $\mathbb{S}$
             Data matrix $\hat{\mathbf{X}}_{\mathbb{S}}$
             Search grid $\boldsymbol{\theta}^g$
             Number of sources $D$
$\hat{R}_{\mathbb{S}} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}}\hat{\mathbf{X}}_{\mathbb{S}}^H$
$\hat{R}_{\mathbb{S}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H$, with eigenvalues in descending order
// diagonalization
$\mathbf{U}_N \leftarrow \mathbf{U}(:, D+1 : N)$
for $i \leftarrow 1$ to $g$ do
    $P_{\text{SPECT-MUSIC}}(\theta_i) = \frac{1}{\mathbf{a}_{\mathbb{S}}^H(\theta_i)\mathbf{U}_N\mathbf{U}_N^H\mathbf{a}_{\mathbb{S}}^H(\theta_i)}$

// assemble the spectrum
Find the $D$ largest peaks of $P_{\text{SPECT-MUSIC}}(\theta_i)$
// spectrum analysis
Retain the corresponding DoAs of the peaks as $\theta_i$;// peaks
    finding
Compute the DoAs as $\theta_i = \angle z_i/\pi$
    Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

$$z = \exp\left(j(2\pi d)/\lambda\theta\right) = \exp\left(j\pi\theta\right) \tag{2-67}$$

assuming $d = \lambda/2$. As we discussed earlier in Section 2.3.5, the noise subspace consists of eigenvectors that are orthogonal to the steering vectors of the array manifold. Specifically, $\mathcal{R}(\mathbf{U}_N) \perp \mathbf{f}(z)$, where $\mathbf{U}_N$ represents an orthonormal basis for the noise subspace of the data received by the coherent ULA and $\mathbf{f}(z) = \mathbf{a}_{\mathbb{S}}(\theta)$ is the array steering vector for the direction θ. The inner product between these two quantities must therefore be zero. From this, we can write the following polynomial:

$$\begin{aligned} Q(z) &= \|\mathbf{f}^H(z)\mathbf{U}_N\|_2^2 \\ &= \mathbf{f}^T(1/z)\mathbf{U}_N\mathbf{U}_N^H\mathbf{f}(z) \end{aligned} \tag{2-68}$$

which is the polynomial for the coherent array associated with the projection matrix $\mathbf{U}_N\mathbf{U}_N^H$. Based on the orthogonality conditions, this polynomial must evaluate to zero at the true DoAs. The roots of this polynomial are then used to estimate the DoAs, denoted as z_i. We retain the D roots that lie inside and closest to the unit circle. From these roots, using (2-67), we compute the D DoAs as

$$\theta_i = \angle z_i/\pi, \ \forall\ i \in [D] \tag{2-69}$$

The algorithm is formally defined in Algorithm 2.6.

Algorithm 2.6: root-MUSIC

```
Input  : Array geometry $\mathbb{S}$
         Data matrix $\hat{\mathbf{X}}_{\mathbb{S}}$
         Search grid $\boldsymbol{\theta}^g$
         Number of sources $D$
$\hat{R}_{\mathbb{S}} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}}\hat{\mathbf{X}}_{\mathbb{S}}^H$
$\hat{R}_{\mathbb{S}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H$, with eigenvalues in descending order
// diagonalization
$\mathbf{U}_N \leftarrow \mathbf{U}(:, D+1 : N)$
$Q(z) \leftarrow \mathbf{f}^T(1/z)\mathbf{U}_N\mathbf{U}_N^H\mathbf{f}(z)$
// assemble the polynomial
Find the $D$ roots of $Q(z)$ inside and closest to the unit circle as $z_i$
// root-pruning
Compute the DoAs as $\theta_i = \angle z_i/\pi$
Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

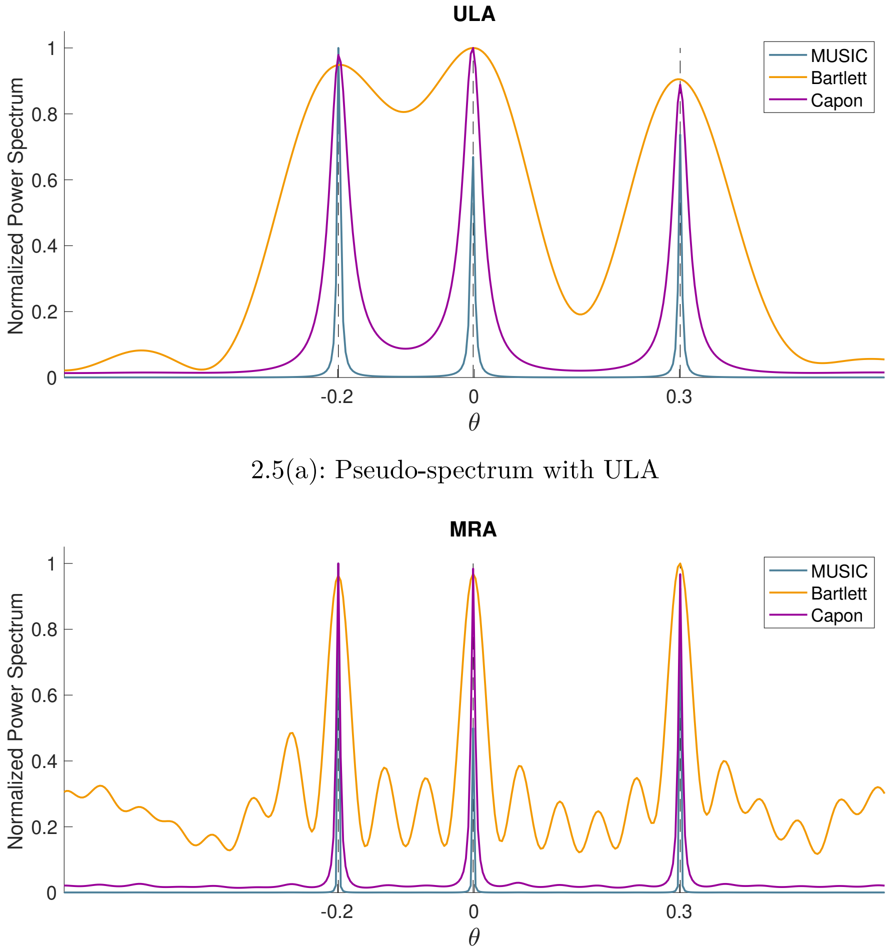
2.4 Numerical examples

To illustrate some of the concepts discussed in the previous sections and to motivate the use of sparse arrays for DoA estimation through various array design strategies and algorithms, we present the power spectra shown in Figure 2.5(a) and Figure 2.5(b).

Each figure was generated using a different array design technique with $N = 10$ physical sensors, under the same SNR and number of snapshots (SNR = 8 dB, $T = 200$ snapshots). In the former, a ULA geometry was employed, while the latter was simulated using an MRA spatial sampling pattern. The true DoAs $\boldsymbol{\theta} = [-0.2, 0, 0.3]$ are indicated by vertical black dashed lines. For each geometry, we compare a subspace technique (MUSIC) with two beamforming-based algorithms, namely the Capon (MVDR) and Bartlett.

It is clear that the pseudo-spectra for all three techniques are significantly narrower and more accurate for estimating the true DoAs in the case of the MRA geometry, due to its increased aperture compared to the ULA. Furthermore, when comparing the algorithms, MUSIC is by far the most accurate, exhibiting the narrowest peaks, followed by Capon, and finally the less accurate Bartlett beamformer. This provides a clear and simple demonstration of the impact of array design and algorithm choice on the accuracy of DoA estimates.

In addition, we present the RMSE curves, computed over 10,000 trials. These curves also include the rMUSIC (root-MUSIC) algorithm, which does not have a pseudo-spectrum. The RMSE curves can be seen in Figures 2.6(a)



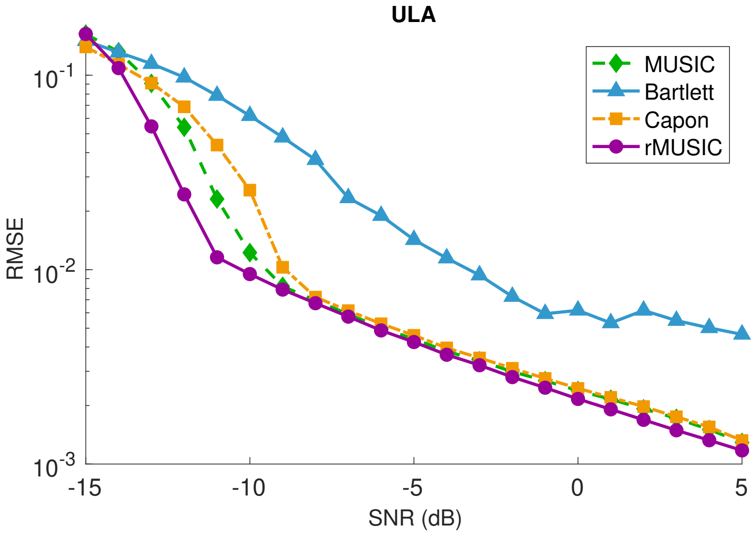
2.5(a): Pseudo-spectrum with ULA

2.5(b): Pseudo-spectrum with MRA

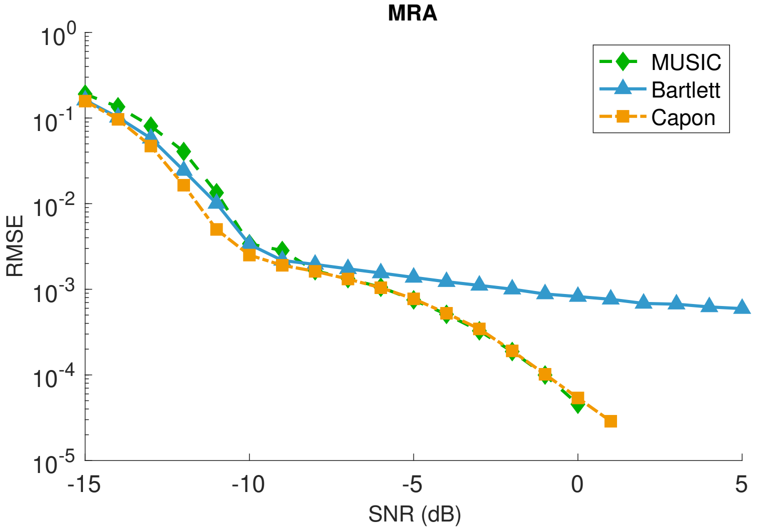
Figure 2.5: Pseudo-spectra using subspace (MUSIC), and beamforming (Bartlett and Capon) techniques to estimate the DoAs in $\boldsymbol{\theta} = [-0.2, 0, 0.3]$. $N = 10$ sensors, $T = 200$ snapshots, and SNR $= 8$ dB.

and 2.6(b).

It can be observed that rMUSIC is the most accurate algorithm in the case of the ULA for almost all the range, especially in low SNR regime. For the MRA, in this scenario, the performance of MUSIC and Capon is very similar, with Capon showing slightly better behaviour at lower SNR values (less than -10 dB). For this geometry, observe that the RMSE values reached the order of 10^{-5} for higher SNR values, which is significantly smaller than the 10^{-3} magnitude observed in the ULA case. Lastly, we observe that, although the Capon algorithm demonstrates a lower RMSE in this specific scenario compared to its counterparts, we select the MUSIC algorithm as the foundation of our work due to its superior superresolution capabilities.



2.6(a): RMSE with ULA



2.6(b): RMSE with MRA

Figure 2.6: RMSE using subspace (MUSIC and rMUSIC), and beamforming (Bartlett and Capon) techniques to estimate the DoAs in $\boldsymbol{\theta} = [-0.2, 0, 0.3]$. $N = 10$ sensors, $T = 200$ snapshots, and 10,000 trials.

2.5 Summary

In this chapter, we reviewed the fundamentals of sensor array signal processing, focusing on spatial sampling patterns (array geometries), signal models, and algorithms for DoA estimation. As we have briefly discussed, DoA estimation is essential in various fields, including mobile communications, military applications, automotive radar, sonar detection, astronomy, and drone surveillance. Accurate DoA estimation enhances both target identification and communication beamforming.

A key consideration in array processing is the geometric configuration of sensor arrays, which significantly affects spatial sampling and DoA estimation accuracy. Sparse arrays, in particular, were discussed as a means of balancing performance and hardware costs by using fewer sensors than traditional uniform arrays while maintaining high resolution. Various geometries, such as ULAs, MRAs, and other non-uniform configurations, were explored to improve spatial resolution without increasing the number of array elements.

The algorithms covered in this chapter focused on leveraging the structure of sensor data to improve DoA estimation accuracy. Techniques such as beamforming (Bartlett and Capon), Compressive Sensing (BP, QCBP, OMP, IHT), Stochastic ML, and subspace-based methods (MUSIC and ESPRIT) were outlined. Subspace-based methods, which decompose the received signal into signal and noise subspaces and exploit their orthogonality to efficiently estimate incoming signal directions, were emphasized, as they form the basis of the techniques proposed in the following chapters.

3
Non-coherent sparse subarrays with enhanced degrees of freedom

In this chapter, we present novel structures for the design of sparse subarrays termed *type-I* and *type-II* subarrays. The properties of those subarrays are studied and some insights are derived, which include their number of degrees of freedom. We start by describing some common linear sparse arrays techniques that will be used throughout this work. Additionally, we derive two DoA estimation algorithms designed for partially-calibrated systems with sparse subarrays: Generalized Coarray Multiple Signal Classification (GCA-MUSIC) and Generalized Coarray Root Multiple Signal Classification (GCA-rMUSIC).

3.1
Sparse linear subarrays design

There are many ways to perform the design of a sensor array that consists of some subarrays. For example, one would rely on the split of a hypothetical geometry into smaller parts such that the aperture of the original array is greater than the sum of the apertures of the subarrays. The inequality instead of equality comes from the fact that the subarrays are separated by a distance of at least one unity of intersensor spacing. This is what we call type-I subarray.

Definição 3.1 (Type-I Sparse Linear Array) *A type-I Sparse Linear Array (type-I SLA) corresponds to an array of predefined sparse linear geometry $\mathbb{S}$. The subsets defining the subarrays geometries are generated from partitions $\mathbb{S}_l \subset \mathbb{S}$ such that if $s \in \mathbb{S}_i$ and $f \in \mathbb{S}_j$, $i < j$, then $s < f$.*

The second most intuitive way of generating a subarray would be to take many subarrays with a predefined geometry and combine them to generate an array of a given aperture. This is what we call type-II subarray. The type-II subarrays analyzed in this work are considered to have the same geometric pattern, i.e., the parent array consists of a union of subarrays with the same geometry pattern. In what follows, we establish the formal definition.

Definição 3.2 (Type-II Sparse Linear Array) *A type-II Sparse Linear Array (Type-II SLA) corresponds to a union of subarrays with predefined sparse*

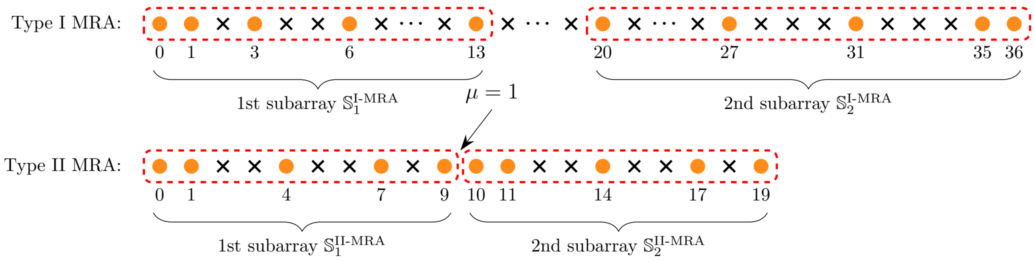

Type I MRA:
0
1
3
6
13
20
27
31
35
36
1st subarray $\mathbb{S}_1^{\text{I-MRA}}$
2nd subarray $\mathbb{S}_2^{\text{I-MRA}}$
$\mu = 1$
Type II MRA:
0
1
4
7
9
10
11
14
17
19
1st subarray $\mathbb{S}_1^{\text{II-MRA}}$
2nd subarray $\mathbb{S}_2^{\text{II-MRA}}$

Figure 3.1: Type-I and type-II MRA. Notice that the type-II array was generated with a parameter $\mu = 1$ (spacing between subarrays).

linear geometries $\mathbb{S}_l$ (partitions of the array geometry $\mathbb{S}$). The set $\mathbb{S}_1$ defines the reference subarray. The remaining subarrays are derived from $\mathbb{S}_1$ through $\mathbb{S}_i = \mathbb{S}_{i-1}^{\Delta_{i-1}}$, where Δ_{i-1} is a translation factor for all the elements of $\mathbb{S}_{i-1}$ and is given by $\Delta_{i-1} = \mu + \kappa_{i-1}$, where μ is the normalized distance between subarrays (in terms of integer multiples of d, that represents the minimum intersensor spacing) and κ_{i-1} is the aperture of the $(i-1)$-th subarray.

In what follows, we consider that the sets $\mathbb{S}$ and $\mathbb{S}_l$ have cardinality N and N_l, respectively, i.e., $|\mathbb{S}| = N$ and $|\mathbb{S}_l| = N_l$ or equivalently $N = \sum_{l=1}^{L} N_l$. Notice that the definitions in both cases restrict the subarrays such that they obey to the following rules: i) they do not share any sensors (partitions are pairwise disjoint); ii) they are collinear; and iii) the array is given by the union of the subarrays $\left(\mathbb{S} = \bigcup_{l=1}^{L} \mathbb{S}_l\right)$. Additionally, observe that if we apply the additional constraint $N_l = N/L$ (i.e., constant κ for all the subarrays) $\forall l \in [L]$[1], then the subarrays used to generate type-II SLAs become multiple invariant [45].

The key difference between both sparse array definitions is that for type-I SLAs, the subarrays are generated from a splitting of the array with a predefined sparse geometry. On the other hand, for type-II SLA, the array is generated by the union of sparse linear subarrays with a predefined geometry.

To illustrate the concepts of type-I and type-II arrays, consider the sparse geometry defined according to the set $\mathbb{S}^{\text{MRA}} = \{0, 1, 3, 6, 13, 20, 27, 31, 35, 36\}$, which corresponds to a MRA with $N = 10$ sensors. We adopt $L = 2$ subarrays. The corresponding type-I array is the original array itself $\left(\mathbb{S}^{\text{I-MRA}} = \mathbb{S}^{\text{MRA}}\right)$ (for a intersubarray displacement equal to (20-13)=7) and its subarrays are given by $\mathbb{S}_1^{\text{I-MRA}} = \{0, 1, 3, 6, 13\}$ and $\mathbb{S}_2^{\text{I-MRA}} = \{20, 27, 31, 35, 36\}$. On the other hand, consider two MRAs with $N = 5$ sensors each given by $\mathbb{S}_1^{\text{II-MRA}} = \{0, 1, 4, 7, 9\}$ and $\mathbb{S}_2^{\text{II-MRA}} = \mathbb{S}_1^{\Delta_1;\text{II-MRA}} = \{10, 11, 14, 17, 19\}$, with translation factor $\Delta_1 = 1 + (9 - 0) = 10$. Thus, $\mathbb{S}^{\text{MRA}} = \mathbb{S}^{\text{I-MRA}} \neq \mathbb{S}^{\text{II-MRA}}$. For didactic purposes, Figure 3.1 shows a graphical representation of this example.

[1] The symbol $[m]$ represents the set $\{1, 2, \ldots, m\}$.

3.1.1 Analysis of degrees of freedom

A natural question that arises from the above introduced (sub)array geometries is related to the manifold structure, as well as the DoFs for a difference coarray scenario [21, 36]. In this section, we shed some light upon these two important aspects.

3.1.1.1 Type-I

Clearly, for type-I arrays there is no *a priori* analytical relation between the subarray manifolds, due to the fact that the geometries change dramatically between subarrays. Also, the number of DoF for the predefined geometry and its type-I counterpart is obviously the same, for the case in which we keep the original intersubarray displacement and both geometries coincide. Moreover, there is no general rule to predict the DoF associated to the subarrays. This requires an analysis case by case for the sparse geometry that is chosen to be employed.

3.1.1.2 Type-II

Some analytical relations can be straightforwardly derived. In this case, to simplify the equations, we will assume that the subarrays are multiple invariant, i.e., they attend the sufficient condition of having the same number of physical sensors N/L and thus the same aperture. Using the array manifold of the whole array expressed in matrix form, we can write

$$\mathbf{A}_{\mathbb{S}} = \left[\mathbf{A}_{\mathbb{S}_1}^T, \left(\mathbf{A}_{\mathbb{S}_1}\mathbf{\Lambda}^{\Delta}\right)^T, \ldots, \left(\mathbf{A}_{\mathbb{S}_1}\mathbf{\Lambda}^{(L-1)\Delta}\right)^T\right]^T \tag{3-1}$$

where we dropped the dependence of the manifold on the DOAs to simplify the notation. The matrix $\mathbf{\Lambda}$ is defined as

$$\mathbf{\Lambda} = \operatorname{diag}\left(e^{j\pi\theta_1}, \ldots, e^{j\pi\theta_D}\right) \tag{3-2}$$

where the variable θ_d represents the sine of one of the DOAs. The multiple exponents of $\mathbf{\Lambda}$ in (3-1) represent the subarrays translations along the straight line and are assumed to be $\Delta = \mu + \kappa$ (see Definition 3.2).

In what follows, we establish one of our most important results [46]: the number of DoF for type-II arrays is upper-bounded by a function of the DoF of the subarrays (sDoF), which can be theoretically calculated for a variety of preconceived geometries. This result is quantitatively described in Theorem 3.3.

Teorema 3.3 *Consider a type-II array with geometry as defined in Definition 3.2 with equal-aperture subarrays. If* $1 \leq \mu \leq \kappa$*, then the number of* DoF *of the array* $\mathbb{S}$ *is upper-bounded by* $L(\mathrm{sDoF}-1)+2(L-1)\mu+1$*, where* sDoF *is the number of* DoF *for each subarray. If* $\mu > \kappa$*, then the number of* DoF *is equal to* $(2L-1)\mathrm{sDoF}$.

Prova. Let $c(n)$ *be a discrete-valued function that assumes the value of 1 if there is a sensor at* nd *or 0 otherwise. This function for a type-II array is given by* $c(n) = c_1(n) + \ldots + c_L(n)$*. Since the subarrays are translated versions of the reference array,* $c_l(n) = c_1(n - \Delta_l)$*, where* $\Delta_l = (l-1)\Delta$*. The weight function associated to that array (counts the number of spatial correlations with lag* n *is defined through [1]* $w(n) = c(n) \circledast c^-(n)$*, where* $c^-(n)$ *is the time reversal version of* $c(n)$*. Then, one can write*

$$\begin{aligned} w(n) &= \left(\sum_{i=1}^{L} c_i(n)\right) \circledast \left(\sum_{j=1}^{L} c_j^-(n)\right) \\ &= \sum_{i,j=1}^{L} c_1(n-(i-1)\Delta) \circledast c_1^-(n-(j-1)\Delta) \\ &= \sum_{i,j=1}^{L} c_1(n) \circledast c_1^-(n) \circledast \delta(n-\Delta_i+\Delta_j) \\ &= \sum_{i,j=1}^{L} w_1(n-\Delta_i+\Delta_j) \end{aligned} \tag{3-3}$$

The number of DoF is the cardinality of the support of $w(n)$*. This support set has always an odd number of elements, given that* $w(n)$ *is an even function. Clearly, from (3-3), the weight function of the reference array* $w_1(n)$ *is repeated along the domain with displacement factors given by* $\Delta_j - \Delta_i$*. Since* $|\operatorname{supp}(w_1(n))| =$ *sDoF and for* $\mu > \kappa$ *there is no superposition of the weight functions of the subarrays with a different* $\Delta_j - \Delta_i$*, then* $|\operatorname{supp}(w(n))| = (2L-1)|\operatorname{supp}(w_1(n))| \Rightarrow$ *DoF* $= (2L-1)$*sDoF. Note that there are* $2L-1$ *diagonals in a* $L \times L$ *matrix representing all the possible* $\Delta_j - \Delta_i$*. For* $1 \leq \mu \leq \kappa$*, the superposition implies that the support set of* $w(n)$ *ranges from* $-(L-1)\Delta-\kappa$ *to* $(L-1)\Delta+\kappa$*. Since* $\kappa =$ *(sDoF-1)*$/2$*, the support set of* $w(n)$ *has a maximum number of elements equal to* $2\left[(L-1)\Delta+\kappa\right]+1$ *or* $L(\mathit{sDoF}-1)+2(L-1)\mu+1$*. The equality holds if the difference coarray of the subarrays has no holes (no missing lags), as it indeed happens for many geometries like Two-Level Nested and (restricted) Minimum Redundancy Arrays.* ■

3.1.2
Coarray weight function

The concept of weight function presented in Definition 2.5, Chapter 2, is vital to define the degrees of freedom in the context of difference coarrays. We exploit this concept in this section and give further insights on to the intuition behind its properties. As it was mentioned before, the weight function counts the number of unique spatial lags in the covariance matrix for any given linear array geometry. To illustrate, we start by describing two cases:

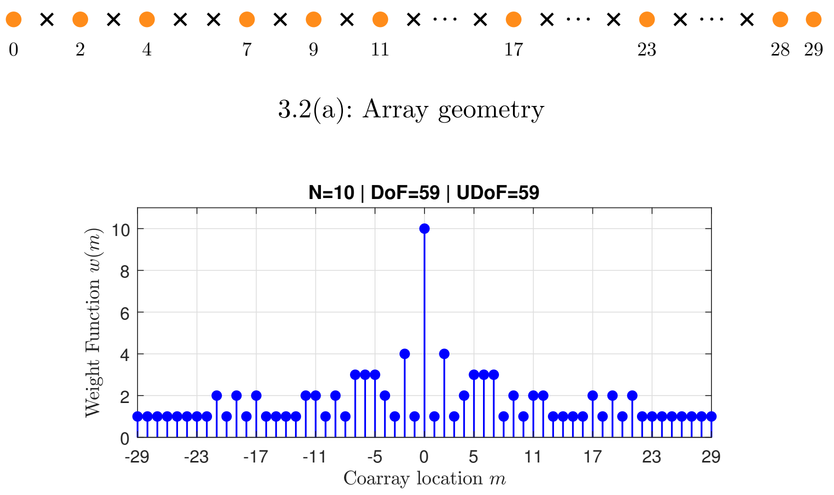
i) 2nd-Order Super Nested Array (SNAQ2) [21] with $N = 10$ sensors located at

$$\mathbb{S}^{\text{SNAQ2}} = \{0, 2, 4, 7, 9, 11, 17, 23, 28, 29\} \tag{3-4}$$

ii) MHA with $N = 10$ sensors and geometry given by

$$\mathbb{S}^{\text{MHA}} = \{0, 1, 6, 10, 23, 26, 34, 41, 53, 55\} \tag{3-5}$$

The arrays and their corresponding weight functions are exhibited in Figures 3.2 and 3.3.

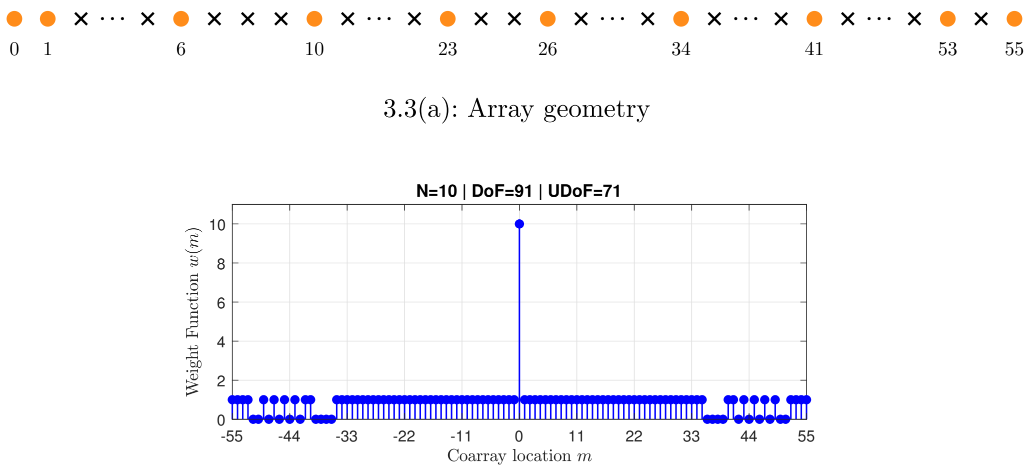


3.2(a): Array geometry

3.2(b): Weight function

Figure 3.2: 2nd-Order Super Nested Array (SNAQ2) with $N = 10$ sensors

Notice that for the two cases previously presented, the weight functions in Figures 3.2(b) and 3.3(b) can be non-zero over its entire support set or defective at some point, i.e., having some zero values. Its support set corresponds to the spatial lags that can be found in the covariance matrix. On the other hand, the points for which $w(m)$ is zero-valued means that the covariance matrix does not possess the corresponding covariance lag. Those defective weight functions are said to have "holes" (Figure 3.3(b)), while in the former case (filled $w(m)$),



3.3(a): Array geometry

3.3(b): Weight function

Figure 3.3: Minimum Hole Array (MHA) with $N = 10$ sensors

we say that the weight function is totally filled, which is our main case of interest (more uniform degrees of freedom). The reason why arrays with filled weight functions are preferable will become clear in the next chapter. when we discuss spatial smoothing in the coarray domain.

3.2 System model and problem statement

In order to illustrate the problem we are dealing with, consider the sensor array system shown in Figure 3.4. In this scheme, a partially-calibrated linear array is composed of subarrays. Plane wave signals that come from far-field sources illuminate the subarrays. Each source possesses a different direction of arrival (DOA) in relation to the subarrays broadside axis (normal axis). The initial preprocessing is independent between the subarrays. The only data available to the central processing unit (CPU) are the second-order statistics from each subarray. The cross-statistics are unknown and the CPU estimates the DOAs through a specific algorithm.

The representation of the data acquisition model with partially-calibrated subarrays for a multiple measurements scenario is given by

$$\mathbf{x}_{\mathbb{S}_l}(t) = \mathbf{A}_{\mathbb{S}_l}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}_l}(t) \tag{3-6}$$

for $l \in [L]$ and $t \in [T]$, where $\mathbf{A}_{\mathbb{S}_l}(\boldsymbol{\theta}) \in \mathbb{C}^{N_l \times D}$ is the l-th subarray manifold with geometry defined by the set of integers $\mathbb{S}_l$ (normalized positions in terms of integer multiples of d - minimum intersensor spacing), the l-th subarray has N_l sensors, and there are D impinging sources with normalized directions given by $\boldsymbol{\theta} \in [-1, 1)^D$ (sine of DOAs), $\mathbf{s}(t) \in \mathbb{C}^D$ is the source signal snapshot indexed by t, $\mathbf{x}_{\mathbb{S}_l}(t) \in \mathbb{C}^{N_l}$ is the l-th subarray received signal snapshot, and $\mathbf{n}_{\mathbb{S}_l}(t)$ is

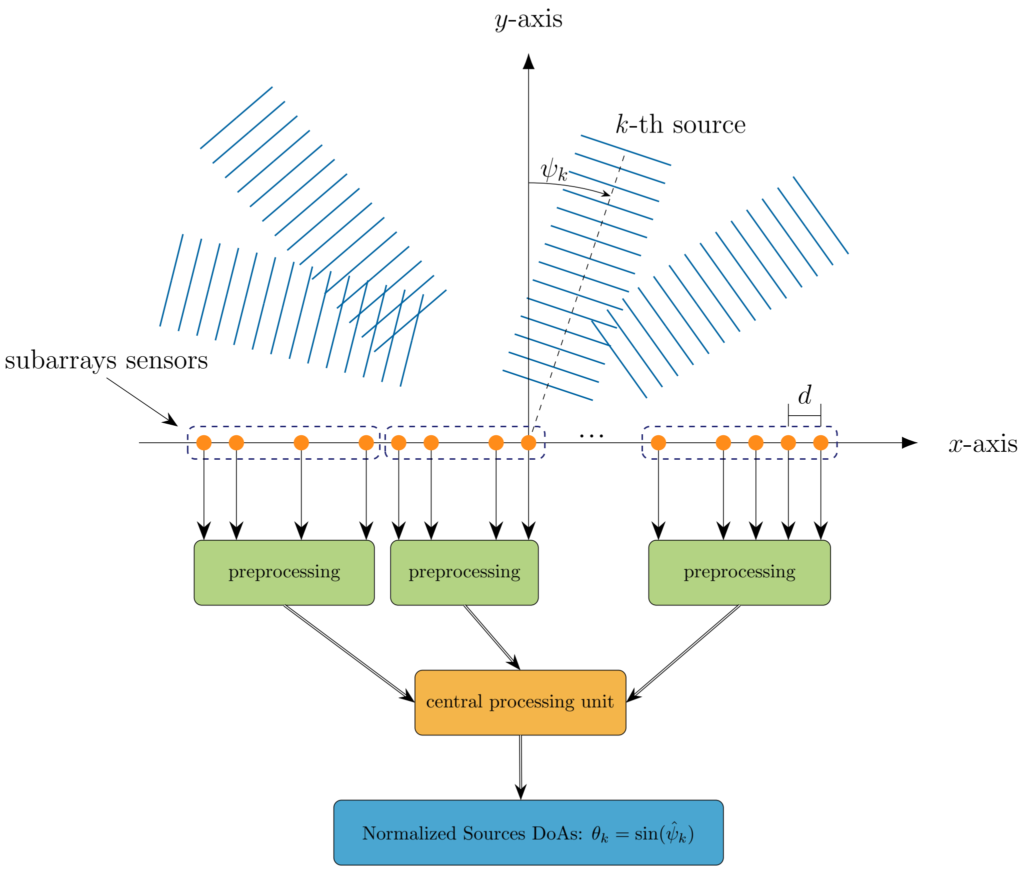


Figure 3.4: Subarrays processing scheme

the subarray noise vector measurement. The noise and the source signals are drawn from a circularly complex multivariate Gaussian distribution. The noise is spatially and temporally white and the sources are uncorrelated. Sources and noise are uncorrelated between themselves. Remark: as abuse of notation, we refer to the set $\mathbb{S}_l$ defining the sensors locations as the array itself.

3.3
Generalized coarray MUSIC

Since we are dealing with sparse subarrays (SpSub), a natural starting point to develop an estimator would be to extend the coarray MUSIC or Spatial Smoothing MUSIC (SS-MUSIC) algorithm [1]. This extension will be termed Generalized Coarray MUSIC (GCA-MUSIC). This estimation procedure has many advantages over other techniques presented in the literature: it is capable of exploiting half of the DoF of the subcoarrays[2], presents super-resolution performance capabilities and has a reasonable trade-off in terms of computational burden. The proposed GCA-MUSIC aims to tackle the case of DoA estimation with partially calibrated sparse subarray geometries.

[2] To be defined in what follows.

To this end, we start by computing the second-order statistics associated to each of the subarrays, according to

$$\mathbf{R}_{\mathbb{S}_l} = \mathbf{A}_{\mathbb{S}_l}(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}^H_{\mathbb{S}_l}(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I},\ l = 1, \ldots, L \tag{3-7}$$

where $\mathbf{R}_{\mathbb{S}_l}$ is the received signal covariance matrix of the l-th subarray and $\mathbf{R}_s = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_D^2)$ is the uncorrelated sources covariance matrix. By vectorizing (3-7), we arrive at

$$\mathbf{z}_{\mathbb{S}_l} = \left(\mathbf{A}^*_{\mathbb{S}_l}(\boldsymbol{\theta}) \circ \mathbf{A}_{\mathbb{S}_l}(\boldsymbol{\theta})\right)\mathbf{p} + \sigma_n^2\bar{\mathbf{i}} \tag{3-8}$$

where $\mathbf{z}_{\mathbb{S}_l} \in \mathbb{C}^{|\mathbb{D}_l|^2}$ is the vectorized covariance matrix in (3-7), $\circ$ denotes the column-wise Khatri-Rao product, $\bar{\mathbf{i}} = \left[\mathbf{e}_1^T, \ldots, \mathbf{e}_{N_l}^T\right]^T$ is the vectorization of the identity matrix, and $\mathbf{p} = [\sigma_1^2, \ldots, \sigma_D^2]$ contains the sources powers. To simplify the equations, we will adopt a total of N/L sensors for each subarray. By removing the repeated rows in $\mathbf{A}^*_{\mathbb{S}_l}(\boldsymbol{\theta}) \circ \mathbf{A}_{\mathbb{S}_l}(\boldsymbol{\theta})$ after their first occurrence (mirroring the operation in $\mathbf{z}_{\mathbb{S}_l}$ and $\bar{\mathbf{i}}$) and sorting the virtual sensors (coarray) elements in ascending order, we have

$$\mathbf{x}_{\mathbb{D}_l} = \mathbf{A}_{\mathbb{D}_l}(\boldsymbol{\theta})\mathbf{p} + \sigma_n^2\mathbf{i} \tag{3-9}$$

where $\mathbb{D}_l$ denotes the difference coarray set associated to the l-th subarray, $[\mathbf{A}_{\mathbb{D}_l}(\boldsymbol{\theta})]_{m,d} = \exp(j\pi n_m \theta_d)$ with $n_m \in \mathbb{D}$, $\mathbf{x}_{\mathbb{D}_l} \in \mathbb{C}^{|\mathbb{D}_l|}$ is the l-th coarray received signal and $\mathbf{i} \in \{0, 1\}^{|\mathbb{D}_l|}$ is an all-zero vector with the exception of a 1 in its half position (element $(|\mathbb{D}_l| + 1)/2$).

From this point, we introduce some terminology according to the following definitions:

Definição 3.4 (Sparse Subarray (SpSub)) *The sparse subarrays are defined as each partially calibrated part of the whole array. The sensors are coherent within each sparse subarray (sampling process is performed in a synchronized basis). They are denoted by $\mathbb{S}_l$, with $l = 1, \ldots, L$.*

Definição 3.5 (Subcoarray (SCA)) *A subcoarray is defined as the Difference Coarray associated to each SpSub. They are denoted by $\mathbb{D}_l$, with $l = 1, \ldots, L$.*

Definição 3.6 (Spatially Smoothed Subcoarray (SS-SCA)) *A spatially smoothed subcoarray (SS-SCA) is a SCA with reduced dimension dictated by the parameter choices of a spatial smoothing-like procedure. They are denoted by $\mathbb{D}_l^i$, with $l = 1, \ldots, L$ and $i = 1, \ldots, M$, where l is the number of subarrays and M is a function of the subcoarrays apertures.*

Notice that each SCA (associated to a specific SpSub), generated after the mathematical procedure described from (3-7) to (3-9), will have a total of M SS-SCA. Then, we have a total of $M \cdot L$ SS-SCA for the whole array.

We consider SpSub and its respective SCA with central contiguous part (virtual ULA) large enough to allow the recovery of all of the sources DOA. To simplify the equations, we will assume that the SpSub has a filled SCA (no holes in virtual domain), i.e., the second-order statistics associated to each SpSub contains all the correlation lags from 0 up to $\kappa = (|\mathbb{D}_l| - 1)/2$ (the aperture of each SpSub is equal between all the subarrays, because we are considering Type-II Arrays with the same number of physical sensors within each SpSub).

By resorting to the rank properties, it is clear that the outer product $\mathbf{x}_{\mathbb{D}_l}\mathbf{x}_{\mathbb{D}_l}^H$ is rank deficient. Then, we build up this rank using $M = \kappa + 1 = (\text{sDoF} + 1)/2$ SS-SCA (forward spatial smoothing), for each SCA, according to

$$\mathbf{R}_{\mathbb{D}_l}^{\text{SS}} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_{\mathbb{D}_l^i}\left(\mathbf{x}_{\mathbb{D}_l^i}\right)^H \tag{3-10}$$

where $\mathbf{x}_{\mathbb{D}_l^i} = \mathbf{J}_i\mathbf{x}_{\mathbb{D}_l} \in \mathbb{C}^M$ is the i-th overlapping SS-SCA of the l-th SCA, $\mathbf{J}_i = \left[\mathbf{0}_{M\times(M-i)}\mathbf{I}_M\mathbf{0}_{M\times(i-1)}\right]$ is a selection matrix, starting ($i = 1$) from the maximum value of the contiguous part of the SCA, and sDoF $= |\mathbb{D}_l|$ is the number of degrees of freedom for each subarray. We remark that although many spatial smoothing techniques can be used in (3-10) [47], depending on the amount of computational resources available in the DSP, we keep the standard SS as presented in [1] because it resulted in a good estimation performance in our numerical results. Additionally, it can be demonstrated that (3-10) has a signal and noise subspace that allows us to obtain the sources DOA by using MUSIC. Then, each $\mathbf{R}_{\mathbb{D}_l}^{\text{SS}}$, originated from each of the partially calibrated subarrays provides rough estimates of the sources DoA.

In order to illustrate the concepts, consider what is on Figure 3.5. We have used the array geometry previously presented in Figure 3.2(a) as an example.

The first problem was to generate subspace-structured matrices through (3-10) from the subarray data in order to obtain rough estimates of the DoAs. The second problem we deal with is how to combine the processing such that we can profit from the estimates of each SpSub in an integrated fashion. This spectrum combination is key to improve the estimation performance, as will be demonstrated further.

To perform the signal decomposition, we adopt a similar strategy as described in [13]. The signal and noise subspace of $\mathbf{R}_{\mathbb{D}_l}^{\text{SS}}$ can be obtained from the following EVD

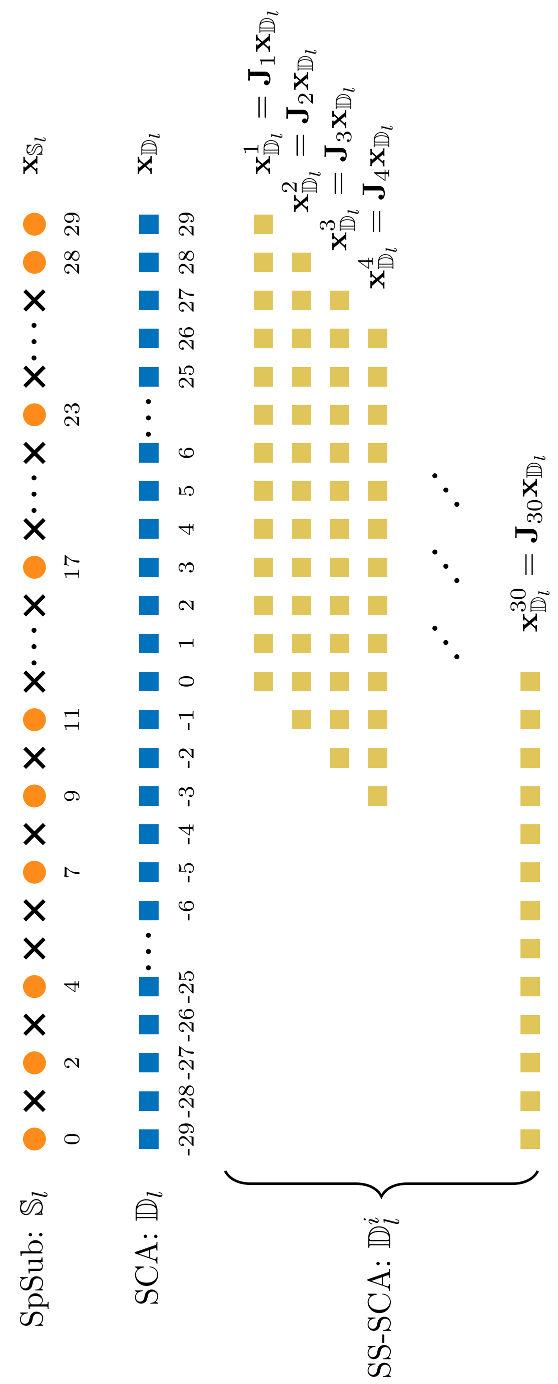

SpSub: $\mathbb{S}_l$
0 2 4 7 9 11 17 23 28 29
$\mathbf{x}_{\mathbb{S}_l}$
SCA: $\mathbb{D}_l$
-29 -28 -27 -26 -25 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 25 26 27 28 29
$\mathbf{x}_{\mathbb{D}_l}$
SS-SCA: $\mathbb{D}_l^i$
$\mathbf{x}_{\mathbb{D}_l}^1 = \mathbf{J}_1 \mathbf{x}_{\mathbb{D}_l}$
$\mathbf{x}_{\mathbb{D}_l}^2 = \mathbf{J}_2 \mathbf{x}_{\mathbb{D}_l}$
$\mathbf{x}_{\mathbb{D}_l}^3 = \mathbf{J}_3 \mathbf{x}_{\mathbb{D}_l}$
$\mathbf{x}_{\mathbb{D}_l}^4 = \mathbf{J}_4 \mathbf{x}_{\mathbb{D}_l}$
$\mathbf{x}_{\mathbb{D}_l}^{30} = \mathbf{J}_{30} \mathbf{x}_{\mathbb{D}_l}$

Figure 3.5: Illustration of SpSub, SCA, and SS-SCA

$$\mathbf{R}^{\mathrm{SS}}_{\mathbb{D}_l} = \begin{bmatrix}\mathbf{U}_l & \mathbf{V}_l\end{bmatrix} \operatorname{diag}\left(\beta_1^l, \ldots, \beta_M^l\right) \begin{bmatrix}\mathbf{U}_l^H \\ \mathbf{V}_l^H\end{bmatrix} \tag{3-11}$$

that has the same eigenvectors (signal and null-space) as those associated to a covariance matrix whose data was obtained with the first array manifold of the spatial smoothing procedure, denoted by $\mathbf{A}_{\mathbb{D}_l^1}(\boldsymbol{\theta})$ (the virtual array manifold corresponding to the last M rows of $\mathbf{A}_{\mathbb{D}_l}(\boldsymbol{\theta})$). Then, we can state that $\mathcal{R}\left(\mathbf{A}_{\mathbb{D}_l^1}(\boldsymbol{\theta})\right) = \mathcal{R}\left(\mathbf{U}_l\right)$ and $\mathcal{N}\left(\mathbf{A}^H_{\mathbb{D}_l^1}(\boldsymbol{\theta})\right) = \mathcal{R}\left(\mathbf{V}_l\right)$. Particularly in this case, the subarrays are partly-calibrated and so the statistics are divided and must be processed separately at some degree.

The i-th SS-SCA of the l-th SCA has its signal part contained in $\mathcal{R}\left(\mathbf{U}_l\right)$. Define

$$\tilde{\mathbf{U}}_l = \operatorname{blkdiag}\left(\mathbf{I}_{M(l-1)}, \mathbf{U}_l, \mathbf{I}_{M(L-l)}\right) \tag{3-12}$$

and

$$\tilde{\mathbf{V}}_l = \operatorname{blkdiag}\left(\mathbf{I}_{M(l-1)}, \mathbf{V}_l, \mathbf{I}_{M(L-l)}\right) \tag{3-13}$$

Clearly, $\mathcal{R}\left(\mathbf{U}_l\right) \subset \mathcal{R}\left(\tilde{\mathbf{U}}_l\right)$, i.e., the mixed block-diagonal structure imposes that the range space of $\tilde{\mathbf{U}}_l$ is a superset of the corresponding signal subspace $\mathbf{U}_l$. However, the signal subspace lies in $\mathcal{R}\left(\tilde{\mathbf{U}}_l\right)$ $\forall\, l \in \{1, \ldots, L\}$.

Every affine set is unbounded and convex. Moreover, each of the subspaces $\mathcal{R}\left(\mathbf{U}_l\right)$ is affine. The corresponding noise subspaces $\mathcal{R}\left(\mathbf{V}_l\right)$ are also affine. Consider the set

$$\mathbb{W} = \bigcap_{l=1}^{L} \tilde{\mathbf{V}}_l \tag{3-14}$$

In order to project a point onto $\mathbb{W}$ (intersection of noise subspaces), we can iterate though successive projections onto each of the $\mathcal{R}\left(\tilde{\mathbf{V}}_l\right)$ by resorting to the method of projection onto convex sets (POCS) [13, 48, 49]. Since the matrices $\tilde{\mathbf{V}}_l$ are orthogonal, their projection matrices are given by

$$\begin{aligned}\mathbf{P}_{\tilde{\mathbf{V}}_l} &= \tilde{\mathbf{V}}_l \tilde{\mathbf{V}}_l^H \\ &= \operatorname{blkdiag}\left(\mathbf{I}_{M(l-1)}, \mathbf{V}_l\mathbf{V}_l^H, \mathbf{I}_{M(L-l)}\right)\end{aligned} \tag{3-15}$$

The successive projections lead us to

$$\begin{aligned}\mathbf{P} &= \prod_{l=1}^{\overset{\curvearrowright}{L}} \mathbf{P}_{\tilde{\mathbf{V}}_l} \\ &= \operatorname{blkdiag}\left(\mathbf{V}_1\mathbf{V}_1^H, \mathbf{V}_2\mathbf{V}_2^H, \ldots, \mathbf{V}_L\mathbf{V}_L^H\right)\end{aligned} \tag{3-16}$$

which is idempotent and thus represents the final matrix after the convergence of the POCS procedure. Lets define the composed steering vector in SS-SCA coarray domain. We can write

$$\mathbf{a}_{\mathbb{D}}^{\mathrm{SS}}(\theta) = \left[\left(\mathbf{a}_{\mathbb{D}_1}^{\mathrm{SS}}(\theta)\right)^T \quad \left(\mathbf{a}_{\mathbb{D}_2}^{\mathrm{SS}}(\theta)\right)^T \quad \ldots \quad \left(\mathbf{a}_{\mathbb{D}_L}^{\mathrm{SS}}(\theta)\right)^T\right]^T \tag{3-17}$$

where

$$\mathbf{a}_{\mathbb{D}_l}^{\mathrm{SS}}(\theta) = \Big[0 \quad \exp\left(j(2\pi d/\lambda)\theta\right) \quad \ldots \quad \exp\left(j(2\pi d/\lambda)(M-1)\theta\right)^T\Big] \tag{3-18}$$

that corresponds to the steering vector of $\mathbf{A}_{\mathbb{D}_l^1}(\boldsymbol{\theta})$ and does not depend on the subarray index l. This comes from the fact that each of the SCA is a filled coarray ULA and have the same number of degrees of freedom $|\mathbb{D}_1| = \ldots = |\mathbb{D}_L|$.

The steering vector in (3-17) is orthogonal to the matrix $\mathrm{blkdiag}\left(\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_L\right)$. Then, the quadratic form built from the matrix $\mathbf{P}$ and the composite steering vector in (3-17) should have a null for the true directions θ, similarly to the MUSIC procedure. From that, the pseudo-spectrum can be obtained through

$$\begin{aligned} P(\theta_i^g) &= \frac{1}{\left(\mathbf{a}_{\mathbb{D}}^{\mathrm{SS}}(\theta)\right)^H \mathbf{P}\left(\mathbf{a}_{\mathbb{D}}^{\mathrm{SS}}(\theta)\right)} \\ &= \frac{1}{\sum_{l=1}^{L}\left(\mathbf{a}_{\mathbb{D}_l}^{\mathrm{SS}}(\theta_i^g)\right)^H \mathbf{V}_l \mathbf{V}_l^H \mathbf{a}_{\mathbb{D}_l}^{\mathrm{SS}}(\theta_i^g)}, \; i \in [g] \end{aligned} \tag{3-19}$$

where $\boldsymbol{\theta}^g \in (-1, 1]^g$ is the grid search. The true DOAs are estimated as the angles on the grid that maximize (3-19).

Remarks:

a) The matrices $\mathbf{R}_{\mathbb{D}_l}^{\mathrm{SS}}$ are orthogonally diagonalizable because of their structure as a sum of outer products (Hermitian symmetric);

b) We stress the fact that for SS-SCA with the same number of DoF (virtual aperture) and filled coarray ULAs, the vectors $\mathbf{a}_{\mathbb{D}_l}^{\mathrm{SS}}(\theta)$ do not depend on l, as stated in (3-18).

c) The algorithm can indeed be extended for *any* linear sparse subarray geometries, once they have the same physical orientation (not necessarily type-II arrays). Indeed, even for algorithms with some SpSub that do not have a filled ULA in the coarray domain, the algorithm can be easily extended to employ only the central contiguous part of the coarray, i.e., the central part corresponding to the UDoF. In the simulations section, an example will illustrate this case.

The algorithm is summarized in Algorithm 3.1.

Lastly, we would like to point out that GCA-MUSIC is capable of identifying more sources than sensors even for the partially calibrated array

```
Algorithm 3.1: GCA-MUSIC
    Input  : subarrays geometries $\mathbb{S}_l$, data matrix for each subarray
             $\hat{\mathbf{X}}_{\mathbb{S}_l}$, grid $\boldsymbol{\theta}^g \in (-1, 1]^g$, number of sources $D$
    for $l \leftarrow 1$ to $L$ do
        $\hat{\mathbf{R}}_{\mathbb{S}_l} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}_l}\hat{\mathbf{X}}_{\mathbb{S}_l}^H$
        $\hat{\mathbf{z}}_{\mathbb{S}_l} \leftarrow \mathrm{vec}\left(\hat{\mathbf{R}}_{\mathbb{S}_l}\right)$
        Remove the repeated elements in $\mathbf{z}_l$ and sort the corresponding
          coarray locations to obtain $\hat{\mathbf{x}}_{\mathbb{D}_l}$
        $\hat{\mathbf{R}}^{\mathrm{SS}}_{\mathbb{D}_l} \leftarrow \frac{1}{M}\sum_{i=1}^{M}\hat{\mathbf{x}}_{\mathbb{D}_l^i}\left(\hat{\mathbf{x}}_{\mathbb{D}_l^i}\right)^H$
        Diagonalize $\hat{\mathbf{R}}^{\mathrm{SS}}_{\mathbb{D}_l}$ through an EVD to find $\hat{\mathbf{V}}_l$
    for $i \leftarrow 1$ to $g$ do
        $\hat{P}(\theta_i^g) = \left(\sum_{l=1}^{L}\left(\mathbf{a}^{\mathrm{SS}}_{\mathbb{D}_l}(\theta_i^g)\right)^H \hat{\mathbf{V}}_l\hat{\mathbf{V}}_l^H \mathbf{a}^{\mathrm{SS}}_{\mathbb{D}_l}(\theta_i^g)\right)^{-1}$
    Find the peaks in $\hat{P}(\theta_i^g)$ and take the corresponding angles as the
      estimated DOAs
    Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

scenario. GCA-MUSIC can identify up to $(\mathrm{sDoF}-1)/2$ sources. To write this quantity as a function of the number of physical sensors, we must define the geometry we are dealing with. For example, for sparse subarrays following a NAQ2 geometry, GCA-MUSIC can identify up to $(N/L)^2/2 + N/(2L) - 1$ sources, which is the same number of sources that coarray MUSIC can identify with a coherent array of N/L sensors with this geometry. Then, GCA-MUSIC takes advantage of the covariance lags to exploit the subarrays sDoF, keeping the super-resolution properties associated to eigenspace-based DOA estimation algorithms.

It would be of great interest if we could identify more sources than the number of sensors in the *whole* array. Under certain conditions, the algorithm indeed can identify more sources than the total number of sensors. For example, for the case of the NAQ2 geometry that we discussed earlier, if we have $L = 3$ subarrays and $N > 16$ in a type-II structure, then the number of sources that can be identified is greater than the number of sensors in the whole arrangement. For example, for $L = 3$ and $N = 18$, GCA-MUSIC allows the identification of $D = 20$ sources. This requires a case by case analysis for other geometries.

Additionally, the subarrays displacement μ does not have to be known and the subarrays do not need to be collinear. The only requirement is that each SCA has a central contiguous part ($\mathbb{U}$) that allows the spatial smoothing to be performed on it. Lastly, we could also have a mixed subarrays scenario with MRA, NAQ2 and other geometries and the algorithm still works although the steering vectors $\mathbf{a}^{\mathrm{SS}}_{\mathbb{D}_l}(\theta)$ would eventually possess a different size and the aperture of each SS-SCA would be different for each subarray.

3.3.1 Complexity analysis

In this section, we start by looking at the algorithm's arithmetic complexity. In what follows, we present the detailed calculus of the corresponding number of additions and multiplications of each relevant step of the algorithm.

a) **Step S1: estimation of covariance matrix**. This step computes each of the subarrays' received signal covariance matrix by means of its sample covariance estimate. Since the outer products involve vectors of dimension $N_l = N/L$ (see (3-6)), then the number of required additions and multiplications is $L[(T-1)(N/L)^2]$ and $L[T(N/L)^2]$, respectively. [3];

b) **Step S2: estimation of spatially-smoothed coarray covariance matrix**. In this case, the coarrays outer products are obtained from vectors of dimension M (see (3-10)). Then, the number of additions and multiplications is $L \cdot [(T-1)M^2]$ and $L \cdot [TM^2]$, respectively;

c) **Step S3: EVD of $\mathbf{R}^{\mathbf{SS}}_{\mathbb{D}_l}$**. Since the decompositions involve a square matrix of dimension M, the EVD takes about $4M^3/3 + \mathcal{O}(M^2)$ operations;

d) **Step S4: noise subspace projection matrices**. The projection matrix $\mathbf{P}_{\mathbf{V}_l} = \mathbf{V}_l\mathbf{V}_l^H = \mathbf{I} - \mathbf{U}_l\mathbf{U}_l^H$ requires $L \cdot (B-1)M^2$ additions and LBM^2 multiplications, as it involves L matrices of size $M \times B$, where $B = M - D$;

e) **Step S5: pseudo-spectrum calculation**. The pseudo-spectrum, for each point in the grid, is calculated through $\left(\mathbf{a}^{\mathrm{SS}}_{\mathbb{D}_l}(\theta^g_i)\right)^H \hat{\mathbf{V}}_l\hat{\mathbf{V}}_l^H \mathbf{a}^{\mathrm{SS}}_{\mathbb{D}_l}(\theta^g_i)$. It involves $LBM^2 - L$ additions and $L(B+1)M^2 + LM$ multiplications for each of the points in the grid. For a grid search with g points the complexity for the whole search is $g(LBM^2 - L)$ additions and $g(L(B+1)M^2 + LM)$ multiplications.

The arithmetic complexity is summarized in Table 3.1.

[3] Notice that we consider the case where all the subarrays have the same number of physical sensors to simplify the final expressions.

Table 3.1: Arithmetic complexity analysis of GCA-MUSIC

step	**type of operation**	
	additions	**multiplications**
S1	$L(T-1)(N/L)^2$	$LT(N/L)^2$
S2	$L(T-1)M^2$	LTM^2
S3	$4M^3/3 + \mathcal{O}(M^2)$	
S4	$L(B-1)M^2$	LBM^2
S5	$g(LBM^2 - L)$	$g(L(B+1)M^2 + LM)$
TOTAL	$4M^3/3 + \mathcal{O}(M^2)$	

3.4 Generalized coarray root-MUSIC

Perhaps the most natural extension of the previous algorithm would be, instead of computing the MUSIC pseudo-spectrum, to resort to the root-MUSIC version. In order to develop an accurate version of root-MUSIC for partially calibrated sparse arrays, we had to resort to a technique that we call subspace noise smoothing.

The derived algorithm was termed Generalized Coarray Root-MUSIC or GCA Root-MUSIC. It consists of a subspace-based algorithm that performs a smoothing on the noise subspace of each subarray in order to provide a combination for the noise subspace of the resulting root-MUSIC algorithm. First of all, we should review the standard root-MUSIC algorithm, as it was originally derived in [40].

3.4.1 The standard root-MUSIC algorithm

The standard root-MUSIC algorithm was originally presented as an improvement to the standard MUSIC algorithm by resorting to a polynomial version of MUSIC instead of a grid-search through a spectrum. Indeed, it is demonstrated that this procedure increases the MUSIC accuracy because the errors introduced in the modulus of the polynomial roots do not interfere with the directions, which only rely on the phase of the polynomial roots [10, 50].

Since from previous sections we have demonstrated that the noise subspace is orthogonal to the steering vectors corresponding to the true directions, the array steering vector can be rewritten in polynomial form as:

$$\mathbf{v}(z) = \left[1, z, z^2, \ldots, z^{N-1}\right]^T \tag{3-20}$$

that consists of the array steering vector evaluated at $z = \exp\left(j(2\pi d)/\lambda\theta\right)$.

Indeed, by resorting to the properties of MUSIC, one can write

$$\begin{aligned} Q(z) &= \|\mathbf{v}^H(z)\mathbf{N}\|_2^2 \\ &= \mathbf{v}^H(z)\mathbf{N}\left(\mathbf{v}^H(z)\mathbf{N}\right)^H \\ &= \mathbf{v}^H(z)\mathbf{N}\mathbf{N}^H\mathbf{v}(z) \\ &= \mathbf{v}^T(1/z)\mathbf{N}\mathbf{N}^H\mathbf{v}(z) \end{aligned} \tag{3-21}$$

that is the polynomial form for the coherent array, for $\mathbf{N}$ representing the noise subspace of the covariance matrix. Notice that the array geometry must be a ULA, in this case. Clearly, in the case of partially calibrated subarrays, where only the second order statistics of each subarray are available and we do not have access to the cross second-order statistics, a general polynomial must be assembled by resorting to some additional tools. This will be treated in the next section.

3.4.2 Coarray root-MUSIC in partially calibrated scenarios

In order to develop a version of root-MUSIC for partially calibrated sparse arrays, we exploit the root-MUSIC polynomial and the GCA-MUSIC of the previous section. For the standard root-MUSIC algorithm in [40].

The steering vector of a coherent ULA can be written as

$$\mathbf{f}(z) = \left[1, z, z^2, \ldots, z^{N-1}\right]^T \tag{3-22}$$

which is evaluated at $z = \exp\left(j(2\pi d)/\lambda\theta\right)$, assuming an array with N physical sensors. It is widely known that the noise subspace has all their eigenvectors orthogonal to the steering vectors of the array manifold, i.e., $\mathcal{R}(\mathbf{V}) \perp \mathbf{f}(z)$ where $\mathbf{V}$ is a basis for the noise subspace of the coherent ULA and $\mathbf{f}(z) = \mathbf{a}_{\mathbb{S}}(\theta)$ denotes the corresponding array steering vector for the direction θ. The inner product between those two quantities should be zero. From this, we can write the polynomial:

$$\begin{aligned} Q(z) &= \|\mathbf{f}^H(z)\mathbf{V}\|_2^2 \\ &= \mathbf{f}^H(z)\mathbf{V}\left(\mathbf{f}^H(z)\mathbf{V}\right)^H \\ &= \mathbf{f}^T(1/z)\mathbf{V}\mathbf{V}^H\mathbf{f}(z) \end{aligned} \tag{3-23}$$

which is the polynomial form for the coherent array associated with the projection matrix $\mathbf{V}\mathbf{V}^H$. Notice that in this case, the array geometry must be a ULA. Following the orthogonality conditions, this polynomial must have a zero if evaluated at the true directions. This is the standard root-MUSIC algorithm.

Clearly, in the case of partially calibrated sparse subarrays, where only the second-order statistics of each subarray are available and we do not have access to the cross second-order statistics, a general polynomial must be assembled by resorting to extra tools with the constraint that the subarrays are sparse.

As with GCA-MUSIC, here we deal with each of the SpSub after the spatial smoothing procedure as described in (3-11). In this case, we will use each of the noise subspaces $\mathbf{V}_l$ to obtain a more accurate estimate of the true noise subspace. The proposed procedure relies on the intuition that a more accurate polynomial would be obtained by using the denominator of (3-19). We assume that each SCA can identify all the sources in the coarray domain.

Inspired by root-MUSIC, we first observe that a vector is in the nullspace of (3-16) if and only if each of its partitions of appropriate size is in the nullspace of the corresponding block. Consider the partitioned steering vector defined as

$$\mathbf{f}_{\mathbb{D}}(z) = [\mathbf{f}_1(z), \ldots, \mathbf{f}_L(z)]^T \tag{3-24}$$

where

$$\mathbf{f}_l(z) = \left[1, z, z^2, \ldots, z^{M_l - 1}\right]^T \tag{3-25}$$

and $M_l = (\mathrm{sDoF}_l + 1)/2$. Since each component of (3-24) is necessarily in the range space of (3-16), then we can write

$$\begin{aligned} Q_{\mathbb{D}}(z) &= \mathbf{f}_{\mathbb{D}}^T(1/z)\mathbf{P}\mathbf{f}_{\mathbb{D}}(z) \\ &= \sum_{l=1}^{L} \mathbf{f}_l^T(1/z)\mathbf{V}_l\mathbf{V}_l^H\mathbf{f}_l(z) \end{aligned} \tag{3-26}$$

This global coarray polynomial can then be employed to find the true DOAs. Notice that when this substitution is performed, some mathematical implications must be considered. We briefly discuss them in what follows.

We can build this polynomial because each of the SpSub becomes a virtual ULA in the coarray domain. Its degree is a function of the degrees of freedom of the subarrays. The global polynomial in (3-26) has degree $n_Q = 2M_i - 2$, where $i = \mathrm{argmax}_{l\in\{1,\ldots,L\}} M_l$, the index corresponding to the subarray with largest UDoF. Clearly, the higher order coefficients are only affected by the subarrays with large virtual aperture. This observation allows us to further simplify (3-26) and gain some insight regarding its structure. Based on this polynomial, we can build the global projection matrix as

$$\tilde{\mathbf{P}} = \mathbf{P}_{\mathbf{V}_i} + \sum_{l=1, l\neq i}^{L} \tilde{\mathbf{P}}_{\mathbf{V}_l} \tag{3-27}$$

where
$$\tilde{\mathbf{P}}_{\mathbf{V}_l} = \begin{bmatrix} \mathbf{0}_{(M_i - M_l)\times M_l} \\ \mathbf{I}_{M_l} \end{bmatrix} \mathbf{P}_{\mathbf{V}_l} \begin{bmatrix} \mathbf{I}_{M_l} & \mathbf{0}_{M_l \times (M_i - M_l)} \end{bmatrix} \tag{3-28}$$

Then, we can state that $\tilde{\mathbf{P}}$ is a global noise projection matrix, that accounts for the sum of polynomials and thus represents the noise subspace more accurately compared to the noise subspace of a single subarray. We point out that (3-27) can even be used in scenarios where the subarrays have different SNRs, by exploiting similar strategies used in antenna diversity techniques [51]. It suffices to apply weights on the noise projections and use an optimization criterion to find the best possible subspace estimate. For the sake of clarity and ease of understanding, we shall confine our analysis to subarrays with the same SNR. By combining (3-25) and (3-27), we have

$$Q_{\mathbb{D}}(z) = \mathbf{f}_i^T(1/z)\tilde{\mathbf{P}}\mathbf{f}_i(z), \tag{3-29}$$

where $i = \operatorname{argmax}_{l\in\{1,\ldots,L\}} M_l$, as previously mentioned. After the root pruning process, only the D roots inside and closest to the unit circle are considered.

Two key differences between GCA-rMUSIC and root-MUSIC should be emphasized. The first and more obvious difference is that GCA-rMUSIC is capable of estimating more sources than the number of sensors in each subarray (the polynomial degree is drastically increased). The second difference is that root-MUSIC requires the array to be fully synchronized while GCA-rMUSIC estimates the directions based on a global estimate of the nullspace. The GCA-rMUSIC algorithm is shown in Algorithm 3.2.

3.4.2.1 Complexity analysis

In order to analyze the GCA-rMUSIC algorithm capabilities, we start by looking at its arithmetic complexity. In order to do so, we count the number of operations needed for each of its relevant steps. In what follows, we present the detailed calculus of the corresponding number of additions and multiplications of each step.

a) **Step S1: estimation of covariance matrix**. This step computes each of the subarrays' received signal covariance matrix by means of its sample covariance estimate. Since the outer products involve vectors of dimension $N_l = N/L$ (see (3-6)), then the number of required additions and multiplications is $L[(T-1)(N/L)^2]$ and $L[T(N/L)^2]$, respectively. [4];

b) **Step S2: estimation of spatially-smoothed coarray covariance matrix**. In this case, the coarrays outer products are obtained from

[4] Notice that we consider the case where all the subarrays have the same number of physical sensors to simplify the final expressions.

Algorithm 3.2: GCA-rMUSIC

```
Input  : subarray geometries $\mathbb{S}_l$, data matrix for each subarray $\hat{\mathbf{X}}_{\mathbb{S}_l}$,
         grid $\boldsymbol{\theta}^g \in (-1, 1]^g$, number of sources $D$
for $l \leftarrow 1$ to $L$ do
    $\hat{\mathbf{R}}_{\mathbb{S}_l} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}_l}\hat{\mathbf{X}}_{\mathbb{S}_l}^H$
    $\mathbf{z}_l \leftarrow \mathrm{vec}\left(\hat{\mathbf{R}}_{\mathbb{S}_l}\right)$
    Average the repeated elements in $\mathbf{z}_l$ and sort the corresponding
      coarray locations to obtain $\mathbf{x}_{\mathbb{D}_l}$
    $\mathbf{R}_{\mathbb{D}_l}^{\mathrm{SS}} \leftarrow \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_{\mathbb{D}_l^i}\left(\mathbf{x}_{\mathbb{D}_l^i}\right)^H$
    Diagonalize $\mathbf{R}_{\mathbb{D}_l}^{\mathrm{SS}}$ through an EVD to find $\mathbf{V}_l$
    $\mathbf{P}_{\mathbf{V}_l} \leftarrow \mathbf{V}_l\mathbf{V}_l^H$
Search for $i = \mathrm{argmax}_{l\in\{1,\ldots,L\}} M_l$
$\tilde{\mathbf{P}} = \mathbf{P}_{\mathbf{V}_i} + \sum_{\substack{l=1\\l\neq i}}^{L} \tilde{\mathbf{P}}_{\mathbf{V}_l}$
Find the roots of $Q_{\mathbb{D}}(z) = \mathbf{f}_i^T(1/z)\tilde{\mathbf{P}}\mathbf{f}_i(z)$
Select the $D$ roots inside and closest to the unit circle as $\hat{z}_d$
Estimate the DOAs according to $\hat{\theta} = (\angle\hat{z}_d)/\pi$
Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

vectors of dimension M (see (3-10)). Then, the number of additions and multiplications is $L \cdot [(T-1)M^2]$ and $L \cdot [TM^2]$, respectively;

c) **Step S3: EVD of $\mathbf{R}_{\mathbb{D}_l}^{\mathbf{SS}}$**. Since the decompositions involve a square matrix of dimension M, the EVD takes about $4M^3/3 + \mathcal{O}(M^2)$ operations;

d) **Step S4: noise subspace projection matrices**. The projection matrix $\mathbf{P}_{\mathbf{V}_l} = \mathbf{V}_l\mathbf{V}_l^H = \mathbf{I} - \mathbf{U}_l\mathbf{U}_l^H$ requires $L \cdot (B-1)M^2$ additions and LBM^2 multiplications, as it involves L matrices of size $M \times B$, where $B = M - D$;

e) **Step S5: averaged projection matrix calculation**. The averaged projection matrix is calculated by (3-27) and involves $(LB-1)M^2$ additions and $(LB+1)M^2$ multiplications.

f) **Step S6: polynomial root-finding**. This technique relies on the SVD of a square matrix (companion matrix) whose dimension coincides with the polynomial degree [52]. Then, since the coarray polynomial in (3-26) has degree $2M-2$, this step takes about $32M^3/3 + \mathcal{O}(M^2)$ operations.

3.5 Identifiability

According to the analysis proposed in [53], which establishes the identifiability conditions for noncoherent processing schemes, we can further investigate the properties of type-II arrays regarding identifiability. For that, we assume that the subarrays have an arbitrary number of sensors. The composed coarray manifold for type-II arrays is given by

$$\mathbf{A}_{\mathbb{D}} = \left[\mathbf{A}_{\mathbb{D}_1}^T, \cdots, \mathbf{A}_{\mathbb{D}_L}^T\right]^T \tag{3-30}$$

The maximum number of identifiable sources is given by the sufficient condition $D \leq \lfloor \rho/2 \rfloor$, where ρ is the Kruskal rank of (3-30). Let us analyze two main cases separately: subarrays with the same and a different number of sensors. For subarrays with the same number of sensors, $\rho_{\mathbf{A}_{\mathbb{D}}} = \rho_{\mathbf{A}_{\mathbb{D}_l}}$ and thus the length of the SCA determines the maximum number of identifiable sources. In this particular case, $(M-1)/2$. However, if the subarrays have a different number of sensors, then the identifiability will be limited by the subarray with the largest aperture. In this case, $D \leq \lfloor \max_l(\rho_{\mathbf{A}_{\mathbb{D}_l}})/2 \rfloor$.

3.6 Theoretical bounds on algorithm performance

In this comprehensive section, we construct and analyze the entire suite of Fisher Information Matrices (FIMs) that encapsulate the informational essence inherent to the parameters of our proposed data model. These parameters are crucial to facilitate an accurate estimation of the directional vectors of the sources under study.

Next, we introduce some core concepts. Specifically, the FIMs for a parameter vector $\boldsymbol{\phi}$, within a given statistical model, is defined by the following expression:

$$\text{FIM}(\boldsymbol{\phi}) = \mathbb{E}\left[\left(\frac{\partial \log f(\mathbf{x};\boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right)\left(\frac{\partial \log f(\mathbf{x};\boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right)^T\right] \tag{3-31}$$

where $f(\mathbf{x};\boldsymbol{\phi})$ means the likelihood function of the observed data $\mathbf{x}$, given the parameter vector $\boldsymbol{\phi}$, and $\mathrm{E}\{\cdot\}$ is the expectation operator.

Our discussion goes towards a detailed exploration of these matrices, highlighting their fundamental role in establishing the theoretical limits of algorithmic performance. This exploration is pivotal, as it lays the groundwork for computing the Cramér-Rao Lower Bounds (CRLB) pertinent to our problem domain. The CRLB offers a quantifiable metric for the minimum variance of any unbiased estimator, providing a standard for evaluating the performance of such estimators. Mathematically, the CRLB is defined as follows:

$$\mathrm{Var}(\hat{\boldsymbol{\phi}}) \geq \mathrm{FIM}^{-1}(\boldsymbol{\phi}) \tag{3-32}$$

That said, we derive a CRLB for partially-calibrated arrays. Unlike [53], the proposed CRLB assumes that the impinging source signals are spatially coherent across all subarrays. In the same way, the bound presented in [15] is not applicable under two conditions: 1) when the number of sources exceeds the total number of sensors in all subarrays and 2) when the assumption of uncorrelated sources is considered.

We start this section by noting that while CRLB expressions for partially-calibrated arrays exist in the literature, these established bounds exhibit certain constraints and are not wholly pertinent to the specific context under investigation.

For example, in [15], the CRLB is derived without considering the prior assumption of the uncorrelated sources, which leads to a higher bound than what one would expect for coarray MUSIC-like algorithms. In addition, the inversion of the Gram matrix of the array manifold ($\mathbf{A}^H\mathbf{A}$) is not well defined if there are more sources than sensors in the whole array. As a last limitation, we would like to emphasize that the CRLB shown in [15] only allows the computation of the bound for the directions and calibration parameters, not providing a solution for the sources and noise powers.

Another bound was derived in [53], which assumes statistical orthogonality between sources between distinct subarrays. This implies that the data matrix for the entire array has a block-diagonal structure. Contrary to this assumption, our formulation operates under the premise that the signal sources impinging on the array are spatially coherent across all subarrays, i.e., the subarrays capture information from the same sources. We consider our assumption to be more practical because it can be compared to a fully calibrated array receiving signals from the same sources.

The received signal model for a multiple measurements scenario is given by

$$\mathbf{x}_{\mathbb{S}}(t) = \tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}}\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t),\ t \in [T] \tag{3-33}$$

where

$$\begin{aligned} \tilde{\mathbf{V}}(\boldsymbol{\theta}) &= [\mathbf{V}(\boldsymbol{\theta}_1)| \dots |\mathbf{V}(\boldsymbol{\theta}_D)] \in \mathbb{C}^{N \times LD} \\ \mathbf{V}(\boldsymbol{\theta}_d) &= \mathrm{blkdiag}\left(\mathbf{a}_1(\theta_d), \dots, \mathbf{a}_L(\theta_d)\right) \in \mathbb{C}^{N \times L} \\ \tilde{\mathbf{H}} &= \mathrm{blkdiag}\left(\mathbf{h}_1, \dots, \mathbf{h}_D\right) \in \mathbb{C}^{LD \times D} \\ \mathbf{h}_d &= [1, h_{2d}, h_{3d}, \dots, h_{Ld}]^T \in \mathbb{C}^{L \times 1} \end{aligned} \tag{3-34}$$

for $l \in \{1, \cdots, L\}$ and $t \in \{1, \cdots, T\}$, where L is the number of subarrays and T is the number of snapshots. The vector $\mathbf{x}_{\mathbb{S}}(t) \in \mathbb{C}^N$ represents the array received signal, $\mathbf{s}(t) \in \mathbb{C}^D$ is the source signal vector, and $\mathbf{n}_{\mathbb{S}}(t)$ is the array noise vector. There are D impinging sources with normalized directions given by $\boldsymbol{\theta} \in [-1, 1)^D$ (sine of DOAs).

The noise and the source signals are drawn from a circularly complex multivariate Gaussian distribution. The noise is spatially and temporally white, and the noise and sources are uncorrelated. The vector $\mathbf{h}_d$ contains the calibration parameters associated with all the subarrays (first subarray as reference) for the d-th direction, and $[\mathbf{a}_l(\theta)]_k = \exp(j\pi n_k \theta)$ is the k-th element of the steering vector of the l-th subarray[5], $n_k \in \mathbb{S}_l$ is the k-th sensor position as an integer multiple of the minimum intersensor distance, the set of integers $\mathbb{S}_l$ and N_l define the sensor locations and number of sensor elements of the l-th subarray, respectively.

The received signal in (3-33) can alternatively be rewritten as

$$\begin{aligned} \mathbf{x}_{\mathbb{S}}(t) &= [\mathbf{V}(\boldsymbol{\theta}_1)\mathbf{h}_1| \dots |\mathbf{V}(\boldsymbol{\theta}_D)\mathbf{h}_D]\, \mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t) \\ &= [\mathbf{A}_1(\boldsymbol{\theta}), \mathbf{A}_2(\boldsymbol{\theta})\mathbf{G}_2, \dots, \mathbf{A}_L(\boldsymbol{\theta})\mathbf{G}_L]\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t) \end{aligned} \tag{3-35}$$

where $\mathbf{A}_l(\boldsymbol{\theta}) \in \mathbb{C}^{N_l \times D}$ is the l-th calibrated subarray manifold[6] with geometry defined by the set of integers $\mathbb{S}_l$, and $\mathbf{G}_l = \text{blkdiag}\left([h_{l1}, \dots, h_{lD}]\right)$. The array received signal $\mathbf{x}_{\mathbb{S}}(t)$ can be partitioned as

$$\mathbf{x}_{\mathbb{S}}(t) = \begin{bmatrix} \mathbf{x}_{\mathbb{S}_1}^T(t) & \mathbf{x}_{\mathbb{S}_2}^T(t) & \dots & \mathbf{x}_{\mathbb{S}_L}^T(t) \end{bmatrix}^T \tag{3-36}$$

From that, it is clear that each subarray received signal in (3-36) can be rewritten as

$$\mathbf{x}_{\mathbb{S}_l}(t) = \mathbf{A}_l(\boldsymbol{\theta})\mathbf{G}_l\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}_l}(t) \tag{3-37}$$

$\mathbf{x}_{\mathbb{S}_l}(t) \in \mathbb{C}^{N_l}$ is the l-th subarray received signal snapshot, and $\mathbf{n}_{\mathbb{S}_l}(t)$ is the subarray noise vector measurement[7]. If we calculate the covariance matrix for

[5] Observe that, in this model, we assume the minimum distance between two sensors is half a wavelength ($\lambda/2$) and that the source angles are given by $\theta = \sin(\psi)$, where ψ is the angle between the impinging source signal and the broadside axis of the sensor array.

[6] Notice that the subarrays are calibrated; however, the whole array is only partially calibrated because the differences in calibration parameters between different subarrays are unknown (e.g., their relative distances).

[7] To simplify the equations, we assume equal SNRs across all subarrays, thereby standardizing the noise levels. Considering subarrays with differing SNRs represents a potential avenue for future research.

each of the subarray data, we have

$$\begin{aligned}\mathbf{R}_{\mathbb{S}_l} &= \mathbf{A}_l(\boldsymbol{\theta})\mathbf{G}_l\mathbf{R}_s\mathbf{G}_l^H\mathbf{A}_l^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I}_{N_l} \\ &= \mathbf{A}_l(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}_l^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I}_{N_l}\end{aligned} \tag{3-38}$$

where σ_n^2 is the noise variance, $\mathbf{R}_{\mathbb{S}_l}$ is the received signal covariance matrix of the l-th subarray. In (3-38) we have used the fact that each calibration parameter h_{ld} has a unitary module, according to what follows:

$$\begin{aligned}\mathbf{G}_l\mathbf{R}_s\mathbf{G}_l^H &= \text{blkdiag}\left(h_{l1}p_1h_{l1}^*, \dots, h_{lD}p_1h_{lD}^*\right) \\ &= \text{blkdiag}\left(|h_{l1}|^2p_1, \dots, |h_{lD}|^2p_1\right) \\ &= \text{diag}(\mathbf{p}) \\ &= \mathbf{R}_s\end{aligned} \tag{3-39}$$

where $\mathbf{p} = [\sigma_1^2, \dots, \sigma_D^2]$ contains the powers of the sources.

Remark 1: This shows that, although the subarrays are uncalibrated relative to each other, the covariance matrix of each subarray does not depend on the calibration parameters. In contrast, while the subarrays are individually calibrated, the full array remains uncalibrated. We refer to this condition as a *partially-calibrated array.*

The calibration vectors $\mathbf{h}_d$ contain the calibration parameters associated with all subarrays (first subarray as reference) for the d-th direction, where $d \in [D]$. In order to derive the CRLB, we define the parameter vector as

$$\boldsymbol{\phi} = \left[\boldsymbol{\theta}^T, \mathbf{p}^T, \sigma_n^2, \boldsymbol{\nu}_2^T, \dots, \boldsymbol{\nu}_L^T, \boldsymbol{\eta}_2^T, \dots, \boldsymbol{\eta}_L^T\right] \tag{3-40}$$

where

$$\begin{aligned}\boldsymbol{\nu}_l &= \mathfrak{Re}\left\{[\mathbf{h}_{l,1}, \dots, \mathbf{h}_{l,D}]^T\right\} \\ \boldsymbol{\eta}_l &= \mathfrak{Im}\left\{[\mathbf{h}_{l,1}, \dots, \mathbf{h}_{l,D}]^T\right\}\end{aligned} \tag{3-41}$$

Let us find the Fisher Information Matrix (FIM) associated with the estimation of the vector $\boldsymbol{\phi}$, that can be alternatively (see (3-31)) defined as

$$\text{FIM} = -E\left\{\frac{\partial^2 \ln p(\mathbf{x}_{\mathbb{S}}(t); \boldsymbol{\phi})}{\partial\phi_i\partial\phi_j}\right\}, \tag{3-42}$$

where $p(\mathbf{x}_{\mathbb{S}}(t); \boldsymbol{\phi})$ is the probability density function of $\mathbf{x}_{\mathbb{S}}(t)$ which depends on the nonrandom parameter vector $\boldsymbol{\phi}$. The covariance matrix of the data model is such that $x_{\mathbb{S}}(t) \sim \mathcal{CN}(0, \mathbf{R}_{\mathbb{S}})$ and

$$\mathbf{R}_{\mathbb{S}} = \tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}}\mathbf{R}_s\tilde{\mathbf{H}}^H\tilde{\mathbf{V}}^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I} \tag{3-43}$$

hereafter denoted as $\mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})$ to emphasize the dependence of the distribution on the parameter vector $\boldsymbol{\phi}$. The log-likelihood for a single snapshot is given by [10]

$$\mathcal{L}_{\mathbf{x}}(\boldsymbol{\phi}) = -\ln|\pi\mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})| - \mathbf{x}^H\mathbf{R}_{\mathbb{S}}^{-1}(\boldsymbol{\phi})\mathbf{x} \tag{3-44}$$

The elements of the FIM are given by

$$\begin{aligned}[\text{FIM}]_{ij} &= -E\left[\frac{\partial^2\mathcal{L}_{\mathbf{x}}(\boldsymbol{\phi})}{\partial\phi_i\partial\phi_j}\right] \\ &= T\,\text{tr}\left[\mathbf{R}_{\mathbb{S}}^{-1}(\boldsymbol{\phi})\frac{\partial\mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial\phi_i}\mathbf{R}_{\mathbb{S}}^{-1}(\boldsymbol{\phi})\frac{\partial\mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial\phi_j}\right]\end{aligned} \tag{3-45}$$

Before we calculate the derivatives in (3-45), we first divide the parameter vector into two blocks, as follows

$$\boldsymbol{\phi} = \begin{bmatrix}\boldsymbol{\phi}_1^T & \boldsymbol{\phi}_2^T\end{bmatrix}^T \tag{3-46}$$

where $\boldsymbol{\phi}_1 = \boldsymbol{\theta}$ and $\boldsymbol{\phi}_2 = \left[\mathbf{p}^T, \sigma_n^2, \boldsymbol{\nu}_2^T, \dots, \boldsymbol{\nu}_L^T, \boldsymbol{\eta}_2^T, \dots, \boldsymbol{\eta}_L^T\right]^T$. The FIM matrix can be partitioned as

$$\text{FIM} = \begin{bmatrix}\text{FIM}_{\boldsymbol{\phi}_1\boldsymbol{\phi}_1} & \text{FIM}_{\boldsymbol{\phi}_1\boldsymbol{\phi}_2} \\ \text{FIM}_{\boldsymbol{\phi}_2\boldsymbol{\phi}_1} & \text{FIM}_{\boldsymbol{\phi}_2\boldsymbol{\phi}_2}\end{bmatrix} \tag{3-47}$$

The CRLB matrix is defined as

$$\mathbf{C}_{\text{CRLB}}(\boldsymbol{\phi}) = \begin{bmatrix}\mathbf{C}_{\text{CRLB}}(\boldsymbol{\phi}_1\boldsymbol{\phi}_1) & \mathbf{C}_{\text{CRLB}}(\boldsymbol{\phi}_1\boldsymbol{\phi}_2) \\ \mathbf{C}_{\text{CRLB}}(\boldsymbol{\phi}_2\boldsymbol{\phi}_1) & \mathbf{C}_{\text{CRLB}}(\boldsymbol{\phi}_2\boldsymbol{\phi}_2)\end{bmatrix} \tag{3-48}$$

Due to the fact that $\mathbf{C}_{\text{CRLB}}(\boldsymbol{\phi}) = \text{FIM}^{-1}$ and using the inverse formula for a block-partitioned matrix, we have

$$\mathbf{C}_{\text{CRLB}}(\boldsymbol{\phi}_1\boldsymbol{\phi}_1) = \left[\text{FIM}_{\boldsymbol{\phi}_1\boldsymbol{\phi}_1} - \text{FIM}_{\boldsymbol{\phi}_1\boldsymbol{\phi}_2}\text{FIM}_{\boldsymbol{\phi}_2\boldsymbol{\phi}_2}^{-1}\text{FIM}_{\boldsymbol{\phi}_2\boldsymbol{\phi}_1}\right]^{-1} \tag{3-49}$$

Since we are mainly interested in the estimation of the direction parameters, we are seeking the matrix $\mathbf{C}_{\text{CRLB}}(\boldsymbol{\phi}_1\boldsymbol{\phi}_1)$. Let us proceed with the computation of each of the FIM blocks in (3-49) using the expression in (3-45) with the corresponding partial derivatives.

In order to do so, we calculate the FIM matrices related to the variables in the parameter vector $\boldsymbol{\phi}$, as well as their crossed FIMs. The key in the derivation process is to:

- Compute the derivatives $\partial\mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})/\partial\phi_i$;
- Put the resulting expression in (3-45) in a compact form. We proceed as follows:

In order to prioritize a more fluid text reading, we state the detailed calculation of the derivatives and FIM formulas in Appendix A. In what follows, we describe only the main results.

FIM matrices

1. FIM matrix with respect to the DOAs $\boldsymbol{\theta}$

$$\begin{aligned}&\mathrm{FIM}_{\boldsymbol{\theta}\boldsymbol{\theta}} = \\ &2T\mathfrak{Re}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\ &+ (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T\}\end{aligned} \tag{3-50}$$

   where $\odot$ denotes the Hadamard product.

2. FIM matrix with respect to the sources powers $\mathbf{p}$

$$\mathrm{FIM}_{\mathbf{p}\mathbf{p}} = T(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}) \odot (\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W})^T \tag{3-51}$$

3. FIM matrix with respect to the noise power σ_n^2

$$\mathrm{FIM}_{\sigma_n^2\sigma_n^2} = T\,\mathrm{tr}\left[\mathbf{R}_{\mathbb{S}}^{-2}\right] \tag{3-52}$$

4. FIM matrix with respect to the real part of the calibration parameters $\boldsymbol{\nu}_l$

$$\begin{aligned}&\mathrm{FIM}_{\boldsymbol{\nu}_l\boldsymbol{\nu}_k} = \\ &2T\mathfrak{Re}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T \\ &+ (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T\}\end{aligned} \tag{3-53}$$

5. FIM matrix with respect to the imaginary part of the calibration parameters $\boldsymbol{\eta}_l$

$$\begin{aligned}&\mathrm{FIM}_{\boldsymbol{\eta}_l\boldsymbol{\eta}_k} = \\ &-2T\mathfrak{Re}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T \\ &- (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T\}\end{aligned} \tag{3-54}$$

6. FIM matrix with respect to $\boldsymbol{\theta}$ and $\mathbf{p}$

$$\mathrm{FIM}_{\boldsymbol{\theta}\mathbf{p}} = 2T\mathfrak{Re}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}) \odot (\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T\} \tag{3-55}$$

7. FIM matrix with respect to $\boldsymbol{\theta}$ and σ_n^2

$$\mathrm{FIM}_{\boldsymbol{\theta}\sigma_n^2} = 2T\,\mathrm{diag}^M(\mathfrak{Re}\left\{\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{D}\right\}) \tag{3-56}$$

where $\text{diag}^M(\mathbf{X})$ is a column vector whose elements consists of the diagonal elements of square-matrix $\mathbf{X}$.

8. FIM matrix with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\nu}_l$

$$\begin{aligned} &\text{FIM}_{\boldsymbol{\theta}\boldsymbol{\nu}_l} = \\ &= T\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\ &+ (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\ &+ (\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T \\ &+ (\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T\} \end{aligned} \tag{3-57}$$

9. FIM matrix with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\eta}_l$

$$\begin{aligned} &\text{FIM}_{\boldsymbol{\theta}\boldsymbol{\eta}_l} = \\ &= jT\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\ &- (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\ &+ (\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T \\ &- (\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T\} \end{aligned} \tag{3-58}$$

10. FIM matrix with respect to $\mathbf{p}$ and $\boldsymbol{\sigma}_n^2$

$$\text{FIM}_{\mathbf{p}\sigma_n^2} = T\,\text{diag}^M(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{W}) \tag{3-59}$$

11. FIM matrix with respect to $\mathbf{p}$ and $\boldsymbol{\nu}_l$

$$\begin{aligned} &\text{FIM}_{\mathbf{p}\boldsymbol{\nu}_l} = \\ &= 2T\mathfrak{Re}\left\{\left(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right) \odot \left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\right)^T\right\} \end{aligned} \tag{3-60}$$

12. FIM matrix with respect to $\mathbf{p}$ and $\boldsymbol{\eta}_l$

$$\begin{aligned} &\text{FIM}_{\mathbf{p}\boldsymbol{\eta}_l} = \\ &- 2T\mathfrak{Im}\{(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W})^T\} \end{aligned} \tag{3-61}$$

13. FIM matrix with respect to σ_n^2 and $\boldsymbol{\nu}_l$

$$\mathrm{FIM}_{\sigma_n^2 \boldsymbol{\nu}_l} = 2T\,\mathrm{diag}^{\mathrm{M}}(\mathfrak{Re}\{\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\})^T \tag{3-62}$$

14. FIM matrix with respect to σ_n^2 and $\boldsymbol{\eta}_l$

$$\mathrm{FIM}_{\sigma_n^2 \boldsymbol{\eta}_l} = -2T\,\mathrm{diag}^{\mathrm{M}}(\mathfrak{Im}\{\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\})^T \tag{3-63}$$

15. FIM matrix with respect to $\boldsymbol{\nu}_l$ and $\boldsymbol{\eta}_k$

$$\begin{aligned}&\mathrm{FIM}_{\boldsymbol{\nu}_l\boldsymbol{\eta}_k} = \\ &-2T\mathfrak{Im}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T \\ &+ (\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T\}\end{aligned} \tag{3-64}$$

Since the general FIM is symmetric, the above expressions are exhaustive in describing the amount of information from all the model parameters embedded in the data. By taking the FIM inverse using (3-49), we have the CRLB for the DOAs. However, beyond yielding the CRLB for DOA estimates, these expressions can be used to calculate the CRLB for any parameter in the data model (3-33).

They can even be used for the calculus of the CRLB of the fully calibrated array if we set $L = 1$ and remove the rows and columns corresponding to the calibration parameters. In addition, they do not have any limitations with the quantity of sensors in relation to the number of sources and consider the uncorrelated prior hypothesis for the sources, which is a core assumption for algorithms like GCA-MUSIC and GCA-rMUSIC.

3.7 Relationship between data models

Given that there might be an impression that the parametric data model used to derive the algorithms differs from the one discussed in this chapter, it is important to establish the connection between both data models. In order to establish a connection between the model (3-6) and the model in (3-33), that shows the calibration parameters, we proceed as follows. The received signal can be written as

$$\begin{aligned}\mathbf{x}_{\mathbb{S}}(t) &= \tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}}\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t) \\ &= \left[\mathbf{V}(\boldsymbol{\theta}_1)\mathbf{h}_1 | \dots |\mathbf{V}(\boldsymbol{\theta}_D)\mathbf{h}_D\right]\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t) \\ &= \begin{bmatrix}\mathbf{A}_1(\boldsymbol{\theta}) \\ \mathbf{A}_2(\boldsymbol{\theta})\mathbf{G}_2 \\ \vdots \\ \mathbf{A}_L(\boldsymbol{\theta})\mathbf{G}_L\end{bmatrix}\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t)\end{aligned} \tag{3-65}$$

where $\mathbf{A}_l(\boldsymbol{\theta})$ is the subarray manifold, which is supposed to be calibrated and $\mathbf{G}_l = \operatorname{blkdiag}([h_{l1}, \dots, h_{LD}])$. The array received signal $\mathbf{x}_{\mathbb{S}}(t)$ can be partitioned as

$$\mathbf{x}_{\mathbb{S}}(t) = \left[\mathbf{x}_{\mathbb{S}_1}^T(t) \quad \mathbf{x}_{\mathbb{S}_2}^T(t) \quad \dots \quad \mathbf{x}_{\mathbb{S}_L}^T(t)\right]^T \tag{3-66}$$

From that, it is clear that each subarray received signal in (3-66) can be rewritten as

$$\mathbf{x}_{\mathbb{S}_l}(t) = \mathbf{A}_l(\boldsymbol{\theta})\mathbf{G}_l\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}_l}(t) \tag{3-67}$$

If we calculate the covariance matrix for each of the subarray data, we have

$$\begin{aligned}\mathbf{R}_{\mathbb{S}_l} &= \mathbf{A}_l(\boldsymbol{\theta})\mathbf{G}_l\mathbf{R}_s\mathbf{G}_l^H\mathbf{A}_l^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I} \\ &= \mathbf{A}_l(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}_l^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I}\end{aligned} \tag{3-68}$$

which is the same as described in (3-38). This is because

$$\begin{aligned}\mathbf{G}_l\mathbf{R}_s\mathbf{G}_l^H &= \operatorname{blkdiag}\left(h_{l1}p_1h_{l1}^*, \dots, h_{lD}p_1h_{lD}^*\right) \\ &= \operatorname{blkdiag}\left(|h_{l1}|^2p_1, \dots, |h_{lD}|^2p_1\right) \\ &= \operatorname{diag}(\mathbf{p}) \\ &= \mathbf{R}_s\end{aligned} \tag{3-69}$$

where we have used the fact that each calibration parameter h_{ld} has a unitary module. This establishes the connection between both data models, i.e., they lead to the same covariance matrices for each of the subarrays.

It is important to highlight the fact that both data models, despite any initial appearance of divergence, ultimately converge in terms of presenting identical second-order statistical characteristics when originating from the same dataset. This convergence is not merely coincidental but rather a core observation, as it implies a fundamental equivalence in the performance of the estimation algorithms under analysis, as well as the derived FIM matrices, with respect to the sources DoA.

The quality of the estimated source directions, as derived from these algorithms, remains invariant across both models. This invariance is not only a critical aspect we aim to demonstrate but also aligns with our prior expectations. It serves as a crucial justification for employing these two models interchangeably within the framework of our analytical derivations. Such interchangeability is crucial, as it underpins the consistency and robustness of the reasoning that guides the selection and application of these models in our context.

3.8 Numerical results

In this section, we employ simulations to assess the algorithms' capabilities and demonstrate the applicability of the CRLB developed in this work. The performance curves are shown in terms of root mean squared-error (RMSE) in two distinct ways: comparison of the algorithms for a given geometry and comparison of the geometries for a given algorithm. All the results were obtained from an average of 50,000 simulation trials.

In what follows, we outline the parameter setting, emphasizing that the number of signal sources surpasses the number of sensor elements within each subarray.

1. Number of snapshots: $T = 2000$ (for the RMSE against SNR plots)

2. Number of subarrays: $L = 2$

3. SNR $= 0$ dB (for the RMSE against T plots)

4. Type-II array with $N_l = 7$ sensors (total of $2N_l = 14$ sensors) and the following geometries for the reference subarray: a) II-MRA: $\mathbb{S}_l = \{0, 1, 4, 10, 12, 15, 17\}$; b) II-SNAQ2: $\mathbb{S}_l = \{0, 2, 3, 6, 9, 13, 14\}$; c) II-NAQ2: $\mathbb{S}_l = \{0, 1, 2, 3, 4, 9, 14\}$

5. Normalized intersubarray displacement: $\mu = 8$, which implies that $\mathbf{h}(\theta_d) = [1, \exp(j\pi(\mu + \kappa)\theta_d)]^T$ corresponds to the vector of calibration parameters for $\theta_d = [\boldsymbol{\theta}]_d$

6. $D = 11$ sources with directions given by: $\boldsymbol{\theta} = [\pm 0.75, \pm 0.6, \pm 0.45, \pm 0.3, \pm 0.15, 0]^T$

3.8.1 Analysis of degrees of freedom of full array and subarrays

In this section, we aim to characterize the weight function of the full array based on the weight function of the subarrays for Type II-MRA. To achieve this, we use the results from Theorem 3.3, derived in Section 3.1.1. We observe that the intersubarray displacement is denoted by $\mu = 8$ and the aperture of each subarray is given by $\kappa = 17 - 0 = 17$. Additionally, the computation of the weight functions of the subarrays, as shown in the first two subplots of Figure 3.6, indicates that the subarrays DoF and UDoF are equal to sDoF = UDoF = 35. Consequently, Theorem 3.3 dictates that the total DoF of the full array is given by

$$\begin{aligned} \text{DoF} &= L(\text{sDoF} - 1) + 2(L-1)\mu + 1 \\ &= 2 \cdot (35 - 1) + 2(2-1)8 + 1 = 85 \end{aligned} \tag{3-70}$$

which agrees with the number of non-zero elements in the weight function of the entire array, as depicted in the fifth subplot of Figure 3.6. It is also important to note also that this final weight function represents the sum of the first four weight functions. The weight functions shown in the third and fourth subplots are simply the weight functions of the subarrays translated by $\Delta = \pm(\mu + \kappa) = \pm 25$, as expected.

3.8.2 Computational complexity

The primary objective of GCA-(r)MUSIC is to perform DOA estimation when the number of sources is greater than that of sensors under two distinct scenarios: a) in fully calibrated arrays, where we advocate for non-coherent data processing to reduce hardware complexity and computational costs; and (b) in scenarios where calibration parameters remain unidentified, recommending the use of these algorithms to capitalize on their capability to estimate the DOAs with high performance, outperforming existing algorithms tailored for partially-calibrated scenarios. Regarding computational complexity, our approach is benchmarked in the first scenario against the SS-MUSIC (or coarray MUSIC) algorithm [20].

For the subsequent scenario, we draw a comparison with Spectral RARE and G-MUSIC, tailored for partially-calibrated arrays [13, 15]. Notably, while other algorithms in the literature address partially-calibrated arrays, they were

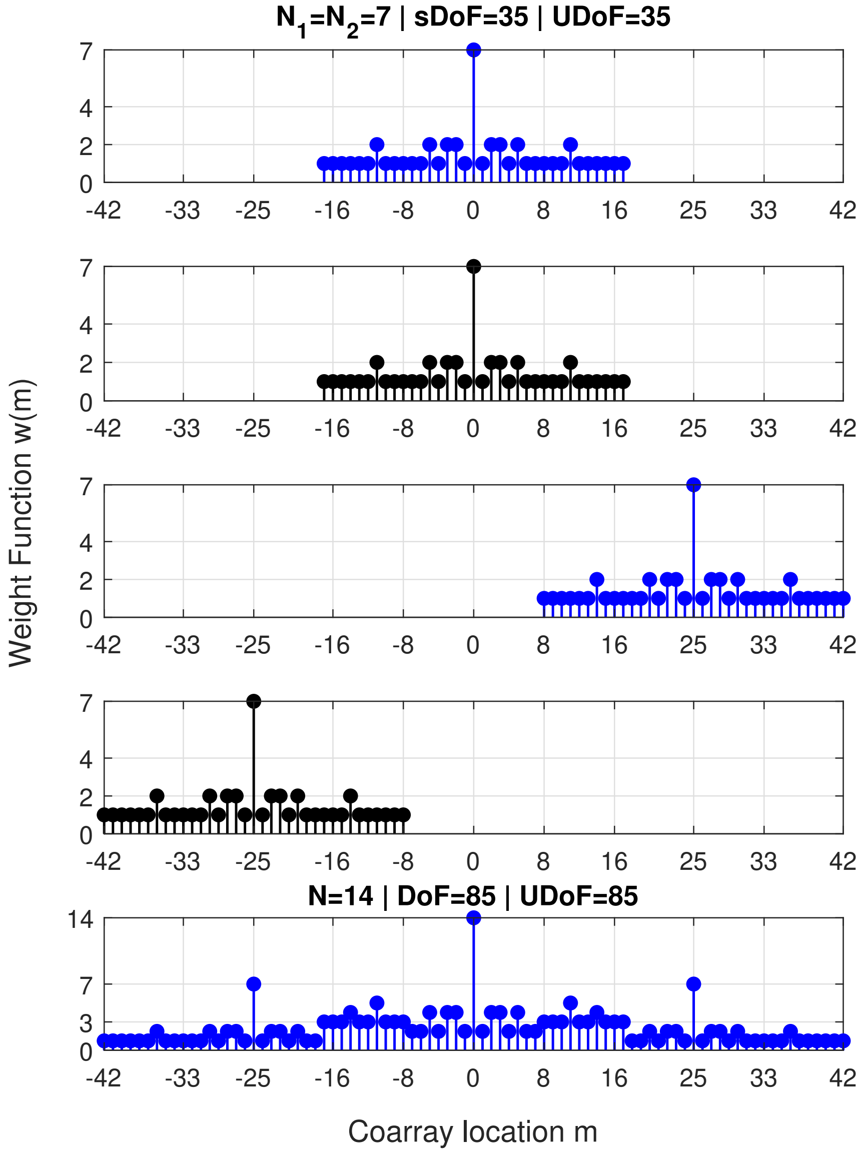


Figure 3.6: Weight function of the full array as a result of the weight function of subarrays for II-MRA.

Table 3.2: Runtime of GCA-(r)MUSIC against established approaches.

Algorithm	Runtime (Norm. Runtime)	DOA RMSE
GCA-rMUSIC	1.61 ms (1.00)	1.0943e-3
GCA-MUSIC	2.10 ms(1.306)	1.0954e-3
SS-MUSIC	12.15 ms (7.54)	0.15e-3
Spectral RARE	6.96 ms (4.32)	8.45e-3
G-MUSIC	2.45 ms (1.52)	0.99629

excluded from our comparison due to their applicability solely to ULAs [54] or because their underlying signal model presupposes that the cross-covariance between subarrays is zero [53], a condition not applicable in our context.

In the subsequent simulation, we utilized a simulation server with 128 GB of RAM and an Intel Core i9-10980XE CPU at 3.0 GHz. The coding for all methods was optimally performed to minimize the number of required flops as much as possible. The normalized spacing between subarrays was assumed known for the SS-MUSIC algorithm (calibrated scenarios only) and unknown for the other four algorithms (GCA-(r)MUSIC, G-MUSIC and Spectral RARE). The spacing between subarrays constitutes our calibration information. The results concerning runtime, averaged over 50,000 simulation trials, are presented in Table 3.2.

It is evident that GCA-MUSIC and GCA-rMUSIC run significantly faster than the other three methods. Moreover, G-MUSIC is not capable of identifying the sources because of its identifiability issues when the number of sources is greater than that of sensors within a subarray. We also highlight that SS-MUSIC is approximately 12 times slower than the proposed approach and only can be used if all of the calibration parameters are known (in our case, the distance between subarrays). Despite those limitations, with a much higher computational and required hardware complexity, it provides the lowest RMSE because it does the processing using the calibration parameters, as expected. Among the algorithms designed for partially-calibrated scenarios, GCA-rMUSIC and GCA-MUSIC run approximately 4.3 and 3.3 times faster than Spectral MUSIC, respectively.

3.8.3 CRLB comparison

Regarding the CRLB, we consider five bounds:

1. CRLB-FC: fully calibrated array without any special structure for the sources covariance matrix [55];

2. CRLB-FC-UP: fully calibrated array assuming the sources are uncorrelated *a priori* - (*Uncorrelated Prior*), originally obtained in [56] and extended to deal with sparse arrays for the underdetermined case in [23]. It assumes the sources covariance matrix is diagonal;

3. CRLB-PC: partially-calibrated array with sources covariance matrix without any special structure [15];

4. **CRLB-PC-UP-PROP**: partially-calibrated array assuming the sources are uncorrelated *a priori* (sources covariance matrix with diagonal structure), which is our *proposed bound*; and

5. CRLB-PC-UXCov: partially-calibrated array assuming the cross-covariances between subarrays are zero-valued, i.e., the sources impinging on different subarrays are uncorrelated, proposed in [53].

The proposed CRLB-PC-UP-PROP bound is a unique contribution to the literature, providing a lower bound for the estimation performance of DOAs. This CRLB uniquely incorporates two critical assumptions: the uncorrelated nature of the sources, which is a fundamental consideration in coarray MUSIC algorithms, and the partial calibration of arrays. Unlike existing bounds such as CRLB-PC, CRLB-FC, and CRLB-PC-UXCov, the proposed bound overcomes numerical challenges encountered when the number of sources exceeds the number of sensors in the full array. This distinction not only underscores the practicality of our approach but also its applicability in scenarios often encountered in array signal processing.

The CRLBs above are depicted in Figure 3.7. Among these bounds, it is evident that those assuming uncorrelated sources (appended by “-UP”), although initially lower, converge to the same value as its counterpart as the SNR increases. This convergence is observed in the pairs CRLB-FC/CRLB-FC-UP and CRLB-PC/CRLB-PC-UP-PROP. This phenomenon indicates that the uncertainty in the off-diagonal parameters of the sources’ covariance matrix becomes negligible for the DOA estimation performance in substantially high SNR scenarios. Note that CRLB-PC-UP-PROP differs significantly from CRLB-PC for SNR values below 10 dB.

We also highlight that CRLB-FC-UP represents the bound with the lowest values because it corresponds to a scenario with a fully calibrated array and uncorrelated sources, resulting in a lower level of uncertainty for the DOA estimation algorithms. Reduced uncertainty leads to improved performance in estimation procedures. Moreover, we note that CRLB-PC-UXCov is the

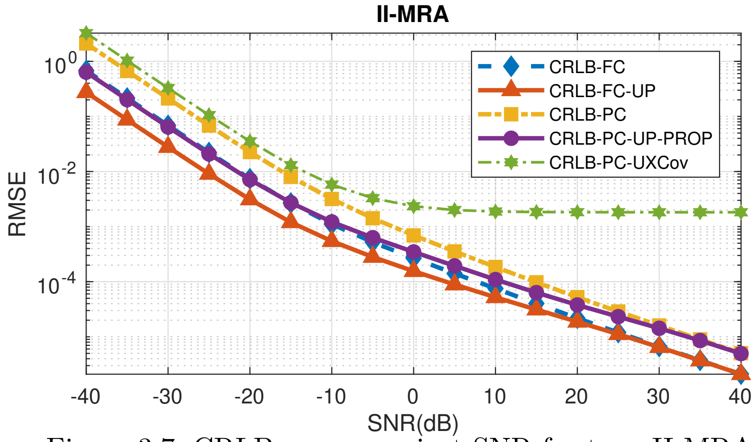


Figure 3.7: CRLB curves against SNR for type-II MRA.

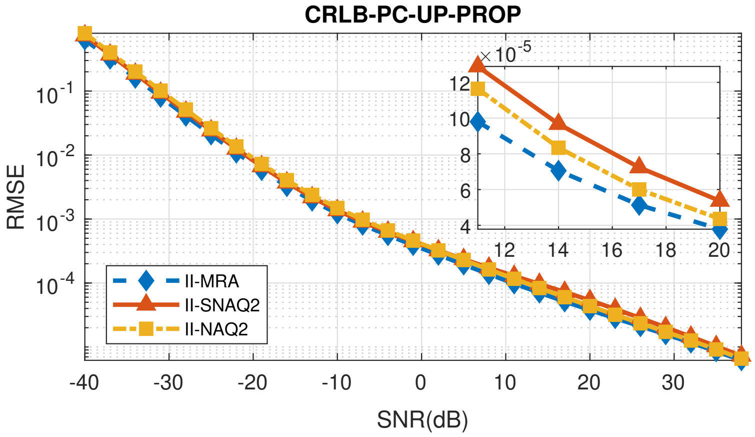


Figure 3.8: Proposed CRLB-PC-UP-PROP against SNR for II-MRA, II-SNAQ2 and II-NAQ2.

highest bound and, contrary to the other CRLB curves, it reaches a plateau after 10 dB. This behaviour, as expected and explained in [53], is due to the subarrays' covariance matrices not being full rank (i.e., more sources than sensors in each subarray). It is also important to mention that by employing the FIM expressions in Section 3.6, we can estimate the CRLB for all parameters of the data model, not only the DOAs. This is because these matrices are explicitly provided, which extends the applicability of the work presented here to many parameter estimation algorithms.

In Figure 3.8, we present a comparison of the three type-II array geometries in terms of the CRLB using CRLB-PC-UP-PROP. The curves, plotted against SNR, suggest that DOA estimation algorithms are likely to perform better with the type II-MRA geometry. This geometry consistently exhibits the lowest CRLB values. This theoretical aspect is corroborated by practical outcomes, as will be evidenced in Section 3.8.4. To allow a clearer comparison, we focused on the SNR range from 10 dB to 20 dB, displaying the curves in the upper-right square box in the figure.

3.8.4 RMSE performance comparison

Before we enter into the RMSE discussion, it is worth exploring the expected RMSE for this scenario. Consider that a given estimation procedure obtains estimates of a random angle $\hat{\theta}_d$ from $(-1, 1]$. Furthermore, suppose that this random variable follows a uniform distribution $\hat{\theta}_d \sim \mathcal{U}(-1, 1)$. Then, the RMSE for this naive algorithm would be given by $\mathrm{RMSE}^{\mathrm{naive}} = \sqrt{\frac{1}{3} + \frac{1}{D}\sum_{d=1}^{D}\theta_d^2}$. Indeed, for this scenario, we have $\mathrm{RMSE}^{\mathrm{naive}} = 0.7472$. Then, for an RMSE smaller than this, the estimator is better than a random choice for the DOAs. This is useful as a benchmark for challenging scenarios in terms of data availability and SNR. Although it is not formally an upper bound on the estimator's RMSE, it is a metric to which we should pay attention.

Next, we present the RMSE curves in Figure 3.9, where the proposed algorithms are able to identify the sources in scenarios with more sources than sensors in each subarray and that GCA-rMUSIC slightly outperforms GCA-MUSIC for lower SNR values. Although SS-MUSIC presents the best performance, it only applies to calibrated schemes, has a higher computational burden, and we show its RMSE only for comparison purposes. The CRLB curve of our data model, as it was derived in Section 3.6, CRLB-PC-UP-PROP, shows that our algorithms get closer to optimum in the interval -17 dB up to -5 dB, saturating the performance as the SNR increases, which agrees with the analysis developed in the fully calibrated case for coarray MUSIC [57].

The RMSE value of 10^{-2} shows that Spectral RARE only achieves the performance of the proposed algorithms after the SNR increases by 15 dB. Additionally, we also observe that G-MUSIC fails to estimate the DOA of the sources because the number of sources exceeds the number of sensors in each subarray. It should be noted that optimization-based algorithms, as presented in [14] and [53], were excluded from our comparative analysis due to their significantly greater computational complexity (convex optimization iterative schemes without closed-form solutions) for the former and assumptions of zero-valued cross-covariance between subarrays for the latter. Furthermore, these algorithms are based on data models that differ from the data model used in this paper. Similar conclusions can be drawn from the RMSE curves against the snapshots in Figure 3.10.

In Figure 3.11 and Figure 3.12, we can verify that the performance varies largely with the geometry for both algorithms and that II-MRA presents the best estimation performance due to its larger number of DoF (virtual aperture), according to what was expected from the analysis of Figure 3.8 in the previous section. The performance of II-MRA is achieved by II-SNAQ2

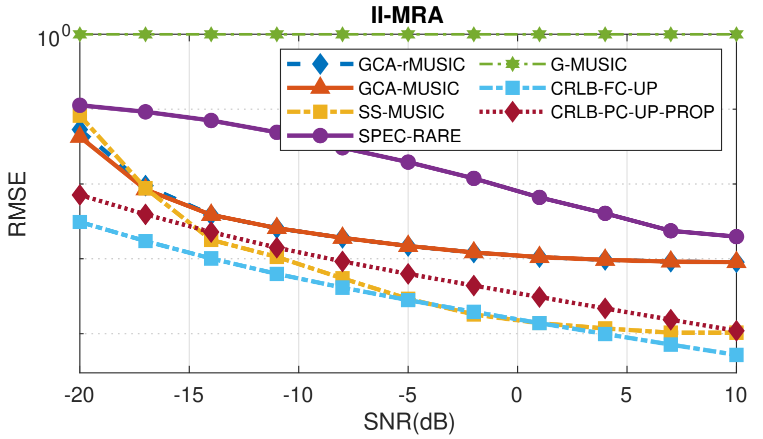


Figure 3.9: Algorithm comparison. RMSE curves against SNR. GCA-MUSIC and GCA-rMUSIC for type-II arrays.

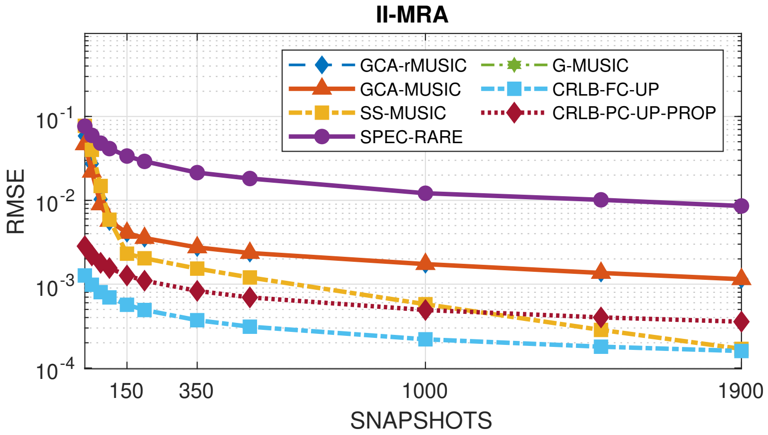


Figure 3.10: Algorithm comparison. RMSE curves against SNAPSHOTS. GCA-MUSIC and GCA-rMUSIC for type-II arrays.

only after the thresholding of -5 dB for the SNR range.

3.9 Summary

In this chapter, we have proposed design methodologies for partially calibrated sparse linear subarrays, along with algorithms for direction-of-arrival (DOA) estimation. In this regard, we introduced array architectures that incorporate two distinct categories, specifically Type-I and Type-II arrays. Type-I arrays partition a known sparse linear geometry into the required number of subarrays, while Type-II arrays configure each subarray to conform to a predefined sparse linear geometry. Furthermore, we also developed two DOA estimation algorithms tailored to partially-calibrated array scenarios within the coarray domain.

These algorithms are capable of estimating a greater number of sources than the available physical sensors within each subarray, while maintaining hardware and computational complexity at levels suitable for real-time

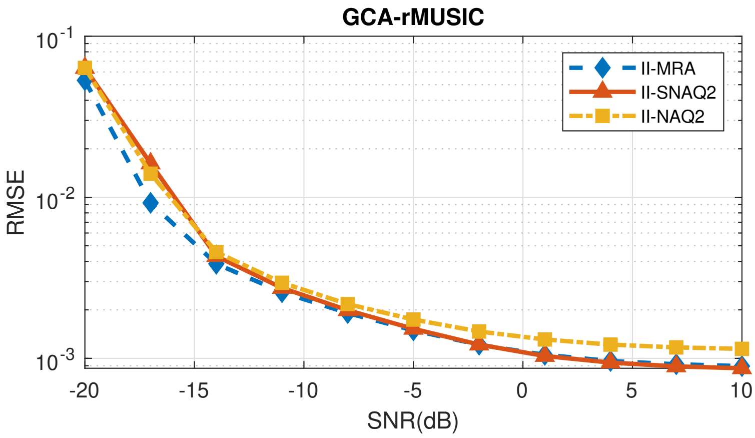


Figure 3.11: Geometry comparison for GCA-root MUSIC. RMSE curves against SNR. Type-II MRA, NAQ2 and SNAQ2.

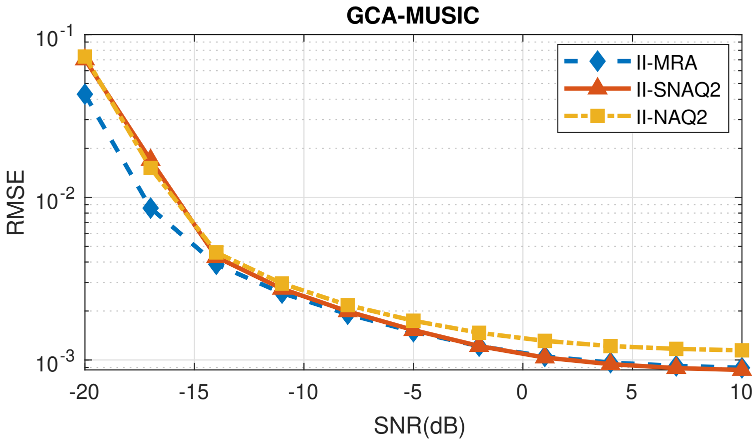


Figure 3.12: Geometry comparison for GCA-MUSIC. RMSE curves against SNR. Type-II MRA, NAQ2 and SNAQ2.

applications. The methods rely on the intersection of projections onto convex sets (affine spaces), as in the case of GCA-MUSIC, in conjunction with a refined projection matrix associated with the noise subspace, as proposed in the GCA-rMUSIC algorithm. Additionally, we analyzed and proposed a closed-form expression for obtaining the subarray degrees of freedom as a function of the overall array's degrees of freedom. Finally, we derived the Fisher Information Matrices and Cramér-Rao Lower Bounds expressions for the adopted data model to establish the theoretical performance limits of the proposed methods.

4
Spatial smoothing techniques for sparse arrays

Spatial smoothing procedures are common methods in sensor array processing that are typically used to deal with fully coherent sources (perfectly correlated). These sources pose significant challenges in signal processing because of the ambiguities introduced in linear algebra-based equations. Without appropriate methods to decorrelate these sources, conventional array processing algorithms often fail, leading to inaccurate results.

In this chapter, we focus on the fundamental principles and methods of spatial smoothing. Spatial smoothing involves the subdivision of an antenna array into overlapping subarrays, as previously discussed in Chapter 3, enabling the effective decorrelation of coherent signals. This process transforms a coherent scenario into an incoherent one, allowing standard array processing techniques to function correctly, as is the case of subspace-based methods for DoA estimation.

The importance of spatial smoothing cannot be overstated. It is a core procedure in environments where signals are likely to be highly correlated, such as multipath propagation scenarios in wireless communications or sonar systems where multiple reflections create coherent paths. By applying spatial smoothing, the performance and reliability of DOA estimation algorithms can be significantly improved.

A substantial body of literature addresses issues related to highly and fully correlated (coherent) sources, focusing on methods to decorrelate them to enable accurate DOA estimation [58]. Such efforts are critical to effectively handle correlated sources [10, 59]. Furthermore, this challenge can also arise in scenarios involving uncorrelated sources, due to the virtual induced correlation, when pre-processing techniques are used to improve the degrees of freedom (DOF) of sparse linear array (SLA) geometries, that is, its ability to estimate a larger number of sources. For example, certain coarray transformations, such as the vectorization of the received signal covariance matrix, increase the number of DOF but also induce coherence among sources in the coarray domain [1], requiring specific techniques to decorrelate them.

In summary, in this work the sources are assumed to be uncorrelated but become fully coherent after a specific linear algebra transformation performed during one of the signal processing stages. We refer to this phenomenon as *virtual correlation.* This virtual correlation arises from the vectorization of the covariance matrix, which is employed to exploit the additional degrees of freedom provided by coarray processing. Whether the coherence arises from the physical or virtual domain, spatial smoothing techniques are employed to enable subspace-based algorithms to function effectively.

We will begin by outlining the specific problem posed by fully coherent sources and why they disrupt conventional processing methods. Following this, we will introduce the concept of spatial smoothing, detailing the mathematical foundations and procedural steps involved. Key spatial smoothing techniques, including forward-backward spatial smoothing and other variants, will be examined in detail, highlighting their application contexts and performance benefits. From that we will migrate those concepts to a sparse geometry scenario and extrapolate some of the results to deal with the more challenging sparse sampling scenario.

To illustrate this, consider the layer-based model of DoA estimation that we have built in Fig. 4.1 for didactic purposes. This figure divides the DoA estimation process into five layers, which are described in Table 4.1.

Table 4.1: Layers of DoA estimation in Fig. 4.1

Index	**Layer**	**Description**	**Implementing Agent**
1	sources	In this layer, the sources are traveling through a homogeneous physical medium	Far-field sources emitting narrow-band propagating signals
2	physical data	This layer comprehends the complex baseband data acquired by array hardware through the RF chains	Array hardware along with its RF chains
3	transformation	Represents the mathematical transformation on statistical data to make it appear as if it was originated from a ULA geometry	Vectorization of covariance matrix, Classical interpolation, Wiener interpolation etc
4	spatial smoothing	Source decorrelation process	FBA, FSS, FBSS, ESS
5	subspace decomposition	Separation of resulting second-order statistics to distinguish between signal and noise subspaces, ultimately resulting in DoA estimates	MUSIC, ESPRIT

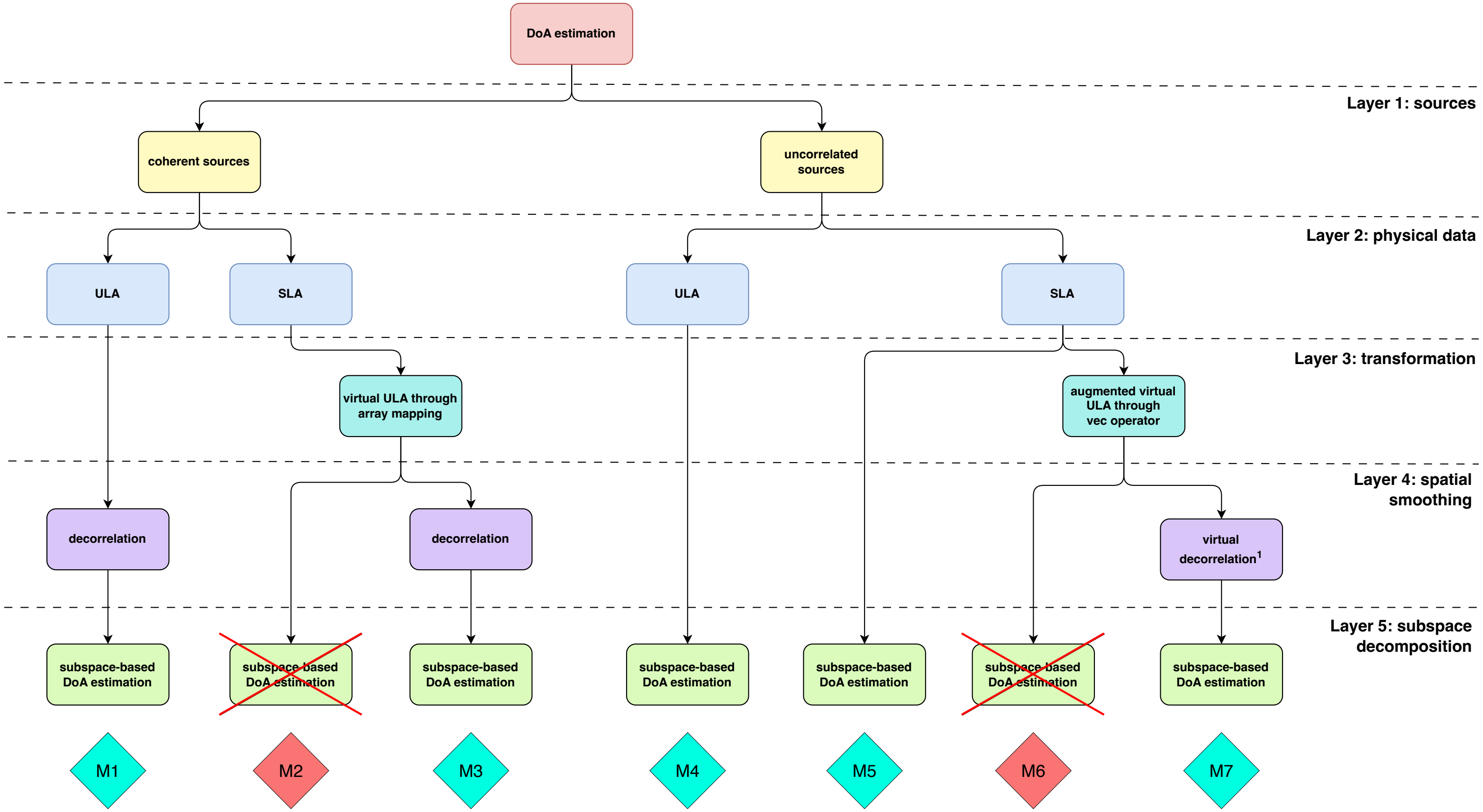

DoA estimation
Layer 1: sources
coherent sources
uncorrelated sources
Layer 2: physical data
ULA
SLA
ULA
SLA
Layer 3: transformation
virtual ULA through array mapping
augmented virtual ULA through vec operator
Layer 4: spatial smoothing
decorrelation
decorrelation
virtual decorrelation1
Layer 5: subspace decomposition
subspace-based DoA estimation
subspace-based DoA estimation
subspace-based DoA estimation
subspace-based DoA estimation
subspace-based DoA estimation
subspace-based DoA estimation
subspace-based DoA estimation
M1
M2
M3
M4
M5
M6
M7

Figure 4.1: Schematic with layers for DOA estimation procedures. [1]The virtual decorrelation corresponds to the decorrelation of the virtually coherent sources $\mathbf{p}$ in (4-8).

Each layer represents a different step for the DoA estimation procedure. In fact, not all estimation procedures utilize all the layers of this framework. Generally, methods for NLAs require more processing, and thus utilize all the layers to process the data. In what follows, we describe this high-level analysis in a concise way.

The first layer is called *sources* because they are the reason the processing task is performed. Algorithms and hardware exist to estimate their parameters, making it essential that sources exist and emit signals on the array.

The second layer is the *physical data* layer, consisting of the array's raw data in the baseband complex stage. This indicates that the sources have impinged on the array, passed through the RF chains, mixers, and other RF circuitry necessary to generate the discrete and quantized data from the baseband for processing. This step essentially depends on the array geometry: whether it is a ULA or NLA geometry will largely affect the following layers. We reiterate that this work focuses on SLAs where sensors are spaced by integer multiples of the minimum intersensor distance. This configuration represents a more restricted subclass of NLAs, as it imposes additional structure on the sensor array geometry.

The third layer is the *transformation layer*, which is useful only if the second layer had an NLA. If the data were acquired by a ULA, this layer is unnecessary. Here, mathematical transformations are performed to make the data appear as if they were obtained from a ULA arrangement. In a less sophisticated procedure, the virtual ULA is obtained by array mapping, filling the spaces between arrays that are too far apart with virtual sensors. The resulting virtual ULA has the same aperture as the physical NLA. This can be achieved with algorithms such as classical array interpolation or Wiener array interpolation. Conversely, when dealing with uncorrelated sources, more complex procedures, such as the vectorization of covariance matrices, can be employed, which indeed increase the aperture of the virtual ULA compared to the physical array. This increased aperture increases the number of degrees of freedom available, allowing us to estimate more sources than the number of physical sensors.

The fourth layer consists of the spatial smoothing procedure. Various techniques have been developed to enhance the sources' covariance matrix, such as Forward-Backward Averaging (FBA), Forward Spatial Smoothing (FSS), Forward-Backward Spatial Smoothing (FBSS), Weighted Spatial Smoothing (WSS), and Enhanced Spatial Smoothing (ESS). In the case of coherent sources in the first layer, the technique decorrelates the sources and allows for the rank increase of the sources' covariance matrix. Conversely, if the sources are uncorrelated, smoothing becomes necessary because the third layer renders the sources fully coherent after the vectorization procedure.

The fifth layer consists of the DoA estimation *per se*. This is where subspace-based algorithms like MUSIC and ESPRIT are applied to the data, treating them as if they come from uncorrelated sources. These algorithms can properly separate both subspaces only if the sources are not fully coherent. The reasoning behind this will be elaborated on later.

In Fig. 4.1, we have named each possible pathway as Mx, where x corresponds to a specific scenario. For example, a scenario with coherent sources, an MRA arrangement that is mapped to a virtual ULA through array interpolation, followed by a smoothing step, and then processed with MUSIC, is called M3. Notice that M2 and M6 methods are inadequate pathways and should be avoided because, in those cases, the subspace-based algorithm is applied to a covariance matrix with coherent sources, making it impossible to distinguish between signal and noise subspaces. We stress that although M6 starts with uncorrelated sources, the transformation layer makes the sources fully coherent.

Clearly, in any of the scenarios from M1 to M7, a better estimation result can be achieved if the processing is improved at some layer or bundle of layers. For example, if the spatial smoothing layer utilizes a better algorithm to decorrelate the sources, then the subsequent layer, the subspace decomposition layer, can more effectively distinguish between the signal and noise subspaces, resulting in better estimation performance. This line of reasoning can be applied to any layer.

To increase the estimation accuracy using sparse arrays, one could consider improving the transformation layer by using alternative methods for filling the virtual sensor data in the presence of coherent sources in the first layer, or developing alternative methods to generate the virtual ULA (coarray) using variations of vectorization or other strategies.

In this chapter, we will present a new technique that aims to increase the performance of DoA estimation with NLAs by modifying the spatial smoothing layer. This modification results in a scenario M7 with superior accuracy, as we will see later. Indeed, one of the breakthrough algorithms in DoA estimation was the development of coarray MUSIC, which acts on the transformation and spatial smoothing layers. By analyzing the equations, we propose an enhancement in the spatial smoothing layer by changing the window aperture of the overlapping subarrays and discuss the fundamentals of this technique.

In this chapter, our goal is to develop a strategy to improve the spatial smoothing performance considering the cases where this is done in coarray domain, aiming to improve the DoA estimation process in scenario M7 of Fig. 4.1, acting on the layer #4, that is, the spatial smoothing layer.

4.1 Signal model

In order to make the understanding more fluid to the reader, we rewrite the signal model for partially-calibrated sparse subarrays from Chapter 3, which can be seen as a general model, encompassing the fully calibrated array case, when we have only one subarray.

Consider the scheme in Figure 4.2, which shows multiple physical subarrays and the plane waves impinging on them. Our goal is to estimate their DoAs. These directions are measured with respect to the broadside axis, and in the model that follows, we estimate their sines, referred to as normalized DoAs. Each subarray performs local processing (DSP$_l$) using its second-order statistics, and the results are forwarded to a centralized DSP for final estimation.

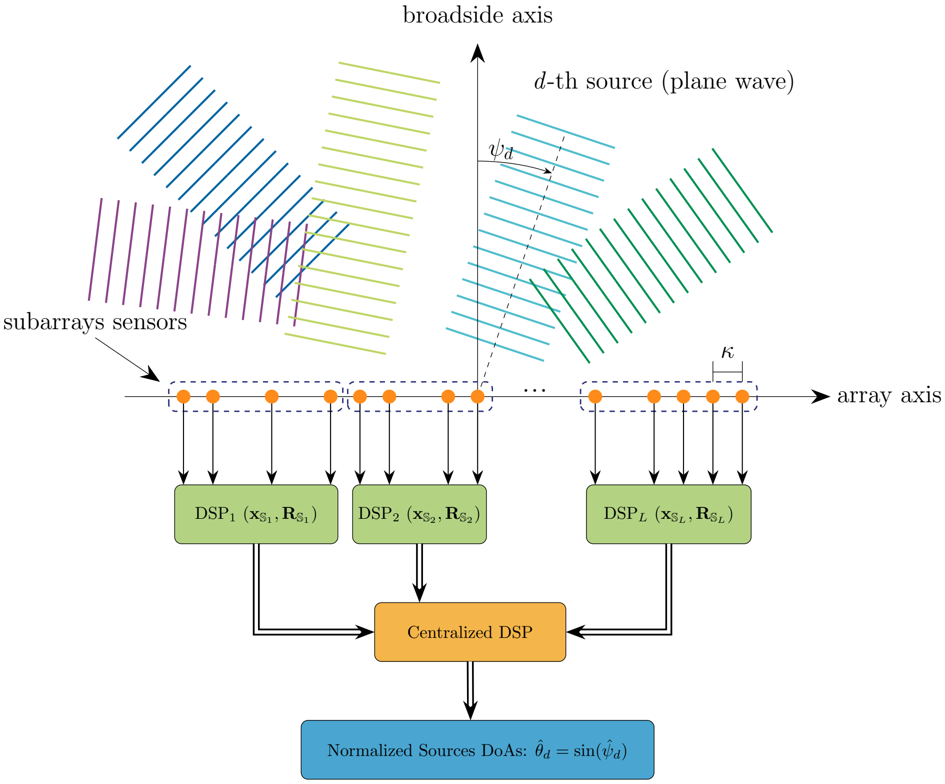

broadside axis
d-th source (plane wave)
ψ_d
subarrays sensors
κ
...
array axis
DSP$_1$ $(\mathbf{x}_{\mathbb{S}_1}, \mathbf{R}_{\mathbb{S}_1})$
DSP$_2$ $(\mathbf{x}_{\mathbb{S}_2}, \mathbf{R}_{\mathbb{S}_2})$
DSP$_L$ $(\mathbf{x}_{\mathbb{S}_L}, \mathbf{R}_{\mathbb{S}_L})$
Centralized DSP
Normalized Sources DoAs: $\hat{\theta}_d = \sin(\hat{\psi}_d)$

Figure 4.2: Partially-calibrated subarrays scheme

The received signal can be written as

$$\mathbf{x}_{\mathbb{S}}(t) = \tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}}\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t),\ \ t \in [T] \tag{4-1}$$

where

$$\begin{aligned}\tilde{\mathbf{V}}(\boldsymbol{\theta}) &= [\mathbf{V}(\boldsymbol{\theta}_1), \ldots, \mathbf{V}(\boldsymbol{\theta}_D)] \in \mathbb{C}^{N\times LD}\\ \mathbf{V}(\boldsymbol{\theta}_d) &= \text{blkdiag}\left(\mathbf{a}_1(\theta_d), \ldots, \mathbf{a}_L(\theta_d)\right) \in \mathbb{C}^{N\times L}\\ \tilde{\mathbf{H}} &= \text{blkdiag}\left(\mathbf{h}_1, \ldots, \mathbf{h}_D\right) \in \mathbb{C}^{LD\times D}\\ \mathbf{h}_d &= [1, h_{2d}, h_{3d}, \ldots, h_{Ld}]^T \in \mathbb{C}^{L\times 1}\end{aligned} \tag{4-2}$$

for $l \in \{1, \cdots, L\}$ and $t \in \{1, \cdots, T\}$, where L is the number of partially-calibrated subarrays, T is the number of snapshots and there are D uncorrelated sources impinging on the array. The vector $\mathbf{x}_{\mathbb{S}}(t) \in \mathbb{C}^N$ denotes the received signal in the whole array, $\mathbf{s}(t) \in \mathbb{C}^D$ is the source signal vector, and $\mathbf{n}_{\mathbb{S}}(t) \in \mathbb{C}^N$ is the measurement noise. In this case, we assume D impinging sources with normalized directions given by $\boldsymbol{\theta} \in [-1, 1)^D$ (sine of DOAs). The noise and the source signals are drawn from a circularly complex multivariate Gaussian distribution. The noise is spatially and temporally white, and the noise and sources are uncorrelated. The vector $\mathbf{h}_d$ contains the calibration parameters associated with all the subarrays (first subarray as reference) for the d-th direction, and $[\mathbf{a}_l(\theta)]_k = \exp(-j\pi n_k \theta)$ is the k-th element of the steering vector of the l-th subarray[1], $n_k \in \mathbb{S}_l$ is the k-th sensor position as an integer multiple of the minimum intersensor distance, the set of integers $\mathbb{S}_l$ and N_l define the sensor locations and number of sensor elements of the l-th subarray, respectively.

The received signal in (4-1) can be rewritten as

$$\begin{aligned}\mathbf{x}_{\mathbb{S}}(t) &= \left[\mathbf{V}(\boldsymbol{\theta}_1)\mathbf{h}_1, \ldots, \mathbf{V}(\boldsymbol{\theta}_D)\mathbf{h}_D\right] \mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t)\\ &= \begin{bmatrix}\mathbf{A}_1(\boldsymbol{\theta})\\ \mathbf{A}_2(\boldsymbol{\theta})\mathbf{G}_2\\ \vdots\\ \mathbf{A}_L(\boldsymbol{\theta})\mathbf{G}_L\end{bmatrix} \mathbf{s}(t) + \mathbf{n}_{\mathbb{S}}(t)\end{aligned} \tag{4-3}$$

where $\mathbf{A}_l(\boldsymbol{\theta}) \in \mathbb{C}^{N_l\times D}$ is the l-th calibrated subarray manifold with geometry defined by the set of integers $\mathbb{S}_l$, and $\mathbf{G}_l = \text{diag}\left([h_{l1}, \ldots, h_{lD}]\right)$. $\mathbf{G}_1$ is the identity matrix of order D, which is why it is not included in (4-3). The array received signal $\mathbf{x}_{\mathbb{S}}(t)$ can be partitioned as

$$\mathbf{x}_{\mathbb{S}}(t) = \left[\mathbf{x}_{\mathbb{S}_1}^T(t), \mathbf{x}_{\mathbb{S}_2}^T(t), \ldots, \mathbf{x}_{\mathbb{S}_L}^T(t)\right]^T. \tag{4-4}$$

Each of the received signals in (4-4) can also be expressed as

[1] Observe that, in this model, we assume the minimum distance between two sensors is half a wavelength ($\lambda/2$) and that the source angles are given by $\theta = \sin(\psi)$, where ψ is the angle between the impinging source signal and the broadside axis of the sensor array.

$$\mathbf{x}_{\mathbb{S}_l}(t) = \mathbf{A}_l(\boldsymbol{\theta})\mathbf{G}_l\mathbf{s}(t) + \mathbf{n}_{\mathbb{S}_l}(t), \tag{4-5}$$

where $\mathbf{x}_{\mathbb{S}_l}(t) \in \mathbb{C}^{N_l}$ is the l-th subarray received signal snapshot, and $\mathbf{n}_{\mathbb{S}_l}(t)$ is the measurement noise for each subarray.

The covariance matrix for each subarray is represented by

$$\begin{aligned}\mathbf{R}_{\mathbb{S}_l} &= \mathrm{E}\{\mathbf{x}_{\mathbb{S}_l}(t)\mathbf{x}_{\mathbb{S}_l}^H(t)\}\\ &= \mathbf{A}_l(\boldsymbol{\theta})\mathbf{G}_l\mathrm{E}\{\mathbf{s}(t)\mathbf{s}^H(t)\}\mathbf{G}_l^H\mathbf{A}_l^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I}_{N_l}\\ &= \mathbf{A}_l(\boldsymbol{\theta})\mathbf{G}_l\mathbf{R}_s\mathbf{G}_l^H\mathbf{A}_l^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I}_{N_l}\\ &= \mathbf{A}_l(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}_l^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I}_{N_l}.\end{aligned} \tag{4-6}$$

where have used the fact that pre-multiplying by $\mathbf{G}_l = \mathrm{diag}(g_{l,1}, ..., g_{l,D})$ and post-multiplying by $\mathbf{G}_l^H = \mathrm{diag}(g_{l,1}^*, ..., g_{l,D}^*)$ scales the d-th diagonal element of $\mathbf{R}_s$ by $g_{l,d}g_{l,d}^* = |g_{l,d}|^2 = 1$. Notice that the subarrays are uncalibrated relative to one another and the covariance matrix of each subarray is independent of the calibration parameters. Conversely, while each subarray is individually calibrated, the full array as a whole remains uncalibrated.

By applying the vectorization operator on (4-6), we arrive at

$$\mathbf{z}_{\mathbb{S}_l} = \left(\mathbf{A}_{\mathbb{S}_l}^*(\boldsymbol{\theta}) \circ \mathbf{A}_{\mathbb{S}_l}(\boldsymbol{\theta})\right)\mathbf{p} + \sigma_n^2\bar{\mathbf{i}}, \tag{4-7}$$

where $\circ$ denotes the Khatri-Rao product, $\bar{\mathbf{i}} = \left[\mathbf{e}_1^T, \ldots, \mathbf{e}_{N_l}^T\right]^T$ is the vectorization of the identity matrix, and $\mathbf{e}_i$ is the i-th canonical basis vector (all-zeros vector except for a 1 at the i-th position). To simplify the equations, we assume N/L sensors for each subarray, where $N = \sum_{l=1}^{L} N_l$ is the total number of sensors. By removing the repeated rows in $\mathbf{A}_{\mathbb{S}_l}^*(\boldsymbol{\theta}) \circ \mathbf{A}_{\mathbb{S}_l}(\boldsymbol{\theta})$ after their first occurrence (mirroring the operation in $\mathbf{z}_{\mathbb{S}_l}$ and $\bar{\mathbf{i}}$)[2] and sorting the virtual sensor (coarray) elements in ascending order, we have

$$\mathbf{x}_{\mathbb{D}_l} = \mathbf{A}_{\mathbb{D}_l}(\boldsymbol{\theta})\mathbf{p} + \sigma_n^2\mathbf{i}, \tag{4-8}$$

where $\mathbb{D}_l$ denotes the difference coarray set associated with the l-th subarray, $[\mathbf{A}_{\mathbb{D}_l}(\boldsymbol{\theta})]_{m,d} = \exp(j\pi n_m\theta_d)$ with $n_m \in \mathbb{D}_l$, $\mathbf{x}_{\mathbb{D}_l} \in \mathbb{C}^{|\mathbb{D}_l|}$ is the l-th coarray received signal and $\mathbf{i} \in \{0,1\}^{|\mathbb{D}_l|}$ is an all-zero vector with the exception of a 1 in its half position (element $(|\mathbb{D}_l| + 1)/2$). The problem of interest consists of estimating the source directions $\boldsymbol{\theta}$ from the preprocessed data in the model (4-8).

Observe that, since the sources are stationary, the vector $\mathbf{p}$ remains constant across all snapshots. Consequently, the power of each source is

[2] In practice, with finite data sample, we average the repeated elements in $\mathbf{z}_{\mathbb{S}_l}$ instead of keeping only the first element. This gives a more accurate estimate of the statistic for this specific covariance lag.

proportional to the power of every other source, i.e., $\sigma_i^2 = k * \sigma_j^2$, for all $i, j \in [D]$. Thus, the vector $\mathbf{p}$ can be interpreted as a single virtual snapshot for the model of virtually correlated sources, where the virtual sources exhibit constant amplitudes. Thus, physically uncorrelated sources become virtually coherent sources after using the vec() operator transformation.

4.2 Standard spatial smoothing algorithms

The standard spatial smoothing algorithm involves averaging submatrices of the full array covariance matrix to increase the rank of the source covariance matrix, particularly in scenarios where the sources are coherent or highly correlated [10, 60]. In such cases, it is not possible to effectively separate the signal and noise subspaces. However, subspace-based algorithms such as MUSIC [6] rely on the full rank of the source covariance matrix, making it essential to reconstruct the rank through the smoothing procedure.

Fig. 4.1 illustrates the role of spatial smoothing using uniform linear arrays (ULAs) and SLAs in the context of physical and coarray data. This schematic segments the DOA estimation into 5 layers, departing from the sources impinging on the array up until the subspace decomposition for the final estimation, having spatial smoothing as an intermediary pre-processing step (Layer #4). We focus on the use of spatial smoothing when the sources are coherent or virtually coherent[3], which can occur with ULA or SLAs. Each scenario for DOA estimation consists of the final stage in Layer #5 and ranges from M1 up until M7, where Mx is the x-th possible path in the scheme.

Clearly stating, in this section, we are reviewing the process performed in Layer #4, scenario M1: coherent sources, ULA geometry, decorrelation, and subspace-based algorithm at the end. This was the most natural and first scenario that employed spatial smoothing in the literature and is vital for our understanding.

The process of averaging the submatrices of the full array covariance matrix is done with some decrease in the array aperture. Thus, as we increase the rank of the source covariance matrix, we tend to lose some resolution capabilities on the resultant reduced-aperture array, given that less equivalent sensors are available after the averaging process. Remarks:

a) in Section 4.2 and its subsections, we assume that our signal model involves only one subarray, which is fully calibrated. The scenario involving

[3] The virtually coherent sources correspond to the virtual single snapshot of source powers $\mathbf{p}$ for the coarray model in (4-8).

partial calibration will be clear from the context. It is important to note that the term subarrays is used in two distinct senses: partially-calibrated subarrays and spatial smoothing subarrays. The former refers to physical subarrays, while the latter pertains to subarrays conceived from a spatial smoothing signal processing perspective.

b) For most equations, we omit the explicit dependence of the covariance matrices on $\mathbb{S}$, as it is unnecessary to specify whether the covariance matrix considers physical or virtual sensors. This is because our initial focus is on smoothing within the physical data, rather than coarray data. Later, in Section 4.3, the covariance matrices will depend on virtual sensors or coarray data, with the set $\mathbb{D}$ explicitly shown in the notation.

4.2.1 Uniform weighting spatial smoothing

For ease of understanding, we will first discuss spatial smoothing approaches that use uniform weights form each of the submatrices being averaged. The uniform weighting scheme relies on the partitioning of the array data $\mathbf{x}(t)$ into data of P overlapping subarrays $\mathbf{x}_1(t)$, $\mathbf{x}_2(t)$, …, $\mathbf{x}_P(t)$ of the aperture M. The p-th subarray consists of samples of $\mathbf{x}(t)$ that range from position p up to position $p + M - 1$. Thus, the (forward) spatially smoothed covariance matrix is calculated through [10]

$$\begin{aligned} \mathbf{R}_f &= \sum_{i=1}^{P} w_p \mathrm{E}\{\mathbf{x}_p(t)\mathbf{x}_p^H(t)\} \\ &= \sum_{p=1}^{P} w_p \mathbf{R}_p \end{aligned} \tag{4-9}$$

Then, $\mathbf{R}_f$ instead of $\mathbf{R}_\mathbb{S}$, the received signal covariance matrix, is used to compute the sources DoA. The uniform weighting considers $w_p = 1/P$ for simplicity and this results in a reasonably stable covariance matrix for most of the applications, which decorrelates source signals up to some degree and allows eigenspace algorithms like MUSIC for example to be used.

The performance of (4-9) can be further improved by means of the addition of a backward covariance matrix factor for each subarray, which is given by [61]

$$\mathbf{R}_b = \sum_{p=1}^{P} w_p \mathbf{J}\mathbf{R}_p^* \mathbf{J} \tag{4-10}$$

which relies on the conjugate symmetry of the array steering matrix in the case of the ULA arrangement.

The forward-backward spatial smoothing would then be obtained through

$$\mathbf{R}_{fb} = \sum_{p=1}^{P} w_p \left(\mathbf{R}_p + \mathbf{J}\mathbf{R}_p^* \mathbf{J}\right)/2 \tag{4-11}$$

or

$$\mathbf{R}_{fbu} = \frac{1}{2P} \sum_{p=1}^{P} \left(\mathbf{R}_p + \mathbf{J}\mathbf{R}_p^* \mathbf{J}\right) \tag{4-12}$$

if we consider uniform weights. We changed the notation to $\mathbf{R}_{fbu}$ with the subscript "fbu" to emphasize the use of uniform weights.

The covariance matrix for each subarray can also alternatively be written as

$$\begin{aligned} \mathbf{R}_p &= \mathbf{A}_r(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}_r^H(\boldsymbol{\theta}) + \sigma^2\mathbf{I} \\ &= \mathbf{A}_r(\boldsymbol{\theta})\boldsymbol{\Lambda}^{(M-N)/2+(p-1)}\mathbf{R}_s\left(\boldsymbol{\Lambda}^{(M-N)/2+(p-1)}\right)^H \mathbf{A}_r^H(\boldsymbol{\theta}) + \sigma^2\mathbf{I} \end{aligned} \tag{4-13}$$

where $\mathbf{A}_r(\boldsymbol{\theta})$ is reference subarray array manifold, M is the spatial smoothing subarray aperture, N is the number of physical sensors and $\boldsymbol{\Lambda}$ is defined as

$$\boldsymbol{\Lambda} = \begin{bmatrix} e^{j\pi\theta_1} & & & \\ & e^{j\pi\theta_2} & & \\ & & \ddots & \\ & & & e^{j\pi\theta_D} \end{bmatrix} \tag{4-14}$$

This matrix multiplies the reference subarray manifold to generate the manifolds of the other subarrays. By replacing (4-13) in (4-11), we can compute the spatially smoothed source signal matrix, that can be written as

$$\mathbf{R}_s^{\text{fbss}} = \sum_{p=1}^{P} w_p \mathfrak{Re}\left\{\boldsymbol{\Lambda}^{(M-N)/2+(p-1)}\mathbf{R}_s\left[\boldsymbol{\Lambda}^{(M-N)/2+(p-1)}\right]^H\right\} \tag{4-15}$$

The spatial smoothing technique aims to make the matrix in (4-15) as diagonal as possible, reducing the correlation between sources. Notice that in the specific case of the FBSS algorithm, the resulting source correlation matrix is real.

Some comments about this method are worth mentioning. Firstly, we emphasize that FBSS is more efficient than FSS, since it requires fewer sensors for the same number of coherent sources, according to

$$D < M \leq \lfloor N - D/2 + 1 \rfloor \tag{4-16}$$

where D is the number of DOAs to be estimated. Also, it is easy to implement,

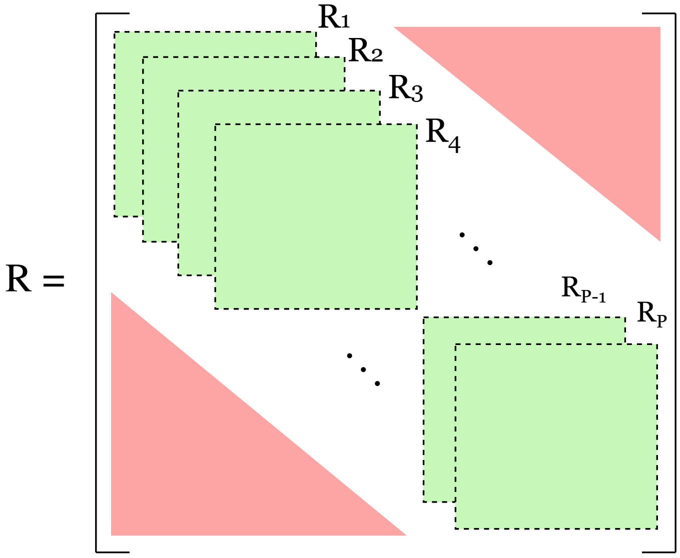


Figure 4.3: Covariance matrices of spatial smoothing subarrays for F(B)SS are shown in green. Discarded statistics are indicated by red triangles, emphasizing the loss of information.

widely employed, and referred in the literature as one of the best methods to smooth spatial data. However, we can also highlight some weaknesses: It can only be used with ULAs if we do not do some preprocessing at the transformation layer and results in loss of array resolution (maximum number of estimated DOAs much smaller than the number of physical sensors).

4.2.1.1
Uniformly weighted improved methods

Still dealing with uniform weighting methods, we can name some more recent and advanced methods for spatial smoothing. These methods utilize more statistics than FSS and FBSS, which discard the statistics corresponding to the larger lags of the covariance matrix, i.e., statistics on the upper-right and lower-left of the covariance matrix. To illustrate this problem and present some of the solutions proposed in the literature, consider what is shown in Fig. 4.3. In this simplified scheme, illustrating F(B)SS, the covariance matrices of the spatial smoothing subarrays are shown in green, while the regions of discarded statistics are indicated by red triangles in the upper-right and lower-left regions, emphasizing the loss of information for higher spatial lags.

Some attempts have been made to use those statistics in the spatial smoothing process, in order to achieve a higher degree of decorrelation between the sources. The first one we would like to refer to is the work developed in [62], where a new rule for the averaging of matrices is proposed as follows:

$$\mathbf{R}_{\mathrm{AL1}} = \frac{1}{2P}\sum_{i=1}^{P}\sum_{j=1}^{P}\left(\mathbf{R}_{ij}\mathbf{R}_{ji} + \mathbf{R}_{ij}^{*}\mathbf{R}_{ji}^{*}\right) \tag{4-17}$$

where $P = N - D + 1$ is the number of spatial smoothing subarrays (4-17), $\mathbf{R}_{ij}$ means the sub-covariance matrix between i-th and j-th subarrays, and the acronym AL1 stands for Additional Lags #1.

Another technique proposed in [63] requires a few modifications on (4-17), resulting in

$$\mathbf{R}_{\mathrm{AL2}} = \frac{1}{2P}\sum_{i=1}^{P}\sum_{j=1}^{P}\left(\mathbf{R}_{ii}\mathbf{R}_{jj} + \mathbf{R}_{ii}^{*}\mathbf{R}_{jj}^{*}\right) \tag{4-18}$$

where the acronym AL2 stands for Additional Lags #2, to differentiate it from (4-17). It does not use the statistics represented in red in Figure 4.3. Instead, it generates numerous fourth-order products based on the green regions.

In [64], a recent algorithm in the same line of reasoning was proposed. It aims to utilize all discarded fourth-order statistics, according to

$$\mathbf{R}_{\mathrm{AL3}} = \frac{1}{2P}\sum_{i=1}^{P}\sum_{j=1}^{P}\left(\mathbf{R}_{ij}\mathbf{R}_{ji} + \mathbf{R}_{ij}^{*}\mathbf{R}_{ji}^{*} + \mathbf{R}_{ii}\mathbf{R}_{jj} + \mathbf{R}_{ii}^{*}\mathbf{R}_{jj}^{*}\right) \tag{4-19}$$

We emphasize that the algorithms named ALx, where $x \in \{1, 2, 3\}$, utilize fourth-order statistics for the computation of the smoothed covariance matrix and require preprocessing in the transformation layer if they are to be used for DoA estimation with NLA geometries. In fact, they are not intrinsically capable of performing DoA estimation using sparse arrays and employ the same weighting scheme for all submatrices.

4.2.2 Non-uniformly weighted spatial smoothing

In this subsection, we discuss one of the most widely referenced methods in the literature associated with applying a weighting spatial smoothing scheme to the summation of submatrices. The work proposed in [65] aims to make the averaged matrix as close as possible to a Toeplitz structure by utilizing the degrees of freedom introduced by the weights.

Indeed, since we are operating in the fourth layer (spatial smoothing layer), our objective is to decorrelate the sources. If the sources are perfectly decorrelated, we will demonstrate that, for a ULA geometry, the resulting array covariance matrix is Toeplitz. The intuition behind this technique is that by adjusting the taps so that the averaged matrix becomes Toeplitz, the sources will be as uncorrelated as possible within the constraint domain for the weights.

The optimum value of vector weights $\mathbf{w}$ to make the resulting matrix as close as possible to a Toeplitz structure is

$$\mathbf{w} = \frac{\mathbf{E}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{E}^{-1}\mathbf{1}} \tag{4-20}$$

where $\mathbf{E}$ is defined as

$$\mathbf{E} = \mathfrak{Re}\left\{\sum_{i=0}^{M-2}\sum_{j=1}^{M-i}\mathbf{f}_{i+j,j}\mathbf{f}_{i+j,j}^T\right\} \tag{4-21}$$

$\mathbf{f}$ is given by

$$[\mathbf{f}_{i+j,j}]_p = [\mathbf{H}_p]_{i+j,j} - \frac{1}{M-i}\sum_{j=1}^{M-i}[\mathbf{H}_p]_{i+j,j} \tag{4-22}$$

and

$$\mathbf{H}_p = \frac{1}{2}\left(\mathbf{R}_p + \mathbf{J}\mathbf{R}_p^*\mathbf{J}\right) \tag{4-23}$$

4.3 Spatial smoothing for NLAs

In this section, we briefly discuss the spatial smoothing algorithm for NLAs, which was previously used in Chapter 3.

4.3.1 Standard coarray MUSIC

Indeed, the main idea of coarray MUSIC is to perform spatial smoothing on coarray domain. This comes from the fact, already discussed, that the overlapping strategy for subarrays can only be implemented if we have ULAs. Since for NLAs this is not the case, we have to turn them into ULAs by applying a transformation using the vec() operator on Layer #3. The covariance matrix for an NLA is given by

$$\mathbf{R}_{\mathbb{S}} = \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}_{\mathbb{S}}^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I} \tag{4-24}$$

We highlight that this equation does not consider multiple subarrays, as in the previous chapter. Instead, we used a fully calibrated array. However, the procedure for coarray MUSIC is the same as if we consider only one subarray. After the vectorization of this covariance matrix, sorting and removal of repeated rows of the coarray manifold, we arrive at

$$\mathbf{x}_{\mathbb{D}} = \mathbf{A}_{\mathbb{D}}(\boldsymbol{\theta})\mathbf{p} + \sigma_n^2\mathbf{i} \tag{4-25}$$

After this, by applying forward spatial smoothing on the coarray received signal $\mathbf{x}_{\mathbb{D}}$ with spatially smoothing coarrays (SS-CA) of size equal to $G = (\text{UDoF} + 1)/2$, we arrive at

$$\tilde{\mathbf{R}}_{\mathbb{D}}^{\text{SS}} = \frac{1}{G}\sum_{g=1}^{G}\mathbf{x}_{\mathbb{D}}^{g}\left(\mathbf{x}_{\mathbb{D}}^{g}\right)^H \tag{4-26}$$

where $\mathbf{x}_{\mathbb{D}}^{g} = \mathbf{J}_g \mathbf{x}_{\mathbb{D}} \in \mathbb{C}^G$ is the g-th overlapping SS-CA,

$$\mathbf{J}_g = \left[\mathbf{0}_{G\times(G-g)}\ \mathbf{I}_G\ \mathbf{0}_{G\times(g-1)}\right] \tag{4-27}$$

One can demonstrate that (4-26) possesses a signal and noise subspaces that can be properly separated using SVD/EVD in order to be used by subspace-based algorithms like MUSIC [1].

4.3.2
Variable Window Size Spatial Smoothing for Coarray DOA estimation

In this section, we present a more accurate approach using coarray data, SLAs and uncorrelated sources, which correspond to scenario M7 in Fig. 4.1. This method builds upon the previously discussed concept of decorrelating sources, which are virtual in this context.

We stress the fact that, in this case, the sources are physically uncorrelated in Layer #1, but become coherent after the transformation through the vec() operator in Layer #3 (Fig. 4.1). Firstly, we briefly discuss the use of coarray MUSIC by resorting to spatial smoothing within the coarray domain with increased DOF. As previously noted, the overlapping subarray strategy of spatial smoothing can only be directly applied when the matrix represents data coming from ULAs. For SLAs, this strategy is infeasible, requiring its transformation into a ULA by applying the vec() operator on Layer #3. The covariance matrix is given by

$$\begin{aligned}\mathbf{R}_{\mathbb{S}} &= \mathrm{E}\{\mathbf{x}_{\mathbb{S}}(t)\mathbf{x}_{\mathbb{S}}^{H}(t)\} \\ &= \mathbf{A}_{\mathbb{S}}(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}_{\mathbb{S}}^{H}(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I}\end{aligned} \tag{4-28}$$

In this case, we assume a fully calibrated array. However, the procedure for coarray MUSIC is the same as if we consider only one subarray in partially-calibrated array scenarios, in terms of the data model. After the vectorization of this covariance matrix, sorting and averaging the repeated rows of the coarray manifold, we arrive at

$$\mathbf{x}_{\mathbb{D}} = \mathbf{A}_{\mathbb{D}}(\boldsymbol{\theta})\mathbf{p} + \sigma_n^2\mathbf{i} \tag{4-29}$$

as it was already mentioned in (4-8). Observe also that (4-29) is deterministic, since (4-28) was defined in terms of the expectation operator. After this, by applying FSS on the coarray received signal $\mathbf{x}_{\mathbb{D}}$ with spatial smoothing coarrays (SS-CA) of size equal to $G = (\mathrm{UDOF}+1)/2$, we arrive at

$$\mathbf{R}_{\mathbb{D}}^{\mathrm{SS}} = \frac{1}{G}\sum_{g=1}^{G}\mathbf{x}_{\mathbb{D}}^{g}\left(\mathbf{x}_{\mathbb{D}}^{g}\right)^{H} \tag{4-30}$$

where UDOF is the number of *uniform* DOF and corresponds to the number of distinct contiguous covariance lags in the covariance matrix [66]. Observe that the resulting spatially smoothed matrix in (4-30) relies on 4-th order statistics, since we form outer products from a vectorized covariance matrix. Also, $\mathbf{x}_{\mathbb{D}}^{g} = \mathbf{J}_g \mathbf{x}_{\mathbb{D}} \in \mathbb{C}^G$ means the g-th overlapping SS-CA, where $\mathbf{J}_g$ is defined as

$$\mathbf{J}_g = \left[\mathbf{0}_{G\times(G-g)}\ \mathbf{I}_G\ \mathbf{0}_{G\times(g-1)}\right] \tag{4-31}$$

One can demonstrate that (4-30) possesses signal and noise subspaces that can be properly separated using eigenvalue decomposition (EVD) to be used by subspace-based algorithms like MUSIC [1, 20].

The standard coarray MUSIC algorithm can identify up to $D = G - 1$ sources. Since $G = \mathcal{O}(N^2)$, and UDOF $= \mathcal{O}(N^2)$, in most scenarios we have $G \gg D$, the number of sources, indicating that the SS-CA aperture could be much larger than the number of detected sources. This intuition introduces a variation of the coarray MUSIC algorithm tailored for scenarios where the number of sources is significantly smaller than the number of available DOF in the coarray domain. This improves the estimation accuracy of subspace-based algorithms such as MUSIC by producing a more refined averaging process for the smoothing average, as the number of rank-one (outer product) matrices to be averaged increases. Mathematically, we can write

$$M = \text{UDOF} - P + 1 \tag{4-32}$$

where P, in this context, is the number of *Spatially-Smoothed Coarrays* (SS-CA) and M is their aperture. The standard coarray MUSIC (CA-MUSIC) algorithm assumes $M = P = G$. Clearly, the sum $M + P = \text{UDOF} + 1$ is a constant because the number of UDOF depends only on the array geometry. Then, we can adjust the SS-CA aperture M by reducing it in order to prioritize increasing the number of matrices P to be averaged. From now on, it is assumed that the variable G has a fixed value of $(\text{UDOF} + 1)/2$, but the values of M and P can vary, i.e., P can increase and M can decrease according to some defined criterion. The process of varying the coarray window is mathematically described as follows.

Consider the standard spatial smoothing process in the coarray domain defined in (4-30). Let us define the difference between M and G as $G - M = a$, where $a \in \mathbb{N}_+$ is a positive constant that indicates compression in the coarray window. Clearly, for the CA-MUSIC algorithm, $a = 0$ (standard SS-MUSIC).

In fact, as we increase the parameter a, the SS-CA aperture M is reduced and the number P of submatrices of order $M = G - a$ increases. This has the effect of obtaining a different trade-off between coarray resolution and

RMSE performance. If the sources are sufficiently separated and we have much more DOF than the number of sources, this leads to a more clever setting of parameters than what SS-MUSIC does.

However, after we mismatch the number P of SS-CA and their aperture M, we should guarantee that the spatial smoothing on the coarray domain still can be used to perform a decomposition between signal and noise subspaces, i.e., the rank of the source covariance matrix due to the outer product of SS-CA is reconstructed, preserving the structure of noise and signal subspace directions given by their eigenvectors.

We stress the fact that in the standard coarray MUSIC algorithm, $G = M$. In the case of VWS, $M = G - a$. Thus, if the window compression parameter a is greater than zero, M is smaller than G. Moreover, G depends only on the degrees of freedom of the arrays and is therefore defined by the geometry. In contrast, M depends on the geometry as well as the hyperparameter a.

Let us develop the equations considering $a > 0$. The case $a = 0$ is straightforward and corresponds to standard CA-MUSIC. We can write the spatially smoothed coarray sum as

$$\tilde{\mathbf{R}}_{\mathbb{D}}^{\mathrm{SS}} = \frac{1}{P}\sum_{p=1}^{P} \tilde{\mathbf{R}}_{\mathbb{D}^p} \tag{4-33}$$

where

$$\tilde{\mathbf{R}}_{\mathbb{D}^p} = \tilde{\mathbf{x}}_{\mathbb{D}^p}\tilde{\mathbf{x}}_{\mathbb{D}^p}^H \tag{4-34}$$

and

$$\tilde{\mathbf{x}}_{\mathbb{D}^p} = \begin{cases} \mathbf{A}_{\mathbb{D}^r}\mathbf{\Phi}^{p-a-1}\mathbf{p} + \sigma_n^2\mathbf{e}_{p-a} \quad \text{if} \\ a+1 \leq p \leq M+a \\ \mathbf{A}_{\mathbb{D}^r}\mathbf{\Phi}^{p-a-1}\mathbf{p} \quad \text{if } 1 \leq p \leq a \text{ or} \\ M+a+1 \leq p \leq M+2a \end{cases} \tag{4-35}$$

where $\mathbf{A}_{\mathbb{D}^r}$ is the coarray manifold corresponding to the $(a+1)$-th subarray, taken as reference, and $\mathbf{\Phi} = \mathbf{\Lambda}^{-1}$, with $\mathbf{\Lambda}$ defined in (4-14). By making the change of variable $i = p - a - 1$, we can rewrite (4-35) as

$$\mathbf{y}_i = \begin{cases} \mathbf{A}_{\mathbb{D}^r}\mathbf{\Phi}^{i}\mathbf{p} + \sigma_n^2\mathbf{e}_{i+1} \quad \text{if } 0 \leq i \leq M-1, \\ \mathbf{A}_{\mathbb{D}^r}\mathbf{\Phi}^{i}\mathbf{p} \quad \text{if } -a \leq i \leq -1 \text{ or} \\ M \leq i \leq M+a-1 \end{cases} \tag{4-36}$$

with $\mathbf{y}_i = \tilde{\mathbf{x}}_{\mathbb{D}^{i+a+1}}$. In (4-36), the first case corresponds to $M = G - a$ noisy SS-CA, whereas the second case corresponds to $2a$ *noiseless* SS-CA. Consequently, this results in a total of $G - a + 2a = G + a$ SS-CA, compared to G noisy SS-CA in the conventional CA-MUSIC algorithm. Then, the VWS scheme increased the number of covariance matrices to be averaged.

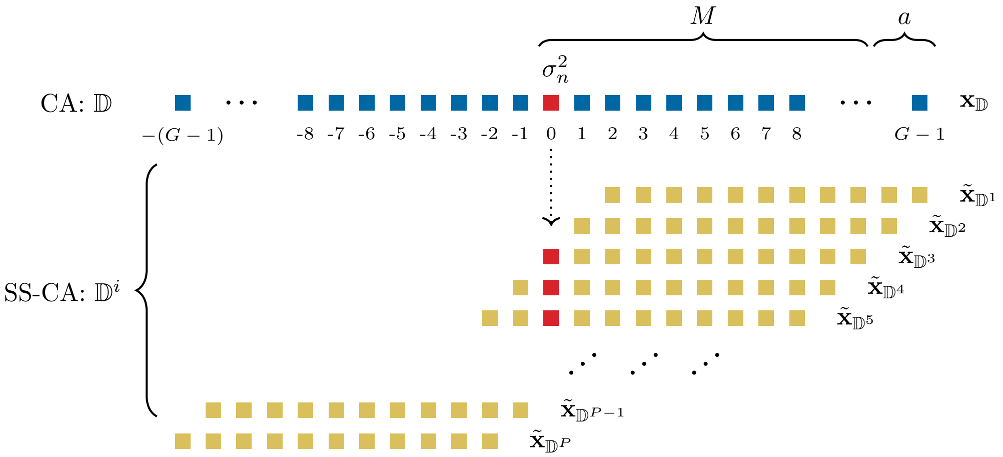

M
a
σ_n^2
CA: $\mathbb{D}$
$\cdots$
$\cdots$
$\mathbf{x}_{\mathbb{D}}$
$-(G-1)$
-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8
$G-1$
SS-CA: $\mathbb{D}^i$
$\tilde{\mathbf{x}}_{\mathbb{D}^1}$
$\tilde{\mathbf{x}}_{\mathbb{D}^2}$
$\tilde{\mathbf{x}}_{\mathbb{D}^3}$
$\tilde{\mathbf{x}}_{\mathbb{D}^4}$
$\tilde{\mathbf{x}}_{\mathbb{D}^5}$
$\tilde{\mathbf{x}}_{\mathbb{D}^{P-1}}$
$\tilde{\mathbf{x}}_{\mathbb{D}^P}$

Figure 4.4: Spatial smoothing using coarrays and VWS scheme: addition of noiseless spatially smoothed coarrays at the very beginning and very end of the coarray.

By doing this, we aim to balance between a smaller SS-CA, which reduces array resolution, and a larger quantity of noiseless coarray data. Indeed, this trade-off forms the basis for the proposed VWS-CA-MUSIC algorithm. Let us now proceed with the development of the VWS-CA-MUSIC algorithm using the relevant linear algebra equations.

To illustrate the intuition behind (4-35), consider the scheme provided in Fig. 4.4. The noise component in the coarray domain is represented by the noise variance. In (4-8), this noise is incorporated into the observed coarray data as a scaling factor for a vector that consists of zeros, except for a 1 at its central position, denoted as $\mathbf{i}$. When the spatial smoothing coarray (SS-CA) has a size smaller than half the dimension of $\mathbf{i}$ or the cardinality of $\mathbb{D}$, as illustrated in Fig. 4.4, the SS-CA does not include the central element of $\mathbf{i}$ at either the start or the end of the spatial smoothing process. In this case, it is clear that the coarray elements represented as filled red squares are the only ones contaminated by noise. Consequently, there exist some SS-CA segments that remain noiseless.

The noisy rank-one spatially smoothed matrices are given by

$$\begin{aligned}\mathbf{Y}_i &= \left(\mathbf{A}_{\mathbb{D}^r}\mathbf{\Phi}^i\mathbf{p}+\sigma_n^2\mathbf{e}_{i+1}\right)\left(\mathbf{A}_{\mathbb{D}^r}\mathbf{\Phi}^i\mathbf{p}+\sigma_n^2\mathbf{e}_{i+1}\right)^H\\ &= \left(\mathbf{A}_{\mathbb{D}^r}\mathbf{\Phi}^i\mathbf{p}+\sigma_n^2\mathbf{e}_{i+1}\right)\left(\mathbf{p}^H\mathbf{\Phi}^{-i}\mathbf{A}_{\mathbb{D}^r}^H+\sigma_n^2\mathbf{e}_{i+1}^H\right)\\ &= \mathbf{A}_{\mathbb{D}^r}\mathbf{\Phi}^i\mathbf{p}\mathbf{p}^H\mathbf{\Phi}^{-i}\mathbf{A}_{\mathbb{D}^r}^H+\sigma_n^2\mathbf{e}_{i+1}\mathbf{p}^H\mathbf{\Phi}^{-i}\mathbf{A}_{\mathbb{D}^r}^H\\ &+\mathbf{A}_{\mathbb{D}^r}\mathbf{\Phi}^i\mathbf{p}\sigma_n^2\mathbf{e}_{i+1}^H+\sigma_n^4\mathbf{e}_{i+1}\mathbf{e}_{i+1}^H,\end{aligned} \tag{4-37}$$

while the noiseless rank-one matrices can be represented as

$$\mathbf{W}_j = \mathbf{A}_{\mathbb{D}^r} \mathbf{\Phi}^j \mathbf{p}\mathbf{p}^H \mathbf{\Phi}^{-j} \mathbf{A}_{\mathbb{D}^r}^H, \tag{4-38}$$

where $i \in \{0, \ldots, M-1\}$ and $j \in \{-a, \ldots, -1\} \cup \{M, \ldots, M+a-1\}$. From this, we can then express the sum in (4-33) as

$$\tilde{\mathbf{R}}_{\mathbb{D}}^{\text{SS}} = \frac{1}{P} \left(\sum_{i=0}^{M-1} \mathbf{Y}_i + \left(\sum_{j=-a}^{-1} \mathbf{W}_j + \sum_{j=M}^{M+a-1} \mathbf{W}_j \right) \right), \tag{4-39}$$

where the noiseless summation has two separate parts: one in the beginning of the coarray data and another at the last portion of the coarray vector.

Next, we investigate the effect of the parameter a on the equations. The expression in (4-39) can be rewritten as follows:

$$\begin{aligned} \tilde{\mathbf{R}}_{\mathbb{D}}^{\text{SS}} = \frac{1}{P} \Big(& \mathbf{A}_{\mathbb{D}^r} \mathbf{C}\mathbf{C}^H \mathbf{A}_{\mathbb{D}^r}^H + \mathbf{A}_{\mathbb{D}^r} \mathbf{B}\mathbf{B}^H \mathbf{A}_{\mathbb{D}^r}^H \\ & + \sigma_n^4 \mathbf{I} + \sigma_n^2 \mathbf{A}_{\mathbb{D}^r} \mathbf{C} + \sigma_n^2 \mathbf{C}^H \mathbf{A}_{\mathbb{D}^r}^H \Big) \end{aligned} \tag{4-40}$$

where the scaled identity matrix $\sigma_n^4 \mathbf{I}$ was obtained through

$$\begin{aligned} \sigma_n^4 \mathbf{I} &= \sum_{i=0}^{M-1} \sigma_n^4 \mathbf{e}_{i+1} \mathbf{e}_{i+1}^H \\ &= \sum_{i=0}^{M-1} \sigma_n^4 \mathbf{P}_{\mathbf{e}_{i+1}} \end{aligned} \tag{4-41}$$

with $\mathbf{P}_{\mathbf{e}_{i+1}}$ being the orthogonal projection operator onto the $(i+1)$-th canonical basis vector. The matrices $\mathbf{A}_{\mathbb{D}^r}^H$, $\mathbf{B}$, $\mathbf{C}$ are defined as

$$\mathbf{A}_{\mathbb{D}^r}^H = \begin{bmatrix} 1 & \omega_1 & \cdots & \omega_1^{G-a-1} \\ 1 & \omega_2 & \cdots & \omega_2^{G-a-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \omega_D & \cdots & \omega_D^{G-a-1} \end{bmatrix} \tag{4-42}$$

$$= \begin{bmatrix} 1 & \omega_1 & \cdots & \omega_1^{M-1} \\ 1 & \omega_2 & \cdots & \omega_2^{M-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \omega_D & \cdots & \omega_D^{M-1} \end{bmatrix} \tag{4-43}$$

$$\mathbf{B} = \operatorname{diag}(\mathbf{p}) \times \tag{4-44}$$

$$\begin{bmatrix} \omega_1^{-a} & \omega_1^{-a+1} & \cdots & \omega_1^{-1} & \omega_1^{G-a} & \cdots & \omega_1^{G-1} \\ \omega_2^{-a} & \omega_2^{-a+1} & \cdots & \omega_2^{-1} & \omega_2^{G-a} & \cdots & \omega_2^{G-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \omega_D^{-a} & \omega_D^{-a+1} & \cdots & \omega_D^{-1} & \omega_D^{G-a} & \cdots & \omega_D^{G-1} \end{bmatrix} \tag{4-45}$$

$$= \operatorname{diag}(\mathbf{p}) \times \tag{4-46}$$

$$\begin{bmatrix} \omega_1^{-a} & \omega_1^{-a+1} & \cdots & \omega_1^{-1} & \omega_1^{M} & \cdots & \omega_1^{M+a-1} \\ \omega_2^{-a} & \omega_2^{-a+1} & \cdots & \omega_2^{-1} & \omega_2^{M} & \cdots & \omega_2^{M+a-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \omega_D^{-a} & \omega_D^{-a+1} & \cdots & \omega_D^{-1} & \omega_D^{M} & \cdots & \omega_D^{M+a-1} \end{bmatrix} \tag{4-47}$$

$$= \operatorname{diag}(\mathbf{p})\, \mathbf{\Phi}^{-a} \times \tag{4-48}$$

$$\begin{bmatrix} 1 & \omega_1 & \cdots & \omega_1^{a-1} & \omega_1^{M+a} & \cdots & \omega_1^{M+2a-1} \\ 1 & \omega_2 & \cdots & \omega_2^{a-1} & \omega_2^{M+a} & \cdots & \omega_2^{M+2a-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \omega_D & \cdots & \omega_D^{a-1} & \omega_D^{M+a} & \cdots & \omega_D^{M+2a-1} \end{bmatrix} \tag{4-49}$$

$$= \operatorname{diag}(\mathbf{p})\mathbf{\Phi}^{-a} \begin{bmatrix} \mathbf{B}_1^H & \mathbf{B}_2^H \end{bmatrix} \tag{4-50}$$

$$= \operatorname{diag}(\mathbf{p})\mathbf{\Phi}^{-a} \begin{bmatrix} \mathbf{B}_1^H & \mathbf{\Phi}^{M+a}\mathbf{B}_1^H \end{bmatrix} \tag{4-51}$$

$$= \operatorname{diag}(\mathbf{p}) \begin{bmatrix} \mathbf{\Phi}^{-a} & \mathbf{\Phi}^{M+a} \end{bmatrix} \begin{bmatrix} \mathbf{B}_1^H & \\ & \mathbf{B}_1^H \end{bmatrix} \tag{4-52}$$

where

$$\mathbf{B}_1^H = \begin{bmatrix} 1 & \omega_1 & \cdots & \omega_1^{a-1} \\ 1 & \omega_2 & \cdots & \omega_2^{a-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \omega_D & \cdots & \omega_D^{a-1} \end{bmatrix} \tag{4-53}$$

$$\mathbf{C} = \operatorname{diag}(\mathbf{p}) \begin{bmatrix} 1 & \omega_1 & \cdots & \omega_1^{G-a-1} \\ 1 & \omega_2 & \cdots & \omega_2^{G-a-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \omega_D & \cdots & \omega_D^{G-a-1} \end{bmatrix} \tag{4-54}$$

$$= \operatorname{diag}(\mathbf{p}) \begin{bmatrix} 1 & \omega_1 & \cdots & \omega_1^{M-1} \\ 1 & \omega_2 & \cdots & \omega_2^{M-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \omega_D & \cdots & \omega_D^{M-1} \end{bmatrix} \tag{4-55}$$

$$= \operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H \tag{4-56}$$

From this, by replacing (4-52) into (4-40), we arrive at

$$\begin{aligned}
\tilde{\mathbf{R}}_{\mathbb{D}}^{\mathrm{SS}} &= \frac{1}{P}\Big(\mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H\mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H + \sigma_n^4\mathbf{I} \\
&+ 2\sigma_n^2\mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H + \mathbf{A}_{\mathbb{D}^r}\mathbf{B}\mathbf{B}^H\mathbf{A}_{\mathbb{D}^1}^H\Big) \\
&= \frac{1}{P}\Big[\Big(\mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H + \sigma_n^2\mathbf{I}\Big)^2 + \\
&\mathbf{A}_{\mathbb{D}^r}\mathbf{B}\mathbf{B}^H\mathbf{A}_{\mathbb{D}^r}^H\Big] \\
&= \frac{1}{P}\left(\mathbf{R}_1^2 + \mathbf{R}_2^2\right)
\end{aligned} \tag{4-57}$$

where $\mathbf{R}_1 = \mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H + \sigma_n^2\mathbf{I}$ and $\mathbf{R}_2^2 = \mathbf{A}_{\mathbb{D}^r}\mathbf{B}\mathbf{B}^H\mathbf{A}_{\mathbb{D}^r}^H$. Clearly, the matrix $\mathbf{R}_1$ has full rank, as does its squared form.

We observe that the quantities involved in this expression are deterministic. Specifically, the cross terms in $(\mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H + \sigma_n^2\mathbf{I})^2$ exist, do not vanish, and are explicitly shown in the first line of (4-57). The effect of the cross terms is to square the eigenvalues of the expression within the parentheses. The perfect square indicates that the signal and noise subspaces remain unchanged in relation to $\mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H + \sigma_n^2\mathbf{I}$, as squaring a matrix affects only the squares of its eigenvalues. Consequently, the first part of the equation remains tractable by subspace-based algorithms. The second term, involving the Gram matrix $\mathbf{B}\mathbf{B}^H$, will be demonstrated to be equally tractable, ensuring that it does not disrupt the signal and noise subspaces established in the first part. This will be further addressed in the analysis section.

The matrix $\mathbf{R}_2$ consists of a low-rank perturbation on $\mathbf{R}_1^2$ and can be rewritten as

$$\begin{aligned}
\mathbf{R}_2^2 &= \mathbf{A}_{\mathbb{D}^r}\mathbf{B}\mathbf{B}^H\mathbf{A}_{\mathbb{D}^r}^H \\
&= \mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\begin{bmatrix}\boldsymbol{\Phi}^{-a} & \boldsymbol{\Phi}^{M+a}\end{bmatrix}\begin{bmatrix}\mathbf{B}_1^H\mathbf{B}_1 & \\ & \mathbf{B}_1^H\mathbf{B}_1\end{bmatrix}\times \\
&\begin{bmatrix}\boldsymbol{\Phi}^{a} \\ \boldsymbol{\Phi}^{-M-a}\end{bmatrix}\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H \\
&= \mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\Big(\boldsymbol{\Phi}^{-a}\mathbf{B}_1^H\mathbf{B}_1\boldsymbol{\Phi}^{a} + \\
&\boldsymbol{\Phi}^{M+a}\mathbf{B}_1^H\mathbf{B}_1\boldsymbol{\Phi}^{-M-a}\Big)\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H \\
&= 2\mathbf{A}_{\mathbb{D}^r}\operatorname{diag}(\mathbf{p})\mathbf{B}_1^H\mathbf{B}_1\operatorname{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H,
\end{aligned} \tag{4-58}$$

where we utilized the result provided in (4-52), along with the definition of $\mathbf{B}_1^H$ as given in (4-53). Given that $\mathbf{B}_1^H$ exhibits a Vandermonde structure, it inherently has a rank that does not exceed $\max(a, D)$. Consequently, the smallest value of the parameter a that ensures $\mathbf{R}_2^2$ achieves full rank is D. In contrast, if the parameter a is smaller than D, as it is usually the case, then the matrix $\mathbf{R}_2^2$ will have a rank of a, indicating that it is rank-deficient by $D - a$.

We observe that even in the scenario where $\mathbf{R}_2^2$ retains full rank, it continues to be a low-rank perturbation of the rank-M matrix $\mathbf{R}_1^2$. Despite this perturbation, the matrix $\tilde{\mathbf{R}}_{\mathbb{D}}^{\text{SS}}$ preserves several valuable properties. The properties of the algorithm are rigorously analyzed in Section 4.4. The VWS-CA-MUSIC algorithm is summarized in Algorithm 4.1.

Algorithm 4.1: VWS-CA-MUSIC

```
Input  : Array geometry $\mathbb{S}$, data matrix $\hat{\mathbf{X}}_{\mathbb{S}}$, grid $\boldsymbol{\theta}^g \in (-1, 1]^g$,
         number of sources $D$, window compression parameter $a$
$\hat{\mathbf{R}}_{\mathbb{S}} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}}\hat{\mathbf{X}}_{\mathbb{S}}^H$
$\hat{\mathbf{z}} \leftarrow \text{vec}\left(\hat{\mathbf{R}}_{\mathbb{S}}\right)$
Remove the repeated elements in $\hat{\mathbf{z}}$ and sort the corresponding
  coarray locations to obtain $\hat{\mathbf{x}}_{\mathbb{D}}$
$\hat{\tilde{\mathbf{R}}}_{\mathbb{D}}^{\text{SS}} \leftarrow \frac{1}{P}\sum_{i=1}^{P} \hat{\tilde{\mathbf{x}}}_{\mathbb{D}^i}\hat{\tilde{\mathbf{x}}}_{\mathbb{D}^i}^H$
Diagonalize $\hat{\tilde{\mathbf{R}}}_{\mathbb{D}}^{\text{SS}}$ through an EVD to find $\hat{\tilde{\mathbf{U}}}_N$
for $i \leftarrow 1$ to $g$ do
    $\hat{\tilde{P}}(\theta_i^g) = \left[\left[\tilde{\mathbf{a}}_{\mathbb{D}}^{\text{SS}}(\theta_i^g)\right]^H \hat{\tilde{\mathbf{U}}}_N\hat{\tilde{\mathbf{U}}}_N^H\tilde{\mathbf{a}}_{\mathbb{D}}^{\text{SS}}(\theta_i^g)\right]^{-1}$
Find the peaks in $\hat{P}(\theta_i^g)$ and take the corresponding angles as the
  estimated DOAs
Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

Since we have the signal and noise subspaces for the reduced-size preprocessed matrix $\tilde{\mathbf{R}}_{\mathbb{D}}^{\text{SS}}$, a natural extension of VWS-CA-MUSIC is a root-MUSIC version, namely, the VWS-CA-rMUSIC algorithm. In order to derive it, we proceed as follows. We assume the following polynomial function

$$\mathbf{f}_{\mathbb{D}}(z) = \left[1, z, z^2, \ldots, z^{M-1}\right]^T \tag{4-59}$$

with $M = (\text{UDOF} + 1)/2 - a$. Equation (4-59) has the form of a steering vector of a ULA with M sensors for $z = \exp(j\pi\theta)$, since we use the common assumption of a minimum inter-sensor spacing of $\lambda/2$. The polynomial in (4-59) evaluated at the true directions θ defines a space that is orthogonal to the eigenvectors of the Hermitian matrix $\mathbf{U}_N\mathbf{U}_N^H$, the projection matrix onto the noise subspace. Thus, we can state that the polynomial in the form of the quadratic form

$$\begin{aligned} Q_{\mathbb{D}} &= \mathbf{f}_{\mathbb{D}}^{H}(z)\mathbf{U}_{N}\mathbf{U}_{N}^{H}\mathbf{f}_{\mathbb{D}}(z) \\ &= \mathbf{f}_{\mathbb{D}}^{T}(1/z)\mathbf{U}_{N}\mathbf{U}_{N}^{H}\mathbf{f}_{\mathbb{D}}(z) \end{aligned} \tag{4-60}$$

has D zeros at the the unit circle, where the DoAs correspond to the angles of these zeros, normalized by π. Given that $\hat{z} = \exp\left(j\pi\hat{\theta}\right)$, it follows that $\hat{\theta} = \angle\hat{z}/\pi$. In practice, we take the D zeros closest to and within the unit circle, which is called the *root pruning process*. Of course, since the polynomial will have $2M-1$ coefficients, it will possess degree $2M-2$, resulting in a total number of $2M-2$ roots. The VWS-CA-rMUSIC algorithm is summarized in Algorithm 4.2.

Algorithm 4.2: VWS-CA-rMUSIC

```
Input : subarray geometries $\mathbb{S}_l$, data matrix for each subarray $\hat{\mathbf{X}}_{\mathbb{S}_l}$,
        grid $\boldsymbol{\theta}^g \in (-1, 1]^g$, number of sources $D$, window
        compression parameter $a$
$\hat{\mathbf{R}}_{\mathbb{S}} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}}\hat{\mathbf{X}}_{\mathbb{S}}^{H}$
$\hat{\mathbf{z}} \leftarrow \text{vec}\left(\hat{\mathbf{R}}_{\mathbb{S}}\right)$
Remove the repeated elements in $\hat{\mathbf{z}}$ and sort the corresponding
  coarray locations to obtain $\hat{\mathbf{x}}_{\mathbb{D}}$
$\hat{\hat{\mathbf{R}}}_{\mathbb{D}}^{\text{SS}} \leftarrow \frac{1}{P}\sum_{i=1}^{P}\hat{\hat{\mathbf{x}}}_{\mathbb{D}^i}\hat{\hat{\mathbf{x}}}_{\mathbb{D}^i}^{H}$
Diagonalize $\hat{\hat{\mathbf{R}}}_{\mathbb{D}}^{\text{SS}}$ through an EVD to find $\hat{\hat{\mathbf{U}}}_{N}$
Find the roots of $Q_{\mathbb{D}}(z) = \mathbf{f}_{\mathbb{D}}^{T}(1/z)\hat{\hat{\mathbf{U}}}_{N}\hat{\hat{\mathbf{U}}}_{N}^{H}\mathbf{f}_{\mathbb{D}}(z)$
Select the $D$ roots inside and closest to the unit circle as $\hat{z}_d$
Estimate the DOAs according to $\hat{\theta} = \left(\angle\hat{z}_d\right)/\pi$
Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

In terms of computational complexity, the dominant step in coarray MUSIC is the EVD of the spatially smoothed data in (4-33), since the cost of EVD is $\mathcal{O}\left(M^3\right)$. In VWA schemes, the value of M is reduced, with a new aperture for the SS-CA being $M' = M - a$. As the computational cost of EVD depends on the size of the decomposed matrix, we can state that the VWS versions of the algorithms reduce the overall computational complexity.

Lastly, we note that, in terms of computational complexity, the dominant step in coarray MUSIC is the EVD of the spatially smoothed data matrix in (4-33), as the cost of EVD scales with $\mathcal{O}(M^3)$. In VWA schemes, the effective value of M is reduced, with the new aperture for the spatially smoothed coarrays being $M' = M - a$. Since the computational cost of EVD depends on the dimension of the matrix being diagonalized, we can conclude that the VWA-based versions of the algorithms reduce the overall computational complexity.

4.3.3 VWS for Patially-Calibrated Scenarios

The VWS version of coarray MUSIC, developed in Section 4.3.2, can be used to produce more accurate variants of the GCA-MUSIC and GCA-rMUSIC algorithms introduced in [2, 67]. These algorithms are specifically tailored for partially-calibrated scenarios, where the whole SLA is composed of smaller pieces of sparse linear subarrays and the array is calibrated within each subarray, but remains uncalibrated between different subarrays.

Specifically, we present a refined version of these algorithms by adjusting the spatial smoothing window size within each spatially-smoothed subcoarray (SS-SCA) and subsequently perform subspace intersection, similar to what was done in the first version of the algorithms presented in [2].

We denote these refined algorithms as Variable Window Size Generalized Coarray MUSIC (VWS-GCA-MUSIC) and Variable Window Size Generalized Coarray root-MUSIC (VWS-GCA-rMUSIC) algorithms. In this case, we say that the algorithms pre-process data from *subcoarrays*, i.e., spatial smoothing performed within each physical subarray.

The underlying intuition behind these schemes is that by introducing a more accurate version of the averaged spatially smoothed matrix within the subspace intersection scheme, we can achieve a more precise estimation of these subspaces, leading to improved RMSE performance in identifying signal and noise subspaces subject to partially-calibrated scenarios with sparse subarrays.

For the l-th SCA, i.e., the coarray derived from the l-th physical subarray, let us rewrite the spatial smoothing equation as

$$\tilde{\mathbf{R}}_{\mathbb{D}_l}^{\mathrm{SS}} = \frac{1}{L} \sum_{p=1}^{P} \tilde{\mathbf{x}}_{\mathbb{D}_l^p} \left(\tilde{\mathbf{x}}_{\mathbb{D}_l^p} \right)^H , \tag{4-61}$$

where $\tilde{\mathbf{x}}_{\mathbb{D}_l^p} = \mathbf{J}_p \tilde{\mathbf{x}}_{\mathbb{D}_l} \in \mathbb{C}^M$ is the p-th overlapping SS-SCA of the l-th SCA, where the vectors $\tilde{\mathbf{x}}_{\mathbb{D}_l^p}$ have aperture M, considering the window compression parameter $a > 0$. Following a similar procedure described in [2], and taking the intersection onto the convex sets of noise subspaces associated to each subarray, one can demonstrate that this projection is given by

$$\tilde{\mathbf{P}} = \mathrm{blkdiag} \left(\tilde{\mathbf{U}}_{N_1} \tilde{\mathbf{U}}_{N_1}^H, \tilde{\mathbf{U}}_{N_2} \tilde{\mathbf{U}}_{N_2}^H, \ldots, \tilde{\mathbf{U}}_{N_L} \tilde{\mathbf{U}}_{N_L}^H \right) \tag{4-62}$$

where $\tilde{\mathbf{U}}_{N_l}$ is the noise subspaces of the l-th subarray computed from (4-61). Taking this into account, the VSW-GCA-MUSIC spectrum can be obtained through

$$\tilde{P}(\theta_i^g) = \left[\tilde{\mathbf{a}}_{\mathbb{D}}^{\mathrm{SS}}(\theta)^H \tilde{\mathbf{P}} \tilde{\mathbf{a}}_{\mathbb{D}}^{\mathrm{SS}}(\theta)\right]^{-1} \tag{4-63}$$

with the combined steering vector given by

$$\tilde{\mathbf{a}}_{\mathbb{D}}^{\mathrm{SS}}(\theta) = [\tilde{\mathbf{a}}_{\mathbb{D}_1}^{\mathrm{SS}}(\theta)^T \ \tilde{\mathbf{a}}_{\mathbb{D}_2}^{\mathrm{SS}}(\theta)^T \ \cdots \ \tilde{\mathbf{a}}_{\mathbb{D}_L}^{\mathrm{SS}}(\theta)^T]^T \tag{4-64}$$

where

$$\tilde{\mathbf{a}}_{\mathbb{D}_l}^{\mathrm{SS}}(\theta) = [1 \ \ \exp\left(j\pi\theta\right) \ \cdots \ \exp\left(j\pi(M-1)\theta\right)]^T \tag{4-65}$$

The VSW-GCA-MUSIC algorithm is summarized in Algorithm 4.3.

Algorithm 4.3: VWS-GCA-MUSIC

```
Input : subarrays geometries $\mathbb{S}_l$, data matrix for each subarray
        $\hat{\mathbf{X}}_{\mathbb{S}_l}$, grid $\boldsymbol{\theta}^g \in (-1,1]^g$, number of sources $D$, window
        compression parameter $a$
for $l \leftarrow 1$ to $L$ do
    $\hat{\mathbf{R}}_{\mathbb{S}_l} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}_l}\hat{\mathbf{X}}_{\mathbb{S}_l}^H$
    $\hat{\mathbf{z}}_l \leftarrow \mathrm{vec}\left(\hat{\mathbf{R}}_{\mathbb{S}_l}\right)$
    Remove the repeated elements in $\hat{\mathbf{z}}_l$ and sort the corresponding
      coarray locations to obtain $\hat{\mathbf{x}}_{\mathbb{D}_l}$
    $\hat{\hat{\mathbf{R}}}_{\mathbb{D}_l}^{\mathrm{SS}} \leftarrow \frac{1}{P}\sum_{i=1}^{P} \hat{\hat{\mathbf{x}}}_{\mathbb{D}_l^i}\hat{\hat{\mathbf{x}}}_{\mathbb{D}_l^i}^H$
    Diagonalize $\hat{\hat{\mathbf{R}}}_{\mathbb{D}_l}^{\mathrm{SS}}$ through an EVD to find $\hat{\hat{\mathbf{U}}}_{N_l}$
for $i \leftarrow 1$ to $g$ do
    $\hat{\hat{P}}(\theta_i^g) = \left[\sum_{l=1}^{L} \left[\tilde{\mathbf{a}}_{\mathbb{D}_l}^{\mathrm{SS}}(\theta_i^g)\right]^H \hat{\hat{\mathbf{U}}}_{N_l}\hat{\hat{\mathbf{U}}}_{N_l}^H \tilde{\mathbf{a}}_{\mathbb{D}_l}^{\mathrm{SS}}(\theta_i^g)\right]^{-1}$
Find the peaks in $\hat{P}(\theta_i^g)$ and take the corresponding angles as the
  estimated DOAs
Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

Departing from the same eigenvalue decomposition of (4-61), one can demonstrate that the global noise projection matrix for all the subarrays is given by [2]

$$\check{\mathbf{P}} = \mathbf{P}_{\check{\mathbf{U}}_{N_i}} + \sum_{l=1, l\neq i}^{L} \mathbf{\Pi}_{\check{\mathbf{U}}_{N_l}} \tag{4-66}$$

where $i = \mathrm{argmax}_{l\in\{1,\ldots,L\}}\, M_l$, in case the subarrays have different apertures, $\mathbf{P}_{\check{\mathbf{U}}_{N_i}} = \tilde{\mathbf{U}}_{N_i}\tilde{\mathbf{U}}_{N_i}^H$, and the projection $\mathbf{\Pi}_{\check{\mathbf{U}}_{N_l}}$ is defined as

$$\mathbf{\Pi}_{\check{\mathbf{U}}_{N_l}} = \begin{bmatrix} \mathbf{0}_{(M_i-M_l)\times M_l} \\ \mathbf{I}_{M_l} \end{bmatrix} \mathbf{P}_{\check{\mathbf{U}}_{N_l}} \begin{bmatrix} \mathbf{I}_{M_l} & \mathbf{0}_{M_l\times(M_i-M_l)} \end{bmatrix} \tag{4-67}$$

with

$$\mathbf{P}_{\check{\mathbf{U}}_{N_l}} = \mathrm{blkdiag}\left(\mathbf{I}_{M(l-1)}, \tilde{\mathbf{U}}_{N_l}\tilde{\mathbf{U}}_{N_l}^H, \mathbf{I}_{M(L-l)}\right) \tag{4-68}$$

By searching the nulls of the root-MUSIC polynomial using the projection matrix in (4-66), one can write

$$\tilde{Q}_{\mathbb{D}}(z) = \mathbf{f}_i^T(1/z)\check{\mathbf{P}}\mathbf{f}_i(z) \tag{4-69}$$

which is used to estimate the DOAs. The VWS-GCA-rMUSIC algorithm is summarized in Algorithm 4.4.

Algorithm 4.4: VWS-GCA-rMUSIC

```
Input : subarray geometries $\mathbb{S}_l$, data matrix for each subarray $\hat{\mathbf{X}}_{\mathbb{S}_l}$,
        grid $\boldsymbol{\theta}^g \in (-1, 1]^g$, number of sources $D$, window
        compression parameter $a$
for $l \leftarrow 1$ to $L$ do
    $\hat{\mathbf{R}}_{\mathbb{S}_l} \leftarrow (1/T)\hat{\mathbf{X}}_{\mathbb{S}_l}\hat{\mathbf{X}}_{\mathbb{S}_l}^H$
    $\mathbf{z}_l \leftarrow \text{vec}\left(\hat{\mathbf{R}}_{\mathbb{S}_l}\right)$
    Average the repeated elements in $\mathbf{z}_l$ and sort the corresponding
      coarray locations to obtain $\mathbf{x}_{\mathbb{D}_l}$
    $\hat{\tilde{\hat{\mathbf{R}}}}_{\mathbb{D}_l}^{\text{SS}} \leftarrow \frac{1}{P}\sum_{i=1}^{P} \hat{\tilde{\hat{\mathbf{x}}}}_{\mathbb{D}_l^i}\hat{\tilde{\hat{\mathbf{x}}}}_{\mathbb{D}_l^i}^H$
    Diagonalize $\tilde{\mathbf{R}}_{\mathbb{D}_l}^{\text{SS}}$ through an EVD to find $\hat{\tilde{\hat{\mathbf{U}}}}_{N_l}$
    $\hat{\mathbf{P}}_{\check{\mathbf{U}}_{N_l}} \leftarrow \hat{\check{\hat{\mathbf{U}}}}_{N_l}\hat{\check{\hat{\mathbf{U}}}}_{N_l}^H$
Search for $i = \text{argmax}_{l\in\{1,\ldots,L\}} M_l$
$\hat{\check{\hat{\mathbf{P}}}} \leftarrow \hat{\tilde{\hat{\mathbf{P}}}}_{\mathbf{U}_{N_i}} + \sum_{l=1,l\neq i}^{L} \hat{\mathbf{\Pi}}_{\mathbf{U}_{N_l}}$
Find the roots of $Q_{\mathbb{D}}(z) = \mathbf{f}_i^T(1/z)\check{\mathbf{P}}\mathbf{f}_i(z)$
Select the $D$ roots inside and closest to the unit circle as $\hat{z}_d$
Estimate the DOAs according to $\hat{\theta} = (\angle\hat{z}_d)/\pi$
Output: Estimated DOAs $\hat{\boldsymbol{\theta}}$
```

4.4 Analysis

In this section, we analyze the subspace properties of the proposed algorithms and provide a mathematical rationale for their ability to exploit the underlying signal model in the extraction of DOA parameters. Furthermore, we establish an upper bound for the window compression parameter. The analysis begins with the latter and concludes with the former, as it represents the most significant contribution of this section.

4.4.1 Analysis of window compression parameter

Regarding the analysis of the parameter a, in what follows we establish the allowable range of values for a to ensure algorithms' effectiveness. In fact, it is clear that to detect up to D sources using a subspace-based algorithm, the minimum window size M must be greater than D.

The condition $M > D$ means

$$G - a > D \Rightarrow a < \frac{\mathrm{UDOF} + 1 - 2D}{2} \tag{4-70}$$

once $M = G - a$ and $G = (\mathrm{UDOF} + 1)/2$. This represents an upper limit on the parameter a. It is crucial to consider this constraint when setting algorithm parameters, given that reducing the window size indiscriminately could impact the minimum dimension necessary for subspace-based algorithms to effectively distinguish between signal and noise subspaces.

4.4.2
Subspace properties of VWS schemes

In this subsection, we analyze and discuss the subspace properties of the VWS scheme, employing formal Lemma statements and a Theorem to rigorously establish our theoretical results.

Lema 4.1 *The matrix* $\mathbf{R}_2^2 = 2\mathbf{A}_{\mathbb{D}^r}\,\mathrm{diag}(\mathbf{p})\mathbf{B}_1^H\mathbf{B}_1\,\mathrm{diag}(\mathbf{p})\mathbf{A}_{\mathbb{D}^r}^H$ *has a unique positive semidefinite square root denoted by* $\mathbf{R}_2$*.*

Prova. The matrix $\mathbf{R}_2^2$ *can be rewritten in Gram form* $\mathbf{R}_2^2 = \mathbf{F}\mathbf{F}^H$*, where* $\mathbf{F} = \sqrt{2}\mathbf{A}_{\mathbb{D}^r}\,\mathrm{diag}(\mathbf{p})\mathbf{B}_1^H$*. Clearly,* $\mathbf{R}_2^2$ *is Hermitian. Moreover, it is also PSD, since* $\mathbf{x}^H\mathbf{F}\mathbf{F}^H\mathbf{x} = (\mathbf{F}^H\mathbf{x})^H\mathbf{F}^H\mathbf{x} = \|\mathbf{F}^H\mathbf{x}\|_2^2 \geq 0$ *for all* $\mathbf{x} \in \mathbb{C}^M$*. The spectral theorem for Hermitian matrices implies that* $\mathbf{R}_2^2$ *can be diagonalized by a unitary matrix* $\mathbf{U}_2$*, such that* $\mathbf{R}_2^2 = \mathbf{U}_2\mathbf{\Lambda}_2^2\mathbf{U}_2^H$*, where* $\mathbf{\Lambda}_2^2$ *is the diagonal matrix with eigenvalues. Since* $\mathbf{R}_2^2$ *is PSD, it has nonnegative eigenvalues. We can then write its unique PSD square root (principal square root) as* $\mathbf{R}_2 = \mathbf{U}_2\mathbf{\Lambda}_2\mathbf{U}_2^H$*.* ■

Lema 4.2 *If a vector* $\mathbf{z}$ *belongs to the noise subspace of* $\mathbf{R}_1^2$*, it also belongs to the noise subspace of* $\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}$*.*

Prova. Suppose that a vector $\mathbf{z} \in \mathbb{C}^M$ *is within the noise subspace of* $\mathbf{R}_1^2$*. It follows that*

$$\begin{aligned}
\mathbf{R}_1^2\mathbf{z} &= \mathbf{U}_{S_1}\mathbf{\Lambda}_{S_1}^2\mathbf{U}_{S_1}^H\mathbf{z} + \mathbf{U}_{N_1}\mathbf{\Lambda}_{N_1}^2\mathbf{U}_{N_1}^H\mathbf{z} \\
&= \mathbf{U}_{S_1}\mathbf{\Lambda}_{S_1}^2\left(\mathbf{U}_{S_1}^H\mathbf{z}\right) + \sigma_n^4\left(\mathbf{U}_{N_1}\mathbf{U}_{N_1}^H\right)\mathbf{z} \\
&= \mathbf{U}_{S_1}\mathbf{\Lambda}_{S_1}^2(0) + \sigma_n^4\mathbf{z} \\
&= \sigma_n^4\mathbf{z}
\end{aligned} \tag{4-71}$$

At the same time, we have

$$\begin{aligned}\mathbf{R}_2^2\mathbf{z} &= \mathbf{U}_{S_2}\boldsymbol{\Lambda}_{S_2}^2\mathbf{U}_{S_2}^H\mathbf{z} \\ &= \mathbf{U}_{S_2}\boldsymbol{\Lambda}_{S_2}^2\left(\mathbf{U}_{S_2}^H\mathbf{z}\right) \\ &= 0,\end{aligned} \tag{4-72}$$

where the last step in (4-72) follows from the fact that $\mathcal{R}\left(\mathbf{U}_{S_2}\right) \subseteq \mathcal{R}\left(\mathbf{U}_{S_1}\right)$*, since* $\mathcal{R}\left(\mathbf{B}_1\right) \subseteq \mathcal{R}\left(\mathbf{A}_{\mathbb{D}^r}\right)$*, given that* $\mathbf{B}_1 = [\mathbf{I}_a \mid \mathbf{0}]\mathbf{A}_{\mathbb{D}^r}$*. Thus, we obtain*

$$\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}\mathbf{z} = \frac{1}{P}\left(\sigma_n^4\mathbf{z} + 0\right) = (\sigma_n^4/P)\mathbf{z}, \tag{4-73}$$

which means that $\mathbf{z}$ *corresponds to the eigenvector associated with the minimum eigenvalue of* $\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}$ *and thus belongs to its noise subspace.* ■

The result presented in Lemma 4.2 is of critical significance because it enables us to utilize the noise eigenvectors of $\tilde{\mathbf{R}}_{\mathbb{D}}^{\mathrm{SS}}$ to accurately estimate the DoAs. Specifically, it shows that the reduction in the size of the coarray window does not distort the signal subspace of $\mathbf{R}_1^2$, which should be preserved in traditional subspace-based schemes.

Lema 4.3 *The matrix* $\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}$ *is of full rank, with a rank equal to* M*.*
Prova.

$$P\tilde{\mathbf{R}}_{\mathbb{D}}^{SS} = \mathbf{U}_1\boldsymbol{\Lambda}_1^2\mathbf{U}_1^H + \mathbf{U}_2\boldsymbol{\Lambda}_2^2\mathbf{U}_2^H \tag{4-74}$$

$$= \begin{bmatrix}\mathbf{U}_{S_1} & \mathbf{U}_{N_1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_{S_1} & \\ & \boldsymbol{\Lambda}_{N_1}\end{bmatrix}\begin{bmatrix}\mathbf{U}_{S_1}^H \\ \mathbf{U}_{N_1}^H\end{bmatrix} \tag{4-75}$$

$$+ \begin{bmatrix}\mathbf{U}_{S_2} & \mathbf{U}_{N_2}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_{S_2} & \\ & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{U}_{S_2}^H \\ \mathbf{U}_{N_2}^H\end{bmatrix} \tag{4-76}$$

$$= \mathbf{U}_{S_1}\boldsymbol{\Lambda}_{S_1}^2\mathbf{U}_{S_1}^H + \mathbf{U}_{S_2}\boldsymbol{\Lambda}_{S_2}^2\mathbf{U}_{S_2}^H + \mathbf{U}_{N_1}\boldsymbol{\Lambda}_{N_1}^2\mathbf{U}_{N_1}^H \tag{4-77}$$

$$= \mathbf{U}_{S_1}\boldsymbol{\Lambda}_{S_1}^2\mathbf{U}_{S_1}^H + \mathbf{U}_{S_1}\mathbf{Q}\boldsymbol{\Lambda}_{S_2}^2\mathbf{Q}^H\mathbf{U}_{S_1}^H + \sigma_n^4\mathbf{I} \tag{4-78}$$

$$= \mathbf{U}_{S_1}\left(\boldsymbol{\Lambda}_{S_1}^2 + \mathbf{Q}\boldsymbol{\Lambda}_{S_2}^2\mathbf{Q}^H\right)\mathbf{U}_{S_1}^H + \sigma_n^4\mathbf{I}, \tag{4-79}$$

where we assumed $\mathbf{U}_{S_2} = \mathbf{U}_{S_1}\mathbf{Q}$*, because* $\mathcal{R}\left(\mathbf{U}_{S_2}\right) \subseteq \mathcal{R}\left(\mathbf{U}_{S_1}\right)$*. Note that the diagonal matrices* $\boldsymbol{\Lambda}_{S_1}^2 \in \mathbb{C}^{D\times D}$ *and* $\boldsymbol{\Lambda}_{S_2}^2 \in \mathbb{C}^{a\times a}$ *both have strictly positive diagonal elements. Also, because* $\mathbf{U}_{S_1}$ *and* $\mathbf{U}_{S_2}$ *are orthonormal,* $\mathbf{Q}$ *is semi-unitary (*$\mathbf{Q}^H\mathbf{Q} = \mathbf{I}_a$*). Of course,* $\mathbf{Q} \in \mathbb{C}^{D\times a}$ *has rank* a*, assuming* $a < D$*. This implies that* $\mathbf{Q}\boldsymbol{\Lambda}_{\mathbf{S_2}}^{\mathbf{2}}\mathbf{Q}^{\mathbf{H}}$ *has rank* a*. Clearly,* $\boldsymbol{\Lambda}_{S_1}^2 + \mathbf{Q}\boldsymbol{\Lambda}_{S_2}^2\mathbf{Q}^H$ *is Hermitian because it results from the sum of two Hermitian matrices. Let us show that adding the second term in parentheses of (4-79) does not reduce the rank* D *of matrix* $\boldsymbol{\Lambda}_{S_1}^2$*.*

The matrix $\mathbf{Q}\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H \in \mathbb{C}^{D\times D}$ *has at most a nonzero eigenvalues, and these eigenvalues are determined by the diagonal elements of* $\mathbf{\Lambda}_{S_2}^2$ *projected on* $\mathbf{Q}$*. However, since* $\mathbf{\Lambda}_{S_1}^2$ *is diagonal with positive entries, these eigenvalues will affect the overall eigenvalue spectrum of* $\mathbf{\Lambda}_{S_1}^2 + \mathbf{Q}\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H$*, but will not introduce zero eigenvalues. In fact,* $\mathbf{\Lambda}_{S_1}^2$ *is positive definite, which means* $\mathbf{x}^H\mathbf{\Lambda}_{S_1}^2\mathbf{x} > 0$ *and* $\mathbf{Q}\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H$ *is positive semidefinite, which means* $\mathbf{x}^H\mathbf{Q}\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H\mathbf{x} \geq 0$*. This implies that* $\mathbf{x}^H\mathbf{\Lambda}_{S_1}^2\mathbf{x} + \mathbf{x}^H\mathbf{Q}\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H\mathbf{x} > 0$ *is positive definite and therefore has full rank D. Consequently, we can state that the first term in (4-79) has rank D, and adding it to the noise covariance matrix ensures that the resulting sum* $P\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}$ *is full-rank because the eigenvalues of the resulting matrix consist of the eigenvalues of the first term plus the noise variances of the second term. Therefore, the rank of* $\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}$ *is* M[4]*. Furthermore, we can write* $\tilde{\mathbf{R}}_{\mathbb{D}}^{SS} = \mathbf{U}_S\mathbf{\Lambda}_S^2\mathbf{U}_S^H\mathbf{z} + \mathbf{U}_N\mathbf{\Lambda}_N^2\mathbf{U}_N^H\mathbf{z}$*, since* $\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}$ *has been shown to possess a subspace structure with noise and signal parts.* ■

To proceed with our analysis and present our most important result, which we enunciate as a theorem, we must first define the measure called *Signal Subspace Strength.*

Definição 4.4 (Signal Subspace Strength) *The signal subspace strength of* $\mathbf{R} \succeq 0$*, denoted* $\operatorname{str}(\mathbf{R})$ *is defined as the sum of the eigenvalues corresponding to signal subspace eigenvectors, i.e.,* $\operatorname{str}(\mathbf{R}) = \operatorname{tr}(\mathbf{\Lambda}_S)$*. The noise subspace strength is defined analogously.*

In the following theorem, we provide the mathematical reasoning for why DoA estimation using $\tilde{\mathbf{R}}_{\mathbb{D}}^{\mathrm{SS}}$ with a reduced-size window is generally more accurate than when using only the matrix $\mathbf{R}_1^2$. More specifically, the addition of the matrix $\mathbf{R}_2^2$ leads to increased estimation accuracy.

Teorema 4.5 *The signal subspace strength* $\operatorname{str}\left(\mathbf{R}_1^2 + \mathbf{R}_2^2\right)$ *is always greater than* $\operatorname{str}\left(\mathbf{R}_1^2\right)$*.*

Prova. Let us start by showing that $\mathbf{R}_1^2 \succ 0$*. The positive semi-definiteness of* $\mathbf{R}_2^2 \succeq 0$ *was shown in Lemma 4.1.*

$$\begin{aligned}\mathbf{x}^H\mathbf{R}_1^2\mathbf{x} &= \mathbf{x}^H\mathbf{U}_1\mathbf{\Lambda}_1^2\mathbf{U}_1^H\mathbf{x}\\ &= \mathbf{w}^H\mathbf{\Lambda}_1^2\mathbf{w},\ \textit{where}\ \mathbf{w} = \mathbf{U}_1^H\mathbf{x} \neq 0\\ &= \sum_{i=1}^{M}\lambda_{1,i}^2|w_i|^2 > 0\end{aligned} \tag{4-80}$$

where the last inequality follows from the fact that all the $\lambda_{1,i}^2 > 0$*. Since the positive semi-definiteness is closed under addition and any positive-definite*

[4] Pre-multiplying by P does not change the rank properties.

matrix is also positive-semidefinite, it follows that $\mathbf{R}_1^2 + \mathbf{R}_2^2$ *is also PSD. Indeed*[5]

$$\begin{aligned}\operatorname{str}\left(P\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}\right) &= \operatorname{str}\left(\mathbf{U}_1\mathbf{\Lambda}_1^2\mathbf{U}_1^H + \mathbf{U}_2\mathbf{\Lambda}_2^2\mathbf{U}_2^H\right) \\ &= \operatorname{str}\left(\mathbf{U}_{S_1}\left(\mathbf{\Lambda}_{S_1}^2 + \mathbf{Q}\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H\right)\mathbf{U}_{S_1}^H + \right. \\ &\left. +\sigma_n^4\mathbf{I}\right)\end{aligned} \tag{4-81}$$

where the columns of $\mathbf{Q}$ *are orthonormal because of the subspace relationships between* $\mathbf{U}_1$ *and* $\mathbf{U}_2$. *The matrix* $\mathbf{\Lambda}_{S_1}^2 + \mathbf{Q}\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H$ *is not diagonal in the general case. Then, we cannot compute the signal eigenvalues directly. However, we can ensure that the first term in the sum corresponds to the signal part and thus its trace corresponds to the signal subspace strength of* $\operatorname{str}\left(P\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}\right)$. *From this, we have*

$$\begin{aligned}&\operatorname{str}\left(P\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}\right) \\ &= \operatorname{tr}\left(\mathbf{U}_{S_1}\left(\mathbf{\Lambda}_{S_1}^2 + \mathbf{Q}\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H\right)\mathbf{U}_{S_1}^H\right) \\ &= \operatorname{tr}\left(\mathbf{U}_{S_1}\mathbf{\Lambda}_{S_1}^2\mathbf{U}_{S_1}^H\right) + \operatorname{tr}\left(\mathbf{U}_{S_1}\mathbf{Q}\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H\mathbf{U}_{S_1}^H\right) \\ &= \operatorname{tr}\left(\mathbf{\Lambda}_{S_1}^2\mathbf{U}_{S_1}^H\mathbf{U}_{S_1}\right) + \operatorname{tr}\left(\mathbf{\Lambda}_{S_2}^2\mathbf{Q}^H\mathbf{U}_{S_1}^H\mathbf{U}_{S_1}\mathbf{Q}\right) \\ &= \operatorname{tr}\left(\mathbf{\Lambda}_{S_1}^2\right) + \operatorname{tr}\left(\mathbf{\Lambda}_{S_2}^2\right) > \operatorname{tr}\left(\mathbf{\Lambda}_{S_1}^2\right) = \operatorname{str}\left(\mathbf{R}_1^2\right)\end{aligned} \tag{4-82}$$

which shows us that the signal subspace strength of the reduced coarray size window $\operatorname{str}\left(P\tilde{\mathbf{R}}_{\mathbb{D}}^{SS}\right)$ *is stronger than its standard version, when the component* $\mathbf{R}_2^2$ *is not present, i.e., the addition of the rank-a perturbation* $\mathbf{R}_2^2$ *adds constructively to the eigenvalues of* $\mathbf{R}_1^2$. ■

An immediate consequence of Theorem 4.5 is that the array SNR (ASNR), defined as

$$\text{ASNR} = 10\log_{10}\frac{\operatorname{tr}(\mathbf{\Lambda}_S)}{\operatorname{tr}(\mathbf{\Lambda}_N)} \tag{4-83}$$

increases as the window size is reduced. This occurs because reducing the window size leads to a more concentrated signal subspace, which enhances the relative significance of the eigenvalues in $\mathbf{\Lambda}_S$.

In contrast, the total noise variance, represented by the sum of the eigenvalues in $\mathbf{\Lambda}_N$, decreases due to a reduction in the dimension of the noise subspace, although the magnitude of the individual eigenvalues of the noise subspace remains unchanged. As a result, the signal eigenvalues dominate more significantly than the noise eigenvalues, thereby amplifying the ASNR.

[5] See Lemma 4.3

4.5
Numerical results

In this section, we use numerical simulations to evaluate the performance of the VWS schemes for the proposed algorithms, namely VWS-CA-(r)MUSIC and VWS-GCA-(r)MUSIC, considering both fully and partially-calibrated scenarios. This analysis demonstrates the performance gains achieved by compressing the smoothing window, i.e., increasing the compression parameter a, resulting in the exploitation of the low-rank perturbations in the smoothed (sub)coarray covariance matrices by means of the addition of the low-rank matrix $\mathbf{R}_2^2$ in (4-57), as described in Section 4.3.2.

The global parameter setting for the simulation scenarios considers

- Number of snapshots: $T = 1,000$
- Uncorrelated sources distributed within the interval $[-0.8, 0.8]$
- Average over $10,000$ trials to ensure well behaved curves

We complement the global parameter settings by providing specific parameter descriptions in each subsection, according to whether the simulations correspond to fully or partially calibrated scenarios.

Moreover, we employ the well established geometries such as Two-Level Nested Arrays (NAQ2) [1], 2nd-Order Super Nested Arrays (SNAQ2) [21], Minimum Redundancy Arrays (MRA) [22], and Minimum Hole Arrays (MHA) [24]. In what follows, we illustrate the RMSE performance gains for each scenario.

4.5.1
RMSE performance in fully calibrated scenarios

In this subsection, we analyze the effects of varying the window compression parameter a in the coarray domain. In what follows, we provide the specific parameter setting:

- Number of sensors: $N = 7$ sensors;
- Number of sources: $D = 4$
- Array geometries: NAQ2, SNAQ2, MRA and MHA.

The RMSE curves against SNR are shown in Fig. 4.5 and Fig. 4.6 for VWS-CA-MUSIC and VWS-CA-rMUSIC, respectively. Notice that the VWS schemes lead to significant improvements in the RMSE performance for most of the SNR range over their competitors, namely coarray MUSIC (CA-MUSIC or SS-MUSIC) and coarray root-MUSIC (CA-rMUSIC). We also observe that

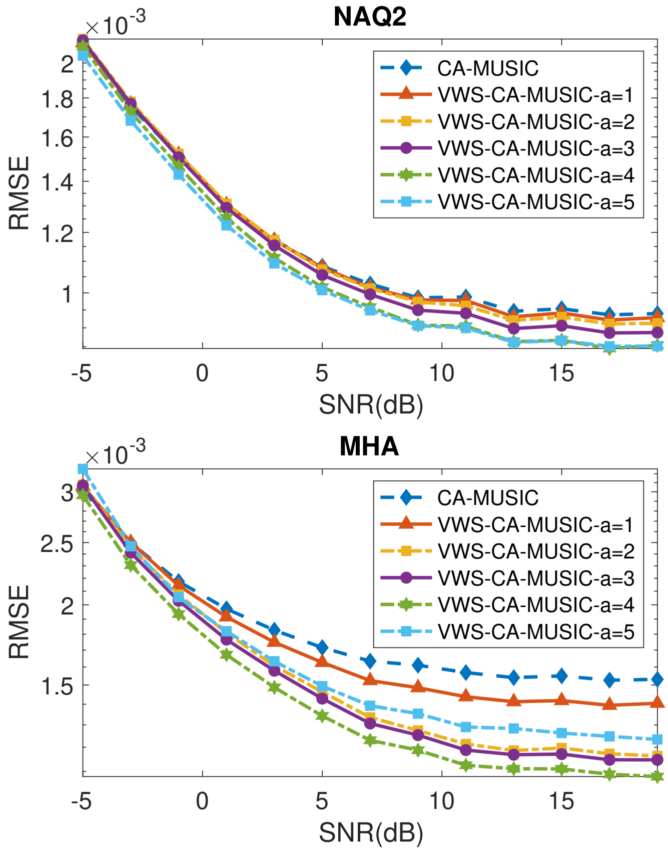


Figure 4.5: RMSE curves of VWS-CA-MUSIC for various values of the parameter a against SNR, using the geometries NAQ2 and MHA. The parameter $a = 0$ corresponds to the CA-MUSIC (coarray MUSIC or SS-MUSIC) algorithm with a fixed window size, as originally presented in [1].

for each geometry, there is an optimal value for the parameter a that minimizes the RMSE response and this value is greater than 0, meaning the addition of the the low-rank perturbations indeed improve the signal subspace strength in comparison to the noise subspace strength, as it was demonstrated in Section 4.4.

4.5.2 RMSE performance in partially-calibrated scenarios

Moving on to partially-calibrated scenarios, we compare the VWS method proposed in Section 4.3.3 with the GCA-MUSIC and GCA-rMUSIC algorithms, both of which employ fixed window apertures. The parameter setting for this scenario is given below:

- Number of subarrays: $L = 3$
- Number of sensors in each subarray: $N_l = 8$ (total of 24 sensors)

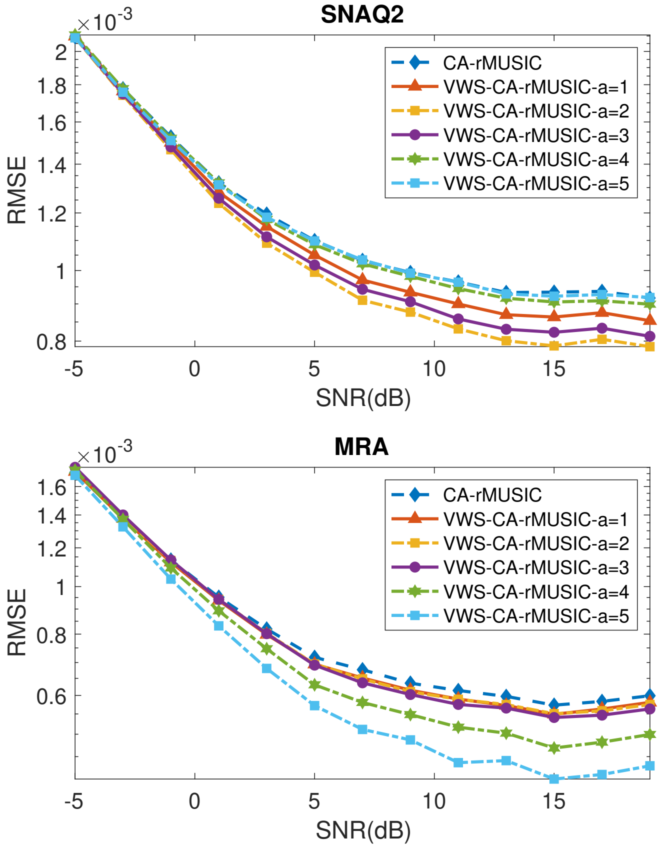


Figure 4.6: RMSE curves of VWS-CA-rMUSIC for various values of the parameter a against SNR, using two different geometries: SNAQ2 and MRA. Note that $a = 0$ corresponds to the CA-rMUSIC algorithm with a fixed window size, using the spatial smoothing scheme originally presented in [1].

– Number of sources: $D = 12$ (more sources than sensors in each physical subarray)

– Array geometry: type-II MRA.

The corresponding performance curves are presented in Fig. 4.7 and Fig. 4.8. Additionally, we observe that the so-called Type-II geometries, originally proposed in [46], provide a framework for constructing sparse subarrays from sparse arrays specifically tailored for partially calibrated scenarios. As it can be clearly seen from the curves, for both the spectrum and search-free versions of the algorithms, low-rank perturbations caused by the reduction in the smoothing window significantly contribute to the achievement of performance gains. This improvement is particularly pronounced in the spectrum version of the VWS-GCA-MUSIC algorithm.

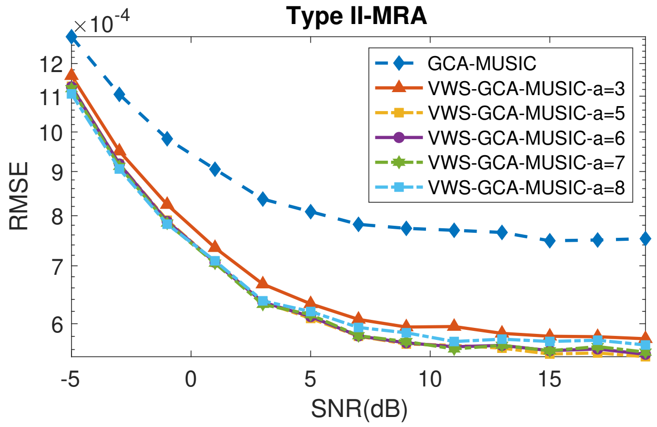


Figure 4.7: RMSE curves of VWS-GCA-MUSIC for various values of the parameter a against SNR, using the type-II MRA geometry in partially-calibrated scenarios. The window compression parameter $a = 0$ corresponds to the GCA-MUSIC algorithm with a fixed window size, as originally proposed in [2].

4.6 Summary

In this Chapter, we proposed VWS spatial smoothing techniques to improve coarray-based DOA estimation for SLA configurations previously introduced in Chapter 3 and [1]. The proposed approach incorporates adjustable spatial smoothing window sizes, which yield significant gains in estimation accuracy for sparse arrays. We also devised the VWS-CA-MUSIC and VWS-CA-rMUSIC algorithms for fully calibrated SLA geometries, and their corresponding generalized versions, the VWS-GCA-MUSIC and VWS-GCA-rMUSIC algorithms for partially-calibrated SLA geometries.

An analysis of the proposed VWS approach is carried out along with an assessment of the computational complexity of the proposed against existing algorithms. By introducing a window compression parameter, we have enhanced the accuracy of spatial smoothing using coarrays in the fully and partially-calibrated parameter settings. A detailed analysis of the noise and signal subspaces was conducted, demonstrating that the VWS method does not distort the structure of the signal and noise subspaces, enhancing the signal subspace strength. Consequently, the proposed algorithms allow for the use of subspace-based techniques in the coarray domain, which can estimate more sources than the number of physical sensors more accurately.

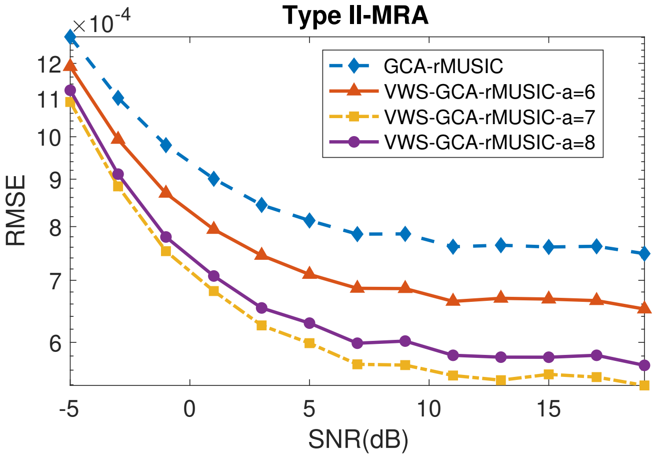


Figure 4.8: RMSE curves of VWS-GCA-rMUSIC for various values of the parameter a against SNR. The parameter $a = 0$ corresponds to the GCA-rMUSIC algorithm with a fixed window size (see [2]).

5
Conclusions and future works

In this work, we have developed and formalized two sparse linear array geometries, classified as type-I and type-II. These two approaches to generating sparse subarrays were thoroughly studied, and their key properties were analytically derived. Additionally, we introduced six novel algorithms: Generalized Coarray MUSIC (GCA-MUSIC) and Generalized Coarray Root-MUSIC (GCA-rMUSIC), along with their Variable Window Size (VWS) versions, and VWS-CA-(r)MUSIC. The GCA family of algorithms was specifically designed for the partially calibrated array scenario, while the CA family was formulated to address fully coherent (calibrated) array configurations.

In Chapter 3, the sparse linear array geometries were studied. Two particular designs were investigated: type-I and type-II sparse linear arrays. While the former consists of a splitting of the array to generate the subarrays, the latter is assembled by merging a given number of sparse subarrays with a predefined sparse linear geometry. Type-I arrays do not possess an explicit *a priori* formula that relates the number of degrees of freedom of the whole array as a function of the degrees of freedom of the subarrays. However, the degrees of freedom of type-II sparse linear arrays do have an explicit mathematical relationship with the number of degrees of freedom of the subarrays. This relationship was established in the form of a theorem and is the most important result of this chapter.

Moreover, two algorithms were developed: GCA-MUSIC and GCA-rMUSIC in Chapter 3. Both algorithms can be used in the context of coarray processing and thus are capable of estimating more sources than the number of physical sensors in each subarray. GCA-MUSIC relies on the intersection of the signal subspaces generated by each subarray, while GCA-rMUSIC estimates eigenvectors of the spatially-smoothed coarray covariance matrix that are based on a global noise projection matrix. Then, it successfully combines the noise subspaces to estimate the DOAs more accurately by using a root-MUSIC-like strategy at the end. The first algorithm consists of a pseudospectrum strategy with a complexity smaller than that of GCA-rMUSIC, which is a subspace-based search-free polynomial rooting algorithm. Numerical results have demonstrated that the root-MUSIC version presents increased accuracy when compared to its pseudo-spectrum counterpart. We rigorously derived fifteen Fisher Information Matrices corresponding to all parameters of the data model employed in our analysis. These matrices facilitate the computation of the Cramér-Rao Lower Bound (CRLB) for the estimation problem under consideration, thereby establishing a theoretical bound on the performance of the GCA-MUSIC and GCA-rMUSIC estimators. In addition, they are instrumental in quantifying the amount of information each parameter contributes to the estimation problem.

In Chapter 4, we developed a spatial smoothing scheme for sparse linear arrays that addresses a key limitation of CA-MUSIC, namely, its fixed spatial smoothing subarray aperture. We introduced the variable window size (VWS) versions of coarray MUSIC and coarray root-MUSIC for fully coherent arrays, followed by analogous extensions for partially calibrated arrays, termed VWS-GCA-MUSIC and VWS-GCA-rMUSIC.

The VWS versions enhance the root mean square error (RMSE) performance of the algorithms by incorporating a low-rank matrix into the spatial smoothing process, which adds signal energy to the signal subspace. This improvement arises because the VWS scheme introduces noiseless covariance matrices into the spatial smoothing sum, increasing the rank of the effective covariance matrix. Moreover, the VWS versions are more general than the original (G)CA-(r)MUSIC algorithms, which can be interpreted as particular cases of VWS-based algorithms when the spatial smoothing window compression parameter is set to zero.

For future work, we suggest some further investigations that could enrich this work by extending the techniques to alternative scenarios and giving a more detailed analysis of the developed algorithms. Indeed, we suggest the following activities:

a) **Statistical analysis of algorithms**. Although we have established the foundations for VWS-CA-(r)MUSIC, (VWS-)GCA-MUSIC, as well as (VWS-)GCA-rMUSIC, it remains to be quantified in terms of RMSE their statistical performance (efficiency) and the associated asymptotic properties. Although the CRLB gives us an idea about this aspect in terms of numerical simulations, this kind of analysis would greatly increase the comprehension of those techniques for a variety of scenarios and could reveal opportunities for further improvements regarding accuracy.

b) **Extensions of algorithms to 2D and 3D arrays**. A significant future contribution to the field of DoA estimation in array signal processing would be the extension of the generalized coarray algorithms to planar (2D) and volumetric (3D) array configurations. For instance, in the planar array case, the extension would enable handling sparse linear subarrays that are not necessarily collocated, allowing for subarrays that can be rotated and displaced arbitrarily within the 2D plane.

This introduces the possibility of subarrays observing the same source from different DoAs, offering increased spatial diversity. Such an extension would be particularly beneficial for type-II sparse subarrays, where position diversity can be exploited to improve array performance. To fully realize this, new theoretical guarantees would need to be developed regarding the degrees of freedom (DoF) afforded by these non-collocated and rotated subarrays, ensuring accurate DoA estimation. Similarly, extending the framework to volumetric (3D) arrays would allow for the deployment of subarrays at arbitrary positions within a 3D space, further increasing the available DoF. This would open up new possibilities for DoA estimation in complex environments and improve performance in scenarios requiring high-resolution spatial sensing. In both 2D and 3D cases, careful analysis is needed to ensure that the coarray structures preserve sufficient DoF and maintain robustness under the proposed array geometries.

c) **Spatial smoothing with increased aperture**. Spatial smoothing techniques are widely used in array signal processing to address the issue of coherent sources, and in this work, they play a key role in increasing the rank of the coarray covariance matrix. However, a well-known trade-off is that spatial smoothing reduces the effective array aperture, thereby limiting the total number of sources that can be identified.

Developing a technique that preserves or even increases the coarray aperture during smoothing would represent a significant advancement. Such a method would enable more efficient utilization of the coarray degrees of freedom, potentially allowing for the identification of a greater number of sources than current spatial smoothing methods permit. By mitigating the loss in aperture, this approach could enhance the performance of DoA estimation algorithms, particularly in scenarios with a high number of coherent or closely spaced sources.

Bibliography

[1] P. Pal and P. P. Vaidyanathan, "Nested arrays: A novel approach to array processing with enhanced degrees of freedom," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4167–4181, 2010.

[2] W. S. Leite, R. C. de Lamare, Y. Zakharov, W. Liu, and M. Haardt, "Direction finding with sparse subarrays: Design, algorithms and analysis," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–15, 2024.

[3] D. Malioutov, M. Çetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[4] Z.-M. Liu, Z.-T. Huang, Y.-Y. Zhou, and J. Liu, "Direction-of-arrival estimation of noncircular signals via sparse representation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, pp. 2690–2698, 7 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6237619/

[5] P. Gerstoft, C. F. Mecklenbrauker, A. Xenaki, and S. Nannuru, "Multisnapshot sparse bayesian learning for doa," *IEEE Signal Processing Letters*, vol. 23, pp. 1469–1473, 10 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7536146/

[6] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, mar 1986. [Online]. Available: http://ieeexplore.ieee.org/document/1143830/

[7] A. Jaffer, "Maximum likelihood direction finding of stochastic sources: a separable solution," in *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1988, pp. 2893–2896. [Online]. Available: http://ieeexplore.ieee.org/document/197258/

[8] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 984–995, 7 1989. [Online]. Available: http://ieeexplore.ieee.org/document/32276/

[9] M. Haardt and J. A. Nossek, “Unitary esprit: How to obtain increased estimation accuracy with a reduced computational burden,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 43, 1995.

[10] H. L. Van Trees, *Optimum Array Processing*. New York, USA: John Wiley & Sons, Inc., mar 2002. [Online]. Available: http://doi.wiley.com/10.1002/0471221104

[11] T. E. Tuncer and B. Friedlander, *Classical and Modern Direction-of-Arrival Estimation*. Orlando, FL, USA: Academic Press, Inc., 2009.

[12] D. Johnson and D. Dudgeon, *Array Signal Processing: Concepts and Techniques*, ser. Prentice-Hall Signal Processing Series. P T R Prentice Hall, 1993.

[13] D. Rieken and D. Fuhrmann, “Generalizing MUSIC and MVDR for multiple noncoherent arrays,” *IEEE Transactions on Signal Processing*, vol. 52, pp. 2396–2406, 9 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1323249/

[14] T. Tirer and O. Bialer, “A method for reducing the performance gap between non-coherent and coherent sub-arrays,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 3358–3370, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9099368/

[15] C. See and A. Gershman, “Direction-of-arrival estimation in partly calibrated subarray-based sensor arrays,” *IEEE Transactions on Signal Processing*, vol. 52, pp. 329–338, 2 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1261321/

[16] M. Pesavento, A. B. Gershman, and K. M. Wong, “Direction finding in partly calibrated sensor arrays composed of multiple subarrays,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 2103–2115, 9 2002.

[17] M. Pesavento, A. Gershman, and K. M. Wong, “Direction of arrival estimation in partly calibrated time-varying sensor arrays,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 5. IEEE, 2001, pp. 3005–3008.

[18] T. Tirer and O. Bialer, “Direction of arrival estimation for non-coherent sub-arrays via joint sparse and low-rank signal recovery,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6 2021, pp. 4395–4399.

[19] W. K. Ma, T. H. Hsieh, and C. Y. Chi, "DOA estimation of quasi-stationary signals via Khatri-Rao subspace," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 2165–2168, 2009.

[20] P. Pal and P. P. Vaidyanathan, "A novel array structure for directions-of-arrival estimation with increased degrees of freedom," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, no. 2. IEEE, 2010, pp. 2606–2609.

[21] C.-L. Liu and P. P. Vaidyanathan, "Super Nested Arrays: Linear Sparse Arrays With Reduced Mutual Coupling—Part I: Fundamentals," *IEEE Transactions on Signal Processing*, vol. 64, no. 15, pp. 3997–4012, aug 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7458883/

[22] A. Moffet, "Minimum-redundancy linear arrays," *IEEE Transactions on Antennas and Propagation*, vol. 16, no. 2, pp. 172–175, mar 1968.

[23] C.-L. Liu and P. Vaidyanathan, "Cramér–Rao bounds for coprime and other sparse arrays, which find more sources than sensors," *Digital Signal Processing*, vol. 61, pp. 43–61, feb 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1051200416300264

[24] E. Vertatschitsch and S. Haykin, "Nonredundant arrays," *Proceedings of the IEEE*, vol. 74, no. 1, pp. 217–217, 1986. [Online]. Available: http://ieeexplore.ieee.org/document/1457703/

[25] A. Dollas, W. Rankin, and D. McCracken, "A new algorithm for Golomb ruler derivation and proof of the 19 mark ruler," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 379–382, 1998. [Online]. Available: http://ieeexplore.ieee.org/document/651068/

[26] Z. Zheng, W.-Q. Wang, Y. Kong, and Y. D. Zhang, "Misc array: A new sparse array design achieving increased degrees of freedom and reduced mutual coupling effect," *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1728–1741, 2019.

[27] W. Shi, Y. Li, and R. C. de Lamare, "Novel sparse array design based on the maximum inter-element spacing criterion," *IEEE Signal Processing Letters*, vol. 29, pp. 1754–1758, 2022.

[28] X. Sheng, D. Lu, Y. Li, and R. C. de Lamare, "Enhanced MISC-based sparse array with high udofs and low mutual coupling," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 2, pp. 972–976, 2024.

[29] N. Song, W. U. Alokozai, R. C. de Lamare, and M. Haardt, "Adaptive Widely Linear Reduced-Rank Beamforming Based on Joint Iterative Optimization," *IEEE Signal Processing Letters*, vol. 21, no. 3, pp. 265–269, mar 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6691928/

[30] H. Ruan and R. C. de Lamare, "Robust Adaptive Beamforming Using a Low-Complexity Shrinkage-Based Mismatch Estimation Algorithm," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 60–64, jan 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6664986/

[31] H. Ruan, "Efficient robust adaptive beamforming algorithms for sensor arrays," Ph.D. Thesis, University of York, York, United Kingdom, 2016.

[32] S. Mallat and Zhifeng Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=258082http://ieeexplore.ieee.org/document/258082/

[33] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, ser. Applied and Numerical Harmonic Analysis. New York, NY: Springer New York, feb 2013, vol. 44, no. 8. [Online]. Available: http://link.springer.com/10.1007/978-0-8176-4948-7http://stacks.iop.org/1751-8121/44/i=8/a=085201?key=crossref.abc74c979a75846b3de48a5587bf708f

[34] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.acha.2009.04.002

[35] Y. C. Eldar and G. Kutyniok, *Compressed Sensing*, Y. C. Eldar and G. Kutyniok, Eds. Cambridge: Cambridge University Press, 2012. [Online]. Available: http://ebooks.cambridge.org/ref/id/CBO9780511794308

[36] W. S. Leite and R. C. de Lamare, "List-Based OMP and an Enhanced Model for DOA Estimation With Nonuniform Arrays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 4457–4464, 2021.

[37] Z. Chen, G. Gokeda, and Y. Yu, *Introduction to Direction-of-arrival Estimation*, ser. Artech House signal processing library. Artech House, 2010. [Online]. Available: https://books.google.com.br/books?id=_63d0OKsDsAC

[38] L. Wang, R. C. de Lamare, and M. Haardt, "Direction finding algorithms based on joint iterative subspace optimization," *IEEE Transactions on*

Aerospace and Electronic Systems, vol. 50, no. 4, pp. 2541–2553, oct 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6978860/

[39] L. Qiu, Y. Cai, R. C. de Lamare, and M. Zhao, "Reduced-Rank DOA Estimation Algorithms Based on Alternating Low-Rank Decomposition," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 565–569, may 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7433376/

[40] A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8, 1983, pp. 336–339.

[41] R. Roy, A. Paulraj, and T. Kailath, "Esprit - a subspace rotation approach to estimation of parameters of cisoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1340–1342, 1986.

[42] S. F. B. Pinto and R. C. de Lamare, "Multistep knowledge-aided iterative esprit: Design and analysis," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2189–2201, 2018.

[43] M. Haardt and J. Nossek, "Unitary esprit: how to obtain increased estimation accuracy with a reduced computational burden," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1232–1242, 1995.

[44] P. Stoica and K. Sharman, "Novel eigenanalysis method for direction estimation," *IEE Proceedings F Radar and Signal Processing*, vol. 137, no. 1, p. 19, 1990, institution of Engineering and Technology (IET).

[45] A. Swindlehurst, P. Stoica, and M. Jansson, "Exploiting arrays with multiple invariances using MUSIC and MODE," *IEEE Transactions on Signal Processing*, vol. 49, no. 11, pp. 2511–2521, 2001.

[46] W. S. Leite and R. C. de Lamare, "Noncoherent sparse subarrays for doa estimation based on low-rank and sparse recovery," *2022 30th European Signal Processing Conference (EUSIPCO)*, set 2022.

[47] W. Du and R. Kirlin, "Improved spatial smoothing techniques for doa estimation of coherent signals," *IEEE Transactions on Signal Processing*, vol. 39, no. 5, pp. 1208–1210, 1991.

[48] D.-Z. Feng, W. X. Zheng, and A. Cichocki, "Matrix-group algorithm via improved whitening process for extracting statistically independent sources from array signals," *IEEE Transactions on*

Signal Processing, vol. 55, pp. 962–977, 3 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4099555/

[49] W. Xia and Z. He, "Active direction finding in imaging sensor nets," in *2008 11th IEEE Singapore International Conference on Communication Systems*. IEEE, 11 2008, pp. 1359–1363.

[50] B. Rao and K. Hari, "Performance analysis of root-music," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1939–1949, 1989. [Online]. Available: http://ieeexplore.ieee.org/document/45540/

[51] S. Al-Juboori and X. N. Fernando, "Multiantenna spectrum sensing over correlated nakagami-m channels with mrc and egc diversity receptions," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2155–2164, 2018.

[52] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 4th ed. Johns Hopkins University Press, 2012.

[53] W. Suleiman, P. Parvazi, M. Pesavento, and A. M. Zoubir, "Non-coherent direction-of-arrival estimation using partly calibrated arrays," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5776–5788, 2018.

[54] W. Suleiman and P. Parvazi, "Search-free decentralized direction-of-arrival estimation using common roots for non-coherent partly calibrated arrays," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 2292–2296.

[55] 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. [Online]. Available: http://ieeexplore.ieee.org/document/17564/

[56] M. Jansson, B. Göransson, and B. Ottersten, "A subspace method for direction of arrival estimation of uncorrelated emitter signals," *IEEE Transactions on Signal Processing*, vol. 47, 1999.

[57] M. Wang and A. Nehorai, "Coarrays, music, and the cramér–rao bound," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 933–946, 2017.

[58] J. Pan, M. Sun, Y. Wang, and X. Zhang, "An enhanced spatial smoothing technique with ESPRIT algorithm for direction of arrival estimation in coherent scenarios," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3635–3643, 2020.

[59] S. Haykin and K. J. Ray Liu, *Handbook on Array Processing and Sensor Networks*. New Jersey, USA: John Wiley & Sons, Inc., 2009.

[60] Z. Yang, P. Stoica, and J. Tang, "Source resolvability of spatial-smoothing-based subspace methods: A hadamard product perspective," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2543–2553, May 2019.

[61] S. Pillai and B. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, pp. 8–15, 1989.

[62] W. Du and R. Kirlin, "Improved spatial smoothing techniques for DOA estimation of coherent signals," *IEEE Transactions on Signal Processing*, vol. 39, pp. 1208–1210, 1991.

[63] D. mei, Z. shouhong, W. xiangdong, and Z. huanying, "A High Resolution Spatial Smoothing Algorithm," in *2007 International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*. IEEE, 2007, pp. 1031–1034.

[64] J. Pan, M. Sun, Y. Wang, and X. Zhang, "An Enhanced Spatial Smoothing Technique With ESPRIT Algorithm for Direction of Arrival Estimation in Coherent Scenarios," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3635–3643, 2020.

[65] K. Takao, N. Kikuma, and T. Yano, "Toeplitzization of correlation matrix in multipath environment," in *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11. Institute of Electrical and Electronics Engineers, 1986, pp. 1873–1876.

[66] C.-l. Liu and P. P. Vaidyanathan, "Super nested arrays: Sparse arrays with less mutual coupling than nested arrays," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, mar 2016, pp. 2976–2980. [Online]. Available: http://ieeexplore.ieee.org/document/7472223/

[67] W. S. Leite and R. C. De Lamare, "Partially-calibrated sparse subarrays for direction-of-arrival estimation with increased degrees of freedom," in *2022 56th Asilomar Conference on Signals, Systems, and Computers*, 2022, pp. 980–984.

A
Appendix - Theoretical limits on RMSE performance

I) Computation of derivatives

- Derivative with respect to $\boldsymbol{\theta}$:

$$\begin{aligned}\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial \theta_i} =& \frac{\partial \tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}}}{\partial \theta_i}\mathbf{R}_s\tilde{\mathbf{H}}^H\tilde{\mathbf{V}}^H(\boldsymbol{\theta}) \\ &+ \tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}}\mathbf{R}_s\frac{\partial \tilde{\mathbf{H}}^H\tilde{\mathbf{V}}^H(\boldsymbol{\theta})}{\partial \theta_i}\end{aligned} \tag{A-1}$$

The derivatives $\partial(\tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}})/\partial\theta_i$ can be represented in compact form as

$$\frac{\partial \tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}}}{\partial \theta_i} = \mathbf{D}\mathbf{e}_i\mathbf{e}_i^T \tag{A-2}$$

where

$$\mathbf{D} = \left[\frac{\partial \mathbf{V}(\boldsymbol{\theta}_1)\mathbf{h}_1}{\partial \theta_1}, \ldots, \frac{\partial \mathbf{V}(\boldsymbol{\theta}_D)\mathbf{h}_D}{\partial \theta_D}\right] \tag{A-3}$$

for which $\mathbf{e}_i$ corresponds to the i-th canonical basis vector with all zeros except for a 1 in the i-th position. The goal of the outer product (projection) of the canonical basis vectors is to isolate the i-th column of $\mathbf{D}$ and simultaneously nullify all its other columns. Notice that the parameters $\tilde{\mathbf{H}}$ are a function of the DOA sources in the general case.

By substituting (A-2) in (A-1), we have:

$$\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial \theta_i} = \mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H + \mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{D}^H \tag{A-4}$$

where

$$\begin{aligned}\mathbf{W} &\triangleq \tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}} \\ &= [\mathbf{V}(\boldsymbol{\theta}_1)\mathbf{h}_1, \ldots, \mathbf{V}(\boldsymbol{\theta}_D)\mathbf{h}_D]\end{aligned} \tag{A-5}$$

$$\mathbf{R}_s = \operatorname{diag}(\mathbf{p}) \tag{A-6}$$

- Derivative with respect to $\mathbf{p}$ (source powers)

$$\begin{aligned}\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial p_i} &= \mathbf{W}\frac{\partial \mathbf{R}_s}{\partial p_i}\mathbf{W}^H \\ &= \mathbf{w}_i\mathbf{w}_i^H\end{aligned} \tag{A-7}$$

where $\mathbf{w}_i$ is the i-th column of $\mathbf{W}$, defined in (A-5).

– Derivative with respect to σ_n^2

Clearly, $\mathbf{R}_{\mathbb{S}}$ can be expressed as

$$\mathbf{R}_{\mathbb{S}} = \tilde{\mathbf{V}}(\boldsymbol{\theta})\tilde{\mathbf{H}}\mathbf{R}_s\tilde{\mathbf{H}}^H\tilde{\mathbf{V}}^H(\boldsymbol{\theta}) + \sigma_n^2\mathbf{I} \tag{A-8}$$

From (A-8), the derivative of the covariance matrix with respect to the noise variance is given by

$$\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial \sigma_n^2} = \mathbf{I} \tag{A-9}$$

– Derivatives with respect to the real part of the calibration parameters $\boldsymbol{\nu}_l$

$$\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial \nu_{l,i}} = \frac{\partial \mathbf{W}}{\partial \nu_{l,i}}\mathbf{R}_s\mathbf{W}^H + \mathbf{W}\mathbf{R}_s\frac{\partial \mathbf{W}^H}{\partial \nu_{l,i}} \tag{A-10}$$

The derivatives $\partial\mathbf{W}/\partial\nu_{l,i}$ can be represented as

$$\frac{\partial \mathbf{W}}{\partial \nu_{l,i}} = \mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\mathbf{e}_i^T \tag{A-11}$$

where

$$\begin{aligned}\mathbf{P}_{\mathbf{E}_l} &= \mathbf{E}_l\mathbf{E}_l^T,\ \mathbf{E}_l = \left[\mathbf{0}_{a_l\times N_l}^T \quad \mathbf{I}_{N_l\times N_l} \quad \mathbf{0}_{b_l\times N_l}^T\right]^T \\ a_l &= \sum_{i=1}^{l-1} N_i,\ b_l = \sum_{i=l+1}^{L} N_i \\ \mathbf{A} &= \left[\mathbf{A}_{\mathbb{S}_1}^T(\boldsymbol{\theta}), \ldots, \mathbf{A}_{\mathbb{S}_L}^T(\boldsymbol{\theta})\right]^T\end{aligned} \tag{A-12}$$

By substituting (A-11) in (A-10), we can write

$$\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial \nu_{l,i}} = \mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H + \mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l} \tag{A-13}$$

– Derivatives with respect to the imaginary part of the calibration parameters $\boldsymbol{\eta}_l$

Due to the structure and relation between $\boldsymbol{\nu}_l$ and $\boldsymbol{\eta}_l$, the derivatives $\partial\mathbf{W}/\partial\eta_{l,i}$ follow immediately from (A-11) as

$$\begin{aligned}\frac{\partial \mathbf{W}}{\partial \eta_{l,i}} &= \jmath\frac{\partial \mathbf{W}}{\partial \nu_{l,i}} \\ &= \jmath\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\mathbf{e}_i^T\end{aligned} \tag{A-14}$$

The derivative of the covariance matrix with respect to $\eta_{l,i}$ is given by

$$\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial \eta_{l,i}} = \frac{\partial \mathbf{W}}{\partial \eta_{l,i}}\mathbf{R}_s\mathbf{W}^H + \mathbf{W}\mathbf{R}_s\frac{\partial \mathbf{W}^H}{\partial \eta_{l,i}} \tag{A-15}$$

By replacing (A-14) into (A-15) results in

$$\begin{aligned}\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial \eta_{l,i}} = j\Big(&\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H - \\ &\mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\Big)\end{aligned} \tag{A-16}$$

II) Calculation of the FIM matrices

Firstly, observe that the elements of the FIM are given by

$$\begin{aligned}[\mathsf{FIM}]_{ij} &= -E\left[\frac{\partial^2\mathcal{L}_{\mathbf{x}}(\boldsymbol{\phi})}{\partial\phi_i\partial\phi_j}\right] \\ &= T\,\mathrm{tr}\left[\mathbf{R}_{\mathbb{S}}^{-1}(\boldsymbol{\phi})\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial\phi_i}\mathbf{R}_{\mathbb{S}}^{-1}(\boldsymbol{\phi})\frac{\partial \mathbf{R}_{\mathbb{S}}(\boldsymbol{\phi})}{\partial\phi_j}\right]\end{aligned} \tag{A-17}$$

which will be used in the derivations that follows.

1. FIM matrix with respect to the DOAs $\boldsymbol{\theta}$

By substituting (A-4) into (A-17), we have

$$\begin{aligned}&[\mathsf{FIM}_{\boldsymbol{\theta\theta}}]_{ij} = \\ &\mathrm{tr}\{\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H \\ &+\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{D}^H \\ &+\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H \\ &+\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{D}^H\}\end{aligned} \tag{A-18}$$

By applying the cyclic property of the trace, we obtain:

$$\begin{aligned}&[\mathsf{FIM}_{\boldsymbol{\theta\theta}}]_{ij} = \\ &T\,\mathrm{tr}\{(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i)(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_j) \\ &+(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i)(\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_j) \\ &+(\mathbf{e}_j^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i)(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j) \\ &+(\mathbf{e}_j^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i)(\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j)\}\end{aligned} \tag{A-19}$$

which can be rewritten without the trace as:

$$\begin{aligned}&[\mathsf{FIM}_{\boldsymbol{\theta\theta}}]_{ij} = \\ &2T\mathfrak{Re}\{(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_j)(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i) \\ &+(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j)(\mathbf{e}_j^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i)\}\end{aligned} \tag{A-20}$$

or in matrix form

$$\begin{aligned}&\mathsf{FIM}_{\boldsymbol{\theta\theta}} = \\ &2T\mathfrak{Re}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\ &+ (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T\}\end{aligned} \tag{A-21}$$

where $\odot$ denotes the Hadamard product. Due to space constraints, only the final results of the subsequently computed FIMs are presented.

2. FIM matrix with respect to the sources powers $\mathbf{p}$

   By replacing (A-7) into (A-17), we can write

$$\begin{aligned}[\mathsf{FIM}_{\mathbf{pp}}]_{ij} &= \mathrm{tr}\left[\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_j\mathbf{w}_j^H\right] \\ &= \mathrm{tr}\left[\left(\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_j\right)\left(\mathbf{w}_j^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_i\right)\right] \\ &= (\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W})_{ij}(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W})_{ji}\end{aligned} \tag{A-22}$$

   where we have used the cyclic property of the trace operator. In matrix form, we can represent this result as

$$\mathsf{FIM}_{\mathbf{pp}} = T(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}) \odot (\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W})^T \tag{A-23}$$

3. FIM matrix with respect to the noise power σ_n^2

   By substituting (A-9) into (A-17), we arrive at

$$\mathsf{FIM}_{\sigma_n^2\sigma_n^2} = T\,\mathrm{tr}\left[\mathbf{R}_{\mathbb{S}}^{-2}\right] \tag{A-24}$$

4. FIM matrix with respect to the real part of the calibration parameters $\boldsymbol{\nu}_l$

   By substituting (A-13) into (A-17), we can write

$$\begin{aligned}&[\mathsf{FIM}_{\boldsymbol{\nu}_l\boldsymbol{\nu}_k}]_{ij} = \\ &\mathrm{tr}\left\{\left(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}\mathbf{e}_j\right)\left(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\right)\right. \\ &+ \left(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\right)\left(\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\right) \\ &+ \left(\mathbf{e}_i^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}\mathbf{e}_j\right)\left(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\right) \\ &+ \left(\mathbf{e}_i^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\right) \times \\ &\left.\left(\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\right)\right\}\end{aligned} \tag{A-25}$$

which can be rewritten as

$$\begin{aligned}
&[\mathsf{FIM}_{\boldsymbol{\nu}_l\boldsymbol{\nu}_k}]_{ij} = 2\mathfrak{Re}\left\{\left(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}\mathbf{e}_j\right)\times\right.\\
&\left(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\right)+\\
&\left(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\right)\times\\
&\left.\left(\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\right)\right\}
\end{aligned}\tag{A-26}$$

or in matrix form

$$\begin{aligned}
&\mathsf{FIM}_{\boldsymbol{\nu}_l\boldsymbol{\nu}_k} =\\
&2T\mathfrak{Re}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A})\odot(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T\\
&+(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)\odot(\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T\}
\end{aligned}\tag{A-27}$$

5. FIM matrix with respect to the imaginary part of the calibration parameters $\boldsymbol{\eta}_l$

   By substituting (A-16) into (A-17), we can write

$$\begin{aligned}
&\left[\mathsf{FIM}_{\boldsymbol{\eta}_l\boldsymbol{\eta}_k}\right]_{ij} =\\
&-\operatorname{tr}\left\{\left(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}\mathbf{e}_j\right)\left(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\right)\right.\\
&-\left(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\right)\left(\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\right)\\
&-\left(\mathbf{e}_i^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}\mathbf{e}_j\right)\left(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\right)\\
&\left.+\left(\mathbf{e}_i^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\right)\left(\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\right)\right\}
\end{aligned}\tag{A-28}$$

which can be rewritten as

$$\begin{aligned}
&\left[\mathsf{FIM}_{\boldsymbol{\eta}_l\boldsymbol{\eta}_k}\right]_{ij} =\\
&-2\mathfrak{Re}\{(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}\mathbf{e}_j)(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i)\\
&-(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j)(\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i)\}
\end{aligned}\tag{A-29}$$

or in matrix form

$$\begin{aligned}
&\mathsf{FIM}_{\boldsymbol{\eta}_l\boldsymbol{\eta}_k} =\\
&-2T\mathfrak{Re}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A})\odot(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T\\
&-(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)\odot(\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T\}
\end{aligned}\tag{A-30}$$

6. FIM matrix with respect to $\boldsymbol{\theta}$ and $\mathbf{p}$

$$\begin{aligned}[\mathsf{FIM}_{\boldsymbol{\theta}\mathbf{p}}]_{ij} &= \operatorname{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_j\mathbf{w}_j^H+\right.\\ &\left.\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_j\mathbf{w}_j^H\right\}\end{aligned} \tag{A-31}$$

However, it should be noticed that the quantity $\mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H$ can be further simplified as $p_i\mathbf{d}_i\mathbf{w}_i^H$, where $\mathbf{d}_i$ is the i-th column of $\mathbf{D}$ and $\mathbf{R}_s = \operatorname{diag}(\mathbf{p})$. Considering this, (A-31) can be rewritten as

$$\begin{aligned}[\mathsf{FIM}_{\boldsymbol{\theta}\mathbf{p}}]_{ij} &= \operatorname{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-1}p_i\mathbf{d}_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_j\mathbf{w}_j^H+\right.\\ &\left.\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_i\mathbf{d}_i^H p_i\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_j\mathbf{w}_j^H\right\}\\ &= \operatorname{tr}\left\{\left(\mathbf{w}_j^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{d}_i\right)\left(p_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_j\right)+\right.\\ &\left.\left(\mathbf{w}_j^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_i p_i\right)\left(\mathbf{d}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_j\right)\right\}\\ &= 2\mathfrak{Re}\left\{\left(p_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_j\right)\left(\mathbf{w}_j^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{d}_i\right)\right\}\\ &= 2\mathfrak{Re}\left\{\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\right)_{ij}\left(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\right)_{ji}\right\}\end{aligned} \tag{A-32}$$

which can be represented in matrix form as

$$\mathsf{FIM}_{\boldsymbol{\theta}\mathbf{p}} = 2T\mathfrak{Re}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W})\odot(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T\} \tag{A-33}$$

7. FIM matrix with respect to $\boldsymbol{\theta}$ and σ_n^2

$$\begin{aligned}\left[\mathsf{FIM}_{\boldsymbol{\theta}\sigma_n^2}\right]_{ij} &= \operatorname{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}+\right.\\ &\left.\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}^{-1}\right\}\\ &= \operatorname{tr}\left\{p_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{d}_i + p_i\mathbf{d}_i^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{w}_i\right\}\\ &= 2\mathfrak{Re}\left\{p_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{d}_i\right\}\\ &= 2\mathfrak{Re}\left\{\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{D}\right)_{ii}\right\}\end{aligned} \tag{A-34}$$

or in matrix form

$$\mathsf{FIM}_{\boldsymbol{\theta}\sigma_n^2} = 2T\operatorname{diag}^M(\mathfrak{Re}\left\{\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{D}\right\}) \tag{A-35}$$

where $\operatorname{diag}^M(\mathbf{X})$ is a column vector whose elements consists of the diagonal elements of square-matrix $\mathbf{X}$.

8. FIM matrix with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\nu}_l$

$$\begin{aligned}
&[\mathsf{FIM}_{\boldsymbol{\theta}\boldsymbol{\nu}_l}]_{ij} = \\
&\mathrm{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\right. \\
&+\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l} \\
&+\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H \\
&\left.+\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\right\} \\
&=\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\right)_{ji}\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right)_{ij} \\
&+\left(\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\right)_{ji}\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\right)_{ij} \\
&+\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\right)_{ji}\left(\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right)_{ij} \\
&+\left(\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\right)_{ji}\left(\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\right)_{ij}
\end{aligned} \tag{A-36}$$

or in matrix form

$$\begin{aligned}
&\mathsf{FIM}_{\boldsymbol{\theta}\boldsymbol{\nu}_l} = \\
&=\mathcal{T}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})\odot(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\
&+(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)\odot(\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\
&+(\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})\odot(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T \\
&+(\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)\odot(\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T\}
\end{aligned} \tag{A-37}$$

9. FIM matrix with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\eta}_l$

$$\begin{aligned}
&\left[\mathsf{FIM}_{\boldsymbol{\theta}\boldsymbol{\eta}_l}\right]_{ij} = \\
&j\,\mathrm{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\right. \\
&-\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\mathbf{e}_i\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l} \\
&+\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H \\
&\left.-\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\mathbf{e}_i^T\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\right\} \\
&=j\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\right)_{ji}\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right)_{ij} \\
&-j\left(\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D}\right)_{ji}\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\right)_{ij} \\
&+j\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\right)_{ji}\left(\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right)_{ij} \\
&-j\left(\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\right)_{ji}\left(\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\right)_{ij}
\end{aligned} \tag{A-38}$$

or in matrix form

$$\begin{aligned}
&\mathsf{FIM}_{\boldsymbol{\theta}\boldsymbol{\eta}_l} = \\
&= jT\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\
&- (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{D})^T \\
&+ (\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}) \odot (\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T \\
&- (\mathbf{D}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s) \odot (\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T\}
\end{aligned} \tag{A-39}$$

10. FIM matrix with respect to $\mathbf{p}$ and $\boldsymbol{\sigma}_n^2$

$$\begin{aligned}
\left[\mathsf{FIM}_{\mathbf{p}\sigma_n^2}\right]_{ij} &= \operatorname{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\right\} \\
&= \operatorname{tr}\left\{\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{w}_i\right\} \\
&= \left(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{W}\right)_{ii}
\end{aligned} \tag{A-40}$$

or in matrix form

$$\mathsf{FIM}_{\mathbf{p}\sigma_n^2} = T\,\operatorname{diag}^M(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{W}) \tag{A-41}$$

11. FIM matrix with respect to $\mathbf{p}$ and $\boldsymbol{\nu}_l$

$$\begin{aligned}
&[\mathsf{FIM}_{\mathbf{p}\boldsymbol{\nu}_l}]_{ij} = \\
&\operatorname{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H \right.\\
&\left. + \mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\right\} \\
&= 2\mathfrak{Re}\left\{\left(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right)_{ij}\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\right)_{ji}\right.
\end{aligned} \tag{A-42}$$

or in matrix form

$$\begin{aligned}
&\mathsf{FIM}_{\mathbf{p}\boldsymbol{\nu}_l} = \\
&= 2T\mathfrak{Re}\left\{\left(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right) \odot \left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\right)^T\right\}
\end{aligned} \tag{A-43}$$

12. FIM matrix with respect to $\mathbf{p}$ and $\boldsymbol{\eta}_l$

$$\begin{aligned}
&\left[\mathsf{FIM}_{\mathbf{p}\boldsymbol{\eta}_l}\right]_{ij} = \\
&j\,\mathrm{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\right. \\
&\left.-\ \mathbf{R}_{\mathbb{S}}^{-1}\mathbf{w}_i\mathbf{w}_i^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\right\} \\
&= -2\mathfrak{Im}\left\{\left(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right)_{ij}\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\right)_{ji}\right\}
\end{aligned} \tag{A-44}$$

or in matrix form

$$\begin{aligned}
&\mathsf{FIM}_{\mathbf{p}\boldsymbol{\eta}_l} = \\
&-2T\mathfrak{Im}\{(\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})\odot(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W})^T\}
\end{aligned} \tag{A-45}$$

13. FIM matrix with respect to σ_n^2 and $\boldsymbol{\nu}_l$

$$\begin{aligned}
\left[\mathsf{FIM}_{\sigma_n^2\boldsymbol{\nu}_l}\right]_{ij} &= \mathrm{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\right. \\
&\left.+\ \mathbf{R}_{\mathbb{S}}^{-2}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\right\} \\
&= 2\mathfrak{Re}\left\{\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right)_{jj}\right\}
\end{aligned} \tag{A-46}$$

or in matrix form

$$\mathsf{FIM}_{\sigma_n^2\boldsymbol{\nu}_l} = 2T\,\mathrm{diag}^{\mathrm{M}}(\mathfrak{Re}\{\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\})^T \tag{A-47}$$

14. FIM matrix with respect to σ_n^2 and $\boldsymbol{\eta}_l$

$$\begin{aligned}
\left[\mathsf{FIM}_{\sigma_n^2\boldsymbol{\eta}_l}\right]_{ij} &= j\,\mathrm{tr}\left\{\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_j\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\right. \\
&\left.-\ \mathbf{R}_{\mathbb{S}}^{-2}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\right\} \\
&= -2\mathfrak{Im}\left\{\left(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\right)_{jj}\right\}
\end{aligned} \tag{A-48}$$

or in matrix form

$$\mathsf{FIM}_{\sigma_n^2\boldsymbol{\eta}_l} = -2T\,\mathrm{diag}^{\mathrm{M}}(\mathfrak{Im}\{\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-2}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\})^T \tag{A-49}$$

15. FIM matrix with respect to $\boldsymbol{\nu}_l$ and $\boldsymbol{\eta}_k$

$$\begin{aligned}
&\left[\mathsf{FIM}_{\boldsymbol{\nu}_l\boldsymbol{\eta}_k}\right]_{ij} = j\,\mathrm{tr}\left\{\left(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}\mathbf{e}_j\right)\times\right.\\
&\left(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\right)\\
&-\left(\mathbf{e}_i^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\right)\left(\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A}\mathbf{e}_i\right)\\
&+\left(\mathbf{e}_i^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A}\mathbf{e}_j\right)\left(\mathbf{e}_j^T\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\right)\\
&-\left(\mathbf{e}_i^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_j\right)\times\\
&\left.\left(\mathbf{e}_j^T\mathbf{A}^H\mathbf{P}_{\mathbf{E}_k}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s\mathbf{e}_i\right)\right\}
\end{aligned} \tag{A-50}$$

$$\begin{aligned}
&\mathsf{FIM}_{\boldsymbol{\nu}_l\boldsymbol{\eta}_k} =\\
&-2T\mathfrak{Im}\{(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A})\odot(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_l}\mathbf{A})^T\\
&+(\mathbf{A}^H\mathbf{P}_{\mathbf{E}_l}\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{P}_{\mathbf{E}_k}\mathbf{A})\odot(\mathbf{R}_s\mathbf{W}^H\mathbf{R}_{\mathbb{S}}^{-1}\mathbf{W}\mathbf{R}_s)^T\}
\end{aligned} \tag{A-51}$$